AFRL-AFOSR-VA-TR-2016-0138

Combinational Optimal Stopping Problems

Pavlo Krokhmal
IOWA UNIV IOWA CITY

04/01/2016
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 03/25/2016	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* 04/01/2012-12/31/2015

4. TITLE AND SUBTITLE
Combinatorial Optimal Stopping Problems

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-12-1-0142

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. Pavlo Krokhmal

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Iowa
105 Jessup Hall
Iowa City, IA 52242-1316

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Science and Research
875 N. Randolph Street, Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Optimal resource utilization is one of the most general “meta”-settings in operations research: many hard optimization problems can be casted as problems of optimal resource utilization. Additional challenges are introduced by uncertainties; the difficulties are further multiplied in a dynamic context. This project has considered a class of discrete and combinatorial optimal resource utilization problems under uncertainties that arise in the context of the optimal stopping problems. In addition, as a generalization of traditional stochastic formulations that optimize the expected payoff or cost, we considered risk averse discrete and combinatorial optimization problems, where the risk of the stopping decision was estimated using a coherent or convex risk measure. In particular, we developed a special class of certainty equivalent (CE) measures of risk that can be represented via solution of a specially formulated (stochastic) optimization problem. A number of solution techniques for discrete and combinatorial problems involving CE measures have been developed, including exact methods based on polyhedral approximations, branch-and-bound and branch-and-cut algorithms, scenario decomposition techniques, and combinatorial branch-and-bound methods for risk-averse combinatorial optimization problems.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Lynn Hudachek
U	U	U	SAR	229	**19b. TELEPHONE NUMBER** *(Include area code)* 319-335-2123

Reset

DISTRIBUTION A: Distribution approved for public release

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

DISTRIBUTION A: Distribution approved for public release

AFOSR grant FA9550-12-1-0142

Final Report

PI: Dr. Pavlo Krokhmal
Associate Professor
Department of Mechanical and Industrial Engineering
University of Iowa
Iowa City, IA 52242

Optimal resource utilization is one of the most general "meta"-settings in operations research: many hard optimization problems can be casted as problems of optimal resource utilization. Additional challenges are introduced by uncertainties; the difficulties are further multiplied in a dynamic context. This project has considered a class of discrete and combinatorial optimal resource utilization problems under uncertainties that arise in the context of the optimal stopping problems. In addition, as a generalization of traditional stochastic formulations that optimize the expected payoff or cost, we considered risk averse discrete and combinatorial optimization problems, where the risk of the stopping decision was estimated using a coherent or convex risk measure. In particular, we developed a special class of certainty equivalent (CE) measures of risk that can be represented via solution of a specially formulated (stochastic) optimization problem. A number of solution techniques for discrete and combinatorial problems involving CE measures have been developed, including exact methods based on polyhedral approximations, branch-and-bound and branch-and-cut algorithms, scenario decomposition techniques, and combinatorial branch-and-bound methods for risk-averse combinatorial optimization problems.

Particularly, the developed class of certainty equivalent (CE) measures of risk allows builds upon a new representation for coherent and convex measures of risk that expresses the risk measure in the form of infimal convolution of some kernel function, and, importantly, formalizes a key idea that *measure of risk is a solution of a stochastic programming problem.* One of the key properties of this new representation is that it admits incorporation in stochastic programming problems in the form of convex constraints. By selecting the kernel function in this representation in the form of the *certainty equivalent*, a well-known construct in utility theory and decision making under uncertainty, we constructed a family of CE convex nonlinear measures of risk, which allow for direct incorporation of decision-makers preferences, as expressed by his/her utility function, into downside risk measure, and also encompass a number of existing in literature risk measure, such as Conditional-Value-at-Risk, Higher-Moment Coherent Risk measures, etc. The corresponding results are presented in [2].

Implementation of the developed measures of risk in decision making problem under uncertainties leads to mathematical programming problems with a specific set of constraints. A number of computational methods for solving such problems were developed in the course of the project. In particular, we considered special cases of CE measures, corresponding to the choice of the utility function in the form of a power function and an exponential function. In the case of power utility function, the corresponding certainty equivalent measures of risk reduce to higher-moment coherent measures of risk, which are implementable in stochastic programming problems via p-order cone constraints. p-Order cones represent a generalization of well-known second-order cones, but unlike the latter, they are not self-dual, which precludes development of fast, long-step self-dual interior point method algorithms for solving p-cone programming problems. To this end, we developed solution methods based on polyhedral approximations of p-order cones and subsequent decomposition of the obtained approximating linear programming (LP) problems. It has been shown that the developed method allows one to formulate an *exact* solution method for p-cone programming problems, with iteration complexity that is on par with state-of-the-art first-order methods for second-order programming problems. The corresponding results are subsequently used in exact branch-and-bound algorithm for discrete and combinatorial p-cone programming problems. The utilization of polyhedral approximations of p-cones at each node of the branch-and-bound tree allows for taking advantage of "warm-start" capabilities of linear programming solvers, and subsequently reduces solution time by orders of magnitude, compared to branch-and-bound schemes based on solving the nonlinear relaxation of integer p-order programming problem at each branch-and-bound node [4]. A separate research thrust was dedicated to development of branch-and-cut techniques for integer p-order cone programming. In this context, mixed-integer rounding cuts and lifted cuts were developed in [6]. A scenario decomposition technique for solving large-scale stochastic programming problems with risk measures represented in the form of infimal convolution, including certainty equivalent measures of risk, was proposed in [1]. Importantly, this method has been proven to terminate in a number of iterations that does not exceed the number of scenarios, a significant advantage over decomposition methods based on supporting hyperplane representations, where number of iterations could be exponential in the size of the scenario set. In [11], a number of methods for handling a special class of nonlinear convex constraints were proposed as a generalization of earlier developed techniques for p-order cone programming problems.

The developed models and solution approaches were applied to problems of data mining and machine learning [8], identification or robust and risk-averse structures in graphs and combinatorial structures [5, 7, 9, 12, 13]. In papers that consider risk-averse combinatorial problems, a number of combinatorial branch-and-bound algorithms were developed that incorporated solving a stochastic programming problem at each node of combinatorial branch-and-bound tree so as to obtain a bound on the risk of combinatorial substructure corresponding to the branch-and-bound node.

Archival publications

[1] Rysz, M., Vinel, A., Krokhmal, P., and E. L. Pasiliao (2015) A scenario decomposition algorithm for stochastic programming problems with a class of downside risk measures, INFORMS Journal on Computing, 27(2), 416-430.

[2] Vinel, A. and P. Krokhmal (2015) Certainty equivalent measures of risk, Annals of Operations Research, DOI:10.1007/s10479-015-1801-0.

[3] Chernikov, D., Krokhmal, P., Zhupanska, O. I., and C. L. Pasiliao (2015) A two-stage stochastic PDE-constrained optimization approach to vibration control of an electrically conductive composite plate subjected to mechanical and electromagnetic loads, Structural and Multidisciplinary Optimization, 52(2), 227-352.

[4] Vinel, A. and P. Krokhmal (2014) Polyhedral approximations in p-order cone programming, Optimization Methods and Software, 29(6), 1210-1237.

[5] Rysz, M., Mirghorbani, M., Krokhmal, P. and E. L. Pasiliao (2014) On risk-averse maximum weighted subgraph problems, Journal of Combinatorial Optimization, 28(1), 167-185.

[6] Vinel, A. and P. Krokhmal (2014) On valid inequalities for mixed integer p-order cone programming, Journal of Optimization Theory and Applications, 160(2), 439-456.

[7] Rysz, M. , Krokhmal, P., and E.L. Pasiliao (2013) Minimum risk maximum clique problem, in: A. Sorokin and P. M. Pardalos (Eds), Dynamics of Information Systems: Algorithmic Approaches, Springer Proceedings in Mathematics & Statistics, vol. 51, 251–267

[8] Morenko, Y., Vinel, A., Yu, Z., and P. Krokhmal (2013) On p-norm linear discrimination, European Journal of Operational Research, 231(3), 784-789.

[9] Rysz, M., Pajouh, F., Krokhmal, P. and E. L. Pasiliao (2014) On risk-averse weighted k-club problems, Examining Robustness and Vulnerability of Critical Infrastructure Networks, NATO Science for Peace and Security Series - D: Information and Communication Security, vol. 37, 231-242.

[10] Mirghorbani, M. and P. Krokhmal (2013) On finding k-cliques in k-partite graphs, Optimization Letters, 7(6), 1155-1165.

Papers in review

[11] Vinel, A. and P. Krokhmal (2015) Mixed-Integer Programming with a Class of Nonlinear Convex Constraints, under review in Discrete Optimization.

[12] Rysz, M., Krokhmal, P., and E. L. Pasiliao (2015) Identifying resilient structures in stochastic networks: A two-stage stochastic optimization approach, under review in Networks.

[13] Rysz, M., Pajouh, F., Krokhmal, P., and E. L. Pasiliao (2015) Identifying risk-averse low-diameter clusters in graphs with stochastic vertex weights, under review in Annals of Operations Research.

Awards, honors, and promotions

Dr. Pavlo Krokhmal (PI) has been accepted a position and promotion to the rank of Full Professor at the Department of Systems and Industrial Engineering at the University of Arizona, effective Jan 04, 2016

Dr. Pavlo Krokhmal (PI) has received the National Research Council’s Senior Research Associateship Award (2015)

Dr. Pavlo Krokhmal (PI) has received the Donald E. Bently Faculty Fellowship of Engineering at the University of Iowa (2013)

Dr. Pavlo Krokhmal (PI) has received AFOSR Summer Faculty Fellowship Award (2014)

Dr. Pavlo Krokhmal (PI) has received the Recognition for Excellence in Teaching and Dedication to Student Success, College of Engineering, University of Iowa (2013)

Dr. Alexander Vinel (former student partially supported by the grant) received a tenure-track Assistant Professor Position in the Department of Industrial and Systems Engineering at Auburn University (2015)

Dr. Maciej Rysz (former student partially supported by the grant) received the National Research Council’s Postdoctoral Associateship Award (2014)

Dr. Alexander Vinel (former student partially supported by the grant) received the Best Research Assistant Award at the Department of Mechanical and Industrial Engineering at the University of Iowa (2015).

Dissertations defended

(by students partially supported by the grant)

Dr. Mohammad Mirghorbani (2013) Graph-theoretical studies of combinatorial optimization problems, PhD Thesis, University of Iowa

Dr. Maciej Rysz (2014) Risk-averse optimization in networks, PhD Thesis, University of Iowa

Dr. Alexander Vinel (2015) Mathematical programming techniques for solving stochastic programming problems with certainty equivalent risk measures, PhD Thesis, University of Iowa

Identifying resilient structures in stochastic networks: A two-stage stochastic optimization approach

Maciej Rysz[1] Pavlo Krokhmal[2]* Eduardo L. Pasiliao[3]

[1]National Research Council, AFRL
Eglin AFB, FL 32542

[2]Department of Mechanical and Industrial Engineering,
University of Iowa, 3131 Seamans Center, Iowa City, IA 52242

[3]Air Force Research Lab, 101 West Eglin Blvd, Eglin AFB, FL 32542

Abstract

We propose a two-stage stochastic programming framework for designing or identifying "resilient", or "repairable" structures in graphs whose topology may undergo a stochastic transformation. The repairability of a subgraph satisfying a given property is defined in terms of a budget constraint, which allows for a prescribed number of vertices to be added to or removed from the subgraph so as to restore its structural properties after the observation of random changes to the graph's set of edges. A two-stage stochastic programming model is formulated and is shown to be $\mathcal{NP}$-complete for a broad range of graph-theoretical properties that the resilient subgraph is required to satisfy. A general combinatorial branch-and-bound algorithm is developed, and its computational performance is illustrated on the example of two-stage stochastic maximum clique problem.

Keywords: Maximum subgraph problem, two-stage stochastic optimization, combinatorial branch-and-bound algorithm, stochastic maximum clique problem.

1 Introduction and motivation

An important feature to incorporate in a networked system's design is an inherent resilience to withstand random structural changes that affect the relationship characteristics between its components. A reliable system should, therefore, possess a high tolerance against a broad range of possible (failure) scenarios, and, moreover, be constructed in such a way that its properties can be restored within available resource limits.

In the present study we pursue an approach that regards a distributed subsystem, or subgraph, to be *resilient* if it can be *"repaired"* at a minimum (or fixed) cost after a random change in the underlying

*Corresponding author, e-mail: krokhmal@engineering.uiowa.edu.

graph's topology. More specifically, many graph-theoretical and network optimization problems consist in finding a subgraph with prescribed properties that has the largest (respectively, smallest) size, weight, etc. Well-known examples include the shortest path problem, maximum clique/independent set problem, minimum vertex cover problem, and so on. In situations when the topology of the underlying graph or network may be subject to changes (e.g., deletions of vertices and/or edges), the "resilience" of the selected subgraph is often of interest. A large body of literature has been accumulated on this subject, where various interpretations of "reliability", "resilience", or "robustness" of subgraphs have been explored (see, among others [11, 13, 22, 25, 28]). Typically, robustness in this context is associated with the ability of the selected subgraph to satisfy (exactly or to a certain degree) a given property, or perform a given function, etc., after deletion of edges and/or vertices. Several examples include network flow control, preservation of vertex and edge connectivity, maximization of overall algebraic connectivity, and prevention of catastrophic cascade failures [6–8, 13].

In this work we adopt the point of view that a structure in a network or graph is "resilient" if it is "repairable" with respect to randomized changes in the graph's topology. Namely, we consider the following general framework: assume that the given graph $G = (V, E)$ may undergo a randomized change in the future, resulting in $G' = (V, E')$, where E' is generally not a subset of E. Then, it is of interest to identify vertex subsets $S, S' \subseteq V$ such that the induced subgraphs $G[S]$ and $G'[S']$ satisfy a prescribed property Π, with additional requirements:

(i) the difference between sets S and S' is within a prescribed bound M, i.e., $|S \setminus S'| + |S' \setminus S| \leq M$;

(ii) the size of S and the expected size of S' should be as large as possible.

In other words, the problem is to identify such a set S that $G[S]$ has property Π and is as large as possible under the condition that, after a random change to the graph's set of edges, the set S may be modified or "repaired" to form a set S', such that $G'[S']$ satisfies Π and the expected size of S' is also as large as possible.

The described framework has obvious interpretations in, for example, the defense domain, where one may be interested in identifying the largest networked or distributed system that can maintain its structure – with, perhaps, necessary repairs – under adversarial attacks.

Mathematically, the described framework lends itself naturally to the context of two-stage stochastic optimization [5, 19], which models decision making process in the presence of uncertainties that involves two sequential decisions. The *first-stage* decision is made before the actual realization of uncertain factors can be observed. The *second-stage,* or *recourse* decision is made upon observing the realization of uncertainties, and takes into account both the preceding first-stage decision and the observed realization of stochastic parameters.

Stochastic recourse problems have gained much attention in the network literature due to their versatility for modeling uncertainties. Particular emphasis has been placed on network problems with random elements evidenced in forms that influence the overall flow distribution, demands, and costs. A number of applications examine stochastic factors in the context of vehicle routing and network flow problems where uncertainties are attributed to arc capacities or node demands (see e.g., [3, 9, 14, 15, 27]). Several similar considerations utilized a two-stage recourse framework to enhance the design of stochastic supply chain networks and network resource allocation [10, 24]. Other studies examined the preservation of connections between vertices when the edge costs are uncertain [7, 16], as well as decision making in routing problems with stochastic edge failures [26].

Although uncertainty in the aforementioned studies mostly influenced decisions related to directed flows and routing, less focus has been put on developing two-stage recourse constructs for designing/identifying graphs that are adept at maintaining their connection properties in situations when random factors affect/alter/damage their original physical characteristics. A notable non-recourse problem of finding the largest subset of vertices that form a clique with a specified probability, given that edges in the graph can fail with some probabilities, was studied in [17]. A similar approach in application to certain clique relaxations was pursued in [31]. In this work, we introduce a two-stage stochastic recourse framework for identifying "sustainable" subgraphs whose structural properties are influenced by definite edge failures and/or construction in each random scenario realization. The proposed model is general and in principle can be adapted to address a broad range of structural graph properties, along with uncertainties in the form of vertex failures.

The remainder of the article is organized as follows. In Section 2 we discuss the deterministic graph-theoretic underpinnings and establish a mathematical programming representation of the *two-stage stochastic recourse maximum subgraph problem*. Section 3 presents an efficient graph-based (combinatorial) branch-and-bound solution algorithm for instances when the desired subgraphs possess hereditary structural properties. Finally, Section 5 considers a numerical case study demonstrating the effectiveness of the proposed algorithm for solving two-stage stochastic recourse maximum clique (i.e., complete graph) problems.

2 Problem definition

In this section we present a formal graph-theoretical description of the discussed framework. Before introducing the stochastic model that represents the focus of the present work, we outline the relevant deterministic concepts, which pertain to problems involving identification of the largest subgraph/subset of a system's vertices that collectively possess a specified structural property.

2.1 Deterministic maximum subgraph problem

Let $G = (V, E)$ represent an undirected graph where each vertex $i \in V$ is a component of the networked system, and an edge $(i, j) \in E$ defines a connection/relation between vertices i and j. Then, the problem of finding the largest (sub)graph $S \subseteq V$ of vertices with a prescribed structural property Π, also known as the *maximum subgraph problem,* or *maximum* Π *problem*, is given by

$$\max_{S \subseteq V} \big\{ |S| : G[S] \ni \Pi \big\}, \tag{1}$$

where $G[S]$ denotes the subgraph of G *induced* by S, i.e., a graph such that any of its vertices i, j are connected by an edge if and only if (i, j) is an edge in graph G. Here and throughout the text the relation $G[S] \ni \Pi$ stands for "$G[S]$ satisfies property Π" (we also say that S *is a* Π*-subgraph of* G); similarly, $G[S] \not\ni \Pi$ represents a converse statement.

In the context of the maximum subgraph problem (1), an important class of graph-theoretical properties Π is represented by *properties that are hereditary with respect to induced subgraphs* (or just *hereditary* for short). Namely, Π is called hereditary with respect to induced subgraphs if for any graph that satisfies Π, removal of any vertex from this graph results in an induced subgraph that also satisfies Π [1, 4, 30]. The class of hereditary properties encompasses many well-known and important graph-theoretical properties, such as *completeness*, *independence*, *planarity*, and so on.

The practical and theoretical significance of the class of hereditary properties in relation to the maximum subgraph problem (1) stems from the fact that a large number of important and difficult graph-theoretical problems are special cases of (1) when Π is hereditary and "meaningful" in some sense. Namely, Π is called *nontrivial* if it is satisfied by a single-vertex graph yet not satisfied by every graph, and is called *interesting* if the order of graphs satisfying Π is unbounded [30]. Then, the following fundamental observation regarding problem (1) holds:

Theorem 1 (Yannakakis [30]) *If property Π is hereditary with respect to induced subgraphs, nontrivial, and interesting, then the maximum subgraph problem* (1) *is $\mathcal{NP}$-complete.*

In many practical applications, the topology of graph G in the maximum subgraph problem (1) may not always be assumed constant, and is subject to unpredictable, or stochastic changes (e.g., edge and/or vertex failures). Once graph G is assumed to be stochastic, however, formulation (1) becomes ill-posed, since it does not provide a guarantee or conditions under which the selected subgraph $G[S]$ satisfies the sought property Π. Therefore, in the presence of uncertainties formulation (1) of the maximum subgraph problem has to be modified so as to explicitly specify the conditions under which its solution can be considered a Π-subgraph of (stochastic) graph G. One common approach in the literature is to require that the solution of an optimization problem with stochastic data satisfies the required properties with a prescribed probability; an application of this approach to a maximum clique problem on stochastic graphs was considered in [17]. In the present endeavor, we require that the solution of the maximum subgraph problem on a stochastic graph is "repairable" in some sense.

2.2 A two-stage stochastic maximum subgraph problem

Here we introduce an approach for determining "resilient" maximum Π-subgraphs in situations when the topology of the underlying graph G may be subject to uncertain (random) future changes that is based on two-stage stochastic programming and which was tentatively outlined in Section 1.

Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the set of random events, $\mathcal{F}$ is the sigma-algebra, and $\mathbb{P}$ is the probability measure, we assume that the topology of a graph $G = (V, E)$ may undergo a random transformation at some moment in the future, resulting in an updated graph $G(\omega) = (V, E(\omega))$, $\omega \in \Omega$. In this work, it is assumed for simplicity that only the set of edges $E = E(\omega)$ may be dependent on the random event ω, while the set of vertices V is constant. As it will be seen next, the proposed formulation and solution method can be generalized to account for possibility of a stochastic set V.

Traditionally to stochastic programming literature, it is assumed that the set Ω is finite, $\Omega = \{\omega_1, \ldots, \omega_N\}$, such that $\mathbb{P}(\omega_k) = p_k > 0$ for $k = 1, \ldots, N$, and $\sum_k p_k = 1$. Consequently, the possible changes to the topology of graph G are observed in the form of N discrete scenarios $\{G(\omega_1), \ldots, G(\omega_N)\}$, where $G(\omega_k) = (V, E(\omega_k))$. For notational convenience, we will denote $G_k = G(\omega_k)$, $E_k = E(\omega_k)$; also, to emphasize that the original graph G represents the unchanged, or "null" state of a distributed system, we denote $G_0 = G = (V, E_0)$, where $E_0 = E$ represents the initial set of edges in the graph.

Characterization of "resilient" substructures in graphs subjected to randomized topology changes via the formalism of two-stage stochastic programming is the key feature of the proposed approach. In general, a two-stage stochastic programming model may be presented in the form

$$\begin{aligned} \min \quad & f_1(\mathbf{x}) + \mathbb{E} f_2(\mathbf{x}, \mathbf{y}(\omega), \omega) \\ \text{s. t.} \quad & \mathbf{h}_1(\mathbf{x}) \leq \mathbf{0}, \\ & \mathbf{h}_2(\mathbf{x}, \mathbf{y}(\omega), \omega) \leq \mathbf{0}, \quad \forall \omega \in \Omega. \end{aligned} \tag{2}$$

Here $\mathbf{x}$ represents the *first-stage* decision/action that is made before the actual realization of the uncertain event ω can be observed. Associated with the first-stage decision are the first-stage cost $f_1(\mathbf{x})$ and the first-stage constraints $\mathbf{h}_1(\mathbf{x}) \leq \mathbf{0}$ that the vector $\mathbf{x}$ has to satisfy. Since the first-stage decision $\mathbf{x}$ may not be optimal for every given possible realization of ω, a recourse, or *second-stage* corrective decision $\mathbf{y} = \mathbf{y}(\omega)$ is made after the actual realization of ω has been observed, so as to minimize some second-stage cost $f_2(\mathbf{x}, \mathbf{y}(\omega), \omega)$. Importantly, the second-stage decision must also satisfy specific second-stage constraints $\mathbf{h}_2(\mathbf{x}, \mathbf{y}(\omega), \omega) \leq \mathbf{0}$ for any given first-stage $\mathbf{x}$. Note that the second-stage decision depends explicitly on the specific realization of ω as well as on the first-stage decision $\mathbf{x}$. In turn, the first-stage decision must take into account *all* possible realizations of the random element ω and the corresponding subsequent recourse decisions $\mathbf{y}(\omega)$. This interdependency is emphasized by the following "nested", or *recourse* representation of the (extensive) form of two-stage stochastic programming formulation (2):

$$\min \big\{ f_1(\mathbf{x}) + \mathbb{E} Q(\mathbf{x}, \omega) : \mathbf{h}_1(\mathbf{x}) \leq \mathbf{0} \big\}, \tag{3a}$$

where Q is the second-stage function that represents the optimal second-stage cost given the first-stage vector $\mathbf{x}$ and the observed realization ω:

$$Q(\mathbf{x}, \omega) = \min \big\{ f_2(\mathbf{x}, \mathbf{y}(\omega), \omega) : \mathbf{h}_2(\mathbf{x}, \mathbf{y}(\omega), \omega) \leq \mathbf{0} \big\}. \tag{3b}$$

According to the above, the following two-stage framework is adopted for identification of "resilient" Π-subgraphs in G_0:

- Given a graph $G_0 = (V, E_0)$, find a set of vertices $S_0 \subseteq V$ such that the induced subgraph $G_0[S_0]$ satisfies Π (*"first stage"*).
- Graph G_0 undergoes a randomized change of topology. It is assumed that the resulting graph $G_k = (V, E_k)$ is chosen at random with probability p_k from a collection of graphs $\{G_1, \ldots, G_N\}$ (*"observation of uncertainty"*).
- For any given realization G_k, select sets $\Delta_k^+ \subseteq V \setminus S_0$ and $\Delta_k^- \subseteq S_0$, such that after "augmentation" or "repair" of the original set S_0 the resulting set S_k,

  $$S_k := (S_0 \setminus \Delta_k^-) \cup \Delta_k^+,$$

  induces a subgraph $G_k[S_k]$ on G_k that satisfies Π (*"second, or recourse stage"*).
- Sets S_0 and $\Delta_k^\pm$ must be chosen in such way that the expected size of Π-subgraph in the first and second stages is maximized, and sets $\Delta_k^\pm$ contain no more than M vertices,

  $$|\Delta_k^+| + |\Delta_k^-| \leq M. \tag{4}$$

Then, the *two-stage stochastic maximum subgraph (TSMS) problem* can be stated in the graph-theoretical formulation as follows:

$$\begin{aligned}
\max \quad & |S_0| + \sum_{k \in \mathcal{N}} p_k |S_k| && \text{(5a)}\\
\text{s. t.} \quad & G_k[S_k] \ni \Pi, \quad \forall k \in \{0\} \cup \mathcal{N} && \text{(5b)}\\
& |S_0 \setminus S_k| + |S_k \setminus S_0| \le M, \quad \forall k \in \mathcal{N} && \text{(5c)}\\
& S_k \subseteq V, \quad \forall k \in \{0\} \cup \mathcal{N}, && \text{(5d)}
\end{aligned}$$

where $\mathcal{N} = \{1, \ldots, N\}$. Obviously, the defined above delta-sets $\Delta_k^{\pm}$ are related to the second-stage sets S_k as

$$\Delta_k^+ = S_k \setminus S_0, \quad \Delta_k^- = S_0 \setminus S_k, \quad k \in \mathcal{N}.$$

The above extended formulation of the two-stage stochastic programming problem can be presented in the recourse form similar to (3):

$$\max_{S_0 \subseteq V} \left\{ |S_0| + \sum_{k \in \mathcal{N}} p_k Q_k(S_0) : G_0[S_0] \ni \Pi \right\}, \tag{6a}$$

where the second-stage function Q_k has the form

$$Q_k(S) = \max_{S_k \subseteq V} \big\{ |S_k| : G_k[S_k] \ni \Pi, \ |S \setminus S_k| + |S_k \setminus S| \le M \big\}. \tag{6b}$$

Complexity of the two-stage stochastic maximum subgraph problem (5)–(6) is established in the next two propositions. For this, consider the decision version of the two-stage stochastic maximum subgraph problem (5)–(6), denoted as $\langle (G_0, \ldots, G_N), (p_1, \ldots, p_N), M, q \rangle$: given a set of $N + 1$ graphs $G_0, \ldots, G_N$ such that $V(G_0) = \ldots = V(G_N)$, a set of positive rational numbers $p_1, \ldots, p_N$ such that $p_1 + \ldots + p_N = 1$, an integer $M \ge 0$, and a rational $q \ge 0$, determine whether graphs G_i contain Π-subgraphs S_i such that $|S_0 \setminus S_i| + |S_i \setminus S_0| \le M$ for all $i = 1, \ldots, N$, and $|S_0| + \sum_{i=1}^{N} p_k |S_k| \ge q$. Similarly, the decision version of the maximum subgraph problem, denoted as $\langle G, m \rangle$, is as follows: given a graph G and a nonnegative integer m, determine whether G contains a Π-subgraph S such that $|S| \ge m$.

Proposition 1 *The decision version of the two-stage stochastic maximum subgraph problem* (5) *is $\mathcal{NP}$-complete, provided that the corresponding maximum subgraph problem is $\mathcal{NP}$-complete.*

Proof: Noting that the two-stage stochastic maximum subgraph problem is obviously in $\mathcal{NP}$, we prove its $\mathcal{NP}$-completeness by reduction from the maximum subgraph problem. Given an instance $\langle G, m \rangle$ of the maximum subgraph problem, let $G_i^* = G$ for $i = 0, \ldots, N$, select arbitrary rational $p_i^* > 0$ such that $p_1^* + \ldots + p_N^* = 1$, an arbitrary integer $M^* \ge 0$, and let $q^* = 2m$. Then, a collection of sets $S_0^* = \ldots = S_N^* \subseteq V(G_i^*)$, $i = 0, \ldots, N$, satisfies the condition $|S_0^* \setminus S_i^*| + |S_i^* \setminus S_0^*| \le M^*$, and, moreover, satisfies $|S_0^*| + \sum_{i=1}^{N} p_i^* |S_i^*| \ge m + m = q^*$ if and only if there exists $S \subseteq V(G)$ of order $|S| \ge m$. □

Next, we observe that for any given first-stage solution, "repairing" it in the second stage via solving the second-stage problem (6b) is $\mathcal{NP}$-complete as well. To this end, the corresponding decision version

$\langle G_k, S, M, q\rangle$ of second-stage maximum subgraph problem is formulated as follows: given a second-stage graph G_k, a first-stage solution $S \subseteq V(G_0) = V(G_k)$, and integer numbers $M \geq 0$ and $q \geq 0$, determine if a Π-subgraph $S_k \subseteq V(G_k)$ of order at least q exists such that $|S \setminus S_k| + |S_k \setminus S| \leq M$. Then, the next observation holds.

Proposition 2 *The decision version of the second-stage maximum subgraph problem problem* (6b) *at any scenario $k \in \mathcal{N}$ is $\mathcal{NP}$-complete if property Π is such that the maximum subgraph problem is $\mathcal{NP}$-complete.*

Proof: First, note that the second-stage maximum subgraph problem is in $\mathcal{NP}$. Next, observe that the order of the Π-subgraph of G_k that satisfies the budget constraint $|S \setminus S_k| + |S_k \setminus S| \leq M$ cannot exceed $\min\{|S| + M, |V|\}$. Then, for a given instance of the maximum subgraph problem $\langle G, m\rangle$, construct an instance $\langle G_k^*, S^*, M^*, q^*\rangle$ of second-stage maximum subgraph problem with $G_k^* = G$, $S^* = \{i\}$ for a fixed $i \in V(G)$, $M^* = m - 1$, and $q^* = m$. The order of the largest Π-subgraph S_k^* of thusly constructed instance $\langle G_k^*, S^*, M^*, q^*\rangle$ of second-stage maximum subgraph problem is always less than or equal to m according to the above observation; moreover, it is equal to m if and only if there exists a Π-subgraph of G of order m that contains vertex i. Therefore, the question of whether a graph G has a Π-subgraph with m vertices can be answered by solving no more than $|V(G)|$ instances $\langle G, \{i\}, m-1, m\rangle$ of second-stage problem as described above. □

Note that while the introduced model assumes a common property Π for the subgraphs selected during both decision stages, possible extensions may include distinct properties at each stage. Further, the model may be enhanced by imposing nonuniform cost structures associated with selecting, adding and removing the vertices; or by introducing different budgetary restrictions in different scenarios.

3 A combinatorial branch-and-bound solution technique for the two-stage stochastic maximum subgraph problem

In this section we introduce an exact graph-based, or combinatorial branch-and-bound (BnB) algorithm for solving problem (5)–(6). We emphasize, however, that the computational efficiency of the proposed method – as with all BnB schemes – depends to a great extent on the specific branching and bounding criteria used for processing of the search space with respect to a particular property Π. An illustration of the proposed procedure is furnished in Section 5 for the case when Π represents the completeness property of a subgraph.

The proposed technique for solving the two-stage stochastic maximum subgraph problem relies on the recourse representation (6a)–(6b) and employs "nested" BnB algorithms for construction of first- and second-stage Π-subgraphs, respectively, that satisfy the interrelationships imposed by the budgetary repair constraints (5c). Namely, a first-stage BnB procedure identifies first-stage subgraphs in G_0 that satisfy property Π while an embedded second-stage BnB is used to determine the largest possible associated Π subgraphs in $G_1, \ldots, G_N$ that can be supported within the repair budget after changes to the original graph G_0 occur. Both algorithms work by navigating between *levels* of the respective BnB trees until the subgraphs of G_0 and G_k, $k = 1, \ldots, N$, that maximize the objective of (5)–(6) are found.

For convenience of notation, it is assumed that S_0 and S_k, $k \in \mathcal{N}$, represent *feasible* solutions (subgraphs) during all but the last iterations of the respective BnB algorithms, upon which they coincides with the *optimal* solution(s) to problem (6).

3.1 First stage branch-and-bound algorithm

The first-stage BnB algorithm begins at level $\ell = 0$ with a partial solution $S_0 := \emptyset$, and a partial and global lower bounds on the objective value of problem (6a), $\mathcal{Z} := -\infty$ and $\mathcal{Z}^* := -\infty$, respectively. Throughout the algorithm the partial solution S_0 contains the vertices in V such that $G_0[S_0]$ satisfies property Π.

At the current node of the BnB tree, level ℓ is associated with the *candidate set* C_ℓ of vertices from which any single vertex can be added to the partial solution S_0 without violating property Π. *Branching* is conducted by removing a branching vertex q from C_ℓ and adding it to the partial solution S_0. The algorithm is initialized with $C_0 := V$, and once a vertex q is selected, the candidate set at level $\ell + 1$ is constructed by eliminating all the vertices from C_ℓ whose inclusion in S_0 would violate the property Π:

$$C_{\ell+1} := \{i \in C_\ell : G_0[S_0 \cup i] \ni \Pi\}. \tag{7}$$

As it will be readily seen next, the operation of constructing candidate set $C_{\ell+1}$ from the preceding candidate set C_ℓ constitutes one of the basic steps of the algorithm, and the computational cost of this step can affect significantly the computational performance of the solution method. In this regard, a major question is whether one can efficiently verify property Π for any given subgraph. The associated decision problem is a follows: given a subgraph S, determine whether S satisfies property Π, or whether some fraction of the representation of S can be modified in order for S to satisfy property Π. In the latter case it is said that S is *ϵ-far* from satisfying property Π, where ϵ corresponds to the fraction of modifications that need to be made. With respect to hereditary properties, a substantial body of literature was accumulated in recent years to address this question. For example, Alon and Shapira [2] showed that every hereditary property is testable with one-sided error. Further, several characterizations of hereditary properties have been proposed [12]. As described above, a property Π is said to be *node-hereditary* if it is closed under taking *induced* subgraphs of G, and is *subgraph-hereditary* if it is closed under taking subgraphs of G. A property is *minor-hereditary* if any *graph minor*[1] S of graph G satisfies Π. In a series of seminal studies [20, 21], Robertson and Seymour established the *graph minor theorem* which, among others, predicated polynomial time identification of hereditary properties closed under graph minors.

In what follows, we implicitly assume that the property Π of a graph can be tested in polynomial time.

Bounding[2] of the partial subgraph S_0 involves determining the quality of the solution that can be obtained by further exploring the vertices in $C_{\ell+1}$. Observe that the most opportune realization of uncertainties is such that the structure of edges sets E_k, $k \in \mathcal{N}$, would preserve the property Π of S_0 in each graph G_k; and – provided that sufficiently many favorable edge modifications occur – the budget M can exclusively be used to add new vertices from set $V \setminus S_0$ to subgraph S_0. In other words, under "ideal" circumstances a second-stage solution of size $\min\{|S_0| + M, |V|\}$ is obtained in any given scenario $k \in \mathcal{N}$. For a given $S \subseteq V(G)$, let $\upsilon_\Pi(G[S])$ represent an upper bound on the size of the largest possible Π-subgraph contained in the induced graph $G[S]$, where subscript Π indicates that the properties and computation of this bound depend explicitly on Π. Then, $\min\{\upsilon_\Pi(G_0[S_0 \cup C_{\ell+1}]) + M, |V|\}$ represents an upper bound on the potential contribution of the recourse action, whence the left-hand-side of the expression

$$\upsilon_\Pi\big(G_0[S_0 \cup C_{\ell+1}]\big) + \min\{\upsilon_\Pi(G_0[S_0 \cup C_{\ell+1}]) + M, |V|\} \leq \mathcal{Z}^* \tag{8}$$

[1] A graph S is a minor of G if edge contractions can be performed of a subgraph of G to obtain S.

[2] The specific mechanisms of both branching and bounding should be selected according to the subgraph property Π under consideration.

provides a "best-case" objective value for problem (6a) with respect to the current partial solution S_0 and candidate set $C_{\ell+1}$. Inequality (8) determines whether the algorithm branches further or backtracks, namely, if (8) is violated, the algorithm proceeds to solve the second-stage recourse problems (6b) for all $k \in \mathcal{N}$, otherwise the algorithm backtracks.

In general, the modified sets of edges E_k, $k \in \mathcal{N}$, will not preserve the property Π of S_0 as described. A significant drawback of condition (8) is therefore its disregard for structural variations between G_0 and G_k, particularly relative to how well solution S_0 and the vertices in candidate set $C_{\ell+1}$ will "perform" in any given scenario realization $k \in \mathcal{N}$. It is therefore of interest to introduce several feasibility and reparability conditions towards improving condition (8) in the context posed by the following question: *given a current first-stage solution S_0 and corresponding candidate set $C_{\ell+1}$, what is the minimum number of modifications that must be made to S_0 in any second-stage scenario $k \in \mathcal{N}$ in order to ascertain property Π?*

One possibility is to perform the feasiblity test furnished by the next proposition prior to solving $Q_k(S_0)$ for $k \in \mathcal{N}$.

Proposition 3 *For a given scenario $k \in \mathcal{N}$, let $S_0^{(k)}$ represent a subset of S_0 that induces a Π-subgraph in $G_k[S_0]$. If the following condition is satisfied,*

$$|S_0| - \max_{S_0^{(k)} \subseteq S_0} \left\{ \left|S_0^{(k)}\right| : G_k\left[S_0^{(k)}\right] \ni \Pi \right\} > M, \tag{9}$$

then subgraph S_0 is an infeasible (irreparable) first-stage solution to problem (5)–(6).

Proof: Recall that the induced subgraph $G_0[S_0]$ has property Π by construction. Clearly, since the vertices remain fixed between the decision stages, the largest possible set of vertices $S_0^{(k)}$ such that $G_k\left[S_0^{(k)}\right] \ni \Pi$ is no larger than $|S_0|$ (i.e., $S_0^{(k)} \subseteq S_0$). Hence, the left-hand-side of expression (9) represents the smallest number Δ_k^- of vertices that must be removed from S_0 in order to obtain a subset $S_0^{(k)}$ that induces a subgraph $G_k\left[S_0^{(k)}\right]$ with property Π under scenario $k \in \mathcal{N}$. This immediately implies that if condition (9) holds for any $k \in \mathcal{N}$, the budget constraint in (6b) cannot be satisfied. □

Finding the maximum subset $S_0^{(k)}$ that induces a Π-subgraph in $G_k[S_0]$ by solving a problem of type (1) for each scenario $k \in \mathcal{N}$ in expression (9) is clearly computationally infeasible. Instead, we utilize the fact that $|S_0| \geq \upsilon_\Pi(G_k[S_0]) \geq \left|S_0^{(k)}\right|$, and employ a more efficient condition by replacing the second term in expression (9) by $\upsilon_\Pi(G_k[S_0])$,

$$|S_0| - \upsilon_\Pi(G_k[S_0]) > M. \tag{10}$$

Obviously, condition (9) is satisfied whenever (10) holds. Assuming that subgraph S_0 is deemed feasible under the current assumptions, the left-hand-side of (10) represents an approximation of the *minimum* number of vertices that must be removed from S_0 under scenario $k \in \mathcal{N}$.

By a similar argument, it is possible to determine the number of vertices that will have to be removed from subgraph S_0 in the second stage if a vertex $i \in C_{\ell+1}$ is added to S_0 in the first stage.

Corollary 1 *If inequality* (9) *is satisfied in scenario $k \in \mathcal{N}$, then vertex $i \in C_{\ell+1}$ can be removed from $C_{\ell+1}$ if the condition*

$$|S_0 \cup i| - \max_{S_i^{(k)} \subseteq S_0 \cup i} \left\{ \left|S_i^{(k)}\right| : G_k\left[S_i^{(k)}\right] \ni \Pi \right\} > M, \tag{11}$$

holds for some Π*-subgraph* $S_i^{(k)}$ *in the induced subgraph* $G_k[S_0 \cup i]$.

An analogous approximation to that of (10) is then obtained,

$$|S_0 \cup i| - \upsilon_\Pi(G_k[S_0 \cup i]) > M. \tag{12}$$

All vertices $i \in C_{\ell+1}$ that satisfy (12) are removed prior to computing $\upsilon_\Pi(G_0[S_0 \cup C_{\ell+1}])$; the resulting "refined" candidate set

$$C'_{\ell+1} := \left\{ i \in C_{\ell+1} : |S_0 \cup i| - \upsilon_\Pi(G_k[S_0 \cup i]) \leq M, \ \forall k \in \mathcal{N} \right\},$$

produces a more conservative estimate $\upsilon_\Pi(G_0[S_0 \cup C_{\ell+1}])$ in (8). To simplify the notation, it will hereafter be assumed that $C_{\ell+1}$ denotes the refined candidate set $C'_{\ell+1}$.

In the case when inequality (10) is violated at all scenarios $k \in \mathcal{N}$, then prior to solving problems $Q_k(S_0)$, $k = 1, \ldots, N$, the following bounding condition for the objective value of problem (6a) is verified at the current node of the BnB tree:

$$\upsilon_\Pi(G_0[S_0 \cup C_{\ell+1}]) + \sum_{k \in \mathcal{N}} p_k \min \left\{ \upsilon_\Pi(G_k[S_0 \cup C_{\ell+1}]) + M_k, |V| \right\} \leq \mathcal{Z}^*, \tag{13}$$

where $M_k = M - (|S_0| - \upsilon_\Pi(G_k[S_0]))$, $k \in \mathcal{N}$, represent reduced budgets obtained from (10).

If inequality (13) is violated, then there are two possibilities that can arise with respect to the second-stage problems (6b). First, the second-stage problem (6b) may be infeasible for some k, given the current solution S_0. Then, the corresponding second-stage function $Q_k(S_0)$ and the respective recourse function $\mathsf{E}_\omega[Q(S_0)] = \sum_{k \in \mathcal{N}} p_k Q_k(S_0)$ assume value of $-\infty$. In this case, vertex q is removed from S_0 and the next branching vertex is selected from the candidate set if $C_\ell \neq \emptyset$. An illustration of such a case is given in Figure 1. Alternatively, all second stage problems are feasible and functions $Q_k(S_0)$, $k = 1, \ldots, N$, are finite, whence the current objective value associated with problem (6a) is updated as $\mathcal{Z} = |S_0| + \sum_{k \in \mathcal{N}} p_k Q_k(S_0)$; the global lower bound $\mathcal{Z}^*$ is replaced by $\mathcal{Z}$ if $\mathcal{Z}^* < \mathcal{Z}$. Then, if the candidate set is non-empty, $C_{\ell+1} \neq \emptyset$, the algorithm selects a branching vertex q from the next level $\ell + 1$. The branching vertex q at level ℓ is stored as q_ℓ for backtracking purposes. Alternatively, if $C_{\ell+1} = \emptyset$, the algorithm backtracks by removing vertex q from S_0.

Whenever condition (13) is satisfied, there is no possibility of achieving an improvement over the global lower bound $\mathcal{Z}^*$ by exploring further levels of the BnB tree; vertex q is removed from S_0. If $C_\ell = \emptyset$, the algorithm backtracks to level $\ell - 1$ by removing from S_0 the most recent branching vertex that was used at level $\ell - 1$, namely vertex $q_{\ell-1}$. The described first-stage BnB procedure is formalized in Algorithm 1.

3.2 Second-stage branch-and-bound algorithm

The BnB algorithm for solving the second-stage problem $Q_k(S_0)$, $k \in \mathcal{N}$, identifies the largest subgraph $S_k^* \subseteq V(G_k)$ with property Π that satisfies the budget constraint (5c). As in the first-stage BnB technique, it navigates the levels of the (second-stage) BnB tree by exploring branching vertices from candidate sets that individually satisfy the property Π with respect to the partial solution S_k. The bounding procedure of the second-stage algorithm pertains to eliminating unfavorable search space relative to the budgetary restriction M. Namely, the subgraph selected in the second stage must be feasible with respect to the first-stage partial solution S_0 in the sense that the number of added and removed vertices from S_0 in scenario k do not exceed the budget M.

Algorithm 1: First stage combinatorial BnB method

```
1  Initialize: $\ell := 0;\ C_0 := V;\ S_0 := \emptyset;\ \mathcal{Z} = \mathcal{Z}^* = -\infty;\ M \in \mathbb{Z}_+;$
2  while $\ell \geq 0$ do
3      if $C_\ell \neq \emptyset$ then
4          select a vertex $q \in C_\ell$;
5          $C_\ell := C_\ell \setminus q$;
6          $S_0 := S_0 \cup q$;
7          for $k \in \mathcal{N}$ do
8              if $|S_0| - \upsilon_\Pi(G_k[S_0]) > M$ then
9                  $S_0 := S_0 \setminus q$;
10                 goto Step 3
11             else
12                 $M_k := M - (|S_0| - \upsilon_\Pi(G_k[S_0]))$
13         $C_{\ell+1} := \{i \in C_\ell : G_0[S_0 \cup i] \ni \Pi\}$;
14         $C_{\ell+1} := \{i \in C_{\ell+1} : |S_0 \cup i| - \upsilon_\Pi(G_k[S_0 \cup i]) \leq M,\ \forall k \in \mathcal{N}\}$;
15         if $\upsilon_\Pi(G_0[S_0 \cup C_{\ell+1}]) + \sum_{k\in\mathcal{N}} p_k \min\left\{\upsilon_\Pi(G_k[S_0 \cup C_{\ell+1}]) + M_k, |V|\right\} > \mathcal{Z}^*$ then
16             for $k \in \mathcal{N}$ do
17                 compute $Q_k(S_0)$;
18                 if $Q_k(S_0) = -\infty$ then
19                     $S_0 := S_0 \setminus q$;
20                     goto Step 3;
21                 else
22                     $\mathcal{Z} := |S_0| + \sum_{k\in\mathcal{N}} p_k Q_k(S_0)$;
23             if $\mathcal{Z} > \mathcal{Z}^*$ then
24                 $\mathcal{Z}^* := \mathcal{Z}$;
25             if $C_{\ell+1} \neq \emptyset$ then
26                 $q_\ell := q$
27                 $\ell := \ell + 1$;
28             else
29                 $S_0 := S_0 \setminus q$;
30         else
31             $S_0 := S_0 \setminus q$
32     else
33         $S_0 := S_0 \setminus q_{\ell-1}$;
34         $\ell := \ell - 1$;
35 return $\mathcal{Z}^*$;
```

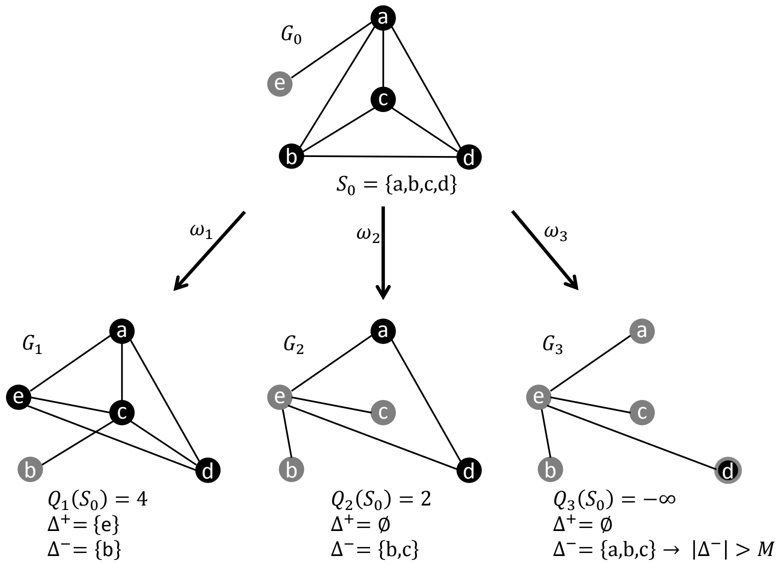


Figure 1: An example with three scenarios demonstrating the reparability of subgraph S_0 with a repair budget $M = 2$ and property Π representing completeness. Black vertices represent those belonging to a complete subgraph. Observe that solution S_0 is feasible (repairable) with respect to scenarios ω_1 and ω_2, but is infeasible (not repairable) with respect to scenario ω_3. Scenario ω_2 also illustrates that the subgraphs in the first or second second stages need not be maximal.

The algorithm begins by selecting a branching vertex q from the candidate set C_ℓ^k; initially $C_0^k := V$. Due to the fact that adding and removing vertices from S_0 imposes a budgetary penalty, the natural tendency is to maintain as similar of a structure as possible in the second stage. Noting that vertices common to C_ℓ^k and the solution S_0 do not utilize the budget M, a vertex $q \in \{S_0 \cap C_\ell^k\}$ is always selected first if $\{S_0 \cap C_\ell^k\} \neq \emptyset$. Once q is added to the second-stage partial solution S_k, the candidate set at the next level $C_{\ell+1}^k$ is constructed by removing all the vertices from C_ℓ^k whose inclusion in S_k would violate property Π.

Given the first- and second-stage partial solutions S_0 and S_k, respectively, the left-hand-side of constraint (5c) can easily be computed so that $\delta = |S_0 \setminus S_k| + |S_k \setminus S_0|$. Observe that the number of vertices in $C_{\ell+1}^k$ that could preserve or reduce the value of δ at consecutive levels of the BnB tree is given by $\gamma = |S_0 \cap C_{\ell+1}^k|$. Several bounding consideration emerge as a result.

The following conditions are possible when $\delta - \gamma \leq M$:

(C1) If $\delta \leq M$, then (5c) is satisfied via vertices in S_k, and S_k replaces S_k^* if $|S_k| > |S_k^*|$. In cases when $\delta = M$ and $\gamma > 0$, a branching vertex $q \in \{S_0 \cap C_{\ell+1}^k\}$ is selected and the algorithm branches to level $\ell := \ell + 1$. On the other hand, if $\gamma = 0$, adding more vertices to S_k will violate (5c); thus, the algorithm backtracks by removing the most recent branching vertex q from S_k. If $\delta < M$ and $C_{\ell+1}^k \neq \emptyset$, the algorithm always branches.

(C2) If $\delta > M$, the partial solution S_k is infeasible with respect to (5c). However, the set $\{S_0 \cap C_{\ell+1}^k\}$ necessarily contains a sufficient number of vertices to (potentially) satisfy M at deeper levels of the BnB tree, i.e., $\gamma \geq \delta - M$. The algorithm branches accordingly.

In cases when $\delta - \gamma > M$, restriction (5c) cannot be satisfied by exploring the vertices in $C_{\ell+1}^k$, and, therefore, the algorithm backtracks as before.

Algorithm 2 outlines the described solution technique for the second-stage problem $Q_k(S_0)$, $k \in \mathcal{N}$. Notice that enhancing the branching and/or bounding scheme is possible by applying structural considerations relative to property Π. However, in an effort to maintain a purely budgetary-based solution procedure that is independent of graph-structural properties, this notion is reserved for future investigations.

Algorithm 2: Second stage combinatorial BnB method for computing $Q_k(S_0)$

```
Input: $G_k$; $S_0$;
Initialize: $\ell := 0$; $C_0^k := V$; $S_k := \emptyset$; $S_k^* := \emptyset$;
while $\ell \geq 0$ do
    if $C_\ell^k \neq \emptyset$ then
        if $|S_0 \cap C_\ell^k| \neq \emptyset$ then
            select a vertex $q \in \{S_0 \cap C_\ell^k\}$;
        else
            select a vertex $q \in C_\ell^k$;
        $C_\ell^k := C_\ell^k \setminus q$;
        $S_k := S_k \cup q$;
        $C_{\ell+1}^k := \{i \in C_\ell^k : G_k[i \cup S_k]$ satisfies $\Pi\}$;
        $\delta := |S_0 \setminus S_k| + |S_k \setminus S_0|$;
        $\gamma := |S_0 \cap C_{\ell+1}^k|$;
        if $\delta - \gamma \leq M$ then
            if $\delta = M$ and $\gamma = 0$ then
                if $|S_k| > |S_k^*|$ then
                    $S_k^* := S_k$;
                $S_k := S_k \setminus q$;
            else
                $q_\ell := q$;
                $\ell := \ell + 1$;
                if $\delta \leq M$ and $|S_k| > |S_k^*|$ then
                    $S_k^* := S_k$;
        else
            $S_k := S_k \setminus q$;
    else
        $S_k := S_k \setminus q_{\ell-1}$;
        $\ell := \ell - 1$;
return $Q_k(S_0) := |S_k^*|$;
```

4 A mathematical programming formulation of the TSMS problem

A mathematical programming formulation of the maximum subgraph problem (1) can be obtained by, for example, defining a binary vector $\mathbf{x} \in \{0,1\}^{|V|}$ that indicates whether vertex $i \in V$ belongs to the sought subset S (i.e., $x_i = 1$ if $i \in S$ and $x_i = 0$ otherwise), and expressing the property Π in the form of "structural" constraints $\mathbf{\Pi}_G(\mathbf{x}) \leq \mathbf{0}$, such that these constraints are satisfied for a given $\mathbf{x}$ if and only if $G[S]$ satisfies Π, where $S = \{i \in V : x_i = 1\}$:

$$\max \left\{ \mathbf{1}^\top \mathbf{x} : \mathbf{\Pi}_G(\mathbf{x}) \leq \mathbf{0},\ \mathbf{x} \in \{0,1\}^{|V|} \right\}. \tag{14}$$

Here $\mathbf{1}$ denotes the vector of ones of an appropriate dimension. The corresponding 0-1 integer programming formulation of TSMS problem (5) then takes the form

$$\begin{aligned} \max \quad & \mathbf{1}^\top \mathbf{x} + \sum_{k \in \mathcal{N}} p_k \mathbf{1}^\top \mathbf{y}_k && \text{(15a)} \\ \text{s. t.} \quad & \mathbf{\Pi}_{G_0}(\mathbf{x}) \leq \mathbf{0} && \text{(15b)} \\ & \mathbf{\Pi}_{G_k}(\mathbf{y}_k) \leq \mathbf{0}, \quad \forall k \in \mathcal{N} && \text{(15c)} \\ & \|\mathbf{x} - \mathbf{y}_k\|_1 \leq M, \quad \forall k \in \mathcal{N} && \text{(15d)} \\ & \mathbf{x}, \mathbf{y}_k \in \{0,1\}^{|V|}, \quad \forall k \in \mathcal{N}, && \text{(15e)} \end{aligned}$$

where the vector $\mathbf{x}$ denotes the first-stage decision variables, and the second-stage variables $\mathbf{y}_k$ are defined for any fixed $k \in \mathcal{N}$ as $y_{ki} = 1$ if $i \in S_k$ and $y_{ki} = 0$ otherwise. Constraints (15d) impose the previously described budgetary restrictions. In correspondence to (6), the above extensive formulation of the TSMS problem can be equivalently presented in recourse form:

$$\begin{aligned} \max \quad & \mathbf{1}^\top \mathbf{x} + \sum_{k \in \mathcal{N}} p_k Q_k(\mathbf{x}) && \text{(16a)} \\ \text{s. t.} \quad & \mathbf{\Pi}_{G_0}(\mathbf{x}) \leq \mathbf{0} && \text{(16b)} \\ & \mathbf{x} \in \{0,1\}^{|V|}, && \text{(16c)} \end{aligned}$$

where the second-stage function is given by

$$\begin{aligned} Q_k(\mathbf{x}) = \max \quad & \mathbf{1}^\top \mathbf{y}_k && \text{(17a)} \\ \text{s. t.} \quad & \mathbf{\Pi}_{G_k}(\mathbf{y}_k) \leq \mathbf{0} && \text{(17b)} \\ & \|\mathbf{x} - \mathbf{y}_k\|_1 \leq M && \text{(17c)} \\ & \mathbf{y}_k \in \{0,1\}^{|V|}. && \text{(17d)} \end{aligned}$$

We next consider a particular instance of the TSMS problem when the property Π defines a clique.

5 Two-stage stochastic maximum clique problem

As an illustrative example of the general TSMS problem and the proposed solution approaches, in this section we consider the *two-stage maximum clique problem,* a special case of the TSMS problem (5) when the property Π represents completeness. Then, the graph-theoretical formulation of the two-stage stochastic maximum clique problem takes the form

$$\max \quad |S_0| + \sum_{k\in\mathcal{N}} p_k|S_k| \tag{18a}$$

$$\text{s. t.} \quad \{S_k \subseteq V : \forall i, j \in S_k, (i, j) \in E_k\}, \quad \forall k \in \{0\} \cup \mathcal{N} \tag{18b}$$

$$|S_0 \setminus S_k| + |S_k \setminus S_0| \leq M, \quad \forall k \in \mathcal{N} \tag{18c}$$

$$S_k \subseteq V, \quad \forall k \in \{0\} \cup \mathcal{N}. \tag{18d}$$

The corresponding mathematical programming formulation that we use in this work employs the well-known *edge formulation* [18] of the structural constraints that guarantee completeness of the selected subgraph, namely

$$\left\{\mathbf{z} \in \{0,1\}^{|V|} : \mathbf{\Pi}_G(\mathbf{z}) \leq \mathbf{0}\right\} = \left\{\mathbf{z} \in \{0,1\}^{|V|} : z_i + z_j \leq 1 \ \text{ for all } \ (i,j) \in \overline{E}\right\},$$

where $\overline{E}$ represents the set of edges of the complement of graph G, i.e., $(i, j) \in E \Leftrightarrow (i, j) \notin \overline{E}$ for any $i, j \in V$. Then, the two-stage stochastic maximum clique problem admits the following 0-1 integer programming from:

$$\max \quad \sum_{i\in V} x_i + \sum_{k\in\mathcal{N}} p_k \left(\sum_{i\in V} y_{ik}\right) \tag{19a}$$

$$\text{s. t.} \quad x_i + x_j \leq 1, \qquad \forall (i, j) \in \overline{E} \tag{19b}$$

$$y_{ik} + y_{jk} \leq 1, \qquad \forall (i, j) \in \overline{E}_k, \ k \in \mathcal{N} \tag{19c}$$

$$\sum_{i\in V} |x_i - y_{ik}| \leq M, \quad \forall k \in \mathcal{N} \tag{19d}$$

$$x_i, y_{ik} \in \{0, 1\}, \qquad \forall i \in V, \ k \in \mathcal{N}. \tag{19e}$$

Formulation (19) can be solved with appropriate integer programming solvers.

The property-specific techniques for finding cliques in all types of graphs via Algorithms 1–2 are described next.

5.1 Candidate set generation, branching and bounding techniques

When property Π defines a clique, a number of efficient techniques has been developed in literature that can be utilized for candidate set generation, branching, and bounding. For example, the candidate sets can be efficiently generated and updated an intersection of neighboring vertices common to the clique elements. Constructing candidate set (7) is performed by pairwise testing any vertex $j \in S_0$ against a vertex $i \in C_\ell$, and removing the vertices from C_ℓ that are not adjacent to subgraph S_0, i.e.,

$$C_{\ell+1} := \{i \in C_\ell : (i, j) \in E_0, \forall j \in S_0\}.$$

A refinement criterion with respect to the second-stage graph scenarios as described by Corollary 1 is furnished by the next proposition.

Proposition 4 *Given a scenario $k \in \mathcal{N}$ and a vertex $i \in C_{\ell+1}$, let $\Gamma_k(i) := \{j \in S_0 : (i, j) \in E_k\}$ represent the (sub)set of vertices such that any two vertices i, j are adjacent in $G_k[S_0 \cup i]$. If the following inequality holds,*

$$|S_0| - |\Gamma_k(i)| > M, \tag{20}$$

then vertex i can be removed from $C_{\ell+1}$.

Proof: If i is added to S_0 in the first stage, then it is easy to see that the vertices $S_0 \setminus \Gamma_k(i)$ must be removed from S_0 in order for $G_k[\Gamma_k(i) \cup i]$ to (possibly) form a complete graph in scenario $k \in \mathcal{N}$. Note that if subgraph $G_k[\Gamma_k(i)]$ is not a clique in $k \in \mathcal{N}$, then at least one vertex from the set $S_0 \setminus \Gamma_k(i)$ must be further removed from S_0 in the first stage. Thus, $|\Gamma_k(i) \cup i|$ provides an upper bound on the size of the maximum clique contained in $G_k[S_0 \cup i]$. Consequently, expression (20) approximates the minimum number of vertices that must be removed from S_0 in the second stage if vertex i is included, which cannot exceed the budget M. □

In this study, we consider two techniques for computing the upper bound $\upsilon_\Pi(\cdot)$ on the size of maximum clique and for selecting a branching vertex $q \in C_\ell$ when property Π represents a clique. We emphasize that proper selection of branching and bounding mechanisms according a graph's structural characteristics and the sought property Π does heavily influence the computational performance of the solution method described in Algorithm 1.

5.1.1 An approximate coloring algorithm

The first technique utilizes principles introduced by Tomita et al. [23] to estimate the size of the maximum clique contained in $G[S]$, $S \subseteq V$, by partitioning S into independent sets, also know as *numbering* or *coloring* classes. The vertices in S are first sorted in degree descending order, and a *minimum* positive integer n_i is assigned to each vertex $i \in S$ such that $n_i \neq n_j$ if the pair $i, j \in S$ are connected by an edge $(i, j) \in E(G)$. Consequently, vertices associated with a number class n_k (i.e., vertices with the same assigned integer value) form an independent set.

Since that the size of any clique embodied in $G[S]$ cannot exceed the number of coloring classes generated from S, one immediately obtains a bound on the maximum clique size as

$$\upsilon_\Pi(G[S]) = \max\{n_i : i \in S\}.$$

We use this expression in Algorithm 1 to obtain the bounds $\upsilon_\Pi(G_k[S_0])$ and $\upsilon_\Pi(G_k[S_0 \cup C_{\ell+1}])$, $k \in \{0\} \cup \mathcal{N}$. Condition (13) then takes the form:

$$|S_0| + \max\{n_i : i \in C_{\ell+1}\} + \sum_{k \in \mathcal{N}} p_k \ \min\Big\{\max\{n_i : i \in S_0 \cup C_{\ell+1}\}_k + M_k, |V|\Big\} \leq \mathcal{Z}^*. \tag{21}$$

The branching rule used in connection with the described approximate coloring scheme is as follows: select a vertex $q \in C_\ell$ with the maximum number $n_q := \max\{n_i : i \in C_\ell\}$. Note that an initial coloring of set $C_0 := V$ is performed prior to Step 2.

5.1.2 Directed acyclical path decomposition

Yamguchi and Masuda [29] proposed a clever technique for finding maximum *weighted* cliques in graphs by transforming $G[S]$, $S \subseteq V$, into a directed acyclical graph $\vec{G}[S]$ such that the lengths of the resulting acyclical paths represent bounds on the size of the maximum clique in $G[S]$. The method proceeds as follows. Without loss of generality, let each vertex $i \in S$ be associated with a unit weight $w_i = 1$, and define set $U(S) := \{u_i : \forall i \in S\}$, where each element u_i is initially equivalent to w_i. Then, the set $U(S)$ is updated by sequentially "*propagating*" the elements u_i, $\forall i \in S$, onto adjacent members in S. Particularly, during each iteration a vertex i that corresponds to the minimum argument u_i in set $U(S)$ is selected, and u_i is propagated by adding it to the weights of vertices $j \in S$ adjacent to vertex i in graph $G[S]$. The elements adjacent u_j are updated are updated as

$$u_j = \begin{cases} u_i + w_j, & \text{if } u_j < u_i + w_j, \\ u_j, & \text{otherwise,} \end{cases} \qquad \text{for all } j \in \{j : (i, j) \in E,\ i, j \in S\}. \tag{22}$$

Once a vertex $i \in S$ has been processed, u_i is fixed and cannot be increased in subsequent propagations from other (unprocessed) adjacent vertices in S. The updating process terminates once all the elements in $U(S)$ have been fixed.

Observe that sequentially fixing elements u_i produces a directed acyclical graph $\vec{G}[S]$, where, once all the elements in $U(S)$ are fixed, any $u_i \in U(S)$ represents the longest acyclical path in $\vec{G}[S]$ whose endpoint is the vertex $i \in S$ (see [29] for details). Utilizing the fact that the length[3] of longest path in $\vec{G}[S]$ is an upper bound on the maximum clique size in $G[S]$, one obtains the bounding condition $\upsilon_\Pi(G[S]) = \max\{u_i \in U(S)\}$. Expression (13) then takes the form:

$$|S_0| + \max\{u_i \in U(C_{\ell+1})\} + \sum_{k \in \mathcal{N}} p_k \ \min\Big\{\max\{u_i \in U(S_0 \cup C_{\ell+1})\}_k + M_k, |V|\Big\} \leq \mathcal{Z}^*. \tag{23}$$

In this case, it is assumed that the vertex with the largest propagated weight from adjacent vertices has a high probability of being a part of the maximum clique. As a result, the algorithm branches by selecting the vertex $q \in C_\ell$ that corresponds to the maximum element in $U(C_\ell)$.

5.2 Numerical experiments and results

Numerical experiments demonstrating the performance of the proposed BnB algorithms for solving the TSMS problem when property Π represents a clique were conducted. Problem (19) was solved for randomly generated Erdös-Rényi graphs of orders $|V| = 25, 50, 75, 100$ with average densities of $d = 0.2, 0.5, 0.8$. The number N of second-stage graph scenarios was selected as $N = 25, 50, 75$. For any given graph configuration, the number of vertices $|V|$ and densities d remained fixed during both decision stages. The value of constant M in the budget constraints was fixed at $M = \lceil \epsilon |V| \rceil$, $\epsilon = 0.15$ throughout.

The combinatorial first- and second-stage BnB algorithms described in Section 3 were coded using C++, and CPLEX 12.5 integer programming (IP) solver was used for solving the mathematical programming formulation (19) of the two-stage stochastic maximum clique problem. The computations were ran on an Intel Xeon 3.30GHz PC with 128GB of RAM, and version 12.5 of the CPLEX solver in Windows 7 64-bit environment was used.

The combinatorial BnB method defined by Algorithms 1 and 2 was implemented in two versions, which use the branching and bounding techniques described in Sections 5.1.1 and 5.1.2, and which are henceforth referred to as "BnB 5.1.1" and "BnB 5.1.2", respectively. The computational performance of both

[3] The path length is given by the aggregate weight of vertices that it coincides with.

variants of Algorithm 1–2 was compared with that of the mathematical programming formulation (19) as solved by the CPLEX solver. The results are reported in Tables 1, 2, and 3, where columns with headings "CPLEX", "BnB 5.1.1", and "BnB 5.1.2" contain the results obtained using the respective methods. Ten instances of each problem/graph configuration were generated and the corresponding solution times and objective values were averaged accordingly. A maximum solution time limit of 3600 seconds was imposed and symbol "—" is used to indicate that the time limit was exceeded for all ten instances for the given graph configuration. If only a portion of the instances were solved within the time limit, the number of instances that achieved a solution and their corresponding average solution times are presented.

Table 1 summarizes the computational times for graphs with average edge densities of $d = 0.2$. Observe that both BnB algorithms provide improvement in running time of at least three orders of magnitude on all problem configurations in comparison to the CPLEX IP solver, and the BnB variant based on acyclical path decomposition produces the best results. It must be noted, however, that sparse graphs put the mathematical programming formulation (19) the two-stage stochastic maximum clique problem at a disadvantage, since the employed "edge formulation" of clique constraints is based on the complement of the graph, which results in a large number of constraints (19b)–(19c) when the underlying graph is sparse. At the same time, the proposed general combinatorial BnB algorithm performs better when the depth, i.e., the number of "levels" of the BnB tree is smaller, which is observed on sparse graphs.

Thus, a more fair comparison of the combinatorial and mathematical programming-based schemes can be accomplished when one considers graphs with densities close to $d = 0.5$, see Table 2. It still can be observed, though, that the combinatorial BnB methods drastically outperform the mathematical programming formulation, where the difference is especially evident on instances that could be solved by all three methods, and the branching and bounding rules based on acyclical path decomposition are still superior. At the same time, graphs of density $d = 0.5$ present a greater challenge to the proposed BnB method, as both its variants were unable to solve to optimality larger problems within the allowed time limit. Note that in the cases when all three methods failed to find an optimal solution within 1 hour, the BnB methods report partial solutions with higher objective value.

Computational results for two-stage stochastic maximum clique problem on graphs with average densities of $d = 0.8$ are presented in Table 3. At these densities, the combinatorial BnB methods are generally inferior to the mathematical programming formulation (19), which can be explained by the fact that the number of clique constraints (19b)–(19c) is relatively small for dense graphs, making problem (19) easier to solve, while the depth of the BnB tree increases with the density of the graph, which leads to deteriorated BnB solution times. On the other hand, it can be seen that the combinatorial methods are still preferable when $|V| = 25$, suggesting that the proposed algorithms may be beneficial for dense graphs when the number of scenarios is large relative the number of vertices.

In the case when $d = 0.8$ and $|V| = 100$, both BnB algorithms failed to generate superior objective values in comparison to CPLEX, an obvious deviation from the trend of preceding results for instances of the same density and $|V| = 50, 75$. Empirical evidence suggests that the majority of computational time for these instances was spent on solving the second-stage problems, while very few first-stage solution were processed. This observation indicates that using a second-stage branching and bounding criteria solely based on budgetary restrictions may not be effective for very dense underlying graphs, particularly, once a certain number of vertices is exceeded. Although the present study aims to define a budgetary-based second-stage BnB approach, it is likely that supplemental graph-structural techniques analogous to those presented in Sections 5.1.1 – 5.1.2 would produce superior results; a task that we reserve for future research.

$d = 0.2$		CPLEX			BnB 5.1.1			BnB 5.1.2		
$\|V\|$	N	#	Time (s)	Obj	#	Time (s)	Obj	#	Time (s)	Obj
25	25	10	24.19	6.29	10	0.02	6.29	10	**0.01**	6.29
	50	10	286.08	6.16	10	0.04	6.16	10	**0.01**	6.16
	75	6	308.47	6.14	10	0.04	6.17	10	**0.02**	6.17
50	25	10	279.91	8.17	10	0.27	8.17	10	**0.17**	8.17
	50	5	1066.66	8.09	10	0.60	8.11	10	**0.34**	8.11
	75	0	—	8.07	10	0.90	8.09	10	**0.55**	8.09
75	25	0	—	9.46	10	1.73	9.47	10	**1.22**	9.47
	50	0	—	9.16	10	3.60	9.20	10	**2.40**	9.20
	75	0	—	9.34	10	5.51	9.49	10	**4.93**	9.49
100	25	0	—	9.96	10	7.90	10.04	10	**7.30**	10.04
	50	0	—	9.88	10	16.09	10.02	10	**15.26**	10.02
	75	0	—	9.32	10	23.09	10.05	10	**21.21**	10.05

Table 1: Average solution times (in seconds) and objective values for problem (19) on random graphs with an edge density of 0.2 and $M = \lceil 0.15|V| \rceil$. All running times are averaged over 10 instances and symbol "—" indicates that the time limit of 3600 seconds was exceeded. Columns corresponding to symbol "#" provide the number of instances solved within the time limit.

$d = 0.5$		CPLEX			BnB 5.1.1			BnB 5.1.2		
$\|V\|$	N	#	Time (s)	Obj	#	Time (s)	Obj	#	Time (s)	Obj
25	25	10	7.65	9.92	10	0.17	9.92	10	**0.04**	9.92
	50	10	32.04	9.75	10	0.28	9.75	10	**0.10**	9.75
	75	7	123.81	9.50	10	0.37	9.51	10	**0.13**	9.51
50	25	0	—	13.39	10	37.68	13.43	10	**27.31**	13.43
	50	0	—	14.07	10	74.53	14.11	10	**44.07**	14.11
	75	0	—	13.46	10	93.95	13.72	10	**64.64**	13.72
75	25	0	—	15.84	10	2631.96	16.06	10	**2216.35**	16.06
	50	0	—	15.74	0	—	16.28	0	—	16.30
	75	0	—	15.03	0	—	16.14	0	—	16.31
100	25	0	—	17.13	0	—	17.29	0	—	17.67
	50	0	—	16.61	0	—	17.07	0	—	17.65
	75	0	—	15.92	0	—	17.04	0	—	17.86

Table 2: Average solution times (in seconds) and objective values for problem (19) on random graphs with an edge density of 0.5 and $M = \lceil 0.15|V| \rceil$. All running times are averaged over 10 instances and symbol "—" indicates that the time limit of 3600 seconds was exceeded. Columns corresponding to symbol "#" provide the number of instances solved within the time limit.

6 Conclusions

We have introduced a new class of two-stage stochastic maximum subgraph problems for finding the maximum expected size of a graph that satisfies a defined structural property Π. Emphasis was put on identifying subgraphs whose properties can be restored within a limited repair budget in the presence of structural uncertainties that manifest in the form of random connection (edge) changes/failures. A combinatorial BnB algorithm exploiting the structure of two-stage stochastic maximum Π subgraph problems was developed. Our technique utilizes two combinatorial BnB algorithms for finding optimal

$d = 0.8$		CPLEX			BnB 5.1.1			BnB 5.1.2		
$\|V\|$	N	#	Time (s)	Obj	#	Time (s)	Obj	#	Time (s)	Obj
25	25	10	5.55	15.87	10	3.35	15.87	10	**1.38**	15.87
	50	10	14.40	15.79	10	11.44	15.79	10	**5.43**	15.79
	75	10	51.61	14.96	10	12.57	14.96	10	**7.22**	14.96
50	25	10	**1078.03**	23.97	0	—	23.25	0	—	23.69
	50	6	**3125.09**	23.43	0	—	23.05	0	—	23.33
	75	0	—	22.50	0	—	22.19	0	—	22.85
75	25	0	—	28.07	0	—	26.57	0	—	28.29
	50	0	—	27.57	0	—	27.14	0	—	28.31
	75	0	—	27.19	0	—	26.24	0	—	27.50
100	25	0	—	30.94	0	—	18.49	0	—	18.44
	50	0	—	30.54	0	—	18.41	0	—	18.39
	75	0	—	30.14	0	—	18.35	0	—	18.27

Table 3: Average solution times (in seconds) and objective values for problem (19) on random graphs with an edge density of 0.8 and $M = \lceil 0.15|V| \rceil$. All running times are averaged over 10 instances and symbol "—" indicates that the time limit of 3600 seconds was exceeded. Columns corresponding to symbol "#" provide the number of instances solved within the time limit.

first- and second-stage subgraph solutions.

The proposed framework applies to a broad range of graph properties, and in this work we illustrated the proposed approach on an example when the property of interest Π defines a clique. Numerical simulations on randomly generated graphs indicate that solution times can be reduced by several orders of magnitude via the proposed BnB algorithm in comparison to an equivalent mathematical programming solver. Namely, for all the tested graph configurations other than ones with high edge density of $d = 0.8$, one or more orders of magnitude in performance improvements were observed.

7 Acknowledgements

This research was performed while the first author held a National Research Council Research Associateship Award at the Air Force Research Laboratory. This work was supported in part by the DTRA grant HDTRA1-14-1-0065, AFOSR grant FA9550-12-1-0142, and the U.S. Department of Air Force grant FA8651-14-2-0003. In addition, support by the AFRL Mathematical Modeling and Optimization Institute is gratefully acknowledged.

References

[1] V. E. Alekseev and D. Korobitsyn. Complexity of some problems on hereditary classes of graphs. *Diskretnaya Matematika*, 4(4):34–40, 1992.

[2] N. Alon and A. Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM Journal on Computing*, 37(6):1703–1727, 2008.

[3] A. Atamtrk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.

[4] H.-J. Bandelt and H. M. Mulder. Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B*, 41(2):182 – 208, 1986.

[5] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1997.

[6] V. L. Boginski, C. W. Commander, and T. Turko. Polynomial-time identification of robust network flows under uncertain arc failures. *Optimization Letters*, 3(3):461–473, 2009.

[7] I. Bomze, M. Chimani, M. Jnger, I. Ljubi, P. Mutzel, and B. Zey. Solving two-stage stochastic steiner tree problems by two-stage branch-and-cut. In O. Cheong, K.-Y. Chwa, and K. Park, editors, *Algorithms and Computation*, volume 6506 of *Lecture Notes in Computer Science*, pages 427–439. Springer Berlin Heidelberg, 2010.

[8] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, 2010.

[9] A. M. Campbell and B. W. Thomas. Probabilistic traveling salesman problem with deadlines. *Transportation Science*, 42(1):1–21, 2008.

[10] R. K. Cheung and C.-Y. Chen. A two-stage stochastic network model and solution methods for the dynamic empty container allocation problem. *Transportation Science*, 32(2):142–162, 1998.

[11] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.

[12] E. Fox-Epstein and D. Krizanc. The complexity of minor-ancestral graph properties with forbidden pairs. In E. Hirsch, J. Karhumki, A. Lepist, and M. Prilutskii, editors, *Computer Science Theory and Applications*, volume 7353 of *Lecture Notes in Computer Science*, pages 138–147. Springer Berlin Heidelberg, 2012.

[13] A. Ghosh and S. Boyd. Growing well-connected graphs. In *Decision and Control, 2006 45th IEEE Conference on*, pages 6605–6611, Dec 2006.

[14] G. D. Glockner and G. L. Nemhauser. A dynamic network flow problem with uncertain arc capacities: Formulation and problem structure. *Operations Research*, 48(2):233–242, 2000.

[15] G. Laporte, F. V. Louveaux, and L. van Hamme. An integer l-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50(3):415–423, 2002.

[16] I. Ljubi, P. Mutzel, and B. Zey. Stochastic survivable network design problems. *Electronic Notes in Discrete Mathematics*, 41(0):245 – 252, 2013.

[17] Z. Miao, B. Balasundaram, and E. Pasiliao. An exact algorithm for the maximum probabilistic clique problem. *Journal of Combinatorial Optimization*, 28(1):105–120, 2014.

[18] P. M. Pardalos and J. Xue. The maximum clique problem. *Journal of Global Optimization*, 4:301–328, 1994.

[19] A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, 1995.

[20] N. Robertson and P. Seymour. Graph minors. i. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39 – 61, 1983.

[21] N. Robertson and P. Seymour. Graph minors. xix. well-quasi-ordering on a surface. *Journal of Combinatorial Theory, Series B*, 90(2):325 – 385, 2004.

[22] M. Rysz, M. Mirghorbani, P. Krokhmal, and E. Pasiliao. On risk-averse maximum weighted subgraph problems. *Journal of Combinatorial Optimization*, 28(1):167–185, 2014.

[23] E. Tomita and T. Seki. An efficient branch-and-bound algorithm for finding a maximum clique. In C. Calude, M. Dinneen, and V. Vajnovszki, editors, *Discrete Mathematics and Theoretical Computer Science*, volume 2731 of *Lecture Notes in Computer Science*, pages 278–289. Springer Berlin Heidelberg, 2003.

[24] P. Tsiakis, N. Shah, and C. C. Pantelides. Design of multi-echelon supply chain networks under demand uncertainty. *Industrial & Engineering Chemistry Research*, 40(16):3585–3604, 2001.

[25] A. Veremyev and V. Boginski. Identifying large robust network clusters via new compact formulations of maximum k-club problems. *European Journal of Operational Research*, 218(2):316 – 326, 2012.

[26] B. Verweij, S. Ahmed, A. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications*, 24(2-3):289–333, 2003.

[27] S. Voccia, A. Campbell, and B. Thomas. The probabilistic traveling salesman problem with time windows. *EURO Journal on Transportation and Logistics*, 2(1-2):89–107, 2013.

[28] J.-W. Wang and L.-L. Rong. Robustness of the western united states power grid under edge attack strategies due to cascading failures. *Safety Science*, 49(6):807 – 812, 2011.

[29] K. Yamaguchi and S. Masuda. A new exact algorithm for the maximum weight clique problem. In *23rd International Conference on Circuits/Systems, Computers and Communictions (ITC-CSCC08)*, 2008.

[30] M. Yannakakis. Node-and edge-deletion np-complete problems. In *STOC'78: Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 253–264, New York, 1978. ACM Press.

[31] O. Yezerska, S. Butenko, and V. Boginski. Detecting robust cliques in the graphs subject to uncertain edge failures. *Working paper*.

Identifying risk-averse low-diameter clusters in graphs with stochastic vertex weights

Maciej Rysz[1] Foad Mahdavi Pajouh[2] Pavlo Krokhmal[3]* Eduardo L. Pasiliao[4]

[1]National Research Council
Air Force Research Labratory
101 West Eglin Blvd, Eglin AFB, FL 32542
E-mail: mwrysz@yahoo.com

[2]Management Science and Information Systems Department
University of Massachusetts Boston
100 Morrissey Blvd., Boston, MA 02125
E-mail: foad.mahdavi@umb.edu

[3]Department of Systems and Industrial Engineering
University of Arizona, 1127 E James E. Rogers Way, Tucson, AZ 85721
E-mail: krokhmal@email.arizona.edu

[4]Munitions Directorate
Air Force Research Laboratory
101 West Eglin Blvd, Eglin AFB, FL 32542
E-mail: eduardo.pasiliao@eglin.af.mil

Abstract

In this work, we study the problem of detecting risk-averse low-diameter clusters in graphs. It is assumed that the clusters represent k-clubs and that uncertain information manifests itself in the form of stochastic vertex weights whose joint distribution is known. The goal is to find a k-club of minimum risk contained in the graph. A stochastic programming framework that is based on the formalism of coherent risk measures is used to quantify the risk of a cluster. We show that the selected representation of risk guarantees that the optimal subgraphs are maximal clusters. A combinatorial branch-and-bound algorithm is proposed and its computational performance is compared with an equivalent mathematical programming approach for instances with $k = 2, 3$, and 4.

Keywords: k-club; low-diameter clusters; stochastic graphs; coherent risk measures; combinatorial branch-and-bound

*Corresponding author

1 Introduction

Graphs are effective tools for modeling many real-world systems and the complex interactions between their components. A typical graph model assigns vertices to represent a system's components and a set of edges to describe the connections and/or relationships between them. Well-known examples of such frameworks are represented by many systems studied in social network analysis, transportation, telecommunications, computational finance, and so on. Additionally, graph-based data mining methods [18] provide powerful techniques for analyzing and understanding systems whose descriptive data may be represented using a graph.

A principal application of graph-based data mining involves the identification of subgraphs, referred to as clusters, corresponding to subsystems with a given structural or functional property. For example, in social networks, detecting highly-connected clusters can be used for advertising and marketing purposes [22, 23, 46]; in stock market graphs, it can be used for identifying diverse portfolios [12]; and in call graphs, it can be used for detecting communicating clusters [1].

One of the basic problems in this context entails finding the largest "perfectly" cohesive group within a network such that the confined members are all interconnected, also known as the maximum clique (complete subgraph) problem. Several prominent studies provided the basis for exact combinatorial solution algorithms for the maximum clique problem [8, 16, 33]. In particular, Carraghan and Pardalos [16] introduced a recursive branch-and-bound method for finding maximum cliques by exploiting the heredity property [42] of complete subgraphs. Subsequent extensions of their work enhanced the process of reducing solution space via vertex coloring schemes for estimation of upper bounds on the maximum achievable subgraph sizes during the branch-and-bound procedure (e.g. [15, 24, 41]).

In many practical applications, the requirement that the desired subgraph must be complete may, however, impose excessive restrictions and therefore warrant some structural relaxation in terms of cluster connectivity. As a consequence, a number of clique relaxation models have been proposed in graph theory literature, which relax the completeness property relative to the degree of the member vertices, their distance from each other, or the density of the subgraph. A comprehensive review of clique relaxation models is provided in [10]. In the present work, we focus on a specific type of clique relaxation, known as the *k-club* [3], which represents a subgraph whose members are connected via at most $k - 1$ intermediary members. The k-club model effectively represents low-diameter clusters that may reveal valuable information embedded in social, financial, and telecommunication networks. Several recent studies proposed combinatorial branch-and-bound methods and presented complexity results associated with finding maximum k-clubs in graphs [13, 17, 34, 39].

An important extension of the described class of problems involves the imposition of topologically exogenous information in the form of deterministic vertex weights, and correspondingly finding a subset of maximum weight that conforms to a defined structural property. Similar exact weight-based branch-and-bound solution techniques have been developed for determining the maximum-weight subgraphs [7, 28, 32].

Numerous circumstances may further justify the imposition of uncertain exogenous information over the graph's edges that influences network flow distribution, robustness, and costs [4, 6, 14, 20, 21, 44]. However, far fewer endeavors consider decision making relative to the optimal allocation of resources over defined subgraph topologies when uncertainties are induced by stochastic factors associated with network vertices [38]. For example, in social networks or call graphs, the uncertainties related to the value or reliability of the information provided by each entity can be modeled by random weights on vertices whose relationships or communications are presented by edges. Similarly, in stock market graphs, the uncertainties associated with returns on investments from different assets can be defined as random weights assigned to their corresponding vertices, with edges linking highly correlated assets (vertices).

In this study, we extend the techniques introduced in [38] to address problems of finding subgraphs of minimum risk that represent a k-club. A probabilistic framework utilizing the distributional information of stochastic vertex weights by means of *coherent measures of risk* [5, 19] is employed to define a *risk-averse k-club* (RA-k) problem as finding the lowest risk k-club in a network. As an illustrative example, we focus on instances when $k = 2, 3, 4$, and utilize a mathematical programming formulation introduced in [43] for finding a maximum k-club in a graph. A combinatorial branch-and-bound method for finding a largest k-club [13, 17, 34] is also modified to accommodate the conditions of RA-k problem via risk-based branching and bounding schema. We compare the solution performance of the proposed branch-and-bound algorithm relative to solving the mathematical programming formulation for the RA-k problem using a state-of-the-art commercial solver.

The remainder of the paper is organized as follows. In Section 2, we examine the general representation of RA-k problem and discuss its properties. Section 3 presents a mathematical programming formulation and a combinatorial branch-and-bound method for solving the RA-k problem. Finally, Section 4 furnishes numerical studies demonstrating the computational performances of the developed branch-and-bound method and the aforementioned mathematical programming approach on problems where risk is quantified using higher-moment coherent risk measures [27].

2 Risk-averse k-club problem

Given an undirected graph $G = (V, E)$ and any subset of its vertices $S \subseteq V$, let $G[S]$ represent the subgraph of G induced by S, i.e., $G[S] = (S, E \cap (S \times S))$. Let $\mathcal{Q}$ denote the desired property which the induced graph $G[S]$ must satisfy. The present work considers the case when $\mathcal{Q}$ represents a certain relaxation of the *completeness* property, such that a subgraph with property $\mathcal{Q}$ represents a *clique relaxation*.

Depending on the characteristic of a complete graph that is relaxed, clique relaxations can be categorized into *density-based* [1, 2, 35], *degree-based* [40], and *distance-based* [3, 29, 30] relaxations. In this work, property $\mathcal{Q}$ represents a special distance-based relaxation of the completeness property. For a formal definition, let $d_G(i, j)$ denote the distance between nodes $i, j \in V$ in graph G, measured as the number of edges in a shortest path between i and j in G. Then, a subset of vertices $S \subseteq V$ of graph G is called a *k-clique* if

$$\max_{i,j \in S} d_G(i, j) \leq k.$$

Note that the definition of the k-clique does not require that the shortest path between $i, j \in S$ belong to $G[S]$. If one requires that the shortest path between any two vertices i, j in S belong to the induced subgraph $G[S]$, then the subset S such that

$$\max_{i,j \in S} d_{G[S]}(i, j) \leq k, \tag{1}$$

is called a *k-club*. Note that a k-club is also a k-clique, while the inverse is not true in general. By definition, 1-cliques and 1-clubs are cliques. Throughout the remainder of this study, we let $\Gamma_G(k)$ denote the set of all k-clubs in graph G:

$$\Gamma_G(k) = \big\{S \subseteq V : d_{G[S]}(i, j) \leq k, \ \forall i, j \in S\big\}. \tag{2}$$

Additionally, a k-club is said to be maximal, if it is not strictly contained in another k-club; and a maximum k-club is a k-club of the largest order in graph G.

A popular class of graph-theoretical problems is represented by the *maximum weight subgraph* problems, which are concerned with finding a subset S of vertices in G such that the induced subgraph satisfies the given property $\mathcal{Q}$ and has the largest weight (defined as the sum of its vertices' weights). The *maximum weight k-club* problem is then formulated as

$$\max\Big\{\sum_{i\in S} w_i : S \in \Gamma_G(k)\Big\}, \tag{3}$$

where $w_i \geq 0$ represents the weight of vertex i and the set $\Gamma_G(k)$ is defined by (2). Clearly, an optimal set S in problem (3) will be *maximal*, but not necessarily *maximum* (of the largest order) set with property $\mathcal{Q}$. If the weight of each vertex is one, the maximum weighted k-club problem is simply referred to as the maximum k-club problem.

In this work, we consider an extension of problem (3) that assumes stochastic vertex weights. In this case, a direct translation into a stochastic framework is not straightforward due to the fact that the maximization of random weights would be ill-posed in context of stochastic programming resulting from the absence of a deterministic optimal solution. Likewise, maximization of the expected weight of the sought set is rather uninteresting in the sense that it reduces to the deterministic version of the problem presented above. A more suitable approach, thus, involves computing the subgraph's weight via a (nonlinear) statistical function that utilizes the distributional information about the weights' uncertainties, rather than a simple sum of its vertices' stochastic weights. In particular, we pursue a *risk-averse* approach so as to find the subgraph of G that has the *lowest risk* and satisfies property $\mathcal{Q}$. Let X_i denote a random variable that represents the costs of losses associated with vertex $i \in V$, such that the joint distribution of vector $\mathbf{X}_G = (X_1, \ldots, X_{|V|})$ is known. Then, the problem of finding the *minimum risk* subgraph in G that has property $\mathcal{Q}$, or the *risk-averse $\mathcal{Q}$ problem* takes the form:

$$\min\big\{\mathscr{R}(S;\mathbf{X}_G) : S \subseteq V \text{ and } G[S] \text{ satisfies } \mathcal{Q}\big\}, \tag{4}$$

where $\mathscr{R}(S;\mathbf{X}_G)$ is the risk associated with set S given the distributional information $\mathbf{X}_G$. In the particular case when property $\mathcal{Q}$ ensures that the subgraph in question is a k-club, formulation (4) defines the *risk-averse k-club problem* (RA-k),

$$\min\big\{\mathscr{R}(S;\mathbf{X}_G) : S \in \Gamma_G(k)\big\}, \tag{5}$$

which represents a risk-averse stochastic generalization of the deterministic maximum weight k-club problem (1), as shown below.

A constructive form of risk function $\mathscr{R}(S;\mathbf{X}_G)$ can be introduced by employing the well-known in stochastic optimization literature concept of *risk measure* [26]. Given a probability space $(\Omega, \mathcal{F}, \mathsf{P})$, where Ω is the set of random events, $\mathcal{F}$ is the σ-algebra, and P is a probability measure, a risk measure ρ is defined as a mapping $\rho : \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X}$ is a linear space of $\mathcal{F}$-measurable functions $X : \Omega \mapsto \mathbb{R}$. In what follows, the space $\mathcal{X}$ is assumed to possess the properties necessary for the risk measures introduced below to be well-defined. Namely, $\mathcal{X}$ is supposed to allow for a sufficient degree of integrability, in particular, $\mathsf{E}|X| < \infty$, and be endowed with an appropriate topology, e.g., the topology induced by convergence in probability. Lastly, we consider risk measures that are *proper* functions on $\mathcal{X}$, i.e., $\rho(X) > -\infty$ for all $X \in \mathcal{X}$ and $\{X \in \mathcal{X} : \rho(X) < \infty\} \neq \emptyset$.

Then, assuming that risk measure ρ is lower semi-continuous (l.s.c.), the risk $\mathscr{R}(S;\mathbf{X}_G)$ of a set $S \subseteq V$ with uncertain vertex weights X_i, $i \in V$, can be defined as the optimal value of the following stochastic programming problem:

$$\mathscr{R}(S;\mathbf{X}_G) = \min\Big\{\rho\Big(\sum_{i\in S} u_i X_i\Big) : \sum_{i\in S} u_i = 1;\ u_i \geq 0,\ i \in S\Big\}. \tag{6}$$

Note that this definition of the set risk function $\mathscr{R}(\cdot)$ admits risk reduction through diversification as illustrated by the following proposition:

Proposition 1 ([38]) *Given a graph $G = (V, E)$ with stochastic weights X_i, $i \in V$, and a l.s.c. risk measure ρ, the set risk function $\mathscr{R}$ defined by* (6) *satisfies*

$$\mathscr{R}(S_2; \mathbf{X}_G) \le \mathscr{R}(S_1; \mathbf{X}_G) \quad \textit{for all} \quad S_1 \subseteq S_2. \tag{7}$$

The following observation regarding the optimal solution of the risk-averse $\mathcal{Q}$ problem (4) stems directly from property (7):

Corollary 1 *There exists an optimal solution of the risk-averse $\mathcal{Q}$ problem* (4) *with $\mathscr{R}(S; \mathbf{X}_G)$ defined by* (6) *that is a maximal set with property $\mathcal{Q}$ in G.*

Additional properties of $\mathscr{R}(S; \mathbf{X}_G)$ as defined by (6) ensue from the assumption that the risk measure ρ belongs to the family of coherent measures of risk [5], i.e., satisfies the properties of monotonicity, $\rho(X) \le 0$ for all $X \le 0$; subadditivity, $\rho(X + Y) \le \rho(X) + \rho(Y)$; transitional invariance, $\rho(X + c) = \rho(X) + c$ for all $c \in \mathbb{R}$; and positive homogeneity, $\rho(\lambda X) = \lambda\rho(X)$ for all $\lambda > 0$. Then, the corresponding set risk function $\mathscr{R}(S; \mathbf{X}_G)$ satisfies analogous properties with respect to the stochastic weights vector $\mathbf{X}_G$,

(G1) *monotonicity*: $\mathscr{R}(S; \mathbf{X}_G) \le \mathscr{R}(S; \mathbf{Y}_G)$ for all $\mathbf{X}_G \le \mathbf{Y}_G$;

(G2) *positive homogeneity*: $\mathscr{R}(S; \lambda\mathbf{X}_G) = \lambda\mathscr{R}(S; \mathbf{X}_G)$ for all $\mathbf{X}_G$ and $\lambda > 0$;

(G3) *transitional invariance*: $\mathscr{R}(S; \mathbf{X}_G + a\mathbf{1}) = \mathscr{R}(S; \mathbf{X}_G) + a$ for all $a \in \mathbb{R}$;

where $\mathbf{1}$ is the vector of ones, and the vector inequality $\mathbf{X}_G \le \mathbf{Y}_G$ is interpreted component-wise.

Observe that $\mathscr{R}(S; \mathbf{X}_G)$ violates in general the subadditivity requirements with respect to the stochastic weights. However, risk reduction via diversification is guaranteed by (7), which ensures that the inclusion of additional vertices to the existing feasible solution is always beneficial. Further, under the assumption of nonnegative stochastic vertex weights, $\mathbf{X}_G \ge \mathbf{0}$, the set risk $\mathscr{R}(S; \mathbf{X}_G)$ can be shown to be subadditive with respect to subsets of V,

$$\mathscr{R}(S_1 \cup S_2; \mathbf{X}_G) \le \mathscr{R}(S_1; \mathbf{X}_G) + \mathscr{R}(S_2; \mathbf{X}_G), \quad S_1, S_2 \subseteq V. \tag{8}$$

Clearly, it is required that S_1, S_2, and $S_1 \cup S_2$ satisfy property $\mathcal{Q}$ in conformance to the context of risk-averse $\mathcal{Q}$ problems.

3 Solution approaches for risk-averse k-club problems

In this section, we first address the computational complexity of the RA-k problem for any fixed positive integer k, and show that this problem is $\mathcal{NP}$-hard. We then propose two exact solution algorithms for this problem. First, we consider a mathematical programming approach for the RA-k problem, where the risk $\mathscr{R}(S; \mathbf{X}_G)$ of a set $S \in \Gamma_G(k)$ is defined by (6). To this end, we take advantage of a recent formulation for the maximum k-club problem developed by Veremyev et al. [43]. Next, we propose a combinatorial branch-and-bound algorithm for solving RA-k problem that utilizes the same solution space processing principles for finding maximum k-clubs as the ones used in [13, 17, 34].

In order to establish the problem's complexity and derive the corresponding solution methods, we need to introduce additional assumptions on the properties of stochastic weights $\mathbf{X}_G$ and risk measure ρ involved in the definition of the risk-averse k-club problem (5). Namely, throughout this section it is assumed that the stochastic weights X_i of vertices $i \in V$ are nonnegative and rational-valued, $X_i : \Omega \mapsto \mathbb{Q}_+$, $i \in V$, where $\mathbb{Q}_+$ denotes the set of nonnegative rational numbers. Also, the corresponding probability measure P is rational-valued, i.e., $\mathsf{P}\{X_i = X_i(\omega)\} \in \mathbb{Q}_+ \cap [0, 1]$ for all $\omega \in \Omega$ and all $i \in V$. Similarly, we assume that the risk measure ρ is such that $\rho(X) \in \mathbb{Q}$ whenever X and the underlying probability measure are rational-valued. In addition, we restrict our attention to risk measures that are *expectation-bounded*[1] [37], i.e., such that $\rho(X) > \mathsf{E}X$ for all non-constant X, and $\rho(X) = \mathsf{E}X$ for all constant X, or such X that $X = \text{const}$ with probability 1.

3.1 Computational complexity

In this section, we derive the computational complexity of the the risk-averse k-club problem from the complexity of the more general class of risk-averse $\mathcal{Q}$ problems (4).

For a given property $\mathcal{Q}$, the decision version of risk-averse $\mathcal{Q}$ problem, denoted by $\langle G, \mathbf{X}_G, \rho, c\rangle$, is as follows. Given a graph $G = (V, E)$, a vector of stochastic weights $\mathbf{X}_G$, a l.s.c. risk measure ρ, and a $c \in \mathbb{Q}$, determine whether there exists a set $S \subseteq V$ such that $G[S]$ satisfies $\mathcal{Q}$ and $\mathscr{R}(S; \mathbf{X}_G) < c$. We also consider the deterministic maximum $\mathcal{Q}$ problem:

$$\max\{|S| : S \subseteq V \text{ and } G[S] \text{ satisfies } \mathcal{Q}\}, \tag{9}$$

and its decision version, denoted as $\langle G, q\rangle$: given a graph $G = (V, E)$ and an integer q, is there a subset of V that has property $\mathcal{Q}$ and order larger than q?

Theorem 1 *If property $\mathcal{Q}$ is such that the decision version of (deterministic) maximum $\mathcal{Q}$ problem is $\mathcal{NP}$-hard, then the decision version of risk-averse Q problem is also $\mathcal{NP}$-hard, provided that the risk measure ρ is proper, l.s.c., and expectation-bounded.*

Proof: The intractability of the risk-averse $\mathcal{Q}$ problem is proved by a polynomial-time reduction from the maximum $\mathcal{Q}$ problem. Given a graph $G = (V, E)$ and a fixed positive integer q, consider the decision version of the maximum $\mathcal{Q}$ problem $\langle G, q\rangle$. For any such maximum $\mathcal{Q}$ decision problem $\langle G, q\rangle$, we replicate $\hat{G} = G$ and let $\hat{X}_i$ for all $i \in V$ be a set of independently and identically distributed random variables with Bernoulli distribution, such that $\mathsf{P}\{\hat{X}_i = 0\} = \mathsf{P}\{\hat{X}_i = 1\} = \frac{1}{2}$ for all $i \in V$. As a risk measure, we select $\hat{\rho}(X) = \sigma^2(X) + \mathsf{E}X$, where $\sigma^2(X)$ denotes the variance of X. Obviously, $\hat{\rho}(X)$ is expectation bounded, as well as l.s.c. and proper, so that the corresponding set risk function $\mathscr{R}$ is well defined. It is easy to see that the set risk function $\mathscr{R}(S, \hat{\mathbf{X}}_{\hat{G}})$ becomes equal to

$$\begin{aligned}\mathscr{R}(S, \hat{X}_{\hat{G}}) &= \min\Big\{\sigma^2\Big(\sum_{i\in S} u_i \hat{X}_i\Big) + \frac{1}{2} : \sum_{i\in S} u_i = 1;\ u_i \geq 0,\ \forall i \in S\Big\}\\ &= \frac{1}{4|S|} + \frac{1}{2}.\end{aligned}$$

This procedure constructs in polynomial time an instance $\langle G, \hat{\mathbf{X}}_G, \hat{\rho}, \frac{1}{4q} + \frac{1}{2}\rangle$ of risk-averse $\mathcal{Q}$ problem such that there exists a $\mathcal{Q}$-subgraph of order larger than q in G if and only if there exists a $\mathcal{Q}$-subgraph

[1] "Expectation-boundedness" is also known as "aversity" [36], but we use the former term in this work so as to avoid semantic confusion when referring to "risk-averse" subgraphs.

S in G such that $\mathscr{R}(S;\hat{\mathbf{X}}_G) < \frac{1}{4q} + \frac{1}{2}$. This shows that the decision version of risk-averse $\mathcal{Q}$ problem is $\mathcal{NP}$-hard if the maximum $\mathcal{Q}$ problem is $\mathcal{NP}$-hard. □

The computational complexity of RA-k problem, which we are concerned with in this work, follows readily from Theorem 1 due to the fact that (deterministic) maximum k-club problem is known to be $\mathcal{NP}$-hard [9]:

Corollary 2 *The decision version of risk-averse k-club problem (RA-k) is $\mathcal{NP}$-hard, provided that risk measure ρ is proper, l.s.c., and expectation-bounded.*

The condition that risk measure ρ in the risk-averse $\mathcal{Q}$ problem (4) be l.s.c. and proper ensures that the resulting set risk function $\mathscr{R}$ is well-defined. Expectation-boundedness, on the other hand, is imposed so as to avoid situations in which the risk-averse $\mathcal{Q}$ problem becomes trivial. In the presented framework we advocate for use of coherent measures of risk when constructing the set risk function (6). It turns out, however, that if one selects $\rho(X) = \mathsf{E}X$, which is formally a coherent risk measure yet does not measure "risk", then the corresponding problem (4) is polynomially solvable, and, moreover, the solution is trivial. This can be viewed as an additional supporting argument for pursuing the *risk-averse* approach when dealing with graph-theoretical problems on graphs with stochastic vertex weights, since the traditional "expectation"-based, or risk-neutral approach to problems with stochastic vertex weights may not yield interesting results. The following proposition formalizes the above observation.

Proposition 2 *Consider the risk-averse $\mathcal{Q}$ problem* (4)*, where the risk measure ρ is such that for any $G = (V, E)$, $\mathbf{X}_G$, and $S \subseteq V$,*

$$\begin{aligned} &\arg\min \left\{ \rho\Bigg(\sum_{i\in S} u_i X_i \Bigg) : \sum_{i\in S} u_i = 1;\ u_i \geq 0,\ \ \forall i \in S \right\} \\ &\quad = \left\{ \mathbf{u} \in \mathbb{R}^{|S|} : u_{i_S} = 1;\ u_i = 0,\ \ \forall i \in S \setminus \{i_S\} \right\}, \end{aligned} \tag{10}$$

and i_S in (10) *is computable in polynomial time. Then, the risk-averse $\mathcal{Q}$ problem is polynomially solvable, provided that property $\mathcal{Q}$ is such that one can determine in polynomial time whether there exists a $\mathcal{Q}$-subgraph of G containing a given $i \in V$.*

Proof: Obviously, condition (10) implies that

$$\mathscr{R}(S, \mathbf{X}_G) = \rho(X_{i_S}) = \min_{i\in S} \rho(X_i).$$

Then, in polynomial time one can compute $\rho(X_{i_0}) = \min_{i\in V} \rho(X_i)$ and it can be verified whether $S_0 \ni i_0$ exists such that $G[S_0]$ satisfies $\mathcal{Q}$. If not, $\rho(i_1) = \min_{i\in V\setminus\{i_0\}} \rho(X_i)$ is computed and existence of $S_1 \ni i_1$ such that $G[S_1]$ satisfies $\mathcal{Q}$ is verified in polynomial time, and so on. Clearly, the risk-averse $\mathcal{Q}$ problem can thus be solved in polynomial time. □

It is easy to see that $\rho(X) = \mathsf{E}X$ constitutes a special case of the risk measure described in Proposition 2, and

$$\mathscr{R}(S, \mathbf{X}_G) = \min_{i\in S} \mathsf{E}X_i.$$

On a related note, Theorem 1 also establishes the computation complexity of risk-averse maximum hereditary subgraph problems that were discussed in our previous work [38]. Recall that property $\mathcal{Q}$ is called *hereditary with respect to induced subgraphs* if for any graph G that satisfies $\mathcal{Q}$, removal of any its vertex creates an induced subgraph that also satisfies $\mathcal{Q}$. Further, property $\mathcal{Q}$ is called *interesting* if the order of graphs that satisfy it is unbounded, and it is called *nontrivial* if it is satisfied by a single-vertex graph and is not satisfied by every graph (see, e.g., [47]).

Corollary 3 *If property $\mathcal{Q}$ is hereditary with respect to induced subgraphs, interesting, and nontrivial, and risk measure ρ is l.s.c., proper, and expectation-bounded, then the risk-averse $\mathcal{Q}$ problem is $\mathcal{NP}$-hard.*

Note that the k-club property is not hereditary with respect to induced subgraphs.

3.2 A mathematical programming formulation

In this section, we formulate the RA-k problem as a (generally nonlinear) mixed integer programming program. To this effect, let binary decision variables x_i indicate whether node $i \in V$ belongs to a subset S:

$$x_i = \begin{cases} 1, & i \in S \\ 0, & \text{otherwise.} \end{cases}$$

When the property $\mathcal{Q}$ denotes a k-club, one can choose the *edge formulation* of the maximum k-club problem proposed by Veremyev et al. [43], whereby the mathematical programming formulation of the RA-k problem takes the form

$$\min \quad \rho\Big(\sum_{i \in V} u_i \, X_i\Big) \tag{11a}$$

$$\text{s. t.} \quad \sum_{i \in V} u_i = 1, \tag{11b}$$

$$u_i \le x_i, \quad i \in V, \tag{11c}$$

$$y_{ij}^{(k)} \ge x_i + x_j - 1, \quad \forall i, j \in V, \; i \ne j, \tag{11d}$$

$$y_{ij}^{(1)} = 0, \quad \forall (i,j) \in \overline{E}, \; i \ne j, \tag{11e}$$

$$y_{ij}^{(l)} = y_{ij}^{(1)}, \quad \forall (i,j) \in E, \; l \in \{2, \ldots, k\}, \tag{11f}$$

$$y_{ij}^{(l)} \le \sum_{t:(i,t) \in E} y_{tj}^{(l-1)}, \quad \forall (i,j) \in \overline{E}, \; l \in \{2, \ldots, k\}, \tag{11g}$$

$$y_{ij}^{(l)} \le x_i, \;\; y_{ij}^{(l)} \le x_j, \;\; y_{ij}^{(l)} = y_{ji}^{(l)}, \quad \forall i, j \in V, \; l \in \{1, \ldots, k\}, \tag{11h}$$

$$x_i \in \{0, 1\}, \;\; u_i \ge 0, \;\; y_{ij}^{(l)} \in [0, 1], \quad \forall i, j \in V, \; l \in \{1, \ldots, k\}, \tag{11i}$$

where $\overline{E}$ represents the set of all complement edges of graph G. Note that nonlinearity in (11) is attributable to the possible nonlinearity of the risk measure ρ. Appropriate nonlinear mixed-integer programming solvers can be used to solve formulation (11) provided that risk measure ρ in admits a suitable mathematical programming representation. A combinatorial branch-and-bound algorithm for solving RA-k problem is described next.

3.3 A combinatorial branch-and-bound algorithm

The combinatorial branch-and-bound (BnB) algorithm for solving problem (11) processes solution space by traversing "levels" of the BnB tree to find a subgraph $G[S]$ that represents a maximal k-club of minimum risk in G as measured by (6). The algorithm begins at level $\ell = 0$ with a partial solution $Q := \emptyset$, incumbent solution $Q^* := \emptyset$, and an upper bound on risk $L^* := +\infty$ (risk induced by Q^*). Partial solution Q is composed of vertices that may potentially become a k-club during latter stages of the algorithm, while Q^* contains vertices corresponding to a maximal k-club whose risk, L^*, is the smallest up to the current stage. A set of "candidate" vertices C_ℓ is maintained at each level ℓ, from which a certain *branching* vertex v_ℓ is selected and added to the partial solution Q, or simply deleted from set C_ℓ without being added to Q. Note that the initial candidate set is $C_0 := V$. To ensure proper navigation between the levels of the BnB tree, the notation P_ℓ^+ or P_ℓ^- is used to indicate whether the last node of the BnB tree at level ℓ was created by adding v_ℓ to Q, or by deleting v_ℓ from C_ℓ without adding it to Q, respectively.

Whenever a BnB tree node is created at the consecutive level $\ell + 1$, a candidate set $C_{\ell+1}$ is constructed by removing all vertices from C_ℓ whose pairwise distances from the vertices in Q exceed k in the induced graph $G[Q \cup C_\ell]$:

$$C_{\ell+1} := \big\{ j \in C_\ell : d_{G[Q \cup C_\ell]}(i, j) \le k, \ \forall i \in Q \big\}.$$

Observe that the refinement of C_ℓ may disrupt the structural integrity of the partial solution if the eliminated candidate vertices serve as distance intermediaries (i.e., comprise the shortest paths) between the vertices in Q. In other words, the distance between at least one pair of vertices $i, j \in Q$ exceeds k upon removal of one or more vertices from C_ℓ when constructing $C_{\ell+1}$. Due to this inherent distance-based dependence of k-clubs, additional considerations are warranted whenever creating a BnB node by either adding or deleting a vertex v_ℓ (i.e., P_ℓ^+ or P_ℓ^-, respectively). Therefore, the necessary structural properties of Q and $C_{\ell+1}$ at each BnB node are

(C1) Q is a k-clique in $G[Q \cup C_{\ell+1}]$, and

(C2) $d_{G[Q \cup C_\ell]}(i, j) \le k, \ \ \forall i \in Q, \ \ \forall j \in C_{\ell+1}$.

After constructing set $C_{\ell+1}$ (condition (C2) is satisfied by definition of $C_{\ell+1}$), if vertices in $C_\ell \setminus C_{\ell+1}$ do serve as distance intermediaries, their removal imposes violations with respect to condition (C1). In such cases, Q cannot become a k-club by exploring deeper levels of the tree and the corresponding node of the BnB tree is fathomed[2] by infeasibility via violation of condition (C1).

Whenever condition (C1) is satisfied, the next step entails evaluating the quality of the solution that can be obtained from the subgraph induced by vertices in $Q \cup C_{\ell+1}$. An exact approach for directly finding a k-club with the lowest possible risk contained in $G[Q \cup C_{\ell+1}]$ would involve solving problem (11) with $x_i = 0$ for all $i \in V \setminus (Q \cup C_{\ell+1})$; we denote the corresponding solution by $\mathcal{S}(Q \cup C_{\ell+1}; \mathbf{X}_G)$. Solving such a (nonlinear) mixed 0–1 program at every node of the BnB tree is clearly impractical. Instead, the following relaxation problem is utilized to obtain a valid lower bound on $\mathcal{S}(Q \cup C_{\ell+1}; \mathbf{X}_G)$:

$$\begin{aligned} \mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G) := \min \quad & \rho\Big(\sum_{i \in V} u_i \, X_i \Big) \\ \text{s. t.} \quad & \sum_{i \in V} u_i = 1, \\ & u_i = 0, \quad i \in V \setminus (Q \cup C_{\ell+1}) \\ & u_i \ge 0, \quad i \in Q \cup C_{\ell+1}. \end{aligned} \tag{12}$$

If $\mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G) \ge L^*$, then the corresponding node of the BnB tree is fathomed by bound due to the fact that sequential refinement can not achieve a further reduction in risk.

In the case when $\mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G) < L^*$ and $Q \cup C_{l+1}$ is a k-club, the new incumbent solution will be $Q^* = Q \cup C_{l+1}$ and the global upper bound on risk is updated, $L^* = \mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G)$. In

[2] Indicated in Algorithm 1 by the assignment "fathom := True".

this case, the current BnB node is fathomed by feasibility. If, however, $\mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G) < L^*$ and $Q \cup C_{l+1}$ is not a k-club, a branching vertex $v_{\ell+1}$ is selected at the next level $\ell + 1$ and BnB node $P^+_{\ell+1}$ will be processed.

After fathoming a BnB node, the algorithm backtracks as follows. If the current BnB node is of type P^+_ℓ, then the vertex v_ℓ is removed from Q, and the node associated with the deletion of v_ℓ, P^-_ℓ, is created. On the other hand, if the BnB node is of type P^-_ℓ, the algorithm sequentially backtracks to the last level, $\ell' < \ell$, associated with a node of type $P^+_{\ell'}$. The node $P^-_{\ell'}$ is then constructed by removing the branching vertex $v_{\ell'}$ from Q. Observe that a node can only be of form P^-_ℓ, after P^+_ℓ has been fathomed/processed.

Empirical observations suggest that branching on a vertex v_ℓ with the smallest value of $\rho(X_{v_\ell})$ or $\mathsf{E}X_{v_\ell}$ can significantly enhance computational performance. To this end, the vertices in any candidate set C_ℓ are ordered in descending order with respect to their risks $\rho(X_i)$ or expected values $\mathsf{E}X_i$, and the last vertex in C_ℓ is always selected when adding vertex v_ℓ to the partial solution Q. The described branch-and-bound algorithm procedure for RA-k problem is formalized in Algorithm 1.

As shown in [17], it is important to mention that the number of leaf nodes in the BnB search tree of Algorithm 1 is $O^*(1.62^{|V|})$, where the modified notation "$O^*\big(g(|V|)\big)$" implies $O\big(g(|V|) \cdot poly(|V|)\big)$ for some polynomial function $poly(|V|)$. Additionally, at each node of the search tree, all pair distances can be computed in $O(|V|^3)$ time and we solve a linear program to obtain a lower bound on the optimal solution of the subtree rooted at that node. Therefore, Algorithm 1 runs in $O^*(1.62^{|V|})$.

Algorithm 1: Combinatorial branch-and-bound algorithm

```
Initialize: $\ell := 0$; $C_0 := V$; $Q := \emptyset$; $Q^* := \emptyset$; $L^* = \infty$; node := $P_0^+$; fathom := False;
while $\ell \geq 0$ do
    if node $= P_\ell^+$ then
        select a vertex $v_\ell \in C_\ell$;
        $C_\ell := C_\ell \setminus \{v_\ell\}$;
        $Q := Q \cup \{v_\ell\}$;
    else
        $Q := Q \setminus \{v_\ell\}$;
    $C_{\ell+1} := \{j \in C_\ell : d_{G[Q \cup C_\ell]}(i, j) \leq k, \ \forall i \in Q\}$;
    if $Q$ is a $k$-clique in $G[Q \cup C_{\ell+1}]$ then
        if $\mathcal{L}(Q \cup C_{\ell+1}) < L^*$ then
            if $Q \cup C_{\ell+1}$ is a $k$-club then
                $Q^* := Q \cup C_{\ell+1}$;
                $L^* := \mathcal{L}(Q \cup C_{\ell+1})$;
                fathom := True;
        else
            fathom := True;
    else
        fathom := True;
    if fathom = True then
        while $\ell \geq 0$ and node $= P_\ell^-$ do
            $\ell := \ell - 1$;
        node := $P_\ell^-$;
        fathom := False;
    else
        $\ell := \ell + 1$;
        node := $P_\ell^+$;
return $Q^*$;
```

4 Case study: Risk-averse k-club problem with higher moment coherent risk measures

In this section, we present a computational framework for problem (11) and conduct numerical experiments demonstrating the computational performance of the proposed BnB algorithm. To this end, we adopt higher moment coherent risk measures to quantify the risk as described next.

4.1 Higher moment coherent risk measures

The class of higher-moment coherent risk (HMCR) measures was introduced in [27] as optimal values to the following stochastic programming problem:

$$\mathrm{HMCR}_{\alpha,p}(X) = \min_{\eta \in \mathbb{R}} \ \eta + (1-\alpha)^{-1} \left\| (X-\eta)^+ \right\|_p, \quad \alpha \in (0,1), \quad p \geq 1, \tag{13}$$

where $X^+ = \max\{0, X\}$ and $\|X\|_p = \big(\mathsf{E}|X|^p\big)^{1/p}$. Mathematical programming problems that contain HMCR measures can be formulated using p-order cone constraints. Typically, in stochastic programming models, the set of random events Ω is assumed to be discrete, $\Omega = \{\omega_1, \ldots, \omega_N\}$, with the probabilities $\mathsf{P}\{\omega_k\} = \pi_k > 0$, and $\pi_1 + \cdots + \pi_N = 1$. The corresponding mathematical programming model (11) with $\rho(X) = \text{HMCR}_{p,\alpha}(X)$ takes the following mixed 0–1 p-order cone programming form:

$$\begin{aligned}
\min \quad & \eta + (1-\alpha)^{-1} t_0 \\
\text{s. t.} \quad & t_0 \geq \|(t_1, \ldots, t_N)\|_p, \\
& \pi_k^{-1/p} y_k \geq \sum_{i \in V} u_i X_{ik} - \eta, \quad k = 1, \ldots, N, \\
& \sum_{i \in V} u_i = 1, \\
& u_i \leq x_i, \ \ i \in V, \\
& \text{(11d)} - \text{(11i)}, \\
& t_k \geq 0, \ k = 0, \ldots, N,
\end{aligned} \tag{14}$$

where X_{ik} represents the realization of the stochastic weight of vertex $i \in V$ under scenario $k \in \{1, \ldots, N\}$. Analogously, the lower bound problem (12) takes the form

$$\begin{aligned}
\mathcal{L}(Q \cup C_{\ell+1}; \mathbf{X}_G) = \min \quad & \eta + (1-\alpha)^{-1} t_0 \\
\text{s. t.} \quad & t_0 \geq \|t_1, \ldots, t_N\|_p, \\
& \pi_k^{-1/p} t_k \geq \sum_{i \in V} u_i X_{ik} - \eta, \quad k = 1, \ldots, N, \\
& \sum_{i \in V} u_i = 1, \\
& u_i \geq 0, \ \ i \in Q \cup C_{\ell+1}, \\
& u_i = 0, \ \ i \in V \setminus (Q \cup C_{\ell+1}), \\
& t_k \geq 0, \ k = 0, \ldots, N.
\end{aligned} \tag{15}$$

For instances when $p \in \{1, 2\}$, problems (14) and (15) reduce to linear programming (LP) and second order cone programming (SOCP) models, respectively. However, in cases when when $p \in (1, 2) \cup (2, \infty)$ the p-cone is not self-dual and there exist no efficient long-step self-dual interior point solution methods. Consequently, we employ solution methods for p-order cone programming problems that are based on polyhedral approximations of p-order cones [45] and representation of rational-order p-cones via second order cones [31].

4.2 Setup of the numerical experiments and results

Numerical experiments of the risk-averse k-club problem for $k = 2, 3, 4$ were conducted on randomly generated Erdös-Rényi graphs of orders $|V| = 50, 100, 200$ with average densities $D(G) = 0.0125$, 0.025, 0.05, 0.1, and 0.15. The specified densities were chosen due to empirical observations indicating that a graph of order $|V| \geq 50$ commonly reduces to a 2-club when the density is in the range [0.15, 0.25]. Clearly, this effect is even more pronounced for $k > 2$. The stochastic weights of graphs' vertices were generated as i.i.d. samples from the uniform $U(0, 1)$ distribution. Scenario sets with $N = 250$ scenarios were generated for each combination of graph order and density. The HMCR risk measures (13) with $p = 1, 2, 3$, and $\alpha = 0.9$ were used.

The BnB algorithm has been coded in C++, and we used the CPLEX Simplex and Barrier solvers for the polyhedral approximations and SOCP reformulations of the p-order cone programming lower bound problem (15), respectively (see [25]). For instances when $p = 1$, the CPLEX Simplex solver was utilized to solve problem (15) directly. The computations were conducted on an Intel Xeon 3.30GHz PC with 128GB RAM, and CPLEX 12.6 solver in Windows 7 64-bit environment was used.

The computational performance of the mathematical programming model (14) was compared with that of developed BnB algorithm. In the case of $p = 1$, problem (14) was solved with CPLEX Mixed Integer Programming (MIP) solver. The CPLEX MIP Barrier solver was used for the SOCP version in the case of $p = 2$, and using the SOCP reformulation in the case of $p = 3$.

Tables 1– 3 present the computational times and the best objective values averaged over five instances for each graph configuration, as well as the number of instances for which an optimal solution was attained within a 3600 second time limit. The reported average time is calculated by only considering the instances where the problem was solved to optimality within the time limit, while the reported average objective value is calculated by only considering the instances in which at least a feasible solution is found within the time limit. The symbol "—" was used to indicate that the time limit was exceeded, and cells containing "NA" correspond to instances for which solution process failed due to CPLEX running out of memory. Table 1 demonstrates that the BnB algorithm significantly outperforms the CPLEX MIP solver over all the listed graph configurations when $k = 2$, achieving up to an order of magnitude of improvement in computational time. Further, observe that the quality of the average best objectives obtained by the BnB algorithm was superior whenever both methods failed to reach an optimal solution within the time limit. In cases when CPLEX failed due to memory capacity issues, the BnB algorithm either attained an optimal solution or an incumbent solution, in which case the average solution associated with the best incumbent solutions are provided. Note that the performance of both algorithms decreases for higher values of p. This becomes particularly pronounced for $p = 3$ and $|V| = 200$ in Table 1, where CPLEX could not manage any of the corresponding instances due to the increased problem size associated with the cutting-plane algorithm for solving polyhedral approximations of p-order cone programming problems, while the BnB algorithm only solved eleven instances within the time limit.

A similar improvement in performance can be observed for $k = 3$ and $k = 4$ in Tables 2–3. As k increases, the number of time limit and memory capacity limit violations for CPLEX increases, further demonstrating the applicability of the proposed BnB method. This observable disadvantage associate with model (11) results from the fact that the number of constraints in model (11) rapidly increases with k, thus overwhelming the solver in many cases. All the instances in Table 3 with $|V| = 200$ are of this type.

Based on the results presented in Tables 1–3, it is worth noting that as $D(G)$ increases for a given p and $|V|$, the average computation time for the BnB algorithm increases, reaches a maximum value, and then decreases. This is due to the fact that once $D(G)$ is large enough, graph G tends to contain larger components of lower diameter that can be detected at the early stages of the BnB algorithm. Another interesting observation is that for a given p and $D(G)$, if $|V|$ is large enough, the average computation time for BnB algorithm decreases as $|V|$ increases. For instance, in Table 2, for $p = 2$ and $D(G) = 0.1$, none of the instances with $|V| = 100$ were solved to optimality, while all the instances with $|V| = 200$ were solved to optimality within 4.05 seconds on average. This observation can be justified by the fact that for a given expected edge density $D(G)$, if $|V|$ is sufficiently large, the diameter of the random graph decreases as $|V|$ increases (see, e.g., [11], p. 62). Therefore, in these cases, the graphs with larger $|V|$

tend to have larger components of low diameter that can likewise be detected during the early stages of the BnB algorithm.

In order to demonstrate the applicability of our algorithms on real-life graphs, Tables 4- 6 present the results obtained from solving various DIMACS graph instances with the same number of scenarios and distribution of uncertain vertex weights as above. Observe that the BnB method outperforms CPLEX over the vast majority of tested instances, and more than two orders of magnitude in improvements were observed for various cases. However, in several cases even the BnB algorithm failed to obtain an incumbent solution within the time limit (denoted by "∞"), underscoring the complex nature of many real-life graphs.

				$p = 1$			$p = 2$			$p = 3$	
				$\|V\|$			$\|V\|$			$\|V\|$	
$D(G)$	Algorithm	Output	50	100	200	50	100	200	50	100	200
0.0125	CPLEX	Time (s)	0.95	4.46	61.64	32.18	91.33	3043.69	129.07	278.05	NA
		Instance	5	5	5	5	5	1	5	5	0
		Objective	0.23	0.21	0.19	0.28	0.25	0.37	0.30	0.25	NA
	BnB	Time (s)	0.23	1.04	5.01	8.63	25.13	80.27	35.64	100.08	301.68
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.23	0.21	0.19	0.28	0.25	0.21	0.30	0.25	0.21
0.025	CPLEX	Time (s)	1.44	7.49	177.34	46.96	233.35	—	93.66	352.55	NA
		Instance	5	5	5	5	5	0	5	5	0
		Objective	0.23	0.20	0.17	0.28	0.23	0.54	0.29	0.23	NA
	BnB	Time (s)	0.24	1.11	7.62	10.90	30.35	167.98	51.66	141.57	746.82
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.23	0.20	0.17	0.28	0.23	0.19	0.29	0.23	0.19
0.05	CPLEX	Time (s)	1.92	14.64	2185.63	60.53	472.10	—	123.17	776.51	NA
		Instance	5	5	5	5	5	0	5	5	0
		Objective	0.20	0.18	0.15	0.23	0.19	0.20	0.24	0.19	NA
	BnB	Time (s)	0.26	2.02	37.88	15.43	75.50	1087.92	65.07	368.43	3051.66
		Instance	5	5	5	5	5	5	5	5	1
		Objective	0.20	0.18	0.15	0.23	0.19	0.16	0.24	0.19	0.16
0.1	CPLEX	Time (s)	4.25	322.01	—	150.96	—	—	423.76	—	NA
		Instance	5	5	0	5	0	0	5	0	0
		Objective	0.18	0.15	0.14	0.20	0.18	0.41	0.20	0.18	NA
	BnB	Time (s)	0.59	28.51	—	38.03	1451.43	—	183.78	—	—
		Instance	5	5	0	5	5	0	5	0	0
		Objective	0.18	0.15	0.14	0.20	0.16	0.15	0.20	0.16	0.15
0.15	CPLEX	Time (s)	9.48	2832.38	—	1055.83	—	—	1862.11	—	NA
		Instance	5	2	0	5	0	0	5	0	0
		Objective	0.17	0.14	0.18	0.18	0.16	0.17	0.18	0.16	NA
	BnB	Time (s)	2.41	2033.67	—	164.06	—	—	707.16	—	—
		Instance	5	3	0	5	0	0	5	0	0
		Objective	0.17	0.14	0.18	0.18	0.14	0.20	0.18	0.15	0.22

Table 1: Average computation times (in seconds), number of instances solved to optimality (out of five) and the average best objective values obtained by solving problem (11) using the proposed BnB algorithm and CPLEX with $k = 2$ and risk measure (13).

			$p = 1$			$p = 2$			$p = 3$		
				$\|V\|$			$\|V\|$			$\|V\|$	
$D(G)$	Algorithm	Output	50	100	200	50	100	200	50	100	200
0.0125	CPLEX	Time (s)	0.88	6.12	NA	14.16	148.36	NA	86.09	258.75	NA
		Instance	5	5	0	5	5	0	5	5	0
		Objective	0.22	0.19	NA	0.27	0.22	NA	0.27	0.21	NA
	BnB	Time (s)	0.23	1.04	6.80	8.84	29.01	162.51	36.64	113.63	708.45
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.22	0.19	0.16	0.27	0.22	0.18	0.27	0.21	0.18
0.025	CPLEX	Time (s)	1.26	12.68	NA	27.07	516.78	NA	69.69	577.47	NA
		Instance	5	5	0	5	5	0	5	5	0
		Objective	0.21	0.18	NA	0.24	0.19	NA	0.25	0.19	NA
	BnB	Time (s)	0.24	1.59	81.50	11.28	65.54	2075.28	52.72	286.93	—
		Instance	5	5	5	5	5	4	5	5	0
		Objective	0.21	0.18	0.15	0.24	0.19	0.15	0.25	0.19	0.15
0.05	CPLEX	Time (s)	2.79	287.23	NA	163.45	—	NA	385.90	—	NA
		Instance	5	5	0	5	0	0	5	0	0
		Objective	0.17	0.14	NA	0.19	0.17	NA	0.19	0.16	NA
	BnB	Time (s)	0.43	44.13	—	29.41	1060.88	—	131.34	1531.64	—
		Instance	5	5	0	5	4	0	5	1	0
		Objective	0.17	0.14	0.14	0.19	0.15	0.18	0.19	0.15	0.17
0.1	CPLEX	Time (s)	14.27	25.41	NA	2656.62	3425.15	NA	2797.45	3311.62	NA
		Instance	5	2	0	5	1	0	2	1	0
		Objective	0.15	0.11	NA	0.15	0.11	NA	0.15	0.11	NA
	BnB	Time (s)	3.00	719.53	3.70	367.80	—	4.05	941.06	—	4.96
		Instance	5	3	5	5	0	5	5	0	5
		Objective	0.15	0.11	0.10	0.15	0.11	0.10	0.15	0.11	0.10
0.15	CPLEX	Time (s)	50.30	480.23	NA	2329.51	—	NA	1003.57	—	NA
		Instance	5	5	0	2	0	0	3	0	0
		Objective	0.13	0.11	NA	0.13	0.12	NA	0.13	0.12	NA
	BnB	Time (s)	2.50	0.29	4.63	762.31	0.50	4.94	998.11	1.51	5.89
		Instance	5	5	5	5	5	5	4	5	5
		Objective	0.13	0.11	0.10	0.13	0.11	0.10	0.13	0.11	0.10

Table 2: Average computation times (in seconds), number of instances solved to optimality (out of five) and the average best objective values obtained by solving problem (11) using the proposed BnB algorithm and CPLEX with $k = 3$ and risk measure (13).

5 Conclusions

We have considered an RA-k problem which entails finding a k-club of minimum risk in a graph. HMCR risk measures were utilized for quantifying the distributional information of the stochastic factors associated with vertex weights. It was shown that the decision version of RA-k problem is $\mathcal{NP}$-hard for any fixed positive integer k, and the optimal solutions are maximal k-clubs. A combinatorial BnB solution algorithm was developed and tested on a special case of RA-k problem when $k = 2, 3, 4$. Numerical experiments on randomly generated graphs of various configurations suggest that the proposed BnB algorithm can significantly reduce solution times in comparison with the mathematical programming model solved using CPLEX MIP solver.

			$p = 1$			$p = 2$			$p = 3$		
				$\lvert V \rvert$			$\lvert V \rvert$			$\lvert V \rvert$	
$D(G)$	Algorithm	Output	50	100	200	50	100	200	50	100	200
0.0125	CPLEX	Time (s)	0.90	7.30	NA	27.09	206.30	NA	118.25	299.46	NA
		Instance	5	5	0	5	5	0	5	5	0
		Objective	0.21	0.17	NA	0.25	0.18	NA	0.26	0.19	NA
	BnB	Time (s)	0.23	1.13	15.13	8.25	31.98	409.93	47.41	160.66	2001.38
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.21	0.17	0.14	0.25	0.18	0.15	0.26	0.19	0.15
0.025	CPLEX	Time (s)	1.60	21.97	NA	50.10	1475.67	NA	79.91	1714.83	NA
		Instance	5	5	0	5	5	0	5	5	0
		Objective	0.19	0.15	NA	0.22	0.16	NA	0.23	0.16	NA
	BnB	Time (s)	0.23	2.46	—	11.58	91.57	—	63.31	514.33	—
		Instance	5	5	0	5	5	0	5	5	0
		Objective	0.19	0.15	0.12	0.22	0.16	0.13	0.23	0.16	0.13
0.05	CPLEX	Time (s)	4.34	—	NA	461.18	—	NA	929.33	—	NA
		Instance	5	0	0	5	0	0	5	0	0
		Objective	0.16	0.12	NA	0.16	0.12	NA	0.16	0.12	NA
	BnB	Time (s)	0.66	728.07	2.71	35.37	—	3.06	177.83	—	4.23
		Instance	5	5	5	5	0	5	5	0	5
		Objective	0.16	0.12	0.10	0.16	0.12	0.10	0.16	0.12	0.10
0.1	CPLEX	Time (s)	33.72	1776.46	NA	898.37	449.72	NA	493.52	500.88	NA
		Instance	5	4	0	3	3	0	4	3	0
		Objective	0.13	0.13	NA	0.13	0.11	NA	0.13	0.11	NA
	BnB	Time (s)	4.63	0.25	3.71	187.73	0.47	4.07	236.05	1.72	5.33
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.13	0.11	0.10	0.13	0.11	0.10	0.13	0.11	0.10
0.15	CPLEX	Time (s)	25.40	2503.89	NA	282.75	—	NA	271.86	—	NA
		Instance	5	5	0	5	0	0	5	0	0
		Objective	0.13	0.11	NA	0.13	0.12	NA	0.13	0.12	NA
	BnB	Time (s)	0.04	0.30	4.63	0.22	0.53	4.97	1.54	1.95	6.42
		Instance	5	5	5	5	5	5	5	5	5
		Objective	0.13	0.11	0.10	0.13	0.11	0.10	0.13	0.11	0.10

Table 3: Average computation times (in seconds), number of instances solved to optimality (out of five) and the average best objective values obtained by solving problem (11) using the proposed BnB algorithm and CPLEX with $k = 4$ and risk measure (13).

6 Acknowledgements

This research was performed while the first author held a National Research Council Research Associateship Award at the Air Force Research Laboratory. This work was supported in part by the AFOSR grant FA9550-12-1-0142, DTRA grant HDTRA1-14-1-0065, and the U.S. Department of Air Force grant FA8651-14-2-0003. In addition, support by the AFRL Mathematical Modeling and Optimization Institute is gratefully acknowledged.

References

[1] J. Abello, P. Pardalos, and M. Resende. On maximum clique problems in very large graphs. In J. Abello and J. Vitter, editors, *External memory algorithms and visualization*, volume 50 of *DI-*

				$p = 1$		$p = 2$		$p = 3$	
Graph Name	$\lvert V\rvert$	$\lvert E\rvert$	Output	CPEX	BnB	CPLEX	BnB	CPLEX	BnB
adjnoun.clq	112	425	Time (s)	22.25	4.95	—	92.12	—	368.83
			Objective	0.20	0.20	∞	0.21	∞	0.21
celegans metabolic.clq	453	2025	Time (s)	3404.14	55.40	—	191.94	—	292.25
			Objective	0.10	0.10	∞	0.11	∞	0.11
celegansneural.clq	297	2148	Time (s)	585.59	—	—	—	—	—
			Objective	0.10	0.10	∞	0.10	∞	0.10
chesapeake.clq	39	170	Time (s)	1.33	0.11	129.44	4.85	106.52	19.50
			Objective	0.23	0.23	0.24	0.24	0.24	0.24
dolphins.clq	62	159	Time (s)	4.90	0.89	193.18	30.05	402.20	161.37
			Objective	0.35	0.35	0.41	0.41	0.40	0.40
email.clq	1133	5451	Time (s)	—	1221.39	NA	—	NA	—
			Objective	∞	0.19	NA	0.19	NA	0.21
football.clq	115	613	Time (s)	179.92	9.21	—	314.28	—	1575.35
			Objective	0.33	0.33	0.37	0.36	NA	0.36
jazz.clq	198	2742	Time (s)	—	—	—	—	—	—
			Objective	∞	∞	∞	∞	∞	∞
karate.clq	34	78	Time (s)	1.83	0.14	60.69	8.22	76.15	41.90
			Objective	0.32	0.32	0.35	0.35	0.35	0.35
lesmis.clq	77	254	Time (s)	7.35	8.19	—	—	—	—
			Objective	0.20	0.20	∞	0.22	∞	0.22
netscience.clq	1589	2742	Time (s)	—	479.99	NA	3380.12	NA	—
			Objective	∞	0.23	NA	0.24	NA	0.24
polblogs.clq	1490	16715	Time (s)	—	—	NA	—	NA	—
			Objective	∞	0.16	NA	0.16	NA	0.16
polbooks.clq	105	441	Time (s)	22.42	4.03	—	62.73	—	303.97
			Objective	0.26	0.26	∞	0.27	∞	0.27

Table 4: Computation times (in seconds) and the best objective values obtained by solving problem (11) for various DIMACS graph instances using the proposed BnB algorithm and CPLEX with $k = 2$ and risk measure (13).

MACS Series on Discrete Mathematics and Theoretical Computer Science, pages 119–130. American Mathematical Society, 1999.

[2] J. Abello, M. Resende, and S. Sudarsky. Massive quasi-clique detection. In S. Rajsbaum, editor, *LATIN 2002: Theoretical Informatics*, pages 598–612, London, 2002. Springer-Verlag.

[3] R. D. Alba. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3:3–113, 1973.

[4] Y. P. Aneja, R. Chandrasekaran, and K. P. K. Nair. Maximizing residual flow under an arc destruction. *Networks*, 38(4):194–198, 2001.

[5] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[6] A. Atamtrk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.

				$p = 1$		$p = 2$		$p = 3$	
DIMACS	$\|V\|$	$\|E\|$	Output	CPEX	BnB	CPLEX	BnB	CPLEX	BnB
adjnoun.clq	112	425	Time (s)	554.92	2911.88	—	—	—	—
			Objective	0.17	0.17	∞	0.17	∞	0.17
celegans metabolic.clq	453	2025	Time (s)	—	—	NA	—	NA	—
			Objective	∞	0.10	NA	0.11	NA	0.11
celegansneural.clq	297	2148	Time (s)	—	20.81	NA	21.19	NA	21.81
			Objective	∞	0.10	NA	0.10	NA	0.10
chesapeake.clq	39	170	Time (s)	1.23	0.03	67.80	0.39	74.99	0.78
			Objective	0.22	0.22	0.23	0.23	0.23	0.23
dolphins.clq	62	159	Time (s)	16.24	1.25	2644.07	59.13	—	221.67
			Objective	0.26	0.26	0.27	0.27	∞	0.27
email.clq	1133	5451	Time (s)	—	—	NA	—	NA	—
			Objective	∞	0.14	NA	0.14	NA	0.14
football.clq	115	613	Time (s)	—	2935.65	—	—	—	—
			Objective	∞	0.19	∞	0.20	∞	0.21
jazz.clq	198	2742	Time (s)	—	—	—	—	—	—
			Objective	∞	0.12	∞	0.12	∞	0.12
karate.clq	34	78	Time (s)	1.42	0.27	—	20.56	—	68.19
			Objective	0.28	0.28	∞	0.29	∞	0.29
lesmis.clq	77	254	Time (s)	5.87	0.12	750.07	0.30	520.12	1.08
			Objective	0.19	0.19	0.19	0.19	0.19	0.19
netscience.clq	1589	2742	Time (s)	—	554.77	NA	—	NA	—
			Objective	∞	0.20	NA	0.20	NA	0.20
polblogs.clq	1490	16715	Time (s)	—	—	NA	—	NA	—
			Objective	∞	∞	NA	∞	NA	∞
polbooks.clq	105	441	Time (s)	202.53	30.70	—	901.68	—	3468.82
			Objective	0.21	0.21	∞	0.22	∞	0.22

Table 5: Computation times (in seconds) and the best objective values obtained by solving problem (11) for various DIMACS graph instances using the proposed BnB algorithm and CPLEX with $k = 3$ and risk measure (13).

[7] L. Babel. A fast algorithm for the maximum weight clique problem. *Computing*, 52(1):31–38, 1994.

[8] E. Balas and C. S. Yu. Finding a maximum clique in an arbitrary graph. *SIAM J. Comput.*, 15(4):1054–1068, Nov. 1986.

[9] B. Balasundaram, S. Butenko, and S. Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10(1):23–39, 2005.

[10] B. Balasundaram and F. M. Pajouh. Graph theoretic clique relaxations and applications. In P. M. Pardalos, D.-Z. Du, and R. Graham, editors, *Handbook of Combinatorial Optimization*, pages 1559–1598. Springer, 2$^{\text{nd}}$ edition, 2013.

[11] A. Barabasi. *Network Science*. Center for Complex Network Research at Northeastern University (http://barabasilab.neu.edu/networksciencebook/downlPDF.html), Boston, MA, 2012.

				$p = 1$		$p = 2$		$p = 3$	
DIMACS	$\|V\|$	$\|E\|$	Output	CPEX	BnB	CPLEX	BnB	CPLEX	BnB
adjnoun.clq	112	425	Time (s)	—	18.57	—	282.90	—	449.63
			Objective	∞	0.15	∞	0.15	∞	0.15
celegans metabolic.clq	453	2025	Time (s)	—	1953.44	NA	2081.61	NA	2123.89
			Objective	∞	0.11	NA	0.12	NA	0.12
celegansneural.clq	297	2148	Time (s)	—	20.73	NA	21.20	NA	21.67
			Objective	∞	0.10	NA	0.10	NA	0.10
chesapeake.clq	39	170	Time (s)	1.70	0.03	95.35	0.23	84.43	0.72
			Objective	0.22	0.22	0.23	0.23	0.23	0.23
dolphins.clq	62	159	Time (s)	37.82	13.21	—	542.94	—	2640.58
			Objective	0.24	0.24	∞	0.24	∞	0.24
email.clq	1133	5451	Time (s)	NA	—	NA	—	NA	—
			Objective	NA	∞	NA	∞	NA	∞
football.clq	115	613	Time (s)	326.48	0.42	435.09	0.87	397.31	1.45
			Objective	0.15	0.15	0.15	0.15	0.15	0.15
jazz.clq	198	2742	Time (s)	—	—	NA	—	NA	—
			Objective	∞	0.12	NA	0.12	NA	0.12
karate.clq	34	78	Time (s)	1.89	0.03	51.11	0.78	57.43	3.82
			Objective	0.23	0.23	0.24	0.24	0.24	0.24
lesmis.clq	77	254	Time (s)	7.86	0.14	948.37	0.38	1067.50	1.17
			Objective	0.19	0.19	0.19	0.19	0.19	0.19
netscience.clq	1589	2742	Time (s)	NA	—	NA	—	NA	—
			Objective	NA	0.19	NA	0.16	NA	0.18
polblogs.clq	1490	16715	Time (s)	NA	—	NA	—	NA	—
			Objective	NA	∞	NA	∞	NA	∞
polbooks.clq	105	441	Time (s)	786.70	—	—	—	—	—
			Objective	0.18	0.19	∞	0.20	∞	0.20

Table 6: Computation times (in seconds) and the best objective values obtained by solving problem (11) for various DIMACS graph instances using the proposed BnB algorithm and CPLEX with $k = 4$ and risk measure (13).

[12] V. Boginski, S. Butenko, and P. Pardalos. Mining market data: a network approach. *Computers & Operations Research*, 33(11):3171–3184, 2006.

[13] J.-M. Bourjolly, G. Laporte, and G. Pesant. An exact algorithm for the maximum k-club problem in an undirected graph. *European Journal of Operational Research*, 138(1):21 – 28, 2002.

[14] A. M. Campbell and B. W. Thomas. Probabilistic traveling salesman problem with deadlines. *Transportation Science*, 42(1):1–21, 2008.

[15] R. Carmo and A. Zge. Branch and bound algorithms for the maximum clique problem under a unified framework. *Journal of the Brazilian Computer Society*, 18(2):137–151, 2012.

[16] R. Carraghan and P. M. Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9(6):375 – 382, 1990.

[17] M.-S. Chang, L.-J. Hung, C.-R. Lin, and P.-C. Su. Finding large k-clubs in undirected graphs. *Computing*, 95(9):739–758, 2013.

[18] D. J. Cook and L. B. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.

[19] F. Delbaen. Coherent risk measures on general probability spaces. pages 1–37, 2002.

[20] G. D. Glockner and G. L. Nemhauser. A dynamic network flow problem with uncertain arc capacities: Formulation and problem structure. *Operations Research*, 48(2):233–242, 2000.

[21] A. Gupta, V. Nagarajan, and R. Ravi. Technical noteapproximation algorithms for vrp with stochastic demands. *Operations Research*, 60(1):123–127, 2012.

[22] S. Hill, F. Provost, and C. Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 22:256–275, 2006.

[23] D. Iacobucci and N. Hopkins. Modeling dyadic interactions and networks in marketing. *Journal of Marketing Research*, 24:5–17, 1992.

[24] J. Konc and D. Janezic. An improved branch and bound algorithm for the maximum clique problem. *proteins*, 4:5, 2007.

[25] P. Krokhmal and P. Soberanis. Risk optimization with p-order conic constraints: A linear programming approach. *European Journal of Operational Research*, 301(3):653–671, 2010.

[26] P. Krokhmal, M. Zabarankin, and S. Uryasev. Modeling and optimization of risk. *Surveys in Operations Researh and Management Science*, 16(2):49–66, 2011.

[27] P. A. Krokhmal. Higher moment coherent risk measures. *Quantitative Finance*, 7:373–387, 2007.

[28] D. Kumlander. A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search. In *Proceedings of the Fourth International Conference on Engineering Computational Technology*, pages 137–138. Civil-Comp Press, 2004.

[29] R. Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15(2):169–190, 1950.

[30] R. Mokken. Cliques, clubs and clans. *Quality and Quantity*, 13(2):161–173, 1979.

[31] Y. Morenko, A. Vinel, Z. Yu, and P. Krokhmal. On p-cone linear discrimination. *European Journal of Operational Research*, 231(3):784–789, 2013.

[32] P. R. J. Östergård. A new algorithm for the maximum-weight clique problem. *Nordic J. of Computing*, 8(4):424–436, Dec. 2001.

[33] P. R. J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1–3):197–207, 2002. Special Issue devoted to the 6th Twente Workshop on Graphs and Combinatorial Optimization.

[34] F. M. Pajouh and B. Balasundaram. On inclusionwise maximal and maximum cardinality k-clubs in graphs. *Discrete Optimization*, 9(2):84 – 97, 2012.

[35] J. Pattillo, A. Veremyev, S. Butenko, and V. Boginski. On the maximum quasi-clique problem. *Discrete Applied Mathematics*, 161(1–2):244–257, 2013.

[36] R. T. Rockafellar and S. Uryasev. The fundamental risk quadrangle in risk management, optimization and statistical estimation. *Surveys in Operations Research and Management Science*, 18:33–53, 2013.

[37] R. T. Rockafellar, S. Uryasev, and M. Zabarankin. Generalized deviations in risk analysis. *Finance and Stochastics*, 10(1):51–74, 2006.

[38] M. Rysz, M. Mirghorbani, P. Krokhmal, and E. Pasiliao. On risk-averse maximum weighted subgraph problems. *Journal of Combinatorial Optimization*, 28(1):167–185, 2014.

[39] A. Schfer, C. Komusiewicz, H. Moser, and R. Niedermeier. Parameterized computational complexity of finding small-diameter subgraphs. *Optimization Letters*, 6(5):883–891, 2012.

[40] S. B. Seidman and B. L. Foster. A graph theoretic generalization of the clique concept. *Journal of Mathematical Sociology*, 6:139–154, 1978.

[41] E. Tomita, Y. Sutani, T. Higashi, S. Takahashi, and M. Wakatsuki. A simple and faster branch-and-bound algorithm for finding a maximum clique. In M. Rahman and S. Fujita, editors, *WALCOM: Algorithms and Computation*, volume 5942 of *Lecture Notes in Computer Science*, pages 191–203. Springer Berlin Heidelberg, 2010.

[42] S. Trukhanov, C. Balasubramaniam, B. Balasundaram, and S. Butenko. Algorithms for detecting optimal hereditary structures in graphs, with application to clique relaxations. *Computational Optimization and Applications*, 56(1):113–130, 2013.

[43] A. Veremyev, O. Prokopyev, and E. Pasiliao. Critical nodes for communication efficiency and related problems in graphs. *Working Paper.*, 2014.

[44] B. Verweij, S. Ahmed, A. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications*, 24(2-3):289–333, 2003.

[45] A. Vinel and P. Krokhmal. Polyhedral approximations in p-order cone programming. *Optimization Methods and Software*, 29(6):1210–1237, 2014.

[46] A. G. Woodside and M. W. DeLozier. Effects of word of mouth advertising on consumer risk taking. *Journal of Advertising*, 5(4):12–19, 1976.

[47] M. Yannakakis. Node-and edge-deletion np-complete problems. In *STOC'78: Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 253–264, New York, 1978. ACM Press.

Mixed-Integer Programming with a Class of Nonlinear Convex Constraints

Alexander Vinel[*] Pavlo A. Krokhmal[†]

Abstract

We study solution approaches to a class of mixed-integer nonlinear programming problems that arise from recent developments in risk-averse stochastic optimization and contain second-order and p-order cone programming as special cases. We explore possible applications of some of the solution techniques that have been successfully used in mixed-integer conic programming and show how they can be generalized to the problems under consideration. Particularly, we consider branch-and-bound method based on outer polyhedral approximations, lifted nonlinear cuts, and linear disjunctive cuts. Results of numerical experiments with discrete portfolio optimization models are presented.

Keywords: Mixed-integer nonlinear programming, measures of risk, branch-and-bound, valid inequalities, conic programming

1 Introduction

In this work we consider solution approaches to a special class of mixed-integer nonlinear optimization problems that includes, among others, mixed integer second- and p-order cone programming problems. Developing the corresponding solution approaches can also be viewed as a way to explore applicability of some of the methods extensively used in mixed-integer conic programming literature in a more general setting. While our interest in the particular class of problems studied here stems from recent developments in risk-averse stochastic optimization (Vinel and Krokhmal, 2014b; Rysz et al., 2014), similar models may arise in other fields of science and engineering in the context of "generalized means" (see below). Namely, in the present study we consider mixed-integer nonlinear programming problems of the form

$$\min \quad \mathbf{c}^\top \mathbf{x} \tag{1a}$$

$$\text{s. t.} \quad v_k^{-1}\left(\sum_{j=1}^{m_k} p_j^k v_k \left(\sum_{i=1}^{n} a_{ij}^k x_i + b_j^k \right) \right) \leq \sum_{i=1}^{n} a_{i0}^k x_i + b_0^k, \quad k = 1, \ldots, K \tag{1b}$$

$$\mathbf{H}\mathbf{x} \leq \mathbf{h} \tag{1c}$$

$$\mathbf{x} \in \mathbb{Z}_+^{n_1} \times \mathbb{R}_+^{n_2}, \tag{1d}$$

[*] Department of Industrial and Systems Engineering, Auburn University, 3301 Shelby Center, Auburn, AL 36849, USA. E-mail: alexandervinel@gmail.com (corresponding author).

[†] Department of Mechanical and Industrial Engineering, University of Iowa, 3131 Seamans Center, Iowa City, IA 52242, USA.

where $n = n_1 + n_2$ is the dimensionality of the mixed-integer decision vector $\mathbf{x}$, and $\mathbf{c}$, $\mathbf{h}$, $\mathbf{H}$ are vectors and a matrix of appropriate dimensions.

The main object of interest in problem (1) is the set of nonlinear constraints (1b), where it is assumed that coefficients p_j^k are positive, $p_j^k > 0$, for all values of j and k, and functions $v_k : \mathbb{R} \mapsto \mathbb{R}, k = 1, \ldots, K$, have the following properties:

(i) $v_k(t) = 0$ for $t \leq 0$,

(ii) $v_k(t)$ are increasing and convex for $t \geq 0$,

(iii) v_k are such that constraints (1b) are convex.

To simplify the exposition and notation, in what follows we are going to suppress index k in (1b), effectively considering problem (1) with a single nonlinear constraint, $K = 1$. Then, given the above assumptions on function v, it is straightforward to see that problem (1) can be rewritten in the form

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} && \text{(2a)} \\ \text{s. t.} \quad & w_0 \geq v^{-1}\Bigg(\sum_{j=1}^m p_j v(w_j) \Bigg) && \text{(2b)} \\ & w_j \geq \sum_{i=1}^n a_{ij} x_i + b_j, \quad j = 1, \ldots, m && \text{(2c)} \\ & w_0 \leq \sum_{i=1}^n a_{i0} x_i + b_0 && \text{(2d)} \\ & \mathbf{Hx} \leq \mathbf{h}, \quad \mathbf{w} \geq \mathbf{0}, \quad \mathbf{x} \in \mathbb{Z}_+^{N_1} \times \mathbb{R}_+^{N_2}. && \text{(2e)} \end{aligned}$$

The expression in the right-hand side of the nonlinear constraint (2b) is well known in the literature under the names of *quasi-arithmetic*, *Kolmogorov*, or *Kolmogorov-Nagumo mean* of the sequence $\{w_1, \ldots, w_m\}$, provided that the positive coefficients p_j satisfy $p_1 + \ldots + p_m = 1$ (see, for example, Bullen et al., 1988; Hardy et al., 1952). In the operations research and economics domains, it is related to the concept of *certainty equivalent* (Wilson, 1979; McCord and Neufville, 1986), or the deterministic quantity such that a rational decision maker with a utility function v would be indifferent between choosing this certain quantity or a random outcome W that may have realizations $w_1, \ldots, w_m$ with probabilities $p_1, \ldots, p_m$.

In the present work, our interest in solving problems of the form (1)–(2) derives from risk-averse stochastic optimization models that employ the *certainty equivalent measures of risk* (Vinel and Krokhmal, 2014b, see also Sections 2 and 5). This application also dictates the above requirements (i)–(iii) on functions v. At the same time, it is easy to see that conditions (i)–(iii) naturally imply that the nonlinear convex constraint (1b) represents a direct generalization of the second-order cone, or, more broadly, p-order cone constraints $w_0 \geq \|(w_1, \ldots, w_m)\|_p$.

Formulation (1) without the integrality constraints has been previously considered in Rysz et al. (2014). That work concentrates on linear constraints (2c), particularly in the case when the value of m is large, which in the stochastic programming setting corresponds to a large number of scenarios (see Section 2). This computational challenge have been addressed by employing an efficient scenario decomposition framework. In the present endeavor we focus our attention on the challenges associated with the nonlinear and integrality constraints in (1). From this point of view, problem (1) can be characterized as a mixed-integer nonlinear programming (MINLP) problem with a convex continuous relaxation, and there exists an extensive body of literature discussing solution methods for either general MINLP or mixed-integer conic programming (MICP). Since the formulation considered here is in some sense "in between" of these two classes, our discussion is concentrated on attempts to utilize the specific structure of the nonlinear constraint. While constraint (2b) is no longer necessary conic, in our discussion below we will show that some of the solution procedures proposed for second- or p-order cone programming (SOCP or pOCP) problems can be extended to this class as well.

Development of both of the most widely used approaches in mixed-integer programming (branch-and-bound algorithm and valid inequalities) in relation to problem (2) will be addressed in this paper. We begin by discussing risk-averse stochastic programming motivation for this problem in Section 2. In Section 3 we present a version of branch-and-bound method targeted at the specific nonlinear constraints considered in this paper. Next, in Section 4 we will address two procedures for generating inequalities valid for the feasible set of (2): lifted nonlinear cuts and disjunctive cuts. Finally in Section 5 we will present some results of numerical experiments. Relevant literature review will be presented in Sections 3 and 4.

In terms of developed solution procedures the main contributions of this paper in our view are the following. First, we show that two techniques (a special implementation of a branch-and-bound and lifted nonlinear valid inequalities) that have been proposed in the context of mixed-integer second-order cone programming (MISOCP) problems can be extended to the more general case considered here. While, both of this extensions do not require novel theoretical developments, heavily relying on the results already established in the literature, the novelty of the problem formulation justifies, in our view, our interest in these extensions. Particularly, we show how these techniques can be reformulated in order to address this new application area, while still allowing for the use of the already existing theoretical basis. Secondly, we propose another numerical approach, which relies on a simple geometric idea for construction of linear disjunctive cuts. To the best of our knowledge this particular scheme has not been considered in the literature before.

2 Risk-Averse Stochastic Programming Motivation

Consider a function $\rho : \mathfrak{X} \mapsto \mathbb{R} \cup \{+\infty\}$, where $\mathfrak{X}$ is an appropriate linear space of $\mathcal{F}$-measurable functions on a probability space $(\Omega, \mathcal{F}, \mathsf{P})$ such that $X = X(\omega) \in \mathfrak{X}$ is interpreted as a random outcome representing a cost or loss associated with the uncertain event $\omega \in \Omega$. Then, function ρ is referred to as a risk measure, and defines a system of preferences on $\mathfrak{X}$ (outcome X is preferred to Y iff $\rho(X) \leq \rho(Y)$). Additionally, suppose that outcome X depends on the value of a decision vector $\mathbf{x} \in \mathcal{X}$. In this case a problem of optimal decision making under uncertainty can be formulated as a (risk-averse) stochastic programming problem

$$\min\{c(\mathbf{x}) \mid \rho(X(\mathbf{x}, \omega)) \leq h(\mathbf{x}),\ \mathbf{x} \in \mathcal{X}\}. \tag{3}$$

Problems of this kind involving various forms of risk measure ρ have been extensively studied in the literature, see, e.g., Krokhmal et al. (2011) for a survey. In the context of this paper we are concerned with a particular type of *certainty equivalent measures of risk,* introduced in Vinel and Krokhmal (2014b), which are defined as

$$\rho(X) := \min_{\eta}\ \eta + \frac{1}{1-\alpha} v^{-1} \mathsf{E} v\big([X - \eta]_+\big),$$

where the *deutility* function v is nondecreasing, convex, such that $v^{-1}\mathsf{E}v(X)$ is convex, and $v(t) = v([t]_+) = v(\max\{0,t\})$. The class of certainty equivalent measures of risk possesses important methodological characteristics, such as convexity, isotonicity with respect to stochastic dominance ordering induced by deutility function v (and, in particular, second-order stochastic dominance), etc., and contains some well-known risk measures as special cases, including CVaR (Rockafellar and Uryasev, 2002) and HMCR (Krokhmal, 2007).

Certainty equivalent measures of risk are amenable to simple implementation in stochastic programming models via constraints of the form (2b) if the set of random events Ω can be assumed finite: $\Omega = \{\omega_1,\ldots,\omega_m\}$ and $\mathsf{P}\{\omega_j\} = p_j > 0$ for $j = 1,\ldots,m$. Then, stochastic programming problem (3) can be equivalently reformulated as

$$\min\Big\{c(\mathbf{x})\;\Big|\;\eta + (1-\alpha)^{-1}v^{-1}\Big(\sum_{j=1}^{m} p_j v\big([X(\mathbf{x},\omega_j)-\eta]_+\big)\Big) \le h(\mathbf{x}),\ \mathbf{x}\in\mathcal{X},\ \eta\in\mathbb{R}\Big\}. \tag{4}$$

If, additionally, it can be assumed that the loss function $X(\mathbf{x},\omega)$ is linear with respect to the decision vector, i.e., $X(\mathbf{x},\omega_j) = \mathbf{a}_j^\top\mathbf{x} + b_j$, and $\mathbf{x}\in\mathbb{Z}^{n_1}\times\mathbb{R}^{n_2}$, then (4) can be written as a special case of MINLP (2)

$$\begin{aligned}
\min\quad & c(\mathbf{x}) && \text{(5a)}\\
\text{s. t.}\quad & \eta + (1-\alpha)^{-1}w_0 \le h(\mathbf{x}) && \text{(5b)}\\
& w_0 \ge v^{-1}\Big(\sum_{j=1}^{m} p_j v(w_j)\Big) && \text{(5c)}\\
& w_j \ge \mathbf{a}_j^\top\mathbf{x} + b_j - \eta, \quad j = 1,\ldots,m && \text{(5d)}\\
& \mathbf{x}\in\mathbb{Z}^{n_1}\times\mathbb{R}^{n_2},\ \mathbf{w}\ge 0,\ \eta\in\mathbb{R}, && \text{(5e)}
\end{aligned}$$

provided that $c(\mathbf{x})$ and $h(\mathbf{x})$ are linear as well. In view of the above, we refer to constraint (2b) as the *certainty equivalent constraint.*

3 Branch-and-Bound based on Outer Polyhedral Approximations

3.1 Existing Methods and Approach due to Vielma et al (2008)

Branch-and-bound (BnB) methods for solving MINLP problems are often divided into two categories depending on the way continuous relaxations are handled. The first group consists of the methods which solve exact non-linear continuous relaxation, usually using some version of an interior point method (see, for example Gupta and Ravindran, 1985; Borchers and Mitchell, 1994; Leyffer, 2001 and references therein). Alternatively, polyhedral approximations can be employed to help with finding approximate solutions of the continuous relaxations (Duran and Grossmann, 1986; Fletcher and Leyffer, 1994; Quesada and Grossmann, 1992; Bonami et al., 2008; Vielma et al., 2008). This approach has been the basis for a few MINLP solvers such as Bonmin (Bonami et al., 2008), FilMINT (Abhishek et al., 2010) or AOA (AIMMS open MINLP solver). For example, outer approximation algorithms (AOA) solve alternating sequence of MILP master problems and NLP subproblems, while in LP-NLP-based BnB methods (Quesada and Grossmann, 1992, FilMINT) the solution of a single master mixed-integer linear programming (MILP) problem is terminated every time an integer valued candidate is found to solve an exact NLP, solution of which is then used to generate new outer approximations.

Another framework has been proposed by Vielma et al. (2008) for the case of mixed-integer second order cone programming (MISOCP) problems. The authors exploit the fact that there exists an extremely efficient lifted outer polyhedral approximation of second order cones, and thus propose to solve full-sized approximating LP at each node of the master MILP, while, as previously, an exact NLP is solved every time a new integer solution is found. Note that in this case, the algorithm is guaranteed to find a solution that is ε-feasible to the relaxation at each node of the BnB tree, as opposed to LP-NLP approach, where NLP solution is used to generate new approximating facets. Hence, one of the key differences between different implementations of such BnB methods can be viewed as a trade-off between the size of approximating LPs (i.e., the accuracy of the approximation) and the number of exact NLPs that need to be solved. Note that an exact NLP, of course, provides tighter lower bounds, and thus, more pruning capabilities, while LPs bring-in superior warm-start efficiencies, consequently speeding up the processing time in each node. In this sense, the approach of Vielma et al. (2008) can be viewed as the most conservative in terms of the use of the exact solvers: NLPs are only solved when absolutely necessary to verify incumbent integer solutions.

The fact that this approach relies on an efficient lifted approximation scheme is essential, since otherwise exponentially large polyhedral approximations may be required to achieve guaranteed ε-feasibility for general nonlinear constraints. The main source of difficulty here can be associated with high dimensionality of the constraint, i.e., it can be seen as a manifestation of the "curse of dimensionality". In Vinel and Krokhmal (2014c) we have shown that this framework can be competitive even when no such efficient approximation scheme is available by designing a branch-and-bound based on polyhedral approximations for mixed-integer p-order cone programming (MIpOCP) problems. The key idea there was the introduction of a cutting plane generation procedure for approximately solving continuous pOCP relaxations. In the next subsection we are going to demonstrate that a similar approach is applicable in the more general setting considered in the current paper. In fact, certainty-equivalent constraints can be naturally viewed as the most general setting which still allows for direct application of the considered dimensionality reduction techniques.

3.2 Lifted Approximation Procedure

In the context of MISOCP problems, efficient (in the dimensionality and number of facets) polyhedral approximations of second-order cones due to Ben-Tal and Nemirovski (2001) are available, which are constructed via a two-step procedure. During the first step, a lifting technique, dubbed by the authors *"tower of variables"*, was used to express the high-dimensional second-order cone set via a number of two-dimensional second-order cones, and then a clever lifting approximation procedure was applied to the resulting low-dimensional second-order cone sets. In our previous work (Vinel and Krokhmal, 2014c) dealing with general p-order cones, the second step of this procedure was replaced by a simpler gradient-based approximation, which could be constructed via an efficient cutting plane procedure. In the current endeavor, we again resort to the first-step lifting procedure due to Ben-Tal and Nemirovski (2001), and then investigate the problem of constructing polyhedral approximations of the resulting low-dimensional sets using a cutting plane technique.

Let us denote set described by constraint (2b) as

$$\mathcal{V}^{(m+1)} := \left\{ \mathbf{w} \in \mathbb{R}^{m+1}_{+} \;\middle|\; w_0 \geq v^{-1}\Big(\sum_{j=1}^{m} p_j v(w_j) \Big) \right\}, \tag{6}$$

where, in order to unclutter the notation, we omit the dependence of $\mathcal{V}^{(m+1)}$ on the parameters p_j and function v. We will call a set of form (6) "V-set". Note also that from here on we assume that $\mathbf{w} \in \mathbb{R}_+^{m+1}$ in order to simplify the exposition. Analogous analysis can be conducted when this condition does not hold.

Proposition 3.1 (Tower-of-variables). *Given $p_j > 0$, $j = 1, \ldots, m$, and a function v that satisfies assumptions* (i)–(iii)*, there exist values $\beta_1, \ldots, \beta_{2m-2} > 0$ such that the projection of the $2m$-dimensional set*

$$\widetilde{\mathcal{V}}^{(2m)} := \big\{\mathbf{w} \in \mathbb{R}_+^{2m} \,\big|\, w_0 = w_{2m-1}, \\ w_{m+j} \geq v^{-1}\big(\beta_{2j-1} v(w_{2j-1}) + \beta_{2j} v(w_{2j})\big),\ j = 1, \ldots, m-1\big\}, \tag{7}$$

onto the space of variables $w_0, \ldots, w_m$ equals the set $\mathcal{V}^{(m+1)}$. Moreover, β_j can be selected in such a way that $\beta_{2j-1} + \beta_{2j} = 1$ for $j = 1, \ldots, m-1$.

Proof. As it has been noted above, the set of inequalities in (7) defines a structure that can be referred to as tower-of-variables, where each variable w_j is represented by a node, and edges connect node w_{j+m} with w_{2j-1} and w_{2j}. Let us define sets Υ_j as $\Upsilon_j = \{j\}$ if $j = 1, \ldots, m$ and $\Upsilon_{m+j} = \Upsilon_{2j-1} \cup \Upsilon_{2j}$ for $j = 1, \ldots, m-1$. In other words, set Υ_j is the subset of indexes $\{1, \ldots, m\}$ corresponding to the initial (non-lifting) variables descending from w_j in the tower-of-variables. In this case, let us take

$$\beta_{2j-1} = \frac{\sum\limits_{k \in \Upsilon_{2j-1}} p_k}{\sum\limits_{k \in \Upsilon_{2j-1} \cup \Upsilon_{2j}} p_k}, \quad \beta_{2j} = \frac{\sum\limits_{k \in \Upsilon_{2j}} p_k}{\sum\limits_{k \in \Upsilon_{2j-1} \cup \Upsilon_{2j}} p_k}, \quad j = 1, \ldots, m-1. \tag{8}$$

Now, the claim of the proposition can be verified directly. □

Remark Proposition 3.1 represents, perhaps, the most general version of the original "tower-of-variables" scheme of Ben-Tal and Nemirovski (2001) proposed for second-order cone sets. Note also that the choice of vector $\boldsymbol{\beta}$ ensuring that the claim above holds is not unique. The particular approach proposed in (8) guarantees that $\beta_{2j-1} + \beta_{2j} = 1$, ensuring that each of the inequalities in (7) describes a proper V-set.

Proposition 3.1 reduces the problem of constructing a polyhedral approximation for $(m+1)$-dimensional V-set (6) to that for $m-1$ three-dimensional (3D) V-sets $\mathcal{V}^{(3)}$ in (7),

$$\mathcal{V}^{(3)} := \big\{\mathbf{w} \in \mathbb{R}_+^3 \,\big|\, w_0 \geq f(w_1, w_2)\big\}, \tag{9}$$

where $f(w_1, w_2) := v^{-1}\big(\beta_1 v(w_1) + \beta_2 v(w_2)\big)$. More importantly, it drastically reduces the dimensionality of the resulting polyhedral approximation; instead of the generally exponential in m number of hyperplanes needed for approximation of set $\mathcal{V}^{(m+1)}$, only $O(mk)$ hyperplanes is required to approximate the lifted set $\widetilde{\mathcal{V}}^{(2m)}$, provided that each 3-dimensional V-set in (7) can be approximated with $O(k)$ hyperplanes.

In this respect, it is necessary to comment on the precise definition of approximation that we will use in this work. Namely, we consider set

$$\mathcal{V}_\varepsilon^{(m+1)} := \left\{\mathbf{w} \in \mathbb{R}_+^{m+1} \,\middle|\, (1+\varepsilon) v(w_0) \geq \sum_{j=1}^m p_j v(w_j)\right\},$$

and, accordingly, its three dimensional version

$$\mathcal{V}_\varepsilon^{(3)} := \big\{\mathbf{w} \in \mathbb{R}_+^3 \,\big|\, (1+\varepsilon)v(w_0) \geq \beta_1 v(w_1) + \beta_2 v(w_2)\big\}. \tag{10}$$

Observe that such a choice of approximating condition allows us to connect the approximation quality of a single three-dimensional constraint in the tower-of-variables construction with the multi-dimensional case.

Proposition 3.2. *Consider set $\mathcal{V}^{(m+1)}$ and its lifted representation $\widetilde{\mathcal{V}}^{(2m)}$. If each of the triples in representation $\widetilde{\mathcal{V}}^{(2m)}$ satisfies $(w_{m+j}, w_{2j-1}, w_{2j})^\top \in \mathcal{V}_\epsilon^{(3)}$ for a given $\epsilon > 0$, then $(w_0, \ldots, w_m)^\top \in \mathcal{V}_\varepsilon^{(m+1)}$, where $\varepsilon \leq (1+\epsilon)^{\lceil \log_2 N \rceil} - 1 = \lceil \log_2 N \rceil \epsilon + O(\epsilon^2)$.*

Proof. The claim can be verified directly by expanding the tower-of-variables (see also Vinel and Krokhmal, 2014c, Proposition 3.2). □

Along with the primary definition $\widetilde{\mathcal{V}}_\varepsilon^{(m+1)}$ of ε-approximation of a V-set, we also consider two additional approximation approaches

$$\overline{\mathcal{V}}_\varepsilon^{(3)} := \big\{\mathbf{w} \in \mathbb{R}_+^3 \,\big|\, v\big((1+\varepsilon)w_0\big) \geq \beta_1 v(w_1) + \beta_2 v(w_2)\big\}, \tag{11}$$

$$\overline{\overline{\mathcal{V}}}_\varepsilon^{(3)} := \big\{\mathbf{w} \in \mathbb{R}_+^3 \,\big|\, v(w_0 + \varepsilon) \geq \beta_1 v(w_1) + \beta_2 v(w_2)\big\}. \tag{12}$$

The set $\overline{\mathcal{V}}_\varepsilon^{(3)}$ is a direct extension of the usual approximation used in the case of conic sets (see, for example, Ben-Tal and Nemirovski (2001)), while set $\overline{\overline{\mathcal{V}}}_\varepsilon^{(3)}$ represents an *absolute error* ε-approximation of V-set. It should be emphasized here that only condition in (10) allows for a natural accuracy propagation analysis for the tower-of-variables construction as in Proposition 3.2. The other two approximating conditions will be used in the discussion establishing finiteness of the proposed computational procedure below.

Since the relaxed feasible set considered in the current work is convex, a cutting plane defined as

$$w_0 \geq f(w_1^*, w_2^*) + f'_{w_1}(w_1^*, w_2^*)(w_1 - w_1^*) + f'_{w_2}(w_1^*, w_2^*)(w_2 - w_2^*), \tag{13}$$

which is tangent to the 3-dimensional set $\mathcal{V}^{(3)}$ at point $(f(w_1^*, w_2^*), w_1^*, w_2^*)$, is globally feasible. Hence, the following general framework can be applied. We will consider a master problem in the form of (2), where nonlinear constraint is substituted with a set of cutting planes (13):

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s. t.} \quad & w_{m+j} \geq f\Big(w_1^{k_j}, w_2^{k_j}\Big) + f'_{w_1}\Big(w_1^{k_j}, w_2^{k_j}\Big)\Big(w_{2j-1} - w_1^{k_j}\Big) \\ & \qquad\qquad + f'_{w_2}\Big(w_1^{k_j}, w_2^{k_j}\Big)\Big(w_{2j} - w_2^{k_j}\Big), \quad j = 1, \ldots, m-1,\ k_j = 1, \ldots, K_j, \\ & \text{(2c)–(2e)}, \end{aligned}$$

where K_j is the number of cutting planes on variables $w_{m+j}, w_{2j-1}, w_{2j}$ for all j, derived around the pairs $\Big(w_1^{k_j}, w_2^{k_j}\Big)$, $k_j = 1, \ldots K_j$. Then, given a current solution $\mathbf{w}^*$ of the master problem, we can add new constraints around pairs $\big(w_{2j-1}^*, w_{2j}^*\big)$, for those j for which the selected approximation condition is violated. Afterwards, the master can be resolved and the iterative process continues. Next we show that this procedure terminates after a finite number of iterations with a solution that satisfies the prescribed approximation accuracy, assuming that the feasible sets considered are bounded. As it turns out, an additional auxiliary approximation scheme may be required.

From here on we will assume that $v(t) = \alpha t^p + o(t^p)$ if $t \to 0$. The following simple lemma will be useful below. We will omit the proof of this result since it can be obtained using standard calculus techniques.

Lemma 3.3. *If function v is finite, strictly increasing and convex on $t \geq 0$ and $v(t) = \alpha t^p + o(t^p)$ as $t \to 0$, then $v^{-1}(\tau) = \alpha^{-1/p}\tau^{1/p} + o(\tau^{1/p})$.*

This claim allows us to establish asymptotic behavior of function f around zero. Indeed, observe that $f(w_1, w_2) = v^{-1}\Big(\beta_1 v(w_1) + \beta_2 v(w_2)\Big) = v^{-1}\Big(\beta_1 \alpha w_1^p + \beta_2 \alpha w_2^p + o(w_1^p + w_2^p)\Big) = (\beta_1 w_1^p + \beta_2 w_2^p)^{1/p} + o\big(\|(w_1, w_2)\|_p\big)$. Moreover, since function f is convex, any plane tangent to this p-order cone defined by constraint $w_0 \geq (\beta_1 w_1^p + \beta_2 w_2^p)^{1/p}$ is a supporting plane for epi f by definition.

In view of this we propose the following auxiliary approximation scheme: whenever the current solution of the master is such that $\|(w_1, w_2)\|_2 \leq \Theta$, then in addition to the regular constraint described above, add a cutting plane tangent to the p-order cone, i.e.,

$$w_0 \geq \beta_1^{1/p} w_1 \frac{\cos^{p-1}\theta^*}{(\cos^p\theta^* + \sin^p\theta^*)^{1-1/p}} + \beta_2^{1/p} w_2 \frac{\sin^{p-1}\theta^*}{(\cos^p\theta^* + \sin^p\theta^*)^{1-1/p}}, \quad \theta^* = \arctan\frac{\beta_1^{1/p} w_2^*}{\beta_2^{1/p} w_1^*},$$

where Θ is a preselected parameter. The analysis above implies that this cut does not violate the original certainty-equivalent constraint, and moreover, as it will be demonstrated below, this approach guarantees convergence of the proposed cutting plane procedure.

Proposition 3.4. *Suppose that for a given solution $\mathbf{w}^*$ of the master, cuts in the form of* (13) *are added around all triples $(w^*_{m+j}, w^*_{2j-1}, w^*_{2j})^\top \notin \mathcal{V}_\varepsilon^{(3)}$, where $j \in \{1, \ldots, m-1\}$ and the described above auxiliary approximation scheme is applied. Assuming that the feasible region is bounded, this cutting plane procedure terminates after a finite number of iterations for any given $\varepsilon > 0$.*

Note that in this proposition we implicitly exclude cases when the original problem is infeasible but its ε-approximation in the sense (10) is feasible for every $\varepsilon > 0$. Conditions that guarantee this are given below in Proposition 3.9.

Before verifying the statement of Proposition 3.4, we establish a few subsidiary lemmas.

Lemma 3.5. *If set $\overline{\overline{\mathcal{V}}}_\varepsilon^{(3)}$ is used in the cutting plane scheme described above instead of $\mathcal{V}_\varepsilon^{(3)}$, then the process terminates in a finite number of iterations even without the auxiliary scheme.*

Proof. The claim follows directly from the fact that a bounded convex set in three-dimensions can be approximately described by a number of supporting planes (one can derive this result directly by considering Taylor's expansion of f). □

Lemma 3.6. *If $v(t) = |t|^p$, then for any $\varepsilon > 0$ there exists $\epsilon > 0$ such that $\overline{\mathcal{V}}_\epsilon^{(3)} \subset \mathcal{V}_\varepsilon^{(3)}$ and vice versa, i.e., for p-order cones the conditions in* (10) *and* (11) *are equivalent.*

Proof. Clearly, for any $\varepsilon > 0$ there exists $\epsilon > 0$ such that $(1+\varepsilon)^p = 1 + \epsilon$, which directly implies the claim of the lemma. □

Lemma 3.7. *For any $\varepsilon > 0$ and $\delta_\varepsilon > 0$ there exists $\epsilon > 0$ such that*

$$\overline{\overline{\mathcal{V}}}_\epsilon^{(3)} \cap \big\{(w_0, w_1, w_2)^\top \in \mathbb{R}^3 \,\big|\, (\beta_1 w_1^p + \beta_2 w_2^p)^{1/p} \geq \delta_\varepsilon\big\} \subset \mathcal{V}_\varepsilon^{(3)}.$$

Proof. Let

$$\epsilon = \min_{(\beta_1 w_1^p + \beta_2 w_2^p)^{1/p} \geq \delta_\varepsilon} v^{-1}\big(\beta_1 v(w_1) + \beta_2 v(w_2)\big) - v^{-1}\bigg(\frac{\beta_1 v(w_1) + \beta_2 v(w_2)}{1+\varepsilon}\bigg).$$

Observe that the minimum above is attained and is strictly positive, since we assume that the feasible region is bounded and v^{-1} is strictly increasing. Now, if $w_0 \geq v^{-1}(\beta_1 v(w_1) + \beta_1 v(w_2)) - \epsilon$ and $(\beta_1 w_1^p + \beta_2 w_2^p)^{1/p} \geq \delta_\varepsilon$, then $w_0 \geq v^{-1}\Big(\frac{\beta_1 v(w_1)+\beta_2 v(w_2)}{1+\varepsilon}\Big)$, which implies the claim of the lemma. □

Lemma 3.8. *Consider set $\mathcal{P}_\epsilon^{(3)} := \big\{\mathbf{w} \in \mathbb{R}^3_+ \,\big|\, (1+\epsilon)w_0^{1/p} \geq \beta_1 w_1^{1/p} + \beta_2 w_2^{1/p}\big\}$, which is analogous to $\mathcal{V}_\epsilon^{(3)}$ for a p-order cone. Then, for any $\varepsilon > 0$ there exist $\epsilon > 0$ and $\delta > 0$ such that*

$$\mathcal{P}_\epsilon^{(3)} \cap \big\{(w_0, w_1, w_2)^\top \in \mathbb{R}^3 \,|\, (\beta_1 w_1^p + \beta_2 w_2^p)^{1/p} \leq \delta\big\} \subset \mathcal{V}_\varepsilon^{(3)}.$$

Proof. In order to establish this result we need to show that there exist ϵ and δ such that if $(\beta_1 w_1^p + \beta_2 w_2^p)^{1/p} \leq \delta$ and $w_0 \geq \frac{(\beta_1 w_1^p + \beta_2 w_2^p)^{1/p}}{(1+\epsilon)^{1/p}}$, then $w_0 \geq v^{-1}\Big(\frac{\beta_1 v(w_1)+\beta_2 v(w_2)}{(1+\varepsilon)^{1/p}}\Big)$. As we have observed above, $v^{-1}(\beta_1 v(w_1) + \beta_2 v(w_2)) = (\beta_1 w_1^p + \beta_2 w_2^p)^{1/p} + o\big(\|(w_1, w_2)\|_p\big)$. Hence, $v^{-1}\Big(\frac{\beta_1 v(w_1)+\beta_2 v(w_2)}{1+\varepsilon}\Big) = \frac{(\beta_1 w_1^p + \beta_2 w_2^p)^{1/p}}{(1+\varepsilon)^{1/p}} + g(w_1, w_2)$, where $g(w_1, w_2) = o\big(\|(w_1, w_2)\|_p\big)$. Then, there exists δ, such that $(\beta_1 w_1^p + \beta_2 w_2^p)^{1/p} \leq \delta$ implies $|g(w_1, w_2)| \leq (\beta_1 w_1^p + \beta_2 w_2^p)^{1/p}\Big(\frac{1}{(1+\varepsilon/2)^{1/p}} - \frac{1}{(1+\varepsilon)^{1/p}}\Big)$. Consequently, it follows that $v^{-1}\Big(\frac{\beta_1 v(w_1)+\beta_2 v(w_2)}{1+\varepsilon}\Big) \leq \frac{(\beta_1 w_1^p + \beta_2 w_2^p)^{1/p}}{(1+\varepsilon/2)^{1/p}}$ for such (w_1, w_2). Then the claim of the lemma is satisfied if we take ϵ such that $(1+\epsilon) = (1+\varepsilon/2)^{1/p}$. □

Proof of the Proposition. Assume to the contrary that the cutting plane procedure does not terminate after a finite number of iterations. Then, for at least one triple $(w_{m+j}, w_{2j-1}, w_{2j})$ the approximation condition is violated infinitely many times and, therefore, infinitely many cutting planes are generated. Let us denote this triple as (w_0, w_1, w_2). First, suppose that there exists Δ such that current solution $\big(w_0^{(i)}, w_1^{(i)}, w_2^{(i)}\big)$ of the master at iteration i satisfies $\big\|\big(w_1^{(i)}, w_2^{(i)}\big)\big\|_p \geq \Delta$ for infinitely many iterations. Consider ϵ defined in Lemma 3.7 for ε and $\delta_\varepsilon = \Delta/2$. By Lemma 3.5, after a finite number of iterations the current solution satisfies $\big(w_0^{(i)}, w_1^{(i)}, w_2^{(i)}\big)^\top \in \overline{\overline{\mathcal{V}}}_\epsilon^{(3)}$, which by Lemma 3.7 implies that $\big(w_0^{(i)}, w_1^{(i)}, w_2^{(i)}\big)^\top \in \mathcal{V}_\varepsilon^{(3)}$ contradicting with our assumption. Hence, this sequence of solutions converges to zero.

In Vinel and Krokhmal (2014c), Proposition 6 we have shown that a finite number cutting planes is sufficient to achieve any preselected accuracy (11) in the case of p-order cone constraints. In the current context it implies that solution $\big(w_0^{(i)}, w_1^{(i)}, w_2^{(i)}\big)^\top \in \mathcal{P}_\epsilon^{(3)}$ for any preselected ϵ after finitely many applications of the auxiliary cuts, where $\mathcal{P}_\epsilon^{(3)}$ is defined in Lemma 3.8. Taking into account Lemmas 3.6

and 3.8 this implies that $\big(w_0^{(i)}, w_1^{(i)}, w_2^{(i)}\big)^\top \in \mathcal{V}_\varepsilon^{(3)}$ for all sufficiently small $\big(w_1^{(i)}, w_2^{(i)}\big)$, i.e, (w_1, w_2) does not converge to zero, completing the proof. □

Observe that the result of Proposition 3.4 essentially provides an exact algorithm for solving problem (2). Indeed, once a solution with a desired accuracy ε is found, an improved solution can be constructed by adding new cutting planes. In Vinel and Krokhmal (2014c) a similar result has been established, namely it has been shown that a cutting plane approximation procedure is guaranteed to terminate with ε-feasible solution in $O(\varepsilon^{-1})$ iterations for p-order cone programing and $O(\varepsilon^{-0.5})$ in the case of second-order cones. Yet, an upper bound on the number of iterations that can be obtained from the proof presented here would be excessively high due to the way this proof is constructed. It can be shown that this bound is at least not better than $O(\varepsilon^{-1.5})$, where the corresponding "big-O" constant is very large. At the same time, all our experiments with both conic and non-conic problems suggest that in practice only a small fraction of all possible facets is generated, i.e., the fact that this bound can be very restrictive may not be detrimental to real-life computational performance.

In conclusion of our discussion, we would like to comment on the relation between the feasible sets of the initial nonlinear model and the presented approximated problem. Let us denote as Feas(V) the set defined by constraints (1b) and (1c) and as Feas(V_ε) the approximation of Feas(V) according to (10). Next, we establish the conditions that guarantee that these feasible sets are "close" to each other (note that it is possible to find examples where Feas(V) is empty, while Feas(V_ε) is not). Following the results presented in Ben-Tal and Nemirovski (2001), Section 4 for the case of second-order cone approximation we can formulate the following result, which we present without a proof since the arguments in Ben-Tal and Nemirovski (2001) apply here as well.

Proposition 3.9. *Assume that the problem under consideration is: (i) strictly feasible, i.e., there exist* $\bar{\mathbf{x}}$ *and* $r > 0$ *such that*

$$\mathbf{H}\bar{\mathbf{x}} \le \mathbf{h},\ v^{-1}\Big(\sum_j p_j v(\mathbf{a}_j^\top \bar{\mathbf{x}} + b_j)\Big) \le \mathbf{a}_0^\top \bar{\mathbf{x}} + b_0 - r, \tag{14a}$$

and (ii) "semibounded", i.e., there exists $R > 1$ *such that*

$$\mathbf{H}\mathbf{x} \le \mathbf{h},\ v^{-1}\Big(\sum_j p_j v(\mathbf{a}_j^\top \mathbf{x} + b_j)\Big) \le \mathbf{a}_0\mathbf{x} + b_0 \quad \Rightarrow \quad \mathbf{a}_0^\top \mathbf{x} + b_0 \le R. \tag{14b}$$

Then for every $\varepsilon > 0$ *such that* $\gamma(\varepsilon) = R\varepsilon/r < 1$, *one has* $\gamma(\varepsilon)\bar{\mathbf{x}} + (1-\gamma(\varepsilon))\ \mathrm{Feas}(V_\varepsilon) \subset \mathrm{Feas}(V) \subset \mathrm{Feas}(V_\varepsilon)$.

3.3 Branch-and-Bound Method

Now that an efficient approximation procedure for solving continuous relaxations is determined, it can be incorporated in a branch-and-bound method due to Vielma et al. (2008). Namely, we consider a master mixed-integer linear programming (MILP) problem (denoted as P_1), which is constructed from problem (2) by substituting (2b) with a set of initial cutting planes of the form (13). The solution procedure consists of applying a regular branch-and-bound method to P_1, with two adjustments. First, lower bounds obtained from the continuous relaxations of P_1 are found by applying the approximation scheme due to Proposition 3.4 with a preselected value of $\varepsilon = \varepsilon_1$. Note that it is not necessary to remove any of the added cutting planes before proceeding to the next node of the solution tree, since these constraints are globally feasible. Second, when an integer-valued solution of P_1 is found, in order to check its feasibility with respect to the exact nonlinear formulation and declare incumbent or branch further, the exact continuous relaxation of P_1 must be solved with bounds on the relaxed values of variables $\mathbf{x}$ determined by the integer-valued solution in question (see, Vielma et al., 2008 for more details and formal analysis). In order to solve the exact relaxation, we once again employ Proposition 3.4, that is to say, we construct a second problem P_2, which represents a continuous relaxation of (2). In this case, we solve it using the same cutting plane procedure due to Proposition 3.4 but with $\varepsilon = \varepsilon_2 \ll \varepsilon_1$ instead. A sufficiently small value of ε_2 guarantees an essentially exact solution.

Note that it has been previously observed (see, Vielma et al., 2008; Vinel and Krokhmal, 2014c) that ε_1 can be selected to be relatively large and still provide promising computational results, which explains the relation $\varepsilon_2 \ll \varepsilon_1$ above. Note also that in this case the described procedure can be viewed as a repetitive resolving of relatively small-scale LP problems P_1, which can benefit from warm-start routines, guided by a regular branch-and-bound, with occasional calls to a larger-scale P_2.

4 Valid Inequalities

4.1 Existing Approaches

It is well-known in the literature that valid inequality theory has been essential in development of efficient solvers, particularly in mixed-integer linear programming (MILP). Building on this success, various approaches to generating valid inequalities have been proposed for mixed-integer nonlinear programming (MINLP) problems. To name a few: Atamtürk and Narayanan (2010, 2011) have proposed mixed-inter rounding (MIR) and conic lifted cuts for conic programming problems; Stubbs and Mehrotra (1999) studied cutting plane theory in 0-1 mixed-convex programming; Çezik and Iyengar (2005) proposed Chvatal-Gomory cuts in conic programming; Bonami (2011) have considered lift-and-project cuts. There have also been a series of publications addressing possible approaches to designing disjunctive (or split) cuts in MINLP (for example, Saxena et al. 2008; Burer and Saxena 2012; Cadoux 2010; Kılınç et al. 2010; Modaresi et al. 2015 among others).

In this section we consider two approaches for generation of valid inequalities for the MINLP problem (2). First, we discuss lifted nonlinear cuts building on the developments in Atamtürk and Narayanan (2011) and Vinel and Krokhmal (2014a). Afterwards, we will present a simple geometric argument that allows us to construct a class of linear disjunctive cuts valid for our feasible set.

4.2 Lifted Nonlinear Cuts

A lifting procedure for conic mixed-integer programming has been proposed in Atamtürk and Narayanan (2011). Authors introduced a lifting scheme, which provides a way of generating new conic valid inequalities for mixed-integer conic sets. We have employed this approach for solving MIpOCP problems in Vinel and Krokhmal (2014a) and obtained promising numerical results for a class of risk-averse portfolio optimization models. While this technique has been proposed as a way to generate conic cuts for conic feasible sets, we show next that it can be extended for V-sets as well. As it will be clear from our discussion below, our main contribution here lies in the reformulation of the procedure in nonconic terms, while all of the proofs directly follow from the previous developments in Atamtürk and Narayanan (2011); Vinel and Krokhmal (2014a).

In this section we will closely follow the notation introduced in Atamtürk and Narayanan (2011). Once again, consider set $\mathcal{V}^{(m+1)}$ defined by (6). It is going to play the role of conic feasible set used in Atamtürk and Narayanan (2011). We can then define

$$T^n(\mathbf{b}) := \left\{ \mathbf{x}^i \in X^i \,\middle|\, \mathbf{b} - \sum_{i=0}^{n} \mathbf{A}_i \mathbf{x}^i \in \mathcal{V}^{(m+1)} \right\}, \tag{15}$$

where each X^i is a mixed-integer set in $\mathbb{R}^{n_i}$ and $\mathbf{A}^i$ and $\mathbf{b}$ are of appropriate dimensions. It is also assumed that $0 \in X^i$ for all i. Suppose that $u : \mathbb{R} \mapsto \mathbb{R}$ satisfies the same assumptions as function v and construct set $\mathcal{U}^{(m+1)}$ analogously to $\mathcal{V}^{(m+1)}$. Let us further assume that inequality $\mathbf{g} - \mathbf{F}^0\mathbf{x}^0 \in \mathcal{U}^{(m+1)}$ is valid for $T^0(\mathbf{b})$. Atamtürk and Narayanan (2011) show how this inequality can be lifted by computing $\mathbf{F}^\ell \in \mathbb{R}^{n_\ell}$ for $\ell = 1, \ldots, i$ so that cut $\mathbf{g} - \sum_{\ell=0}^{i} \mathbf{F}^\ell \mathbf{x}^\ell \in \mathcal{U}^{(m+1)}$ is valid for $T^i(\mathbf{b})$ when sets $\mathcal{V}^{(m+1)}$ and $\mathcal{U}^{(m+1)}$ are proper cones. Then, the following theorem can be shown to hold for a lifting set $\Phi^i(\mathbf{v})$ defined as

$$\Phi^i(\mathbf{v}) := \left\{ \mathbf{d} \in \mathbb{R}^p \,\middle|\, \mathbf{g} - \sum_{i=0}^{n} \mathbf{F}^i \mathbf{x}^i - \mathbf{d} \in \mathcal{U}^{(m+1)} \text{ for all } (\mathbf{x}^0, \ldots, \mathbf{x}^i) \in T^i(\mathbf{b} - \mathbf{v}) \right\}.$$

Recall also that a parametrized set $\Phi(\mathbf{v})$ is called *superadditive* on $\mathbb{R}^m$ if $\Phi(\mathbf{u}) + \Phi(\mathbf{v}) \subset \Phi(\mathbf{u} + \mathbf{v})$ for all $\mathbf{u}$ and $\mathbf{v}$, where $\Phi(\mathbf{u}) + \Phi(\mathbf{v})$ denotes the usual Minkowski sum.

Theorem 4.1. *1. $\Phi^i(\mathbf{v})$ is closed and convex.*

2. $0 \in \Phi^i(0)$

3. $\Phi^{i+1}(\mathbf{v}) \subset \Phi^i(\mathbf{v})$

4. $\mathbf{F}^1, \ldots, \mathbf{F}^{i+1}$ generate a valid inequality for $T^{i+1}(\mathbf{b})$ iff $\mathbf{F}^{i+1}\mathbf{x}^i \in \Phi^i(\mathbf{A}^{i+1}\mathbf{x}^i)$ for all $\mathbf{x}^i$.

5. If $\Omega(\mathbf{v}) \subset \Phi_0(\mathbf{v})$ is superadditive, then $\mathbf{F}^1, \ldots, \mathbf{F}^{i+1}$ generate a valid inequality for $T^{i+1}(\mathbf{b})$ whenever $\mathbf{F}^{i+1}\mathbf{x}^i \in \Omega(\mathbf{A}^{i+1}\mathbf{x}^i)$ for all $\mathbf{x}^i$.

Proof. Since the arguments establishing the analogous results in Atamtürk and Narayanan (2011) do not rely on the conic assumption, we believe it to be unnecessary to repeat those here. □

As it was noted above, we employed an analogous result for the case of p-order cones (i.e., $\mathcal{V}^{(m+1)} = \{(x_0, \mathbf{x}) \in \mathbb{R}^{m+1} \mid x_0 \geq \|\mathbf{x}\|_p\}$) in Vinel and Krokhmal (2014a). As it turns out, our results regarding lifted cuts presented there can be carried through without major changes in case of V-sets. Particularly, we can consider set $\widehat{T}^n(b)$ as

$$\widehat{T}^n(b) := \left\{ (\mathbf{x}, y, t) \in \mathbb{Z}^n_+ \times \mathbb{R}^2_+ \,\middle|\, v\left(\left[\sum_{i=1}^{n} a_i x_i - b \right]_+ \right) + v(y) \leq v(t) \right\},$$

and then show that the following claim holds.

Proposition 4.2. *Inequality*

$$v\left(\left[(1 - f)(x - \lfloor b \rfloor) + \sum_{i=1}^{n} \alpha_i x_i \right]_+ \right) + v(y) \leq v(t) \tag{16}$$

is valid for $\widehat{T}^n(b)$, where $[a]_+ = \max\{0, a\}$, $\alpha_i = \left[\dfrac{a_i - b + \lfloor b \rfloor (1 - f)}{M}\right]_+$, $f = b - \lfloor b \rfloor$, and M is a constant such that $x_i \leq M$ for all i.

This result is a very limited application of Theorem 4.1. Indeed, here we are considering the case when the set $\mathcal{U}^{(m+1)}$ is the same as the initial set $\mathcal{V}^{(m+1)}$ and, moreover, not only all the analysis is restricted to three-dimensional nonlinear constraints, but also the second dimension (represented by variable y) is assumed to be continuous (in other words, integral structure of the second dimension is relaxed). Despite these simplifications, it was demonstrated in Vinel and Krokhmal (2014a) that such an approach may yield promising computational results in MIpOCP problems. In Section 5 we will numerically analyze this procedure in mixed-integer programming with certainty equivalent constraints. In fact, two of these stipulations can be viewed as natural assumptions for the task of deriving valid inequalities in our case. Observe that due to the tower-of-variables technique presented in Section 3 the constraints are already represented in three-dimensional form, and furthermore, it is also highly undesirable from computational perspective to consider $\mathcal{U}^{(m+1)}$ different from initial set $\mathcal{V}^{(m+1)}$, since this would result in additional numerical challenges associated with the new type of nonlinearity introduced to the problem.

4.3 Linear Disjunctive Cuts

Throughout this section we will use the following notation: $\bar{\mathbf{x}} = (x_0, \mathbf{x}) \in \mathbb{R}^{n+1}$. We will also reformulate sets defined y certainty equivalent constraints as

$$\bar{\mathbf{x}} \in \mathcal{K}, \quad \mathcal{K} := \left\{ \bar{\mathbf{x}} \in \mathbb{R}^{n+1} \;\middle|\; F(\mathbf{x}) \leq x_0 \right\}, \quad F(\mathbf{x}) := v^{-1}\Bigg(\sum_{j=1}^{m} v\big(|\mathbf{a}_j^\top \mathbf{x} + b_j|\big) \Bigg), \quad \mathbf{x} \in \mathbb{Z}_+^n. \qquad (17)$$

Note that here we consider $F(\mathbf{x}) := v^{-1}\Big(\sum\limits_{j=1}^{m} v\big(|\mathbf{a}_j^\top \mathbf{x} + b_j|\big) \Big)$ instead of possible $F(\mathbf{x}) := v^{-1}\Big(\sum\limits_{j=1}^{m} v([\mathbf{a}_j^\top \mathbf{x} + b_j]_+) \Big)$, which would be in accordance with the stochastic programming motivations. Such a choice simplifies some of our development below, and since it results in a relaxed set $\mathcal{K}$, any valid inequality obtained for $\mathcal{K}$ will be valid for problem (2) as well.

Disjunctive or split cuts have been extensively studied in the literature, especially when applied to MIP problems (Balas, 1971). This approach is based on a very intuitive idea: consider disjunction $x_k \leq \pi_0 \vee x_k \geq \pi_1 = \pi_0 + 1$ with $\pi_0 \in \mathbb{Z}_+$, where $k \in 1, \ldots, n$ is preselected. Due to integrality condition there are no feasible solutions outside of this disjunction, hence, system (17) implies that

$$\bar{\mathbf{x}} \in \operatorname{conv}\left(\left\{ \begin{matrix} \bar{\mathbf{x}} \in \mathcal{K} \\ x_k \leq \pi_0 \end{matrix} \right\} \bigcup \left\{ \begin{matrix} \bar{\mathbf{x}} \in \mathcal{K} \\ x_k \geq \pi_1 \end{matrix} \right\} \right). \qquad (18)$$

Consequently, any inequality describing this convex hull is valid for the feasible region of (17). Moreover, in the case of mixed-integer linear programming (MILP) all the sets involved (including the convex hull above) are polyhedral, which substantially simplifies the construction procedures, and hence, increases the effectiveness of the cuts. There also exists a considerable amount of literature on generalizing this approach for MINLP problems (Burer and Saxena, 2012; Cadoux, 2010; Kılınç et al., 2010). Recently, various efforts to design nonlinear disjunctive cuts have also been presented (see Andersen and Jensen, 2013; Belotti et al., 2012; Bienstock and Michalka, 2014; Modaresi et al., 2015; Burer and

Kılınç-Karzan, 2014; Kılınç-Karzan, 2015; Kılınç-Karzan and Yıldız, 2015). In some cases (see, for example, Modaresi et al., 2015) it may be possible to describe the convex hull (18) using a single nonlinear constraint; in particular, such a description is available for second-order conic sets. Note that many of the works mentioned above look at the problem in settings considerably more general than described here. Next, we will study applicability of the disjunctive cut framework to sets of form (17).

The first question that we could ask here is whether it might be better to aim at finding a closed-form nonlinear description of (18) following one of the recent developments mentioned above, or whether a simpler linear description could be more useful in this case. Note that if such a nonlinear description is to be found and then used in a numerical procedure to solve problem (2), then it is highly desirable for it to be expressed in the same form as the nonlinear constraint already present in the problem. For example, if the computational procedures used are tailored specifically to the constraints already present in the problem, then addition of a new type of "nonlinearity" that is not comparable with these approaches may be impractical. The descriptions obtained in the literature for mixed-integer second-order cone programming express the convex hull of the disjunction in terms of quadratic sets, essentially preserving the second-order conic nonlinearity in many practical cases, thus justifying the approach.

Consequently, consider (18) with certainty equivalent set (17). We can conclude that it is desirable that its description is itself represented in terms of function F defined in (17). At the same time, consider supporting hyperplanes for (18). It is easy to see that for at least some of such hyperplanes, their intersection with the convex hull is a straight line segment in between $x_k = \pi_0$ and $x_k = \pi_1$. On the other hand, a boundary of a set defined in terms of function F does not in general contain such segments, since it is nonconic. Thus, it is reasonable to expect that such a closed-form description of convex hull (18) cannot be expressed in terms of function F alone. With this in mind, we propose to concentrate on a more modest goal of constructing supporting hyperplanes for (18), or in other words, linear disjunctive cuts.

Next we propose an intuitive idea for a procedure aimed at avoiding difficulties associated with the general disjunctive cut generation techniques available in the literature by exploiting specific structural properties of (18). Suppose that we have selected a point $\bar{\mathbf{x}}^0 \in \mathcal{K}$ such that $x_k^0 = \pi_0$, i.e., $\bar{\mathbf{x}}^0$ is located on one side of the disjunction. Given such a $\bar{\mathbf{x}}^0$, find $\bar{\mathbf{x}}^1 \in \mathcal{K}$ such that $x_k^1 = \pi_1$ and $\partial_{(k)} F(\mathbf{x}^1) \cap \partial_{(k)} F(\mathbf{x}^0) \neq \emptyset$, where subdifferential $\partial_{(k)}$ is taken with respect to variables x_i, $i \neq k$. A linear disjunctive cut is then constructed as a constraint $\sum\limits_i \alpha_i x_i + \beta \leq x_0$, where $(\alpha_1, \ldots, \alpha_{k-1}, \alpha_{k+1}, \ldots, \alpha_n)^\top \in \partial_{(k)} F(\mathbf{x}^0) \cap \partial_{(k)} F(\mathbf{x}^1)$, while α_k and β are selected in such a way that $\sum\limits_i \alpha_i x_i^0 + \beta = x_0^0$ and $\sum\limits_i \alpha_i x_i^1 + \beta = x_0^1$. Geometrically it means that the constructed hyperplane is such that it passes through both $\bar{\mathbf{x}}^0$ and $\bar{\mathbf{x}}^1$ and is supporting to both sides of the disjunction at these points. Clearly, convexity of $\mathcal{K}$ implies that this cut is valid.

The described procedure can be formulated as follows. Given $\bar{\mathbf{x}}^0 \in \mathbb{R}^n \times \mathbb{R}$, $k \in \{1, \ldots, n\}$, $\pi_0, \pi_1 \in \mathbb{Z}$, and function $v : \mathbb{R} \mapsto \mathbb{R}$, find $\bar{\mathbf{x}}^1 \in \mathbb{R}^n \times \mathbb{R}$ and $(\boldsymbol{\alpha}, \beta) \in \mathbb{R}^n \times \mathbb{R}$ such that

$$\begin{cases} \sum_{i=1}^{n} \alpha_i x_i^0 + \beta = x_0^0 & \text{(19a)} \\ \sum_{i=1}^{n} \alpha_i x_i^1 + \beta = x_0^1 & \text{(19b)} \\ x_k^0 = \pi_0 & \text{(19c)} \\ x_k^1 = \pi_1 & \text{(19d)} \\ F(\mathbf{x}^0) = x_0^0 & \text{(19e)} \\ F(\mathbf{x}^1) = x_0^1 & \text{(19f)} \\ (\alpha_1, \ldots, \alpha_{k-1}, \alpha_{k+1}, \ldots, \alpha_n)^\top \in \partial_{(k)} F(\mathbf{x}^0) \cap \partial_{(k)} F(\mathbf{x}^1). & \text{(19g)} \end{cases}$$

Let us denote by $\mathcal{P}$ the half space valid for the linear cut $\sum_{i=1}^{n} \alpha_i x_i + \beta \leq x_0$, i.e., $\mathcal{P} := \{\bar{\mathbf{x}} \in \mathbb{R}^{n+1} \mid \sum_{i=1}^{n} \alpha_i x_i + \beta \leq x_0\}$. By $\partial\mathcal{K}$ and $\partial\mathcal{P}$ we will understand boundaries of these sets. In Proposition 4.5 below we will establish validity of this approach, but first we consider a few useful lemmas.

Lemma 4.3. *The following statements hold:*

(i) $\bar{\mathbf{x}}^i \in \partial\mathcal{K}$ *for* $i = 0, 1$*;*

(ii) $\bar{\mathbf{x}}^i \in \partial\mathcal{P}$ *for* $i = 0, 1$*;*

(iii) if $\bar{\mathbf{x}} \notin \mathcal{P}$ *and* $x_k = \pi_0$*, then* $\bar{\mathbf{x}} \notin \mathcal{K}$*;*

(iv) if $\bar{\mathbf{x}} \notin \mathcal{P}$ *and* $x_k = \pi_1$*, then* $\bar{\mathbf{x}} \notin \mathcal{K}$*.*

Proof. Claims (i) and (ii) follow immediately from (19a)–(19b) and (19e)–(19f). In order to see that (iii) holds, note that (19) implies that on the space restricted by $x_k = \pi_0$ the set $\partial\mathcal{P}$ is a supporting hyperplane for the set $\partial\mathcal{K}$, which immediately implies (iii). Analogous observation holds for (iv). □

Lemma 4.4. *The following statemets hold:*

(i) if $\bar{\mathbf{x}} \in \partial\mathcal{P}$ *and* $x_k < \pi_0$*, then* $\bar{\mathbf{x}} \notin \operatorname{int}\mathcal{K}$*;*

(ii) if $\bar{\mathbf{x}} \in \partial\mathcal{P}$ *and* $x_k > \pi_1$*, then* $\bar{\mathbf{x}} \notin \operatorname{int}\mathcal{K}$*.*

Proof. First, consider claim (i). Suppose that the contrary holds, i.e., $\bar{\mathbf{x}} \in \operatorname{int}\mathcal{K}$. Then, there exists an $\varepsilon > 0$ such that $\bar{\mathbf{y}} = (x_0 - \varepsilon, \mathbf{x}) \in \mathcal{K}$ and $\bar{\mathbf{y}} \notin \mathcal{P}$. Now consider the segment connecting points $\bar{\mathbf{y}}$ and $\bar{\mathbf{x}}^1$, i.e., the set $\{\lambda\bar{\mathbf{y}} + (1 - \lambda)\bar{\mathbf{x}}^1 \mid \lambda \in [0, 1]\} =: \mathcal{T}$. Since both $\bar{\mathbf{y}} \in \mathcal{K}$ and $\bar{\mathbf{x}}^1 \in \mathcal{K}$, then $\mathcal{T} \subset \mathcal{K}$. Since $\pi_0 < \pi_1$ and $x_k < \pi_0$, then there exists $\bar{\mathbf{z}} = (z_0, \mathbf{z}) \in \mathcal{T}$ such that $z_k = \pi_0$. At the same time, $\bar{\mathbf{z}} \notin \mathcal{P}$ as $\bar{\mathbf{y}} \notin \mathcal{P}$ while $\bar{\mathbf{x}}^1 \in \partial\mathcal{P}$. Thus, by Lemma 4.3 (iii) one has that $\bar{\mathbf{x}} \notin \mathcal{K}$, which contradicts the assumption above. Hence, claim (i) holds. Statement (ii) can be proved analogously. □

Proposition 4.5. *If* $\bar{\mathbf{x}} \in \mathcal{K}$ *and* $x_k \notin [\pi_0, \pi_1]$*, then* $\bar{\mathbf{x}} \in \mathcal{P}$*.*

Proof. Assume the contrary, i.e., that $\bar{\mathbf{x}} \notin \mathcal{P}$, which means that $\sum\limits_{i=1}^{n} \alpha_i x_i + \beta > x_0$. Then, there exists $\bar{\mathbf{y}} = (x_0 + \varepsilon, \mathbf{x}) \in \partial\mathcal{P}$ (take $\varepsilon = \sum\limits_{i=1}^{n} \alpha_i x_i + \beta - x_0$). Moreover, by definition $\bar{\mathbf{y}} \in \operatorname{int}\mathcal{K}$. If $x_k < \pi_0$, then this conclusion contradicts Lemma 4.4 (i), otherwise, $x_k > \pi_1$ and the conclusion above contradicts Lemma 4.4 (ii). □

This result guarantees that the cut generated by (19) is feasible for (18). Moreover, it is easy to see that for any $\tilde{\beta} > \beta$ and $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}$ the corresponding cut is not feasible due to Lemma 4.3 (i). Hence, system (19) produces a *tight cut* in the sense that it cannot be improved by an affine transformation.

Observe that $\bar{\mathbf{x}}^{(1)} \in \mathbb{R}^n \times \mathbb{R}$ and $(\boldsymbol{\alpha}, \beta) \in \mathbb{R}^n \times \mathbb{R}$ are the unknowns in the system (19). Given a specific value of $\mathbf{x}^1 \in \mathbb{R}^n$ such that $\partial_{(k)} F(\mathbf{x}^{(0)}) \cap \partial_{(k)} F(\mathbf{x}^{(1)}) \neq \emptyset$ it is easy to determine the rest. Indeed, x_0^1 is uniquely defined by (19f), vector $(\alpha_1, \dots, \alpha_{k-1}, \alpha_{k+1}, \dots, \alpha_n)^\top$ can be selected according to (19g), and α_k and β are fixed by (19a) and (19f). Thus, the most challenging step in this procedure is the selection of $\mathbf{x}^1$ satisfying $\partial_{(k)} F(\mathbf{x}^{(0)}) \cap \partial_{(k)} F(\mathbf{x}^{(1)}) \neq \emptyset$. In the end of this section we will show that such $\mathbf{x}^1$ can always be found, but let us first note that this step can be numerically cumbersome since function F defined in (17) is only piecewise continuously differentiable. Consequently, we propose to employ another approximation procedure in order to achieve this goal. Namely, we consider substituting $|t| \approeq \sqrt{t^2 + \epsilon}$, and hence, defining $\tilde{F}(\mathbf{x}) := v^{-1}\Big(\sum\limits_{j=1}^{m} v\Big(\sqrt{(\mathbf{a}_j^\top \mathbf{x} + b_j)^2 + \epsilon}\Big)\Big)$. Then, $\tilde{F}$ is continuously differentiable and in order to find $\mathbf{x}^1$ we need to solve a system of nonlinear equations with a given $\mathbf{x}^0$ $\frac{\partial \tilde{F}}{\partial x_i}(\mathbf{x}^1) = \frac{\partial \tilde{F}}{\partial x_i}(\mathbf{x}^{(0)})$, $i = 1, \dots, k-1, k+1, \dots, n$. After this system is solved, the validity of the found $\mathbf{x}^1$ can be verified directly by comparing $\partial_{(k)} F(\mathbf{x}^{(0)})$ and $\partial_{(k)} F(\mathbf{x}^{(1)})$.

In order to establish existence of vector $\mathbf{x}^1$, let us introduce some additional notation. Without loss of generality we will assume that $k = 1$, and let us define $\tilde{\mathbf{a}}_j = (a_{2j}, \dots, a_{nj})^\top$ for all j, $\tilde{\mathbf{A}} = (\tilde{a}_1^\top; \dots; \tilde{a}_m^\top)$ and $\tilde{\mathbf{x}} = (x_2, \dots, x_n)^\top$, i.e, expressions with tildes represent the values restricted to variables $(x_2, \dots, x_n)$. Let us also define $F^\ell(\tilde{\mathbf{x}}) := v^{-1}\Big(\sum\limits_j p_j v\big(|\tilde{\mathbf{a}}_j^\top \tilde{\mathbf{x}} + b_j + a_{1j}\pi_\ell|\big)\Big)$ for $\ell = 0, 1$.

Proposition 4.6. *Assuming that $\tilde{\mathbf{A}}$ is full rank, for any $\tilde{\mathbf{x}}^0$ there exists $\tilde{\mathbf{x}}^1$ such that $\partial F^1(\tilde{\mathbf{x}}^1) \cap \partial F^0(\tilde{\mathbf{x}}^0) \neq \emptyset$.*

Proof. Let us consider vector $\boldsymbol{\alpha} \in \partial F^0(\tilde{x}^1)$. By definition $\boldsymbol{\alpha}$ gives us a supporting hyperplane to epi F^0 at point $\tilde{x}^0$. Let us denote this hyperplane as P^0. The full rank of $\tilde{\mathbf{A}}$ guarantees that this hyperplane is non-degenerate, i.e., both functions F^ℓ substantially depend on all variables $\tilde{\mathbf{x}}$. In order to show that the claim of the proposition holds, we need to establish that there exists a supporting hyperplane to epi F^1 which is parallel to P^0.

First, let us assume that there exists a constant M such that $\operatorname{epi}(F^1 + M) \subset \operatorname{epi} F^0$. Then, the hyperplane P^0 is valid for $\operatorname{epi}(F^1 + M)$ (meaning that it does not intersect with it). In this case, since $\operatorname{dom} F^\ell = \mathbb{R}^{n-1}$, there exists a vertical translation of P^0, which is supporting to $\operatorname{epi}(F^1 + M)$. Clearly, this immediately implies the claim of the proposition.

Now, we will show that such a constant exists. Let us introduce an auxiliary variable vector $\mathbf{w} \in \mathbb{R}^m$ and functions $G^\ell(\mathbf{w}) := v^{-1}\Big(\sum\limits_j p_j v\big(|w_j + b_j + a_{1j}\pi_\ell|\big)\Big)$, i.e., $G^\ell(\mathbf{w}) = F^\ell(\tilde{\mathbf{x}})$ if $w_j = \tilde{\mathbf{a}}_j^\top \tilde{\mathbf{x}}$ for $j = 2, \dots, m$ and $\ell = 0, 1$. Clearly, both G^ℓ are proper and convex and $\operatorname{dom} G^\ell = \mathbb{R}^m$. Consider recession function $G^\ell 0^+$ of G^ℓ (see, e.g., Rockafellar, 1997 for details), $(G^\ell 0^+)(\mathbf{w}) = \lim_{\lambda \downarrow 0} \lambda G^\ell(\lambda^{-1}\mathbf{w}) = \lim_{\lambda\downarrow 0} \lambda v^{-1}\Big(\sum_j p_j v\big(|\lambda^{-1}w_j + b_j + a_{1j}\pi_\ell|\big)\Big)$. Observe that since $v^{-1}(0) = 0$ and v^{-1} is nondecreasing and concave, then $v^{-1}(mt) \le mv^{-1}(t)$. With this in mind,

$$\begin{aligned}(G^\ell 0^+)(\mathbf{w}) &\le \lim_{\lambda\downarrow 0} \lambda v^{-1}\Big(m \max_j \Big\{v\big(|\lambda^{-1}w_j + b_j + a_{1j}\pi_\ell|\big)\Big\}\Big) \\ &\le \lim_{\lambda\downarrow 0} m \max_j \Big\{|w_j + \lambda(b_j + a_{1j}\pi_\ell)|\Big\} = m\max\{|w_j|\} < +\infty.\end{aligned}$$

Hence, $\operatorname{dom}(G^l 0^+) = \mathbb{R}^m$, which implies that G^ℓ is Lipschitz continuous on $\mathbb{R}^m$ with Lipschitz constant $L = \sup\{(G^\ell 0^+)(\mathbf{w}) \mid \|\mathbf{w}\| = 1\}$ (see, for example, Auslender and Teboulle, 2003, Proposition 2.5.5). Further, note that $G^0(\mathbf{w}) = G^1(\mathbf{w} + \mathbf{\Delta})$, where $\Delta_j = a_{1j}\pi_0 - a_{1j}\pi_1$. Thus, $|G^1(\mathbf{w}) - G^0(\mathbf{w})| = |G^1(\mathbf{w}) - G^1(\mathbf{w}+\mathbf{\Delta})| \le L\|\mathbf{\Delta}\|$. Finally, if we set $w_j = \tilde{\mathbf{a}}_j^\top \tilde{\mathbf{x}}$, then $|F^1(\tilde{\mathbf{x}}) - F^0(\tilde{\mathbf{x}})| = |G^1(\mathbf{w}) - G^0(\mathbf{w})| \le M$, if $M = L\|\Delta\| < +\infty$, which completes the proof. □

Finally, it is necessary to discuss practical selection of k, π_0, π_1 and $\bar{\mathbf{x}}^0$. If the cut generation procedure is implemented in a branch-and-bound setting, it can be assumed that a solution of a relaxed problem $\bar{\mathbf{x}}^{\text{relax}}$ is known beforehand. Hence, it is natural to select $k \in \big\{\{1,\dots,n\} \,\big|\, x_k^{\text{relax}} \notin \mathbb{Z}\big\}$, $\pi_0 = \lfloor x_k^{\text{relax}} \rfloor$ and $\pi_1 = \pi_0 + 1$. Since the goal of generating a valid inequality is to cutoff $\bar{\mathbf{x}}^{\text{relax}}$, then it is also natural to pick $\bar{\mathbf{x}}^0$ according to $x_i^0 = x_i^{\text{relax}}$, for $i \ne k$, $x_k^0 = \pi_0$ and $x_0^0 = F(\mathbf{x}^0)$.

Before concluding this section, it is worth noting that the proposed procedure does not represent a general way to generate a split closure for the feasible set (17). Alternatively, it can be seen as a quick and simple numerical procedure to find a valid inequality that can cuts off the current non-integral solution.

5 Numerical Experiments

In this section we will report the results of numerical case studies designed to evaluate the performance the proposed techniques. As it has been discussed in the introduction, our main interest in the problem class considered in this paper stems from risk-averse approaches to stochastic programming, and hence we base our numerical experiments on this application area. Next, we will discuss the particular formulation used in our study.

5.1 Model Formulation

According to discussion in Section 2, a scenario-based formulation for risk-minimization problem $\min\{\rho(X(\mathbf{x},\omega)) \mid \mathbf{x} \in \mathcal{X}\}$, where ρ is a certainty equivalent measure of risk, $X(\mathbf{x},\omega_j) = \mathbf{a}_j^\top \mathbf{x} + b_j$, and $\mathbf{x} \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2}$, reduces to

$$\begin{aligned}\min \quad & \eta + \frac{1}{1-\alpha} t && \text{(20a)}\\ \text{s. t.} \quad & t \ge v^{-1}\Big(\sum_{j=1}^m p_j v(w_j)\Big) && \text{(20b)}\\ & w_j \ge \mathbf{a}_j^\top \mathbf{x} + b_j - \eta, \quad j = 1,\dots,m && \text{(20c)}\\ & \mathbf{x} \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2},\ \mathbf{w} \ge 0,\ \eta \in \mathbb{R}. && \text{(20d)}\end{aligned}$$

Here, function v is nondecreasing, convex and such that $v^{-1}\Big(\sum\limits_{j=1}^{m} p_j v(w_j)\Big)$ is convex in $\mathbf{w}$. Some of the promising choices for v have been discussed in Vinel and Krokhmal (2014b). Particularly, $v(t) = [t]_+$ leads to the definition of Conditional-Value-at-Risk (CVaR), a popular risk measure in many stochastic programming applications (see, Rockafellar and Uryasev, 2000, 2002 for more details). It has also been observed in the literature (see, Krokhmal, 2007; Vinel and Krokhmal, 2014b) that $v(t) = [t]_+^p$ for $p > 1$ and $v(t) = e^{[t]_+} - 1$ can lead to some encouraging decision making performance in the presence of the so-called heavily tailed distributions of risks.

Computationally, problem (20) with linear $v(t) = [t]_+$ leads to a linear mixed-integer programming problem, while $v(t) = [t]_+^p$ results in a MIpOCP problem, both of which have been studied before as previously discussed. Since the solution approaches proposed in the current paper are targeted towards the more general non-conic cases, in our numerical experiments here we concentrate on the case of $v(t) = e^{[t]_+} - 1$, which has been referred to in Vinel and Krokhmal (2014b) as *Log-Exponential Convex Risk (LogExpCR) measure*.

We utilize financial portfolio optimization model as the basic decision making problem in our study. It is often used as a test model in the risk-averse stochastic programming literature, and additionally, enjoys abundance of real-life historic data that can be used in various case studies.

In a standard risk-reward portfolio selection problem, a set of n financial assets is considered. Then, the loss is defined as the negative portfolio return, $X(\mathbf{x}, \omega) = -\mathbf{r}(\omega)^\top \mathbf{x}$, where $\mathbf{x}$ stands for the vector of portfolio weights, and $\mathbf{r} = \mathbf{r}(\omega)$ is the uncertain vector of assets' returns. Consequently, the goal is to select portfolio weights $\mathbf{x}$ in such a way that the risk associated with this choice, as evaluated by a risk measure ρ, is minimized, while maintaining a certain predefined value of the expected return (reward):

$$\min_{\mathbf{x}\in\mathbb{R}^n_+} \Big\{\rho(-\mathbf{r}^\top\mathbf{x}) \;\Big|\; \mathsf{E}(\mathbf{r}^\top\mathbf{x}) \geq \bar{r},\ \mathbf{1}^\top\mathbf{x} \leq 1\Big\}, \tag{21}$$

where $\bar{r}$ is the prescribed level of the expected return, $\mathbf{x} \in \mathbb{R}^n_+$ denotes the no-short-selling requirement, and $\mathbf{1} = (1, \ldots, 1)^\top$.

We consider two types of investment constraints that lead to mixed-integer portfolio optimization problems. Many floor trading systems mandate that assets can only be bought in "lots" of shares (for instance, in multiples of 1,000 shares), which leads to a *lot-buying constrained portfolio optimization* model:

$$\min_{\mathbf{x}\in\mathbb{R}^n_+,\ \mathbf{z}\in\mathbb{Z}^n_+} \left\{\rho(-\mathbf{r}^\top\mathbf{x}) \;\middle|\; \mathsf{E}(\mathbf{r}^\top\mathbf{x}) \geq \bar{r}, \quad \mathbf{1}^\top\mathbf{x} \leq 1, \quad \mathbf{x} = \frac{L}{C}\,\mathrm{Diag}(\boldsymbol{\varpi})\,\mathbf{z}\right\}, \tag{22}$$

where L is the size of the lot, C is the investment capital (in dollars), and vector $\boldsymbol{\varpi} \in \mathbb{R}^n$ represents the prices of assets. Similarly, it may be desirable for a portfolio to contain no more than a prescribed number of assets, which leads to *cardinality-constrained portfolio optimization* model:

$$\min_{\mathbf{x}\in\mathbb{R}^n_+,\ \mathbf{z}\in\{0,1\}^n} \Big\{\rho(-\mathbf{r}^\top\mathbf{x}) \;\Big|\; \mathsf{E}(\mathbf{r}^\top\mathbf{x}) \geq \bar{r},\ \ \mathbf{1}^\top\mathbf{x} \leq 1,\ \ \mathbf{x} \leq \mathbf{z},\ \ \mathbf{1}^\top\mathbf{z} \leq Q\Big\}, \tag{23}$$

where Q is the maximum number of assets in the portfolio.

If historical data (scenarios) for the assets' returns is known, then problem (21) with a LogExpCR risk measure can be formulated in the form of (20):

$$\begin{aligned}
\min \quad & \eta + (1-\alpha)^{-1} t \\
\text{s. t.} \quad & t \geq \ln\Big(\sum_{j=1}^{m} p_j e^{w_j} \Big), \\
& \mathbf{w} + (\mathbf{r}_1, \ldots, \mathbf{r}_N)^\top \mathbf{x} + \mathbf{1}\eta \ \geq \ \mathbf{0}, \\
& \mathbf{x}^\top \Big(\sum_j p_j \mathbf{r}_j \Big) \geq \bar{r}, \\
& \mathbf{1}^\top \mathbf{x} \leq 1, \ \mathbf{x} \geq \mathbf{0}, \ \mathbf{w} \geq \mathbf{0}.
\end{aligned} \tag{24}$$

Problems (22) and (23) can be easily reformulated accordingly.

We used historical data for n assets chosen at random from the stocks traded on NYSE, such that historical prices are available for 5100 consecutive trading periods preceding December, 2012. Returns over m consequent 10-day periods starting at a (common) randomized date were used to construct the set of m equiprobable scenarios for the stochastic vector $\mathbf{r}$. The values of parameters were set as follows: $L = 1000, C = 100{,}000, Q = 5, \alpha = 0.9, \bar{r} = 0.005$. Historical values of the assets' returns have been scaled by multiplying parameter MUL, i.e., $r_{ij} = \text{MUL}\frac{\varpi_{i,j+10} - \varpi_{i,j}}{\varpi_{i,j}}$, where $\varpi_{i,j}$ is the close price of asset i at day j. Note that since LogExpCR measure is not positively homogeneous the value of MUL has an impact on the decisions preferred (see Vinel and Krokhmal, 2014b for more details).

5.2 Preliminary Study: Polyhedral Approximation Method for Convex Portfolio Optimization

As a preliminary computational study, we performed experiments with the convex formulation (21). Our goal here was to test the performance of the proposed cutting plane approximation procedure compared to the existing exact approaches. Namely, we implemented the iterative algorithm presented in Section 3.2 using CPLEX LP solver for iteratively resolving the master problem and compared it against MOSEK NLP interior point solver.

In addition, we also implemented a simpler version of the iterative approximation approach, which is applicable for the case of exponential constraints. This scheme follows the same iterative master cutting plane approach with the only difference being the structure of the tangent planes utilized. Observe that using simple variable substitution, the log-exponential constraint in (24): $t \geq \ln\Big(\sum_{j=1}^{m} p_j e^{w_j} \Big)$ can be equivalently expressed as

$$\sum_j p_j \xi_j \leq 1, \quad e^{w_j - t} \leq \xi_j, \quad \xi_j \geq 0, \quad j = 1, \ldots, m.$$

Consequently, the cutting planes (which are actually lines) in this case can be constructed as tangent to the nonlinear constraint of the form $e^x \leq y$. In the remainder of this section we will refer to this approach as *simple approximation procedure*. Our aim here is to verify whether the lifting procedure presented in Section 3.2 is superior to this more straightforward approach.

Results of this study are summarized in Table 1. For each combination of number of assets n, number of scenarios m and the value of scaling parameter MUL we generated 20 random instances based on the historical data as described above. The columns "MOSEK", "Simple" and "CP" correspond to the average solution time used by MOSEK NLP solver, iterative procedure based on simple approximation and lifted cutting planes procedure presented in Section 3.2 respectively. Column “MOSEK-CP” reports maximum absolute difference in portfolio values obtained with MOSEK and lifted cutting planes procedure, while column “MOSEK-SA” contains the maximum absolute difference between MOSEK and the simple approximation. The approximation accuracy as well as CPLEX and MOSEK feasibility and optimality tolerances were set to 10^{-6}.

Observe that for all instances the lifted approach outperforms the simple approximation both in terms of solution time and accuracy, which confirms the theoretical advantages of the lifting procedure discussed in Section 3.2. The lifted cutting plane approach returns portfolios that are mostly within the prescribed tolerance from the exact solutions due to MOSEK. Note that since the values reported are absolute differences, they naturally increase with the increase in the value of parameter MUL. Moreover, the approximation procedure finds these solutions significantly faster than MOSEK for the instances with $\text{MUL} = 10000$ and large-scale instances with $\text{MUL} = 1$ (recall that a change in the scaling parameter in fact changes the optimal solution, i.e., this scaling is more than a simple computational convenience). Additionally, MOSEK could not find an optimal solution returning an infinite portfolio value for some instances with $\text{MUL} = 10000$, whereby the polyhedral approximation-based procedure may be more stable numerically.

The goal of this preliminary study was to check whether the essentially exact algorithm based on the approximation procedure can be competitive against the state-of-the-art NLP solvers, and whether the introduction of lifting leads to computational improvement. We clearly observed that at least for the class of instances considered here, the cutting-plane method performs favorably compared to MOSEK NLP solver, outperforming it significantly for some instances. At the same time, the cutting plane procedure based on simple approximation scheme does not possess any favorable computational properties.

These results, obtained on convex problems, are also of significant importance in the context of branch-and-bound process for the corresponding mixed-integer programming models. It has been demonstrated in the literature that in the case of second-order cone programming a branch-and-bound method based on polyhedral approximation procedure can still outperform conventional approaches even while the approximation scheme itself may not result in computational improvement for the convex model (Glineur, 2000; Vielma et al., 2008). Our previous experience with solving p-order cone programming problems suggested a similar conclusion. In view of this, the results of this preliminary study allow us to confirm that the proposed approach to the mixed-integer model is promising. In the next subsection we will study this case directly.

5.3 Discrete Portfolio Optimization

CPLEX MIP and LP solvers have been used to implement the branch-and-bound method described in Section 3. Namely, callback routines have been employed in order to add approximating hyperplanes at each node of the solution tree, while a goal framework was utilized to guide branching. The exact algorithm based on approximation scheme presented in Section 3 has been used to verify incumbent solutions. The two families of valid inequalities have been employed by means of CPLEX callback routines. In our experiments we only added cuts at the root node of the branch-and-bound tree. A quasi-Newton’s method has been used to solve the underlying nonlinear systems of equations when finding linear split cuts presented in Section 4.3.

Two sets of experiments have been conducted to estimate the effects of the techniques proposed in the

		MUL = 1					MUL = 10000				
n	*m*	MOSEK	Simple	CP	MOSEK-CP	MOSEK-SA	MOSEK	Simple	CP	MOSEK-CP	MOSEK-SA
20	100	0.11	0.13	**0.07**	3.00E-06	1.64E-04	0.11	0.09	**0.05**	0.00E+00	6.00E-04
	200	**0.07**	0.32	0.08	2.00E-07	1.14E-05	0.21	0.25	**0.08**	1.00E-04	1.50E-03
	500	**0.14**	1.72	0.20	1.00E-06	3.30E-05	0.50	0.92	**0.17**	0.00E+00	1.20E-03
	1000	**0.28**	7.03	0.55	1.00E-06	3.10E-05	1.69	2.89	**0.30**	0.00E+00	7.00E-03
	2000	**0.66**	29.26	1.57	2.00E-06	3.40E-05	2.44	11.01	**0.58**	0.00E+00	6.00E-03
	5000	5.83	209.20	**5.54**	1.00E-06	1.71E-03	8.74	77.57	**1.41**	0.00E+00	4.00E-02
50	100	0.08	0.09	**0.03**	0.00E+00	2.52E-04	0.09	0.10	**0.05**	0.00E+00	1.00E-03
	200	0.17	0.36	**0.07**	1.00E-06	4.60E-05	0.19	0.28	**0.09**	0.00E+00	8.00E-04
	500	**0.25**	2.44	0.28	1.30E-06	8.50E-06	0.99	1.12	**0.24**	0.00E+00	1.00E-03
	1000	**0.53**	10.73	0.94	6.00E-07	1.10E-05	3.56	3.26	**0.48**	0.00E+00	9.00E-03
	2000	**0.99**	47.65	2.29	1.00E-06	4.20E-05	6.09	12.15	**0.99**	0.00E+00	3.00E-03
	5000	**5.83**	347.72	9.21	2.00E-06	8.03E-04	25.93	74.60	**2.41**	***	***
100	100	0.20	0.11	**0.03**	0.00E+00	1.12E-04	0.11	0.10	**0.04**	3.00E-03	3.00E-03
	200	0.23	0.47	**0.08**	2.00E-06	0.00E+00	0.23	0.29	**0.09**	0.00E+00	1.00E-03
	500	0.87	3.46	**0.37**	8.00E-07	5.40E-06	3.13	1.28	**0.31**	2.00E-04	1.50E-03
	1000	1.81	17.73	**1.55**	8.00E-07	1.05E-05	8.82	3.79	**0.69**	0.00E+00	8.10E-03
	2000	**2.53**	79.85	2.72	1.00E-06	1.50E-05	12.12	14.99	**1.65**	1.00E-03	5.00E-03
	5000	**14.67**	458.29	22.37	2.00E-06	6.88E-04	56.00	77.15	**2.97**	***	***
200	100	0.12	0.11	**0.03**	1.00E-06	7.10E-05	0.17	0.11	**0.05**	2.00E-03	2.00E-03
	200	0.32	0.45	**0.08**	1.00E-06	1.32E-04	0.43	0.35	**0.12**	1.00E-03	2.00E-03
	500	1.93	4.02	**0.31**	9.00E-07	3.20E-06	1.72	1.54	**0.35**	1.00E-04	1.30E-03
	1000	5.63	23.79	**1.38**	5.30E-07	5.49E-06	17.71	5.24	**1.02**	1.00E-05	7.64E-03
	2000	4.21	125.77	**3.07**	5.00E-07	9.70E-06	16.67	17.18	**2.48**	0.00E+00	5.00E-03
	5000	**21.41**	700.39	59.05	1.00E-06	5.17E-04	104.00	74.34	**3.90**	***	***
500	100	0.16	0.14	**0.05**	1.00E-06	6.32E-04	0.42	0.13	**0.07**	0.00E+00	1.00E-03
	200	0.75	0.58	**0.12**	3.00E-06	8.40E-05	0.52	0.35	**0.13**	4.00E-03	4.00E-03
	500	3.86	5.78	**0.48**	3.00E-07	1.40E-06	3.23	1.96	**0.51**	1.00E-04	1.20E-03
	1000	21.02	42.64	**2.00**	4.00E-07	-2.00E-07	14.62	7.31	**1.68**	1.00E-04	5.40E-03
	2000	87.45	308.63	**4.59**	2.00E-07	1.30E-06	332.82	27.64	**7.31**	***	***
	5000	143.58	1312.82	**27.77**	1.00E-06	2.75E-04	564.27	76.05	**9.01**	***	***

Table 1: Performance of the solvers for convex portfolio optimization problems. Columns MOSEK, Simple and CP represent average solution time (in seconds) over 20 instances of MOSEK NLP solver, cutting planes method based on simple approximation scheme and cutting planes method with a lifted scheme. The maximum absolute difference in the portfolio value is reported in columns MOSEK-SA (comparing solution due to MOSEK with the one due to simple approximation) and MOSEK-CP (MOSEK and lifted cutting planes method). *n* is the number of assets, *m* is the number of scenarios, MUL is the scaling parameter. “***” corresponds to the instances for which MOSEK returned an infinite portfolio value.

n	5			10			20		
m	10	50	100	10	50	100	10	50	100
CGBNB	**0.93**	**0.87**	**0.34**	**0.80**	**1.46**	**1.62**	**1.51**	**2.70**	**3.99**
AIMMS	49.17	67.10	73.55	104.43	151.35	221.19	195.29	618.61	7710.85

Table 2: Running time of AIMMS-AOA and the proposed implementation of the branch-and-bound method in lot-buying constrained portfolio optimization. Results averaged over 20 instances.

n	10			20			50		
m	500	1000	2000	500	1000	2000	500	1000	2000
CG-BNB	**0.74**	**1.72**	**5.10**	**12.03**	**22.57**	**50.64**	**108.67**	**240.38**	**263.57**
AIMMS	11.65	35.90	96.88	294.74	459.21	639.43	863.50	1489.65	2071.98

Table 3: Running time of AIMMS-AOA and the proposed implementation of the branch-and-bound method in cardinality constrained portfolio optimization. Results averaged over 20 instances.

paper. First, the implementation of the branch-and-bound method from Section 3 has been compared against AIMMS AOA implementation. The results for lot-buying and cardinality constrained problems are summarized in Tables 2 and 3, respectively. Observe that our custom implementation significantly outperforms AOA method for all choices of the parameters n and m. It is worth noting that, as it is stated in AIMMS manual, their implementation is much more efficient for binary variables, which is the case in our cardinality constrained problems. This observation explains the fact that in our experiments the improvement over AOA method has been less significant for this class of problems. Overall, we can conclude that these experiments confirm that the branch-and-bound approach presented here can be seen as a viable strategy for solving the considered class of MINLP problems.

In the second stage of our case study, we aimed at evaluating the effect that valid inequalities defined in Section 4 can play in solving problems (22) and (23). Results of this case study are summarized in Table 4 and 5. Note that for each problem size 20 instances were generated and solved with a 1 hour time limit. We report the number of instances solved within the time limit, solution time and number of nodes in the branch-and-bound tree averaged over the instances that have been solved in 1 hour by all three approaches, and the average integrality gap among instances not solved to optimality.

We can observe that in both of the models the usage of the proposed valid inequalities leads to improved solution performance, especially for larger problems sizes. It is, in our view, particularly important to note that we are able to solve more problem instances within the time limit, as well as significantly reduce the integrality gap. It can also be noted that while in the case of lot-buying constrained problems the lifted cuts presented in Section 4.2 exhibit the best overall performance, in cardinality constrained optimization, this approach does not provide any improvement over pure branch-and-bound.

6 Conclusions

In this paper we discussed solution approaches for a class of mixed-integer nonlinear programming problems, which arise from some recent developments in risk-averse stochastic optimization. In our study, we revisit some of the methods that have been previously proposed in the literature, and show that

n	*m*	number solved			running time			nodes in solution tree			gap after time limit		
		lifted	split	no cuts	lifted	split	no cuts	lifted	split	no cuts	nonlin	split	no cuts
50	500	20	20	20	11.57	**9.92**	11.01	5864.50	**4309.05**	5905.65	—	—	—
	1000	20	20	20	41.07	38.45	**28.57**	9307.70	8265.75	**6453.65**	—	—	—
	2000	20	20	20	**68.12**	**68.11**	138.37	7411.30	**6559.15**	13016.30	—	—	—
	5000	19	19	19	695.14	622.18	**581.49**	18903.58	16145.32	**15368.53**	**2.41%**	5.19%	6.25%
100	500	**19**	14	14	**400.22**	436.02	467.32	**129745.46**	173480.42	190997.69	—	—	—
	1000	**15**	13	13	**456.84**	502.90	1300.26	**77967.64**	86555.38	221685.91	**2.68%**	14.02%	6.06%
	2000	19	**20**	15	**179.06**	337.18	223.93	**11908.73**	24955.93	16974.87	3.01%	**—**	5.46%
	5000	19	20	18	673.90	**670.20**	731.66	16101.59	**13831.22**	17026.82	—	—	—
200	500	**6**	1	0	—	—	—	—	—	—	87.92%	**46.49%**	191.83%
	1000	0	0	0	—	—	—	—	—	—	**16.31%**	24.34%	22.99%
	2000	**8**	6	5	**498.57**	787.33	2153.11	**25654.00**	35918.50	138485.00	8.33%	**3.84%**	6.50%
	5000	**17**	12	12	**1408.58**	1804.24	1539.48	**19271.44**	20442.11	22581.56	—	—	—
500	500	0	0	0	—	—	—	—	—	—	**128.91%**	**128.89%**	200.04%
	1000	0	0	0	—	—	—	—	—	—	**109.42%**	114.03%	116.02%
	2000	0	0	0	—	—	—	—	—	—	29.27%	29.34%	**28.95%**
	5000	**2**	1	0	—	—	—	—	—	—	124.45%	**113.67%**	213.15%
1000	500	0	0	0	—	—	—	—	—	—	**97.01%**	98.57%	106.20%
	1000	0	0	0	—	—	—	—	—	—	**227.93%**	**227.73%**	316.26%
	2000	0	0	0	—	—	—	—	—	—	**54.65%**	55.90%	65.86%
	5000	0	0	0	—	—	—	—	—	—	**111.06%**	214.31%	219.85%

Table 4: Performance of two valid inequality families in lot-buying constrained portfolio optimization. The rows refer to: *no cuts* – pure branch-and-bound presented in Section 3, *lifted* – lifted cuts from Section 4.2, *split* – disjunctive cuts introduced in Section 4.3. Results averaged over 20 instances. *Running time* and *nodes in solution tree* columns reflect only instances solved within 1 hour time limit by all three approaches. Similarly *gap after time limit* corresponds to instances for which no optimal solution was found within the time limit for each of the methods.

n	*m*	number solved			running time			nodes in solution tree			gap after time limit		
		lifted	split	no cuts	lifted	split	no cuts	lifted	split	no cuts	nonlin	split	no cuts
		nonlin	split	no cuts	nonlin	split	no cuts	nonlin	split	no cuts	nonlin	split	no cuts
50	500	20	20	20	**108.84**	122.91	**108.67**	**25574.20**	26636.55	**25574.20**	—	—	—
	1000	20	20	20	**240.62**	252.45	**240.38**	19634.00	**19239.50**	19634.00	—	—	—
	2000	20	20	20	**263.00**	288.33	**263.57**	7651.90	**7506.10**	7651.90	—	—	—
	5000	20	20	20	152.99	**76.31**	151.91	1274.30	**994.70**	1274.30	—	—	—
100	500	6	**7**	6	2001.51	**1795.24**	1998.73	293837.33	**111602.00**	293837.33	23.63%	**20.40%**	23.62%
	1000	0	**3**	0	—	—	—	—	—	—	29.48%	**28.22%**	29.36%
	2000	3	**5**	3	2770.52	**2440.59**	2796.48	54008.00	**42317.75**	54008.00	13.08%	**11.84%**	13.07%
	5000	18	**19**	18	1043.44	**991.63**	1047.82	7770.28	**6734.63**	7770.28	4.63%	4.60%	4.61%
200	500	0	**1**	0	—	—	—	—	—	—	85.93%	**74.56%**	85.78%
	1000	0	0	0	—	—	—	—	—	—	71.87%	**52.10%**	71.86%
	2000	0	0	0	—	—	—	—	—	—	37.56%	**17.68%**	37.56%
	5000	0	0	0	—	—	—	—	—	—	8.87%	**8.82%**	8.87%
500	500	0	**1**	0	—	—	—	—	—	—	178.71%	**79.19%**	178.56%
	1000	0	0	0	—	—	—	—	—	—	126.57%	**26.28%**	126.58%
	2000	0	0	0	—	—	—	—	—	—	67.29%	**37.03%**	67.30%
	5000	0	0	0	—	—	—	—	—	—	21.63%	**13.15%**	21.63%
1000	500	0	0	0	—	—	—	—	—	—	223.31%	**123.95%**	223.31%
	1000	0	0	0	—	—	—	—	—	—	163.56%	**65.14%**	163.57%
	2000	0	0	0	—	—	—	—	—	—	92.95%	**73.52%**	92.96%
	5000	0	0	0	—	—	—	—	—	—	219.35%	**124.00%**	219.36%

Table 5: Performance of two valid inequality families in cardinality constrained portfolio optimization. The rows refer to: *no cuts* – pure branch-and-bound presented in Section 3, *lifted* – lifted cuts from Section 4.2, *split* – disjunctive cuts introduced in Section 4.3. Results averaged over 20 instances. *Running time* and *nodes in solution tree* columns reflect only instances solved within 1 hour time limit by all three approaches. Similarly *gap after time limit* corresponds to instances for which no optimal solution was found within the time limit for each of the methods.

these approaches can be naturally generalized to the MINLP problems in question. In addition, we also propose a new simple procedure for generating disjunctive cuts. The conducted numerical experiments produce promising results.

7 Acknowledgements

This work was supported in part by the AFOSR grant FA9550-12-1-0142 and DTRA HDTRA1-14-1-0065. In addition, support by the AFRL Mathematical Modeling and Optimization Institute is gratefully acknowledged.

References

Abhishek, K., Leyffer, S., and Linderoth, J. (2010) “FilMINT: an outer approximation-based solver for convex mixed-integer nonlinear programs,” *INFORMS J. Comput.*, **22** (4), 555–567.

Andersen, K. and Jensen, A. N. (2013) “Intersection cuts for mixed integer conic quadratic sets,” in: “Integer programming and combinatorial optimization,” volume 7801 of *Lecture Notes in Comput. Sci.*, 37–48, Springer, Heidelberg.

Atamtürk, A. and Narayanan, V. (2010) “Conic mixed-integer rounding cuts,” *Math. Program.*, **122** (1, Ser. A), 1–20.

Atamtürk, A. and Narayanan, V. (2011) “Lifting for conic mixed-integer programming,” *Math. Program.*, **126** (2, Ser. A), 351–363.

Auslender, A. and Teboulle, M. (2003) *Asymptotic cones and functions in optimization and variational inequalities*, Springer Monographs in Mathematics, Springer-Verlag, New York.

Balas, E. (1971) “Intersection cuts—a new type of cutting planes for integer programming,” *Operations Res.*, **19**, 19–39.

Belotti, P., Góez, J. C., Pólik, I., Ralphs, T. K., and Terlaky, T. (2012) “A Conic Representation of the Convex Hull of Disjunctive Sets and Conic Cuts for Integer Second Order Cone Optimization,” *Submitted*.

Ben-Tal, A. and Nemirovski, A. (2001) “On polyhedral approximations of the second-order cone,” *Math. Oper. Res.*, **26** (2), 193–205.

Bienstock, D. and Michalka, A. (2014) “Cutting-planes for optimization of convex functions over nonconvex sets,” *SIAM J. Optim.*, **24** (2), 643–677.

Bonami, P. (2011) “Lift-and-project cuts for mixed integer convex programs,” in: “Integer programming and combinatorial optimization,” volume 6655 of *Lecture Notes in Comput. Sci.*, 52–64, Springer, Heidelberg.

Bonami, P., Biegler, L. T., Conn, A. R., Cornuéjols, G., Grossmann, I. E., Laird, C. D., Lee, J., Lodi, A., Margot, F., Sawaya, N., and Wächter, A. (2008) “An algorithmic framework for convex mixed integer nonlinear programs,” *Discrete Optim.*, **5** (2), 186–204.

Borchers, B. and Mitchell, J. E. (1994) “An improved branch and bound algorithm for mixed integer nonlinear programs,” *Comput. Oper. Res.*, **21** (4), 359–367.

Bullen, P. S., Mitrinović, D. S., and Vasić, P. M. (1988) *Means and their inequalities*, volume 31 of *Mathematics and its Applications (East European Series)*, D. Reidel Publishing Co., Dordrecht, translated and revised from the Serbo-Croatian.

Burer, S. and Kılınç-Karzan, F. (2014) “How to Convexify the Intersection of a Second Order Cone and a Nonconvex Quadratic,” *Technical report*, `http://www.optimization-online.org/DB_HTML/2014/06/4383.html`.

Burer, S. and Saxena, A. (2012) “The MILP Road to MIQCP,” in: J. Lee and S. Leyffer (Eds.) “Mixed Integer Nonlinear Programming,” volume 154 of *The IMA Volumes in Mathematics and its Applications*, 373–405, Springer New York.

Cadoux, F. (2010) “Computing deep facet-defining disjunctive cuts for mixed-integer programming,” *Math. Program.*, **122** (2, Ser. A), 197–223.

Çezik, M. T. and Iyengar, G. (2005) “Cuts for mixed 0-1 conic programming,” *Math. Program.*, **104** (1, Ser. A), 179–202.

Duran, M. A. and Grossmann, I. E. (1986) “An outer-approximation algorithm for a class of mixed-integer nonlinear programs,” *Math. Programming*, **36** (3), 307–339.

Fletcher, R. and Leyffer, S. (1994) “Solving mixed integer nonlinear programs by outer approximation,” *Math. Programming*, **66** (3, Ser. A), 327–349.

Glineur, F. (2000) “Computational experiments with a linear approximation of second order cone optimization,” *Technical Report 0001*, Service de Mathématique et de Recherche Opérationnelle, Faculté Polytechnique de Mons, Mons, Belgium.

Gupta, O. K. and Ravindran, A. (1985) “Branch and bound experiments in convex nonlinear integer programming,” *Management Sci.*, **31** (12), 1533–1546.

Hardy, G. H., Littlewood, J. E., and Pólya, G. (1952) *Inequalities*, Cambridge, at the University Press, 2d ed.

Kılınç, M., Linderoth, J., and Luedtke, J. (2010) “Effective separation of disjunctive cuts for convex mixed integer nonlinear programs,” *Technical report*.

Kılınç-Karzan, F. (2015) “On minimal inequalities for mixed integer conic programs,” *Mathematics of Operations Research*, to appear.

Kılınç-Karzan, F. and Yıldız, S. (2015) “Two Term Disjunctions on the Second-Order Cone,” *Mathematical Programming*, DOI:10.1007/s10107–015–0903–4.

Krokhmal, P., Zabarankin, M., and Uryasev, S. (2011) “Modeling and optimization of risk,” *Surveys in Operations Research and Management Science*, **16** (2), 49 – 66.

Krokhmal, P. A. (2007) “Higher moment coherent risk measures,” *Quant. Finance*, **7** (4), 373–387.

Leyffer, S. (2001) “Integrating SQP and branch-and-bound for mixed integer nonlinear programming,” *Comput. Optim. Appl.*, **18** (3), 295–309.

McCord, M. and Neufville, R. d. (1986) “”Lottery Equivalents”: Reduction of the Certainty Effect Problem in Utility Assessment,” *Management Science*, **32** (1), pp. 56–60.

Modaresi, S., Kılınç, M. R., and Vielma, J. P. (2015) “Split cuts and extended formulations for Mixed Integer Conic Quadratic Programming,” *Oper. Res. Lett.*, **43** (1), 10–15.

Quesada, I. and Grossmann, I. E. (1992) “An LP/NLP based branch and bound algorithm for convex MINLP optimization problems,” *Computers & chemical engineering*, **16** (10), 937–947.

Rockafellar, R. T. (1997) *Convex analysis*, Princeton Landmarks in Mathematics, Princeton University Press, Princeton, NJ, reprint of the 1970 original, Princeton Paperbacks.

Rockafellar, R. T. and Uryasev, S. (2000) “Optimization of Conditional Value-at-Risk,” *Journal of Risk*, **2**, 21–41.

Rockafellar, R. T. and Uryasev, S. (2002) “Conditional Value-at-Risk for General Loss Distributions,” *Journal of Banking and Finance*, **26** (7), 1443–1471.

Rysz, M., Vinel, A., Krokhmal, P., and Pasiliao, E. L. (2014) “A scenario decomposition algorithm for stochastic programming problems with a class of downside risk measures,” *INFORMS Journal on Computing*, to appear.

Saxena, A., Bonami, P., and Lee, J. (2008) “Disjunctive cuts for non-convex mixed integer quadratically constrained programs,” in: “Integer programming and combinatorial optimization,” volume 5035 of *Lecture Notes in Comput. Sci.*, 17–33, Springer, Berlin.

Stubbs, R. A. and Mehrotra, S. (1999) “A branch-and-cut method for 0-1 mixed convex programming,” *Math. Program.*, **86** (3, Ser. A), 515–532.

Vielma, J. P., Ahmed, S., and Nemhauser, G. L. (2008) “A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs,” *INFORMS J. Comput.*, **20** (3), 438–450.

Vinel, A. and Krokhmal, P. (2014a) “On valid inequalities for mixed integer p-order cone programming,” *J. Optim. Theory Appl.*, **160** (2), 439–456.

Vinel, A. and Krokhmal, P. A. (2014b) “Certainty equivalent measures of risk,” *submitted for publication*, **29** (6), 1210–1237.

Vinel, A. and Krokhmal, P. A. (2014c) “Polyhedral approximations in p-order cone programming,” *Optim. Methods Softw.*, **29** (6), 1210–1237.

Wilson, R. (1979) “Auctions of Shares,” *The Quarterly Journal of Economics*, **93** (4), pp. 675–689.

Certainty Equivalent Measures of Risk

Alexander Vinel   Pavlo A. Krokhmal*

Department of Mechanical and Industrial Engineering
University of Iowa, 3131 Seamans Center, Iowa City, IA 52242, USA

Abstract

We study a framework for constructing coherent and convex measures of risk that is inspired by infimal convolution operator, and which is shown to constitute a new general representation of these classes of risk functions. We then discuss how this scheme may be effectively applied to obtain a class of certainty equivalent measures of risk that can directly incorporate preferences of a rational decision maker as expressed by a utility function. This approach is consequently employed to introduce a new family of measures, the log-exponential convex measures of risk. Conducted numerical experiments show that this family can be a useful tool for modeling of risk-averse preferences in decision making problems with heavy-tailed distributions of uncertain parameters.

Keywords: Coherent risk measures, convex risk measures, stochastic optimization, risk-averse preferences, utility theory, certainty equivalent, stochastic dominance, log-exponential convex measures of risk

1 Introduction

Informally, a decision making problem under uncertainties can be stated as the problem of selecting a decision $\mathbf{x} \in C \subset \mathbb{R}^n$, given that the cost X of this decision depends not only on $\mathbf{x}$, but also on a random event $\omega \in \Omega$: $X = X(\mathbf{x}, \omega)$. A principal modeling challenge that one faces in this setting is to select an appropriate ordering of random outcomes X, or, in other words, define a way to choose one uncertain outcome, $X_1 = X(\mathbf{x}_1, \omega)$, over another, $X_2 = X(\mathbf{x}_2, \omega)$. A fundamental contribution in this context is represented by the expected utility theory of von Neumann and Morgenstern (1944), which argues that if the preferences of a decision maker are *rational*, i.e., they satisfy a specific system of properties (axioms), then there exists a utility function $u : \mathbb{R} \mapsto \mathbb{R}$, such that a decision under uncertainty is optimal if it maximizes the expected utility of the payoff. Equivalently, the random elements representing payoffs under uncertainty can be ordered based on the corresponding values of expected utility of these payoffs. Closely connected to the expected utility theory is the subject of stochastic orderings (see Levy, 1998), and particularly stochastic dominance relations, which have found applications in economics, decision theory, game theory, and so on.

An alternative approach to introducing preference relations over random outcomes $X(\mathbf{x}, \omega)$, which has traditionally been employed in optimization and operations research literature, and which is followed

[*] Corresponding author. E-mail: krokhmal@engineering.uiowa.edu

in the present work, is to introduce a function $\rho : \mathcal{X} \mapsto \overline{\mathbb{R}}$, where $\mathcal{X}$ is an appropriately defined space containing X, such that X_1 is preferred to X_2 whenever $\rho(X_1) < \rho(X_2)$. The decision making problem in the presence of uncertainties can then be expressed as a mathematical program

$$\min\{\rho(X) : X = X(\mathbf{x}, \omega) \in \mathcal{X},\ \mathbf{x} \in C\}, \tag{1}$$

where function ρ is usually referred to as a *risk measure*. In stochastic programming literature, the objective of a minimization problem like (1) has traditionally been chosen in the form of the expected cost, $\rho(X) = \mathsf{E}X$ (Prékopa, 1995; Birge and Louveaux, 1997), which is commonly regarded as a representation of risk-neutral preferences. In the finance domain, the pioneering work of Markowitz (1952) has introduced a *risk-reward* paradigm for decision making under uncertainty, and variance was proposed as a measure of risk, $\rho(X) = \sigma^2(X)$. Since then, the problem of devising risk criteria suitable for quantification of specific risk-averse preferences has received significant attention (see a survey in Krokhmal et al., 2011). It was noticed, however, that "ad-hoc" construction of ρ may yield risk functionals that, while serving well in a specific application, are flawed in a general methodological sense. Artzner et al. (1999) suggested an axiomatic approach, similar to that of von Neumann and Morgenstern (1944), to defining a well-behaved risk measure ρ in (1), and introduced the concept of *coherent measures of risk*. Subsequently, a range of variations and extensions of the axiomatic framework for designing risk functionals have been proposed in the literature, such as *convex* and *spectral* measures of risk (Föllmer and Schied, 2004; Acerbi, 2002), *deviation measures* (Rockafellar et al., 2006), and so on, see an overview in Krokhmal et al. (2011) and Rockafellar and Uryasev (2013). Since many classes of axiomatically defined risk measures represent risk preferences that are not fully compatible with the rational risk-averse preferences of utility theory, of additional interest are risk measures that possess such a compatibility in a certain sense.

In this paper we propose a new representation for the classes of coherent and convex measures of risk, which builds upon a previous work of Krokhmal (2007). This representation is then used to introduce a class of coherent or convex measures of risk that can directly incorporate *rational* risk preferences as prescribed by the corresponding utility function, through the concept of certainty equivalent. This class of certainty equivalent measures of risk contains some of the existing risk measures, such as the popular Conditional Value-at-Risk (Rockafellar and Uryasev (2000, 2002)) as special cases. As an application of the general approach, we introduce a two-parameter family of log-exponential convex risk measures, which quantify risk by emphasizing extreme losses in the tail of the loss distribution. Two case studies illustrate the practical merits of the log-exponential risk measures; in particular, it is shown that these nonlinear measures of risk can be preferable to more traditional measures, such as Conditional Value-at-Risk, if the loss distribution is heavy-tailed and contains catastrophic losses.

The rest of the paper is organized as follows. In Section 2.1 we briefly discuss the classes of coherent and convex measures of risk as well as some of their properties. Section 2.2 establishes that the constructive formula of Krokhmal (2007) does actually constitute a representation for coherent risk measures and can be generalized to the case of convex measures of risk. Using this representation, in Section 2.3 we introduce a class of coherent or convex measures of risk that are based on certainty equivalents of some utility functions. In Section 2.4 we further study some of the properties of this class of risk measures. Finally, Section 3 discusses the log-exponential convex measures of risk, and illustrates their properties with two case studies.

2 Risk Measures Based on Infimal Convolution

2.1 Coherent and Convex Measures of Risk

Consider a random outcome $X \in \mathcal{X}$ defined on an appropriate probability space $(\Omega, \mathcal{F}, \mathsf{P})$, where $\mathcal{X}$ is a linear space of $\mathcal{F}$-measurable functions $X : \Omega \mapsto \mathbb{R}$. A function $\rho : \mathcal{X} \mapsto \overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$ is said to be a *convex measure of risk* if it satisfies the following axioms:

(A0) *lower semicontinuity (l.s.c.)*;

(A1) *monotonicity*: $\rho(X) \le \rho(Y)$ for all $X \le Y$;

(A2) *convexity*: $\rho\big(\lambda X + (1-\lambda)Y\big) \le \lambda\rho(X) + (1-\lambda)\rho(Y), \quad \lambda \in [0,1]$;

(A3) *translation invariance*: $\rho(X + a) = \rho(X) + a, \quad a \in \mathbb{R}$.

Similarly, function $\rho : \mathcal{X} \to \overline{\mathbb{R}}$ is said to be a *coherent measure of risk* if it satisfies (A0)–(A3), and, additionally,

(A4) *positive homogeneity*: $\rho(\lambda X) = \lambda\rho(X), \quad \lambda > 0$.

Remark 2.1. We assume that the space $\mathcal{X}$ is endowed with necessary properties so that the corresponding risk measures are well defined. Specifically, $\mathcal{X}$ is a space of integrable functions, $\mathsf{E}|X| < +\infty$, and is equipped with an appropriate topology, which is assumed to be the topology induced by convergence in probability, unless stated otherwise. Also, it is assumed throughout the paper that all considered functions are proper (recall that function $f : \mathcal{X} \mapsto \overline{\mathbb{R}}$ is proper if $f(X) > -\infty$ for all $X \in \mathcal{X}$, and $\operatorname{dom} f = \{X \in \mathcal{X} \mid f(X) < +\infty\} \neq \emptyset$).

Remark 2.2. In this work we adopt the traditional viewpoint of engineering literature that a random quantity X represents a cost or a loss, in the sense that smaller realizations of X are preferred. In economics literature it is customary to consider X as wealth or payoff variable, whose larger realizations are desirable. In most cases, these two approaches can be reconciled by inverting the sign of X, which may require some modifications to the properties discussed above. For example, the translation invariance axiom (A3) will have the form $\rho(X + a) = \rho(X) - a$ in the case when X is a payoff function.

Remark 2.3. Without loss of generality, we also assume that a convex measure of risk satisfies normalization property: $\rho(0) = 0$ (observe that coherent measures necessarily satisfy this property). First, such a normalization requirement is natural from methodological and practical viewpoints, since there is usually no risk associated with zero costs or losses. Second, due to translation invariance any convex ρ can be normalized by setting $\tilde{\rho}(X) = \rho(X) - \rho(0)$.

Remark 2.4. It is worth noting that normalized convex measures of risk satisfy the so-called *subhomogeneity* property:

(A4′) *subhomogeneity:* $\rho(\lambda X) \le \lambda\rho(X)$ for $\lambda \in (0,1)$ and $\rho(\lambda X) \ge \lambda\rho(X)$ for $\lambda > 1$.

Indeed, in order to see that the first inequality in (A4′) holds, observe that $\lambda\rho(X) = \lambda\rho(X) + (1-\lambda)\rho(0) \ge \rho(\lambda X + (1-\lambda)0) = \rho(\lambda X)$ for $\lambda \in (0,1)$. Similarly, if $\lambda > 1$, then $\frac{1}{\lambda}\rho(\lambda X) = \frac{1}{\lambda}\rho(\lambda X) + \big(1 - \frac{1}{\lambda}\big)\rho(0) \ge \rho(X)$.

Artzner et al. (1999) and Delbaen (2002) have proposed a general representation for the class of coherent measures by showing that a mapping $\rho : \mathcal{X} \mapsto \mathbb{R}$ is a coherent risk measure if and only if $\rho(X) = \sup_{Q \in \mathcal{Q}} \mathsf{E}_Q X$, where $\mathcal{Q}$ is a closed convex subset of P-absolutely continuous probability measures. Föllmer and Schied (2002) have generalized this result to convex measures of risk. Since then, other representations have been proposed, see Kusuoka (2001, 2012); Frittelli and Rosazza Gianin (2005); Dana (2005); Acerbi (2002). For example, Acerbi (2002) has suggested a spectral representation: $\rho(X) = \int_0^1 \mathrm{VaR}_\lambda(X)\psi(\lambda)\mathrm{d}\lambda$, where $\psi \in \mathcal{L}^1([0,1])$. While many of these results led to important theoretical insights and methodological conclusions, relatively few of them provided practical ways for construction of new risk measures in accordance with specified risk preferences, which are also conducive to implementation in mathematical programming problems. Below we discuss a representation that may be better suited for this purpose.

2.2 An Infimal Convolution Representation for Coherent and Convex Measures of Risk

An approach to constructing coherent measures of risk that was based on the operation of infimal convolution was proposed in Krokhmal (2007). Given a function $\phi : \mathcal{X} \mapsto \overline{\mathbb{R}}$, consider a risk measure ρ, which we will call a *convolution-based measure of risk*, in the form

$$\rho(X) = \inf_{\eta} \ \eta + \phi(X - \eta). \tag{2}$$

Then, the following claim has been shown to hold.

Proposition 2.1 (Krokhmal, 2007, Theorem 1)**.** *Suppose that function ϕ satisfies axioms* (A0)–(A2) *and* (A4)*, and, additionally, is such that $\phi(\eta) > \eta$ for all constant $\eta \neq 0$. Then the infimal convolution in* (2) *is a proper coherent measure of risk. Moreover, the infimum in* (2) *is attained for all X, and can be replaced with a minimization operator.*

In this section we show that this approach can be substantially generalized, which leads us to formulate Theorem 2.5 below. Before moving to this general result, we establish a few subsidiary lemmas. First, we demonstrate that expression (2) is a *representation*, i.e., any coherent measure of risk can be expressed in the form of (2).

Lemma 2.2. *Let ρ be a coherent measure of risk. Then, there exists a proper function $\phi : \mathcal{X} \mapsto \overline{\mathbb{R}}$ that satisfies axioms* (A0)–(A2) *and* (A4)*, $\phi(\eta) > \eta$ for all constant $\eta \neq 0$, and is such that $\rho(X) = \min_\eta \eta + \phi(X - \eta)$.*

Proof. For a given proper and coherent ρ consider $\phi_\rho(X) = 2[\rho(X)]_+$, where $[X]_+ = \max\{X, 0\}$, and observe that ϕ_ρ is proper and satisfies (A0)–(A2) and (A4) if ρ is coherent, and, moreover, $\phi_\rho(\eta) = 2[\eta]_+ > \eta$ for all real $\eta \neq 0$. Finally, $\min_\eta \eta + \phi_\rho(X - \eta) = \min_\eta \eta + 2[\rho(X - \eta)]_+ = \min_\eta \eta + 2[\rho(X) - \eta]_+ = \rho(X)$, i.e., any coherent ρ can be represented in the form of (2). □

Remark 2.5. It is easy to see from the proof of Lemma 2.2 that the function ϕ in representation (2) is not determined uniquely for any given coherent measure ρ. Indeed, one can choose (among possibly others) $\phi(X) = \alpha[\rho(X)]_+$ for any $\alpha > 1$.

Next, we show that the infimal convolution representation (2) can be generalized to convex measures of risk. Technically, the proof of Proposition 2.1 in Krokhmal (2007) relies heavily on the positive homogeneity property (A3) of coherent risk measures, but as we demonstrate below, it can be amended in order to circumvent this issue. Recall that, given a proper, l.s.c., convex function f on $\mathbb{R}^n$ and $\mathbf{x} \in \operatorname{dom} f$, its *recession function* $(f0^+)(\mathbf{y})$ can be defined as

$$(f0^+)(\mathbf{y}) = \lim_{\tau\to\infty} \frac{f(\mathbf{x}+\tau\mathbf{y}) - f(\mathbf{x})}{\tau}.$$

Note that the above expression does not depend on $\mathbf{x} \in \operatorname{dom} f$, hence $(f0^+)(\mathbf{y})$ is well-defined (Rockafellar, 1997, Theorem 8.5). The result established below mirrors that of Proposition 2.1 in the case of convex measures of risk.

Lemma 2.3. *Suppose that a proper function ϕ satisfies axioms (A0)–(A2), and, additionally, is such that $\phi(\eta) > \eta$ for all constant $\eta \neq 0$ and $\phi(0) = 0$. Then the infimal convolution $\rho(X) = \inf_\eta \ \eta + \phi(X - \eta)$ is a proper convex risk measure. Moreover, the infimum is attained for all X, and can be replaced with $\min_\eta$.*

Proof. For any fixed $X \in \mathcal{X}$ consider function $\phi_X(\eta) = \eta + \phi(X - \eta)$. Clearly, since ϕ is proper, l.s.c. and convex, ϕ_X is l.s.c., convex in η and $\phi_X > -\infty$ for all η. Next we will show that the infimum in the definition of ρ is attained for any X. First, suppose that $\operatorname{dom}\phi_X = \emptyset$, hence $\rho(X) = +\infty$, and the infimum in the definition is attained for any $\eta \in \mathbb{R}$. Now, assume that there exists $\tilde\eta \in \operatorname{dom}\phi_X$, and consequently both $\phi(X - \tilde\eta) < +\infty$ and $\rho(X) < +\infty$. Recall that a proper, l.s.c. function ϕ_X attains its infimum if it has no directions of recession (see Rockafellar, 1997, Theorem 27.2), or in other words, if $\phi_X 0^+(\xi) > 0$ for all $\xi \neq 0$. Observe that

$$\begin{aligned}(\phi_X 0^+)(\xi) &= \lim_{\tau\to\infty} \frac{\tilde\eta + \tau\xi + \phi(X - \tilde\eta - \tau\xi) - \tilde\eta - \phi(X - \tilde\eta)}{\tau} \\ &= \xi + \lim_{\tau\to\infty} \frac{\phi(X - \tilde\eta - \tau\xi)}{\tau} \geq \xi + \lim_{\tau\to\infty} \phi\Big(\frac{X - \tilde\eta}{\tau} - \xi\Big),\end{aligned}$$

where the last inequality follows from Remark 2.4 for sufficiently large τ. Since ϕ is l.s.c. and $\phi(\xi) > \xi$ for all $\xi \neq 0$, we can conclude that $\lim_{\tau\to\infty} \phi\Big(\frac{X-\tilde\eta}{\tau} - \xi\Big) \geq \phi(-\xi) > -\xi$, whereby $(\phi_X 0^+)(\xi) > 0$ for all $\xi \neq 0$, which guarantees that the infimum in the definition is attained, and $\rho(X) = \min_\eta \ \eta + \phi(X - \eta)$. Next, we verify that axiom (A0) holds. As shown above, for any $X \in \mathcal{X}$ there exists η_X such that $\rho(X) = \eta_X + \phi(X - \eta_X)$. Consequently,

$$\begin{aligned}\liminf_{Y\to X} \rho(Y) &= \liminf_{Y\to X} \Big(\eta_Y + \phi(Y - \eta_Y)\Big) \geq \liminf_{Y\to X} \Big(\eta_X + \phi(Y - \eta_X)\Big) \\ &= \eta_X + \liminf_{Y\to X} \phi(Y - \eta_X) \geq \eta_X + \phi(X - \eta_X) = \rho(X),\end{aligned}$$

where the last inequality holds due to lower semicontinuity of ϕ. Whence, by definition, ρ is l.s.c. Verification of properties (A1)–(A3) is straightforward and can be taken from Krokhmal (2007), Theorem 1. □

Lemma 2.4. *Let ρ be a convex measure of risk such that $\rho(0) = 0$. Then there exists a proper function $\phi\colon \mathcal{X} \mapsto \overline{\mathbb{R}}$ that satisfies axioms of monotonicity and convexity, is lower semicontinuous, $\phi(\eta) > \eta$ for all $\eta \neq 0$, and such that $\rho(X) = \min_\eta \eta + \phi(X - \eta)$.*

Proof. Analogously to Lemma 2.2, one can take $\phi_\rho(X) = 2[\rho(X)]_+$. □

Combining the above results, we obtain a general conclusion.

Theorem 2.5. *A proper, l.s.c. function $\rho\colon \mathcal{X} \mapsto \overline{\mathbb{R}}$ is a convex (respectively, coherent) measure of risk if and only if there exists a proper, l.s.c. function $\phi\colon \mathcal{X} \mapsto \overline{\mathbb{R}}$, which satisfies the axioms of monotonicity and convexity (and, respectively, positive homogeneity), $\phi(\eta) > \eta$ for all $\eta \neq 0$, $\phi(0) = 0$, and such that $\rho(X) = \min_\eta \eta + \phi(X - \eta)$.*

The importance of infimal convolution representation (2) for convex/coherent risk measures lies in the fact that it is amenable for use in stochastic programming problems with risk constraints or risk objectives (note that the problem does not necessarily have to be convex).

Lemma 2.6. *Let ρ be a coherent measure of risk, and for some $F, H : \mathcal{X} \mapsto \mathbb{R}$ and $C(\mathcal{X}) \subset \mathcal{X}$ consider the following risk-constrained stochastic programming problem:*

$$\min\{F(X) : \rho(X) \leq H(X),\ X \in C(\mathcal{X})\}. \tag{3}$$

Then, for a given convolution representation (2) *of ρ, problem* (3) *is equivalent to a problem of the form*

$$\min\{F(X) : \eta + \phi(X - \eta) \leq H(X),\ X \in C(\mathcal{X}),\ \eta \in \mathbb{R}\}, \tag{4}$$

in the sense that if (3) *is feasible, they achieve minima at the same values of the decision variable X and their optimal objective values coincide. Moreover, if risk constraint is binding at optimality in* (3)*, then (X^*, η^*) delivers a minimum to* (4) *if and only if X^* is an optimal solution of* (3) *and $\eta^* \in \arg\min\{\eta + \phi(X^* - \eta)\}$.*

Proof. Analogous to that in Krokhmal (2007), Theorem 3. □

Additionally, representation (2) conveys the idea that *a risk measure represents an optimal value or optimal solution of a stochastic programming problem of special form.*

2.3 Convolution Representation and Certainty Equivalents

The infimal convolution representation (2) allows for construction of convex or coherent measures of risk that directly employ risk preferences of a decision maker through a connection to the expected utility theory of von Neumann and Morgenstern (1944). Assuming without loss of generality that the loss/cost elements $X \in \mathcal{X}$ are such that $-X$ represents *wealth* or *reward*, consider a non-decreasing, convex *deutility* function $v : \mathbb{R} \mapsto \mathbb{R}$ that quantifies dissatisfaction of a risk-averse rational decision maker with a loss or cost. Obviously, this is equivalent to having a non-decreasing concave utility function $u(t) = -v(-t)$. By the inverse of v we will understand function $v^{-1}(a) = \sup\{t \in \mathbb{R} : v(t) = a\}$.

Remark 2.6. Note that if a non-decreasing, convex $v(t) \not\equiv$ const then, according to the definition above, the inverse is finite, and moreover, if there exists t, such that $v(t) = a < +\infty$, then $v^{-1}(a) = \max\{t \in \mathbb{R} \mid v(t) = a\}$. Additionally, let $v^{-1}(+\infty) = +\infty$.

Then, for any given $\alpha \in (0, 1)$, consider function ϕ in the form

$$\phi(X) = \frac{1}{1-\alpha} v^{-1} \mathsf{E} v(X), \tag{5}$$

where we use an operator-like notation for v^{-1}, i.e., $v^{-1}\mathsf{E}v(X) = v^{-1}(\mathsf{E}v(X))$.

Expression $\mathrm{CE}(X) = v^{-1}\mathsf{E}v(X)$ represents the *certainty equivalent* of an uncertain loss X, a deterministic loss/cost such that a rational decision maker would be indifferent between accepting $\mathrm{CE}(X)$ or an uncertain X; it is also known as *quasi-arithmetic mean*, *Kolmogorov mean*, or *Kolmogorov-Nagumo mean* (see, among others, Bullen et al., 1988; Hardy et al., 1952). Certainty equivalents play an important role in the decision making literature (see, for example, Wilson, 1979; McCord and Neufville, 1986); in the context of modern risk theory, certainty equivalents were considered in the work of Ben-Tal and Teboulle (2007).

In order for function ϕ as defined by (5) to comply with the conditions of Theorem 2.5, the deutility function should be such that $\phi(\eta) = \frac{1}{1-\alpha}v^{-1}v(\eta) \;\; > \eta$ for $\eta \neq 0$. This necessarily implies that $v(\eta) = v(0)$ for all $\eta \leq 0$, provided that v is proper, non-decreasing and convex. Indeed, if $v(\eta^*) < v(0)$ for some $\eta^* < 0$, then according to the above remarks $v^{-1}v(\eta^*) = \max\{\eta : v(\eta) = v(\eta^*)\} = \eta^{**}$, where η^{**} is such that $\eta^* \leq \eta^{**} < 0$ and $v^{-1}v(\eta^{**}) = \eta^{**}$, whence $\phi(\eta^{**}) = (1-\alpha)^{-1}\eta^{**} < \eta^{**}$.

Additionally, without loss of generality it can be postulated that $v(0) = 0$, i.e., zero loss means zero dissatisfaction. Indeed, $\tilde{v}^{-1}\mathsf{E}\tilde{v}(X) = v^{-1}\mathsf{E}v(X)$ for $\tilde{v}(t) = v(t) - v(0)$, i.e., such a transformation of the deutility function does not change the value of the certainty equivalent. Similarly, it is assumed that $v(t) > 0$ for all $t > 0$. This condition represents a practical consideration that positive losses entail positive deutility/dissatisfaction, and is not restrictive from methodological point of view. Indeed, it can be shown that if one allows for $t_0 = \max\{t : v(t) = 0\} > 0$, then the risk measures based on ϕ as given in (5) with deutilities $v(t)$ and $v_0(t) = v(t + t_0)$, such that $v_0(t) > 0$, $t > 0$, will differ only by a constant.

To sum up, we consider non-decreasing, convex deutility function $v : \mathbb{R} \mapsto \mathbb{R}$ such that

$$v(t) = v\big([t]_+\big) = \begin{cases} v(t) > 0, & t > 0, \\ 0, & t \leq 0. \end{cases}$$

We will refer to such a function as a *one-sided deutility*. Then, using the corresponding function ϕ in representation (2) one obtains a class of *certainty equivalent measures of risk:*

$$\begin{aligned} \rho(X) &= \min_\eta \; \eta + \frac{1}{1-\alpha}v^{-1}\mathsf{E}v(X-\eta) \\ &= \min_\eta \; \eta + \frac{1}{1-\alpha}v^{-1}\mathsf{E}v\big([X-\eta]_+\big). \end{aligned} \tag{6}$$

Next we analyze the conditions under which formulation (6) yields a coherent or convex measure of risk. Recall that we assume the space $\mathcal{X}$ to be such that certainty equivalent above is well-defined, particularly, integrability condition is satisfied.

Proposition 2.7. *If v is a one-sided deutility function, then $\phi(X) = \frac{1}{1-\alpha}v^{-1}\mathsf{E}v(X)$ is proper, l.s.c., satisfies the axiom of monotonicity and $\phi(\eta) > \eta$ for all $\eta \neq 0$.*

Proof. Clearly, such a ϕ is proper and l.s.c. The monotonicity property of ϕ defined by (5), $\phi(X) \leq \phi(Y)$ for all $X \leq Y$, follows from both v and v^{-1} being non-decreasing. Finally, note that

$$\begin{aligned} \phi(\eta) &= \frac{1}{1-\alpha}v^{-1}v(\eta) = \frac{1}{1-\alpha}v^{-1}v([\eta]_+) \\ &= \frac{1}{1-\alpha}\sup\big\{t : v(t) = v([\eta]_+)\big\} \geq \frac{1}{1-\alpha}[\eta]_+ > \eta \end{aligned}$$

for all $\eta \neq 0$. □

From Proposition 2.7 we can conclude that in order for the conditions of Theorem 2.5 to be satisfied we only need to guarantee convexity of the certainty equivalent (5) (note that axiom (A4) is satisfied if certainty equivalent itself is positive homogeneous). A sufficient condition of this type has been established in Ben-Tal and Teboulle (2007).

Proposition 2.8 (Ben-Tal and Teboulle, 2007)**.** *If $v \in \mathcal{C}^3(\mathbb{R})$ is strictly convex and $\dfrac{v'}{v''}$ is convex, then the certainty equivalent $v^{-1}\mathsf{E}v$ is also convex.*

The following observation adapts this result to establish convexity of certainty equivalents in the case of one-sided deutility functions.

Corollary 2.9. *If $v \in \mathcal{C}^3[0,\infty)$ is strictly convex and $\dfrac{v'}{v''}$ is convex on $[0,+\infty)$, then certainty equivalent $v_+^{-1}\mathsf{E}v_+$ is convex, where $v_+(t) = v([t]_+)$.*

Proof. Indeed, note that $v_+^{-1}\mathsf{E}v_+(X) = v_+^{-1}\mathsf{E}v([X]_+) = v^{-1}\mathsf{E}v([X]_+)$, which is convex as a superposition of a convex (Proposition 2.8 for function v) and a non-decreasing convex functions. □

Remark 2.7. Conditions of Proposition 2.8 are only sufficient, i.e., it is possible for a certainty equivalent to be convex without satisfying these conditions (as is shown in Corollary 2.9). Moreover, these conditions are rather restrictive. Thus, it is worth noting that if v is a one-sided deutility function such that the corresponding certainty equivalent is convex, then the certainty equivalent measure ρ defined by (6) is a convex (or coherent) measure of risk, regardless of whether Corollary 2.9 holds. At the same time, this result can be useful, as it is demonstrated by Proposition 3.1.

Observe that if function ϕ is taken in the form (5), where v is a one-sided deutility, the structure of the resulting risk measure (6) allows for an intuitive interpretation, similar to that proposed by Ben-Tal and Teboulle (2007). Consider, for instance, a resource allocation problem where X represents an unknown in advance cost of resources necessary to cover future losses or damages. Assume that it is possible to allocate amount η worth of resources in advance, whereby the remaining part of costs, $[X-\eta]_+$, will have to be covered after the actual realization of X is observed. To a decision maker with deutility v, the uncertain cost remainder $[X-\eta]_+$ is equivalent to the deterministic amount of certainty equivalent $v^{-1}\mathsf{E}v([X-\eta]_+)$. Since this portion of resource allocation is "unplanned", an additional penalty is imposed. If this penalty is modeled using a multiplier $\frac{1}{1-\alpha}$, then the expected additional cost of the resource is $\frac{1}{1-\alpha}v^{-1}\mathsf{E}v([X-\eta]_+)$. Thus, the risk associated with the mission amounts to $\eta + (1-\alpha)^{-1}v^{-1}\mathsf{E}v([X-\eta]_+)$, and can be minimized over all possible values of η, leading to definition (6). Moreover, when applied to the general definition (2), this argument provides an intuition behind the condition $\phi(\eta) > \eta$ above. Indeed, the positive difference $\phi(\eta) - \eta$ can be seen as a penalty for an unplanned loss.

We also note that certainty equivalent representation (6) for coherent or convex measures of risk is related to the *optimized certainty equivalents* (OCEs) due to Ben-Tal and Teboulle (2007),

$$\mathrm{OCE}(X) = \sup_{\eta}\ \eta + \mathsf{E}u(X-\eta). \tag{7}$$

While interpretations of formulas (6) and (7) are similar, and moreover, it can be shown that, under certain conditions on the utility function, $\rho(X) = -\mathrm{OCE}(X)$ is a convex measure of risk, there are important differences between these representations. In (7), the quantity being maximized is technically not a certainty equivalent, while the authors have argued that specific conditions on the utility function u allowed them to consider it as one. In addition, representation (7) entails addition of values with generally inconsistent units, e.g., dollars and utility. Finally, as shown above, representation (6) allows for constructing both coherent and convex measures of risk, while the OCE approach yields a coherent risk measure if and only if the utility function is piecewise linear.

Remark 2.8. It is straightforward to observe that by choosing the one-sided deutility function in (6) in the form $v(t) = [t]_+$ one obtains the well-known Conditional-Value-at-Risk (CVaR) measure (Rockafellar and Uryasev, 2002), while one-sided deutility $v(t) = [t]_+^p$ yields the Higher-Moment Coherent Risk (HMCR) measures (Krokhmal, 2007).

Remark 2.9. In general, risk measure ρ is called a *tail measure of risk* if it quantifies the risk of X through its right tail, $[X - c]_+$, where the tail cutoff point c can be adjusted according to risk preferences (Krokhmal et al., 2011). Observe that the above analysis implies that coherent or convex risk measures based on certainty equivalents (6) are necessarily tail measures of risk (see also Propositions 2.14 and 2.15 below).

Another key property of the certainty equivalent measures of risk (6) is that they "naturally" preserve stochastic orderings induced on the space $\mathcal{X}$ of random outcomes by the utility function u or, equivalently, deutility v. Assuming again that $\mathcal{X}$ is endowed with necessary integrability properties, consider the properties of *isotonicity with respect to second order stochastic dominance (SSD)* (see, e.g., De Giorgi, 2005; Pflug, 2006):

(A1′) *SSD isotonicity:* $\rho(X) \le \rho(Y)$ for all $X, Y \in \mathcal{X}$ such that $-X \succcurlyeq_{\mathrm{SSD}} -Y$,

and, more generally, *isotonicity with respect to k-th order stochastic dominance (kSD)*:

(A1″) *kSD isotonicity:* $\rho(X) \le \rho(Y)$ for all $X, Y \in \mathcal{X}$ such that $-X \succcurlyeq_{k\mathrm{SD}} -Y$,
for a given $k \ge 1$.

Recall that random outcome X is said to dominate outcome Y with respect to second-order stochastic dominance, $X \succcurlyeq_{\mathrm{SSD}} Y$, if

$$\int_{-\infty}^{t} F_X(\xi)\,\mathrm{d}\xi \le \int_{-\infty}^{t} F_Y(\xi)\,\mathrm{d}\xi \quad \text{for all } \ t \in \mathbb{R},$$

where $F_Z(t) = \mathsf{P}\{Z \le t\}$ is the c.d.f. of a random element $Z \in \mathcal{X}$. Similarly, outcome X dominates outcome Y with respect to k-th order stochastic dominance, $X \succcurlyeq_{k\mathrm{SD}} Y$, if

$$F_X^{(k)}(t) \le F_Y^{(k)}(t), \quad \text{for all } \ t \in \mathbb{R},$$

where $F_X^{(k)}(t) = \int_{-\infty}^{t} F_X^{(k-1)}(\xi)\mathrm{d}\xi$ and $F_X^{(1)}(t) = \mathsf{P}\{X \le t\}$ (see, for example, Ogryczak and Ruszczyński, 2001). Stochastic dominance relations in general, and SSD in particular have occupied a prominent place in decision making literature (see, for a example, Levy (1998) for an extensive account), in particular due to a direct connection to the expected utility theory. Namely, it is well known

(Rothschild and Stiglitz, 1970) that $X \succcurlyeq_{\mathrm{SSD}} Y$ if and only if $\mathsf{E}u(X) \geq \mathsf{E}u(Y)$ for all non-decreasing and concave utility functions u, i.e., if and only if Y is never preferred over X by any rational risk-averse decision maker. In general, it can be shown that $X \succcurlyeq_{k\mathrm{SD}} Y$ if and only if $\mathsf{E}u(X) \geq \mathsf{E}u(Y)$ for all $u \in U^{(k)}$, where $U^{(k)}$ is a specific class of real-valued utility functions; particularly, $U^{(1)}$ consists of all non-decreasing functions, $U^{(2)}$ contains all non-decreasing and concave functions, $U^{(3)}$ amounts to all non-decreasing, concave functions with convex derivative, and so on (see, for example, Fishburn (1977) and references therein). This characterization of kSD dominance relation naturally implies that the proposed certainty equivalent representation yields risk measures that are necessarily kSD-isotonic, given that the set of considered deutility functions is appropriately restricted.

Proposition 2.10. *If deutility function v is such that $-v(-t) \in U^{(k)}$, then risk measure ρ given by the certainty equivalent representation* (6) *is kSD-isotonic, i.e., satisfies* (A1$''$).

Proof. Follows immediately from the definitions of kSD dominance, kSD isotonicity, and the above discussion. □

Corollary 2.11. *If a real-valued function v is a one-sided deutility, then* (6) *defines a risk measure that is isotonic with respect to second order stochastic dominance.*

Note that Proposition 2.10 does not require the certainty equivalent in (6) to be convex. In this context, the certainty equivalent representation (6) ensures that the risk-averse preferences expressed by a given utility (equivalently, deutility) function are “transparently” inherited by the corresponding certainty equivalent measure of risk.

2.4 Optimality Conditions and Some Properties of Optimal η

Consider the definition of Conditional Value-at-Risk,

$$\mathsf{CVaR}_\alpha(X) = \min_\eta \eta + \frac{1}{1-\alpha}\mathsf{E}[X-\eta]_+.$$

The lowest value of η that delivers minimum in this definition is know in the literature as Value-at-Risk (VaR) at confidence level α, and while VaR in general is not convex, it is widely used as a measure of risk in practice, especially in financial applications (Jorion, 1997; Duffie and Pan, 1997). Thus, it is of interest to investigate some properties of $\eta^*(X) \in \arg\min\big\{\eta + \frac{1}{1-\alpha}v^{-1}\mathsf{E}v(X-\eta)\big\}$. First, we formulate the necessary and sufficient optimality conditions.

Proposition 2.12. *Suppose that v is a non-decreasing and convex function, certainty equivalent $v^{-1}\mathsf{E}v$ is convex and $\mathsf{E}\partial_\pm v(X-\eta^*)$ is well defined, then $\eta^* \in \arg\min\big\{\eta + \frac{1}{1-\alpha}v^{-1}\mathsf{E}v(X-\eta)\big\}$ if and only if*

$$\begin{aligned}\partial_- v^{-1}\big(\mathsf{E}v(X-\eta^*)\big)\cdot \mathsf{E}\partial_- v(X-\eta^*)&\\ \leq 1-\alpha \leq \partial_+ v^{-1}\big(\mathsf{E}v(X-\eta^*)\big)\cdot \mathsf{E}\partial_+ v(X-\eta^*),&\end{aligned}$$

where $\partial_\pm v$ denote one-sided derivatives of v with respect to the argument.

Proof. Let us denote $\phi_X(\eta) = \eta + \frac{1}{1-\alpha}v^{-1}\mathsf{E}v(X-\eta)$. Since certainty equivalent $v^{-1}\mathsf{E}v$ is convex, ϕ_X is also convex, and thus, it has left and right derivatives everywhere on $\operatorname{dom}\phi_X \neq \emptyset$, and η delivers a minimum to ϕ_X if and only if $\partial_-\phi_X(\eta) \le 0 \le \partial_+\phi_X(\eta)$. In what follows, we determine closed form expressions for left and right derivatives of ϕ_X. By definition, if $\eta \in \operatorname{dom}\phi_X$ then

$$\begin{aligned}\partial_+\phi_X(\eta) &= \lim_{\varepsilon\downarrow 0}\frac{\phi_X(\eta+\varepsilon)-\phi_X(\eta)}{\varepsilon}\\ &= 1+\frac{1}{1-\alpha}\lim_{\varepsilon\downarrow 0}\frac{v^{-1}\mathsf{E}v(X-\eta-\varepsilon)-v^{-1}\mathsf{E}v(X-\eta)}{\varepsilon}.\end{aligned}$$

Repeating a usual argument used to prove the chain rule of differentiation (see, e.g., Randolph, 1952), we can define

$$Q(y) = \begin{cases} \dfrac{v^{-1}(y)-v^{-1}\mathsf{E}v(X-\eta)}{y-\mathsf{E}v(X-\eta)}, & y < \mathsf{E}v(X-\eta),\\ \partial_- v^{-1}\big(\mathsf{E}v(X-\eta)\big), & \text{otherwise,}\end{cases}$$

in which case

$$\partial_+\phi_X(\eta) = 1+\frac{1}{1-\alpha}\lim_{\varepsilon\downarrow 0}\left\{Q\big(\mathsf{E}v(X-\eta-\varepsilon)\big)\frac{\mathsf{E}v(X-\eta-\varepsilon)-\mathsf{E}v(X-\eta)}{\varepsilon}\right\}.$$

Clearly, $\lim_{\varepsilon\downarrow 0} Q\big(\mathsf{E}v(X-\eta-\varepsilon)\big) = \partial_- v^{-1}\big(\mathsf{E}v(X-\eta)\big)$ by monotone convergence theorem, and the only part left to find is

$$\lim_{\varepsilon\downarrow 0}\frac{\mathsf{E}v(X-\eta-\varepsilon)-\mathsf{E}v(X-\eta)}{\varepsilon} = -\lim_{\varepsilon\downarrow 0}\frac{\mathsf{E}v(X-\eta)-\mathsf{E}v(X-\eta-\varepsilon)}{\varepsilon}.$$

Observe that $\lim\limits_{\varepsilon\downarrow 0}\dfrac{v(x-\eta)-v(x-\eta-\varepsilon)}{\varepsilon} = \partial_- v(x-\eta)$ for any fixed $x \in \mathbb{R}$ (note that $\partial_- v(x-\eta)$ exists since v is convex). Moreover,

$$\frac{v(x-\eta)-v(x-\eta-\varepsilon)}{\varepsilon} \nearrow \partial_- v(x-\eta) \quad \text{as} \quad \varepsilon \searrow 0,$$

where $\nearrow$ denotes monotonic convergence from below (Rockafellar, 1997, Theorem 23.1). Thus, by monotone convergence theorem, we can interchange the limit and expectation:

$$\begin{aligned}\lim_{\varepsilon\downarrow 0}\frac{\mathsf{E}v(X-\eta)-\mathsf{E}v(X-\eta-\varepsilon)}{\varepsilon} &= \mathsf{E}\lim_{\varepsilon\downarrow 0}\frac{v(X-\eta)-v(X-\eta-\varepsilon)}{\varepsilon}\\ &= \mathsf{E}\partial_- v(X-\eta),\end{aligned}$$

i.e., $\partial_+\phi_X(\eta) = 1-\frac{1}{1-\alpha}\partial_- v^{-1}\big(\mathsf{E}v(X-\eta)\big)\cdot\mathsf{E}\partial_- v(X-\eta)$. Similar arguments can be invoked to evaluate $\partial_-\phi_X(\eta)$ in order to complete the proof. □

Corollary 2.13. *Condition*

$$(v^{-1})'(\mathsf{E}v(X-\eta))\mathsf{E}v'(X-\eta) = 1-\alpha$$

is sufficient for η to deliver the minimum in (5)*, given that $(v^{-1})'$ and v' are well-defined.*

Conditions established above show that for a fixed X, the location of $\eta^*(X)$ is determined by the parameter α. Two propositions below illustrate this observation.

Proposition 2.14. *Given an* $X \in \operatorname{dom}\rho$ *for all* $\alpha \in (0,1)$*, if* $\eta^*_\alpha(X) \in \arg\min \eta + \frac{1}{1-\alpha} v^{-1}\mathsf{E}v(X-\eta)$*, where* v *is a one-sided deutility function, and certainty equivalent* $v^{-1}\mathsf{E}v$ *exists for any* X*, and is convex, then* $\eta^*_{\alpha_1}(X) \leq \eta^*_{\alpha_2}(X)$ *for any* $\alpha_1 < \alpha_2$.

Proof. Below we will use $\eta^*_\alpha(X)$ and η^*_α interchangeably in order to simplify the notation. Let $\alpha_1 < \alpha_2$. Since v is a one-sided deutility, then $v(X-\eta) = v([X-\eta]_+)$, and by the definition of $\eta^*_\alpha(X)$,

$$\eta^*_{\alpha_1} + \frac{1}{1-\alpha_1} v^{-1}\mathsf{E}v\big([X-\eta^*_{\alpha_1}]_+\big) \leq \eta^*_{\alpha_2} + \frac{1}{1-\alpha_1} v^{-1}\mathsf{E}v\big([X-\eta^*_{\alpha_2}]_+\big).$$

Suppose that $\eta^*_{\alpha_1} > \eta^*_{\alpha_2}$, then one has

$$\begin{aligned} 0 < \eta^*_{\alpha_1} - \eta^*_{\alpha_2} &\leq \frac{1}{1-\alpha_1}\Big(v^{-1}\mathsf{E}v\big([X-\eta^*_{\alpha_2}]_+\big) - v^{-1}\mathsf{E}v\big([X-\eta^*_{\alpha_1}]_+\big)\Big) \\ &< \frac{1}{1-\alpha_2}\Big(v^{-1}\mathsf{E}v\big([X-\eta^*_{\alpha_2}]_+\big) - v^{-1}\mathsf{E}v([X-\eta^*_{\alpha_1}]_+)\Big). \end{aligned}$$

This immediately leads to

$$\eta^*_{\alpha_1} + \frac{1}{1-\alpha_2} v^{-1}\mathsf{E}v\big([X-\eta^*_{\alpha_1}]_+\big) < \eta^*_{\alpha_2} + \frac{1}{1-\alpha_2} v^{-1}\mathsf{E}v\big([X-\eta^*_{\alpha_2}]_+\big),$$

which contradicts with the definition of $\eta^*_{\alpha_2}$, thus furnishing the statement of the proposition. □

Proposition 2.15. *Given an* $X \in \operatorname{dom}\rho$ *for all* $\alpha \in (0,1)$*, if* $\eta^*_\alpha(X) \in \arg\min \eta + \frac{1}{1-\alpha} v^{-1}\mathsf{E}v(X-\eta)$*, where* v *is a one-sided deutility function, and certainty equivalent* $v^{-1}\mathsf{E}v$ *exists for any* X *and is convex, then*

$$\lim_{\alpha\to 1} \eta^*_\alpha(X) = \text{ess.sup}(X).$$

Proof. Again, let us consider function $\phi_X(\eta) = \eta + \frac{1}{1-\alpha} v^{-1}\mathsf{E}v(X-\eta)$, and since v is a one-sided deutility, $\phi_X(\eta) = \eta + \frac{1}{1-\alpha} v^{-1}\int_{X\geq\eta} v(X-\eta)\mathrm{d}\mathsf{P}$. Suppose that $\text{ess.sup}(X) = A < +\infty$, consequently $\mathsf{P}(X \geq A-\varepsilon) > 0$ for any $\varepsilon > 0$. Note that $\phi_X(A) = A$. Now,

$$\begin{aligned} \phi_X(A-\varepsilon) &= A - \varepsilon + \frac{1}{1-\alpha} v^{-1}\int_{X\geq A-\varepsilon} v(X-A+\varepsilon)\mathrm{d}\mathsf{P} \\ &\geq A - \varepsilon + \frac{1}{1-\alpha} v^{-1}\int_{X\geq A-\frac{\varepsilon}{2}} v(X-A+\varepsilon)\mathrm{d}\mathsf{P} \\ &\geq A - \varepsilon + \frac{1}{1-\alpha} v^{-1}\Big(v\Big(\frac{\varepsilon}{2}\Big)\mathsf{P}\Big(X \geq A - \frac{\varepsilon}{2}\Big)\Big) = A - \varepsilon + \frac{1}{1-\alpha} M_\varepsilon, \end{aligned}$$

where $M_\varepsilon = v^{-1}\Big(v\big(\frac{\varepsilon}{2}\big)\mathsf{P}\big(X \geq A - \frac{\varepsilon}{2}\big)\Big) > 0$. Hence, $\phi_X(A-\varepsilon) > \phi_X(A)$ for any sufficiently large values of α, which means that in this case any $\eta^*_\alpha(X) \in \arg\min \eta + \frac{1}{1-\alpha} v^{-1}\mathsf{E}v(X-\eta)$ has to satisfy $\eta^*_\alpha(X) \in (A-\varepsilon, A]$, and thus $\lim_{\alpha\to 1}\eta^*_\alpha(X) = A = \text{ess.sup}(X)$.

Now, let $\text{ess.sup}(X) = +\infty$. Note that $\int_{X\geq\eta} v(X-\eta)\mathrm{d}\mathsf{P}$ is a non-increasing function of η. Let $A \in \mathbb{R}$ and $\phi_X(A) = A + \frac{1}{1-\alpha} v^{-1}\int_{X\geq A} v(X-A)\mathrm{d}\mathsf{P}$. Since $\text{ess.sup}(X) = +\infty$, there exists $\tilde{A} > A$ such that $0 < \int_{X\geq\tilde{A}} v(X-\tilde{A})\mathrm{d}\mathsf{P} < \int_{X\geq A} v(X-A)\mathrm{d}\mathsf{P}$. Thus, $\phi_X(\tilde{A}) = \tilde{A} + \frac{1}{1-\alpha} v^{-1}\int_{X\geq\tilde{A}} v(X-\tilde{A})\mathrm{d}\mathsf{P} < \phi_X(A)$ for any sufficiently large α, which yields $\eta^*_\alpha(X) > A$. Since the value of A has been selected arbitrarily, $\lim_{\alpha\to 1}\eta^*_\alpha(X) = +\infty = \text{ess.sup}(X)$. □

3 Application: Log-Exponential Convex Measures of Risk

As it was already mentioned above, CVaR and HMCR measures can be defined in terms of the proposed certainty equivalent-based representation (6). Note that both cases correspond to positively homogeneous functions ϕ, and, therefore, are coherent measures of risk. Next we consider a convex measure of risk resulting from the certainty equivalent representation (6) with an exponential one-sided deutility function $v(t) = -1 + \lambda^{[t]_+}$:

$$\rho_\alpha^{(\lambda)}(X) = \min_\eta \ \eta + \frac{1}{1-\alpha} \log_\lambda \mathsf{E}\lambda^{[X-\eta]_+}, \quad \text{where} \quad \lambda > 1 \quad \text{and} \quad \alpha \in (0,1). \tag{8}$$

We refer to such $\rho_\alpha^{(\lambda)}$ as the family of *log-exponential convex risk (LogExpCR) measures*. First, using the general framework developed above, it can be readily seen that LogExpCR family are convex measures of risk.

Proposition 3.1. *Functions $\rho_\alpha^{(\lambda)}(X)$ defined by* (8) *are proper convex measures of risk.*

Proof. Follows immediately from Theorem 2.5, Proposition 2.7, and Corollary 2.9. □

A particular member of the family of LogExpCR measures is determined by the values of two parameters, α and λ. Recall that in Section 2.4 we have established that parameter α plays a key role in determining the position of $\eta_\alpha^*(X) \in \arg\min \eta + \frac{1}{1-\alpha} v^{-1} \mathsf{E}v(X-\eta)$, particularly, $\alpha_1 < \alpha_2$ leads to $\eta_{\alpha_1}^*(X) \leq \eta_{\alpha_2}^*(X)$, and $\lim_{\alpha\to 1} \eta_\alpha^*(X) = \text{ess.sup}(X)$. These two properties allow us to conclude that α determines the "length" of the tail of distribution of X, or, in other words, determines which part of the distribution should be considered "risky". This is in accordance with a similar property of the CVaR measure, which, in the case of a continuous loss distribution, quantifies the risk as the expected loss in the worst $1-\alpha$ percent of the cases. See Krokhmal (2007) for a similar argument for HMCR measures.

Furthermore, one has

$$\begin{aligned}
\rho_\alpha^{(\lambda)}(X) &= \min_\eta \eta + \frac{1}{1-\alpha} \log_\lambda \mathsf{E}\lambda^{[X-\eta]_+} = \min_\eta \eta + \frac{1}{1-\alpha}\frac{1}{\ln\lambda} \ln \mathsf{E}e^{\ln\lambda[X-\eta]_+} \\
&= \frac{1}{\ln\lambda} \min_\eta \eta\ln\lambda + \frac{1}{1-\alpha} \mathsf{E}e^{[X\ln\lambda - \eta\ln\lambda]_+} = \\
&\frac{1}{\ln\lambda} \min_{\eta'} \eta' + \frac{1}{1-\alpha} \mathsf{E}e^{[X\ln\lambda - \eta']_+} = \frac{1}{\ln\lambda}\rho_\alpha^{(e)}(X\ln\lambda).
\end{aligned}$$

This implies that LogExpCR measures satisfy a "quasi positive homogeneity" property:

$$\rho_\alpha^{(\lambda)}(X)\ln\lambda = \rho_\alpha^{(e)}(X\ln\lambda),$$

where parameter $\ln\lambda$ plays the role of a scaling factor. Thus, in the case of log-exponential convex measures of risk (8), scaling can be seen as a way to designate the total range of the loss variable. Consequently, a combination of the parameters α and λ determines both the region of the loss distribution that should be considered "risky", and the emphasis that should be put on the larger losses. Note that the specific values of these parameters depend on the decision maker's preferences and attitude towards risk. In practice, they may be determined and/or calibrated through preliminary computational experiments.

It is of interest to note that LogExpCR measures are isotonic with respect to *any* order $k \geq 1$ of stochastic dominance:

Proposition 3.2. *The family of log-exponential convex measures of risk* (8) *are kSD-isotonic for any $k \geq 1$, i.e., $\rho_\alpha^{(\lambda)}(X) \leq \rho_\alpha^{(\lambda)}(Y)$ for all $X, Y \in \mathcal{X}$ such that $-X \succcurlyeq_{kSD} -Y$.*

Proof. Follows immediately from Proposition 2.10 for v defined above. □

Based on these observations and the preceding discussion, we can conclude that the introduced family of LogExpCR measures possesses a number of desirable properties from both optimization and methodological perspectives. It is widely acknowledged in the literature that risk is associated with "heavy" tails of the loss distribution; for example, in Krokhmal (2007) it has been illustrated that evaluating risk exposure in terms of higher tail moments can lead to improved decision making in financial applications with heavy-tailed distributions of asset returns. Furthermore, there are many real-life applications where risk exposure is associated with catastrophic events of very low probability and extreme magnitude, such as natural disasters, which often turn out to be challenging for traditional analytic tools (see, for example, Kousky and Cooke, 2009; Cooke and Nieboer, 2011 and references therein, Iaquinta et al., 2009, or Kreinovich et al., 2012). By construction, LogExpCR measures quantify risk by putting extra emphasis on the tail of the distribution, which allows us to hypothesize that they could perform favorably compared to conventional approaches in situations that involve heavy-tailed distributions of losses and catastrophic risks. This conjecture has been tested in two numerical case studies that are presented next. The idea is to evaluate the quality of solutions based on the risk estimates due to nonlinear LogExpCR measure with those obtained using linear CVaR measure, which can now be considered as a standard approach in risk-averse applications. Particularly, we were interested in assessing the influence that the behavior of the tails of the underlying losses distributions has in this comparison.

3.1 Case Study 1: Flood Insurance Claims Model

Dataset description For the first part of the case study we used a dataset managed by a non-profit research organization *Resources for the Future* (Cooke and Nieboer, 2011). It contains flood insurance claims, filed through National Flood Insurance Program (NFIP), aggregated by county and year for the State of Florida from 1980 to 2006. The data is in 2000 US dollars divided by personal income estimates per county per year from the Bureau of Economic Accounts (BEA), in order take into account substantial growth in exposure to flood risk. The dataset has 67 counties, and spans for 355 months.

Model formulation Let random vector $\boldsymbol{\ell}$ represent the dollar values of insurance claims (individual elements of this vector correspond to individual counties), and consider the following stochastic programming problem, where ρ is a risk measure:

$$\min \quad \rho(\boldsymbol{\ell}^\top \mathbf{x}) \tag{9a}$$

$$\text{s. t.} \quad \sum_i x_i = K \tag{9b}$$

$$x_i \in \{0, 1\}. \tag{9c}$$

Such a formulation allows for a straightforward interpretation, namely, the goal here is to identify K counties with a minimal common insurance risk due to flood as estimated by ρ. Clearly, such a simplified model does not reflect the complexities of real-life insurance operations. At the same time, since the purpose of this case study is to analyze the properties of risk measures themselves, a deliberately simple formulation was chosen so as to highlight the differences between solutions of (9) due to different choices of the risk measure ρ in (9a).

Given that the distribution of $\boldsymbol{\ell}$ is represented by equiprobable scenario realizations $\ell_{i,j}$, and m is the number of scenarios (time periods), model (9) with risk measure chosen as the Conditional Value-at-Risk, $\rho(X) = \mathsf{CVaR}_\alpha(X)$, can be expressed as

$$\min \quad \eta + \frac{1}{1-\alpha_{\text{CVaR}}} \sum_j \frac{1}{m} \Big[\sum_i x_i \ell_{i,j} - \eta \Big]_+ \tag{10a}$$

$$\text{s. t.} \quad \sum_i x_i = K \tag{10b}$$

$$x_i \in \{0, 1\}. \tag{10c}$$

Similarly, if a LogExpCR measure is used, $\rho(X) = \rho_\alpha^{(e)}(X)$, then (9) can be formulated as

$$\min \quad \eta + \frac{1}{1-\alpha_{\text{LogExpCR}}} \log \sum_j \frac{1}{m} e^{[\sum_i x_i \ell_{i,j} - \eta]_+} \tag{11a}$$

$$\text{s. t.} \quad \sum_i x_i = K \tag{11b}$$

$$x_i \in \{0, 1\}. \tag{11c}$$

Normal data In order to evaluate the effect of the tail behavior of the loss distribution on the obtained solutions of decision making problems, we additionally generated a similar dataset based on normal distribution. Particularly, we draw 355 realizations from 67-dimensional normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are mean and covariance estimates of NFIP data respectively. Our goal here is to make sure that the main difference between the datasets lays in the tails (normal distribution is a well-known example of a light-tailed distribution), and by preserving mean vector and covariance matrix we secure that this dataset captures the leading trends present in the original data. Now, by comparing the decisions due to CVaR and LogExpCR measures for these two datasets we can make conclusions on the effects that the tails of the distributions have on the quality of subsequent decisions.

Implementation details Problems (10) and (11) represent a mixed-integer linear programming (MIP) and a mixed-integer non-linear programming (MINLP) problems respectively. MIP problems were solved using IBM ILOG CPLEX 12.5 solver accessed through C++ API. For the MINLPs of the form (11) we implemented a custom branch-and-bound algorithm based on outer polyhedral approximation approach, which utilized CPLEX 12.5 MIP solver and MOSEK 6.0 for NLP subproblems (Vinel and Krokhmal, 2014).

In order to evaluate the quality of the decisions we employed a usual training-testing framework. Given a preselected value m, the first m scenarios were used to solve problems (10) and (11), then for the remaining $N - m$ scenarios the total loss was calculated as $L^\rho = \sum_{j=m+1}^N \sum_i \ell_{i,j} x_i^\rho$, where $\mathbf{x}^\rho$ represents an optimal solution of either problem (10) or problem (11), and N is the total number of scenarios in the dataset. In other words, the decision vector $\mathbf{x}^\rho$ is selected based on the first m observations of the historical data (training), and the quality of this solution is estimated based on the "future" realizations (testing).

In this case study we have set $\alpha_{\text{CVaR}} = 0.9$, which is a typical choice in portfolio optimization and can be interpreted as cutting off 90% of the least significant losses. A preliminary test experiment has been performed to select α_{LogExpCR} in such a way that approximately same portion of the distribution was cut off, which yielded $\alpha_{\text{LogExpCR}} = 0.5$. For the sake of simplicity, we set $\lambda = e$.

Discussion of results Tables 1 and 2 summarize the obtained results for NFIP and simulated normal data sets, respectively. Discrepancy in the quality of the decisions based on LogExpCR and CVaR measures is estimated using the value

$$\gamma = \frac{L^{\text{LogExpCR}} - L^{\text{CVaR}}}{\min\left\{L^{\text{LogExpCR}}, L^{\text{CVaR}}\right\}},$$

which represents the relative difference in total losses L^{LogExpCR} and L^{CVaR} associated with the respective decisions. For example, $\gamma = -100\%$ corresponds to the case when losses due to CVaR-based decision were twice as large as losses due to LogExpCR-based decision.

First of all, we can observe that there is a definite variation between the results obtained with NFIP data on one hand and with simulated normal data on the other. Particularly, the absolute values of γ in Table 2 on average are considerably smaller compared to those in Table 1, which indicates that in the case of normal data the risk measures under consideration result in similar decisions, while heavy-tailed historical data leads to much more differentiated decisions.

Secondly, Table 1 suggests that LogExpCR measure yields considerably better solutions for certain sets of parameter values. Most notably, such instances correspond to smaller values of both K and m. Intuitively, this can be explained as follows. Recall that m is the number of scenarios in the training set, and $N - m$ is the number of scenarios in the testing set, which means that larger values of m correspond to shorter testing horizon. Clearly, the fewer scenarios there are in the testing set, the fewer catastrophic losses occur during this period, and vice versa, for smaller values of m there are more exceptionally high losses in the future. Thus, the observed behavior of γ is in accordance with our conjecture that LogExpCR measures are better suited for instances with heavy-tailed loss distributions. Parameter K, in turn, corresponds to the number of counties to be selected, thus, the larger its value is, the more opportunities for diversification are available for the decision-maker, which, in turn, allows for risk reduction.

To sum up, the results of this case study suggest that under certain conditions, such as heavy-tailed loss distribution, relatively poor diversification opportunities, and sufficiently large testing horizon, risk-averse decision strategies based on the introduced log-exponential convex measures of risk can substantially outperform strategies based on linear risk measures, such as the Conditional Value-at-Risk.

3.2 Case Study 2: Portfolio Optimization

As heavy-tailed loss distributions are often found in financial data, we conducted numerical experiments with historical stock market data as the second part of the case study.

Model description As the underlying decision making model we use the traditional risk-reward portfolio optimization framework introduced by Markowitz (1952). In this setting, the cost/loss outcome X is usually defined as the portfolio's negative rate of return, $X(\mathbf{x}, \omega) = -\mathbf{r}(\omega)^\top \mathbf{x}$, where $\mathbf{x}$ stands for the vector of portfolio weights, and $\mathbf{r} = \mathbf{r}(\omega)$ is the uncertain vector of assets' returns. Then, a portfolio allocation problem can be formulated as the problem of minimizing some measure of risk associated with the portfolio while maintaining a prescribed expected return:

$$\min_{\mathbf{x}\in\mathbb{R}^n_+} \left\{ \rho(-\mathbf{r}^\top\mathbf{x}) \,\middle|\, \mathsf{E}(\mathbf{r}^\top\mathbf{x}) \geq \bar{r},\ \mathbf{1}^\top\mathbf{x} \leq 1 \right\}, \tag{12}$$

where $\bar{r}$ is the prescribed level of expected return, $\mathbf{x} \in \mathbb{R}^n_+$ denotes the no-short-selling requirement, and $\mathbf{1} = (1, \dots, 1)^\top$. If the risk measure used is convex, it is easy to see that (12) is a convex optimization problem. In this case study, we again select ρ in (12) as either a LogExpCR or CVaR measure.

Dataset description We utilized historical stock market data available through Yahoo!Finance. We picked 2178 listings traded at NYSE from March, 2000 through December, 2012 (total of 3223 trading days). As it was noted above, financial data often exhibit highly volatile behavior, especially higher-frequency data, while long-term data is usually relatively normal. In order to account for such differences, we generated three types of datasets of loss distribution, which were based on two-day, two-week and one-month historical returns. Particularly, if $p_{i,j}$ is the historical close price of asset i on day j, then we define the corresponding two-day, ten-day, and one-month returns as $r_{i,j} = \frac{p_{i,j}-p_{i,j-\Delta}}{p_{i,j-\Delta}}$, where Δ takes values $\Delta = 2$, 10, and 20, respectively.

Implementation details We utilize a training-testing framework similar to the one used in the previous section, but additionally, we also employ "rolling horizon" approach, which aims to simulate a real-life self-financing trading strategy. For a given time moment, we generate a scenario set containing, respectively, m two-day, ten-day, and one-month returns immediately preceding this date. Then, the portfolio optimization problem (12) is solved for each type of scenario set in order to obtain the corresponding optimal portfolios; the "realized" portfolio return over the next two-day, ten-day, or one-month time period, respectively, is then observed. The portfolio is then rebalanced using the described procedure. This rolling-horizon procedure was ran for 800 days, or about 3 years.

Recall that parameter $\bar{r}$ in (12) represents the "target return", i.e., the minimal average return of the portfolio. For our purposes parameter $\bar{r}$ was selected as $\bar{r} = \tau \max_i\{\mathsf{E}_\omega r_i(\omega)\}$, i.e., as a certain percentage of the maximum expected return previously observed in the market (within the timespan of the current scenario set). Parameter τ has been set to be "low", "moderate", or "high", which corresponds to $\tau = 0.1, 0.5, 0.8$. For each pair of n and m we repeat the experiment 20 times, selecting n stocks randomly each time. The parameters α_{LogExpCR}, α_{CVaR}, and λ have been assigned the same values as in Case Study 1.

Discussion of results Obtained results are summarized in Table 3, and a typical behavior of the portfolio value over time is presented in Figure 1. As in the previous case, we report relative difference in the return over appropriate time period (2-day, 2-week, or 1-month) averaged over the testing horizon of 800 days and over 20 random choices of n assets. Note that since in this case the quality of the decision is estimated in terms of rate of return, i.e., gain, positive values in Table 3 correspond to the cases when the LogExpCR-based portfolio outperforms the CVaR-based portfolio.

Similarly to the previous case, we can observe that the behavior of the tails of the distribution plays a key role in the comparison: under 1-month trading frequency the differences between CVaR and LogExpCR portfolios are relatively insignificant, compared to the 2-day case. Moreover, we can again conclude that for heavy-tailed loss distributions the introduced LogExpCR measure may compare favorably against

CVaR; in particular, conditions of restricted diversification options (relatively small value of n) make utilization of LogExpCR measures more beneficial compared to a linear measure such as CVaR.

4 Concluding Remarks

In this paper we introduced a general representation of the classes of convex and coherent risk measures by showing that any convex (coherent) measure can be defined as an infimal convolution of the form $\rho(X) = \min_\eta \eta + \phi(X - \eta)$, where ϕ is monotone, convex, and $\phi(\eta) > \eta$ for all $\eta \neq 0$, $\phi(0) = 0$ (and positive homogeneous for coherence), and vice versa, constructed in such a way function ρ is convex (coherent). Another way to look at this result is to observe that a monotone and convex ϕ only lacks translation invariance in order to satisfy the definition of a convex risk measure, and infimal convolution operator essentially forges this additional property, while preserving monotonicity and convexity. According to this scheme, a risk measure is represented as a solution of an optimization problem, hence it can be readily embedded in a stochastic programming model.

Secondly, we apply the developed representation to construct risk measures as infimal convolutions of certainty equivalents, which allows for a direct incorporation of risk preferences as given by the utility theory of von Neumann and Morgenstern (1944) into a convex or coherent measure of risk. This is highly desirable since, in general, the risk preferences induced by convex or coherent measures of risk are inconsistent with risk preferences of rational expected-utility maximizers. It is also shown that the certainty equivalent-based measures of risk are "naturally" consistent with stochastic dominance orderings.

Finally, we employ the proposed scheme to introduce a new family of risk measures, which we call the family of log-exponential convex risk measures. By construction, LogExpCR measures quantify risk by placing emphasis on extreme or catastrophic losses; also, the LogExpCR measures have been shown to be isotonic (consistent) with respect to stochastic dominance of arbitrary order. The results of the conducted case study show that in highly risky environments characterized by heavy-tailed loss distribution and limited diversification opportunities, utilization of the proposed LogExpCR measures can lead to improved results comparing to the standard approaches, such as those based on the well-known Conditional Value-at-Risk measure.

Acknowledgements

This work was supported in part by the Air Force Office of Scientific Research grant FA9550-12-1-0142 and National Science Foundation grant EPS1101284. In addition, support by the Air Force Research Laboratory Mathematical Modeling and Optimization Institute is gratefully acknowledged.

References

Acerbi, C. (2002) "Spectral measures of risk: A coherent representation of subjective risk aversion," *Journal of Banking and Finance*, **26** (7), 1487–1503.

Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. (1999) "Coherent measures of risk," *Math. Finance*, **9** (3), 203–228.

Table 1: Relative difference (in %) in total loss $\gamma = \frac{L^{\text{LogExpCR}} - L^{\text{CVaR}}}{\min\{L^{\text{LogExpCR}}, L^{\text{CVaR}}\}}$ for *NFIP data* for various values of the parameters K and m. Entries in bold correspond to the instances for which LogExpCR measure outperformed CVaR.

$K\backslash m$	20	60	100	140	180	220	260	300
1	**–45038.3**	**–3944.4**	**–4652.2**	**–3663.7**	**–3663.7**	**–220.2**	**–220.2**	**–220.2**
3	**–1983.7**	**–971.0**	**–211.5**	**–146.7**	**–146.7**	**–68.4**	0.0	0.0
5	**–1284.2**	**–464.2**	**–85.7**	**–13.1**	**–13.1**	0.0	**–6.6**	0.0
7	**–853.9**	**–342.5**	0.0	0.0	0.0	**–0.4**	**–4.5**	**–13.1**
9	**–387.1**	**–282.9**	0.0	0.0	0.0	0.0	**–3.1**	10.8
11	**–369.9**	**–181.0**	0.0	**–18.2**	14.0	**–27.8**	5.5	**–2.2**
13	**–360.4**	**–33.5**	0.0	**–13.0**	1.0	4.3	0.0	41.0
15	**–353.9**	**–27.9**	**–3.2**	3.1	0.0	**–3.6**	4.8	20.6
17	**–129.8**	**–1.1**	**–0.2**	3.7	**–26.5**	11.5	25.4	25.4
19	**–66.3**	21.6	0.9	0.0	0.0	**–2042.1**	35.4	23.1
21	**–64.0**	5.0	0.0	**–279.0**	2.5	**–1.6**	35.0	8.0
23	**–57.4**	4.8	0.0	0.7	**–65.6**	**–0.1**	20.3	81.8
25	**–49.5**	0.0	**–82.4**	0.0	**–39.2**	4.4	76.9	84.7
27	**–48.2**	0.0	**–52.2**	0.0	4.7	4.1	68.7	84.1
29	**–47.0**	**–34.3**	33.0	**–254.3**	**–463.8**	4.0	81.8	83.5
31	**–41.1**	**–31.2**	8.7	**–218.4**	**–309.8**	8.5	79.3	83.7
33	**–10.6**	46.4	**–10.0**	**–162.7**	**–161.7**	8.9	19.6	84.6
35	**–9.5**	0.0	**–12.2**	**–142.9**	**–153.1**	37.8	53.9	47.6
37	**–7.7**	12.0	**–81.7**	5.3	2.7	57.0	15.0	9.9
39	0.0	5.3	**–102.8**	45.8	45.4	43.4	8.6	5.6
41	0.0	11.4	**–77.3**	30.9	43.8	34.8	20.7	4.8
43	0.0	**–13.4**	**–11.0**	53.8	4.0	50.0	19.8	**–3.4**
45	0.0	0.0	**–28.1**	54.5	**–36.1**	26.2	14.2	8.5
47	**–9.1**	9.0	4.5	19.4	6.4	17.5	28.2	**–2.2**
49	0.0	6.4	27.8	**–3.5**	**–20.1**	7.0	0.5	**–8.5**
51	0.0	5.1	49.6	**–2.6**	1.0	**–9.5**	**–5.0**	2.1
53	39.9	**–16.2**	24.7	28.6	23.0	2.9	**–4.9**	**–137.2**
55	**–7.1**	**–8.7**	28.0	21.6	**–0.9**	5.6	**–83.5**	**–19.1**
57	0.0	3.0	28.5	3.6	4.1	2.6	**–9.1**	**–9.2**
59	0.0	20.0	9.0	2.0	**–0.6**	0.1	25.3	27.4

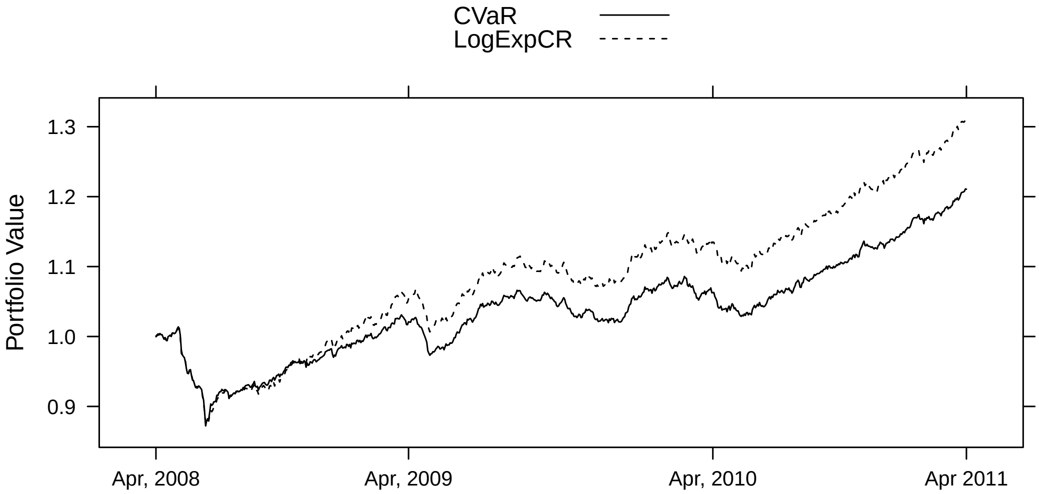
Table 2: Relative difference (in %) in total loss $\gamma = \frac{L^{\text{LogExpCR}} - L^{\text{CVaR}}}{\min\{L^{\text{LogExpCR}}, L^{\text{CVaR}}\}}$ for *normal data* for various values of the parameters K and m. Entries in bold correspond to the instances for which LogExpCR measure outperformed CVaR.

$K \backslash m$	20	60	100	140	180	220	260	300
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	2.2	**–107.1**	0.0	0.0	**–17.6**	22.8	**–58.1**
7	0.0	14.0	0.0	85.3	28.8	84.8	86.4	21.9
9	0.0	14.2	27.2	11.0	2.9	0.0	0.0	0.0
11	17.2	12.9	0.0	34.8	33.9	66.8	36.0	**–50.4**
13	19.2	16.6	**–3.3**	**–11.1**	2.6	18.0	**–12.3**	0.0
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	0.0	**–3.4**	34.0	0.0	0.0	312.1	0.0	**–355.0**
19	43.0	**–8.2**	8.9	52.3	76.7	**–65.9**	**–20.3**	0.0
21	0.0	4.3	21.5	**–45.4**	506.1	**–123.1**	**–119.2**	1.9
23	27.3	**–32.1**	48.8	75.2	242.3	**–63.1**	3.0	**–112.1**
25	**–317.3**	**–8.0**	16.9	74.7	151.1	**–71.3**	**–129.0**	**–64.7**
27	9.7	**–34.1**	31.1	**–50.8**	96.3	154.3	163.8	**–16.8**
29	7.4	13.7	19.4	78.4	44.6	272.6	**–15.3**	–31.4
31	1.8	10.3	5.3	6.4	52.6	234.0	44.8	**–5.5**
33	10.5	**–14.7**	**–15.2**	**–31.2**	**–32.8**	11.5	**–15.0**	10.1
35	9.1	6.0	0.0	0.0	36.8	36.9	437.2	0.0
37	5.1	**–1.0**	0.0	18.0	39.1	20.8	119.3	0.0
39	0.0	**–0.8**	**–1.4**	10.3	13.2	**–14.6**	**–109.7**	73.6
41	19.8	18.3	0.0	24.7	22.1	0.0	44.0	762.8
43	7.6	8.7	6.4	0.0	**–6.1**	0.0	0.0	0.0
45	6.9	5.9	11.4	7.9	6.1	16.9	–20.6	**–99.3**
47	0.0	1.1	16.6	4.0	13.0	0.0	21.5	46.7
49	**–2.8**	22.5	17.7	**–7.5**	–11.2	**–2.3**	0.0	–294.8
51	0.0	5.1	17.8	5.0	10.4	**–28.4**	–0.1	–47.4
53	**–1.1**	0.0	**–6.7**	**–0.5**	25.4	0.0	0.0	**–39.7**
55	6.8	0.0	17.5	18.3	0.0	–9.3	37.8	**–87.4**
57	1.3	0.0	0.0	**–14.5**	**–21.8**	0.0	0.0	0.0
59	6.3	0.0	0.0	0.0	0.0	0.0	0.0	**–10.0**

Table 3: Relative difference (in %) in average portfolio return due to LogExpCR measure and CVaR. Parameter n represents the total number of assets on the market, m is the number of time intervals in the training horizon, τ defines the prescribed expected rate of return as the percentage of the maximum expected return previously observed in the market. Labels "2-day", "2-week", and "1-month" correspond to portfolio rebalancing periods.

n	m	τ	2-day	2-week	1-month
20	2000	0.1	**57.3**	**29.5**	**8.3**
		0.5	**138.3**	**1.1**	–12.9
		0.8	**5.9**	–24.1	–7.4
200	2000	0.1	–17.9	–14.6	–2.2
		0.5	**11.1**	–21.1	**5.4**
		0.8	**17.6**	–13.5	–2.2

Figure 1: Typical behavior of portfolio value, as a multiple of the initial investment (1.0), over time.



Ben-Tal, A. and Teboulle, M. (2007) "An Old-New Concept of Convex Risk Measures: An Optimized Certainty Equivalent," *Mathematical Finance*, **17** (3), 449–476.

Birge, J. R. and Louveaux, F. (1997) *Introduction to Stochastic Programming*, Springer, New York.

Bullen, P. S., Mitrinović, D. S., and Vasić, P. M. (1988) *Means and their inequalities*, volume 31 of *Mathematics and its Applications (East European Series)*, D. Reidel Publishing Co., Dordrecht, translated and revised from the Serbo-Croatian.

Cooke, R. M. and Nieboer, D. (2011) "Heavy-Tailed Distributions: Data, Diagnostics, and New Developments," *Resources for the Future Discussion Paper*, (11-19).

Dana, R.-A. (2005) "A representation result for concave Schur concave functions," *Math. Finance*, **15** (4), 613–634.

De Giorgi, E. (2005) "Reward-Risk Portfolio Selection and Stochastic Dominance," *Journal of Banking and Finance*, **29** (4), 895–926.

Delbaen, F. (2002) “Coherent risk measures on general probability spaces,” in: K. Sandmann and P. J. Schönbucher (Eds.) “Advances in Finance and Stochastics: Essays in Honour of Dieter Sondermann,” 1–37, Springer.

Duffie, D. and Pan, J. (1997) “An Overview of Value-at-Risk,” *Journal of Derivatives*, **4**, 7–49.

Fishburn, P. C. (1977) “Mean-Risk Analysis with Risk Associated with Below-Target Returns,” *The American Economic Review*, **67** (2), 116–126.

Föllmer, H. and Schied, A. (2002) “Convex measures of risk and trading constraints,” *Finance Stoch.*, **6** (4), 429–447.

Föllmer, H. and Schied, A. (2004) *Stochastic finance: An introduction in discrete time*, Walter de Gruyter, Berlin, 2nd edition.

Frittelli, M. and Rosazza Gianin, E. (2005) “Law invariant convex risk measures,” in: “Advances in mathematical economics. Volume 7,” volume 7 of *Adv. Math. Econ.*, 33–46, Springer, Tokyo.

Hardy, G. H., Littlewood, J. E., and Pólya, G. (1952) *Inequalities*, Cambridge, at the University Press, 2d ed.

Iaquinta, G., Lamantia, F., Massab, I., and Ortobelli, S. (2009) “Moment based approaches to value the risk of contingent claim portfolios,” *Annals of Operations Research*, **165** (1), 97–121.

Jorion, P. (1997) *Value at Risk: The New Benchmark for Controlling Market Risk*, McGraw-Hill.

Kousky, C. and Cooke, R. M. (2009) “The unholy trinity: fat tails, tail dependence, and micro-correlations,” *Resources for the Future Discussion Paper*, 09–36.

Kreinovich, V., Chiangpradit, M., and Panichkitkosolkul, W. (2012) “Efficient algorithms for heavy-tail analysis under interval uncertainty,” *Annals of Operations Research*, **195** (1), 73–96.

Krokhmal, P., Zabarankin, M., and Uryasev, S. (2011) “Modeling and optimization of risk,” *Surveys in Operations Research and Management Science*, **16** (2), 49 – 66.

Krokhmal, P. A. (2007) “Higher moment coherent risk measures,” *Quant. Finance*, **7** (4), 373–387.

Kusuoka, S. (2001) “On law invariant coherent risk measures,” in: “Advances in mathematical economics, Vol. 3,” volume 3 of *Adv. Math. Econ.*, 83–95, Springer, Tokyo.

Kusuoka, S. (2012) “A remark on Malliavin calculus: uniform estimates and localization,” *J. Math. Sci. Univ. Tokyo*, **19** (4), 533–558 (2013).

Levy, H. (1998) *Stochastic Dominance*, Kluwer Academic Publishers, Boston-Dodrecht-London.

Markowitz, H. M. (1952) “Portfolio Selection,” *Journal of Finance*, **7** (1), 77–91.

McCord, M. and Neufville, R. d. (1986) “”Lottery Equivalents”: Reduction of the Certainty Effect Problem in Utility Assessment,” *Management Science*, **32** (1), pp. 56–60.

Ogryczak, W. and Ruszczyński, A. (2001) “On consistency of stochastic dominance and mean-semideviation models,” *Mathematical Programming*, **89**, 217–232.

Pflug, G. C. (2006) “Subdifferential representations of risk measures,” *Math. Program.*, **108** (2-3, Ser. B), 339–354.

Prékopa, A. (1995) *Stochastic Programming*, Kluwer Academic Publishers.

Randolph, J. (1952) *Calculus*, Macmillan.

Rockafellar, R. T. (1997) *Convex analysis*, Princeton Landmarks in Mathematics, Princeton University Press, Princeton, NJ, reprint of the 1970 original, Princeton Paperbacks.

Rockafellar, R. T. and Uryasev, S. (2000) “Optimization of Conditional Value-at-Risk,” *Journal of Risk*, **2**, 21–41.

Rockafellar, R. T. and Uryasev, S. (2002) “Conditional Value-at-Risk for General Loss Distributions,” *Journal of Banking and Finance*, **26** (7), 1443–1471.

Rockafellar, R. T. and Uryasev, S. (2013) “The fundamental risk quadrangle in risk management, optimization and statistical estimation,” *Surveys in Operations Research and Management Science*, **18** (12), 33 – 53.

Rockafellar, R. T., Uryasev, S., and Zabarankin, M. (2006) “Generalized deviations in risk analysis,” *Finance Stoch.*, **10** (1), 51–74.

Rothschild, M. and Stiglitz, J. (1970) “Increasing risk I: a definition,” *Journal of Economic Theory*, **2** (3), 225–243.

Vinel, A. and Krokhmal, P. (2014) “Mixed integer programming with a class of nonlinear convex constraints,” *Working paper*.

von Neumann, J. and Morgenstern, O. (1944) *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, 1953rd edition.

Wilson, R. (1979) “Auctions of Shares,” *The Quarterly Journal of Economics*, **93** (4), pp. 675–689.

A scenario decomposition algorithm for stochastic programming problems with a class of downside risk measures

Maciej Rysz[1] Alexander Vinel[1] Pavlo A. Krokhmal[1]* Eduardo L. Pasiliao[2]

[1]Department of Mechanical and Industrial Engineering,
The University of Iowa, 3131 Seamans Center, Iowa City, IA, 52242

[2]Air Force Research Lab, 101 West Eglin Blvd, Eglin AFB, FL 32542

Abstract

We present an efficient scenario decomposition algorithm for solving large-scale convex stochastic programming problems that involve a particular class of downside risk measures. The considered risk functionals encompass coherent and convex measures of risk that can be represented as an infimal convolution of a convex certainty equivalent, and include well-known measures, such as conditional value-at-risk, as special cases. The resulting structure of the feasible set is then exploited via iterative solving of relaxed problems, and it is shown that the number of iterations is bounded by a parameter that depends on the problem size. The computational performance of the developed scenario decomposition method is illustrated on portfolio optimization problems involving two families of nonlinear measures of risk, the higher-moment coherent risk measures and log-exponential convex risk measures. It is demonstrated that for large-scale nonlinear problems the proposed approach can provide up to an order of magnitude of improvement in computational time in comparison to state-of-the-art solvers, such as CPLEX, Gurobi, and MOSEK.

Keywords: Stochastic optimization, risk measures, utility theory, certainty equivalent, scenario decomposition, higher moment coherent risk measures, log-exponential convex risk measures.

1 Introduction and Motivation

Quantification of uncertainties and risk via axiomatically defined statistical functionals, such as the *coherent measures of risk* of Artzner et al. (1999), has become a widely accepted practice in stochastic optimization and decision making under uncertainty (Shapiro et al., 2009; Krokhmal et al., 2011; Uryasev and Rockafellar, 2013). Many of such risk measures admit effective utilization in "scenario-based" formulations of stochastic programming models, i.e., the stochastic optimization problems where the random parameters are assumed to have a known distribution over a finite support that is commonly

*Corresponding author, krokhmal@engineering.uiowa.edu called the *scenario set*. A typical instance of such a problem can be written as

$$\min_{\mathbf{x}\in C} \ \rho(X(\mathbf{x},\omega)), \tag{1}$$

where ρ is the risk measure, $X(\mathbf{x},\omega)$ represents a stochastic loss or cost function dependent on the decision vector $\mathbf{x} \in C \subset \mathbb{R}^n$ and a random event ω from the finite set $\Omega = \{\omega_1, \ldots, \omega_N\}$. In many practical applications accurate approximations of uncertainties may, however, require very large scenario sets ($N \gg 1$), thus potentially leading to substantial computational difficulties.

In this work, we propose an efficient algorithm for solving large-scale stochastic optimization problems involving a class of "downside", or "tail" risk measures that are constructed via *certainty equivalents*, a well known concept in the utility theory. The presented scenario decomposition algorithm exploits the special structure of the feasible set induced by the respective risk measures as well as the properties common to the considered class of risk functionals. As an illustrative example of the general approach, we consider stochastic optimization problems with higher-moment coherent risk measures (HMCR), which quantify risk via higher moments of cost or loss distributions (Krokhmal, 2007), making them advantageous in the presence of "heavy-tailed" uncertainty. We also apply the proposed method to problems with log-exponential convex risk (LogExpCR) measures (Vinel and Krokhmal, 2015).

Perhaps, the most frequently implemented risk measure in problems of type (1) is the well known Conditional Value-at-Risk (CVaR) (Rockafellar and Uryasev, 2000, 2002). When X is piecewise linear in $\mathbf{x}$ and set C is polyhedral, formulation (1) with CVaR objective or constraints reduces to a linear programming (LP) problem. Several recent studies addressed the solution efficiency of LPs with CVaR objectives or constraints for cases when the number of scenarios is large. Lim, Sherali, and Uryasev (2010) noted that (1) in this case may be viewed as a nondifferentiable optimization problem and implemented a two-phase solution approach to solve large-scale instances. In the first phase, they exploit descent-based optimization techniques to circumvent nondifferentiable points by perturbing the solution to differentiable solutions within their "relative neighborhood". The second phase employs a deflecting subgradient search direction with a step size established by an adequate target value. They further extended this approach with a third phase that resorts to the simplex algorithm after achieving convergence by employing an advanced crash-basis dependent on solutions obtained from the first two phases.

Künzi-Bay and Mayer (2006) developed a solution technique for problem (1), with measure ρ chosen as the CVaR, that utilized a specialized L-shaped method after reformulating it as a two-stage stochastic programming problem. However, Subramanian and Huang (2008) noted that the problem structure does not naturally conform to the characteristics of a two-stage stochastic program and introduced a polyhedral reformulation of the CVaR constraint with a statistics based CVaR estimator to solve a closely related version of the problem. In a followup study (Subramanian and Huang, 2009), they retained Value-at-Risk (VaR) and CVaR as unknown variables in the CVaR constraints, enabling a more efficient decomposition algorithm, as opposed to Klein Haneveld and van der Vlerk (2006), where the problem was solved as a canonical integrated chance constraint problem with preceding estimates of VaR. Espinoza and Moreno (2012) proposed a solution method for problems (1) with CVaR measures that entailed generation of aggregated scenario constraints to form smaller relaxation problems whose optimal outcomes were then used to directly evaluate the respective upper bound on the objective of the original problem.

In what follows, we develop a general scenario decomposition solution framework for solving stochastic optimization problems with certainty equivalent-based risk measures by utilizing principles related to those in Espinoza and Moreno (2012). The rest of the paper is organized as follows: A class of certainty equivalent-based risk measures that are in the focus of this study and their implementation in mathematical programming problems are discussed in Section 2. In Section 3 we propose the scenario decomposition algorithm for stochastic programming problems with structure that is induced by the risk measures described in Section 2. Lastly, experimental studies on portfolio optimization problems with large-scale data sets that demonstrate the effectiveness of the developed technique are presented in Section 4 .

2 A Class of Downside Risk Measures Based on Certainty Equivalents

In this section we describe a class of risk measures that encompasses some popular instances in risk management literature. A general solution algorithm that utilizes special properties of this class of measures will be presented in the sequel. Specifically, this algorithm applies to the so-called *coherent* and *convex* measures of risk that can be represented as an infimal convolution of certainty equivalent of some utility function. Below we recall the definitions of coherent and convex risk measures and describe the representation that motivated the present development.

In general, a risk measure $\rho(X)$ over a random outcome (specifically, a cost or a loss) X from probability space $(\Omega, \mathcal{F}, \mathsf{P})$ is defined as a lower semi-continuous (l.s.c.) mapping $\rho : \mathcal{X} \mapsto \mathbb{R}$, with $\mathcal{X}$ being the space of bounded $\mathcal{F}$-measurable functions $X : \Omega \mapsto \mathbb{R}$. In order to avoid an excessively technical discussion, we will implicitly assume that $\mathcal{X}$ is endowed with the properties necessary in the given context (e.g., integrability, and so on). Additional properties of ρ are introduced to make the corresponding risk measure well-suited for a specific application area.

Artzner et al. (1999) and Delbaen (2002) proposed the following four axioms as the desirable characteristics that a “good”, or *coherent* measure of risk should possess:

(A1) *monotonicity:* $\rho(X) \le \rho(Y)$ *for all* $X, Y \in \mathcal{X}$ *such that* $X \le Y$*;*

(A2) *convexity:* $\rho(\lambda X + (1-\lambda)Y) \le \lambda\rho(X) + (1-\lambda)\rho(Y)$ *for all* $X, Y \in \mathcal{X}$ *and* $0 \le \lambda \le 1$*;*

(A3) *positive homogeneity:* $\rho(\lambda X) = \lambda\rho(X)$ *for all* $X \in \mathcal{X}$ *and* $\lambda > 0$*;*

(A4) *translation invariance:* $\rho(X + a) = \rho(X) + a$ *for all* $X \in \mathcal{X}$ *and* $a \in \mathbb{R}$.

The following interpretations may be given to the above axioms: Axiom (A1) ensures that smaller losses lead to lower risk. From the risk management point of view, the convexity axiom (A2) promotes risk reduction via diversification; it is also of fundamental importance in the optimization context. The positive homogeneity property (A3) postulates that scaling losses by a positive factor scales risk correspondingly. Axiom (A4) allows for eliminating risk of an uncertain cost/loss profile X by adding a deterministic hedge, $\rho(X - \rho(X)) = 0$.

Since being proposed in Artzner et al. (1999) and Delbaen (2002), the axiomatic approach to defining risk measures has been widely adopted in literature, and a number of risk functionals tailored to particular preferences emerged thereafter (see, e.g., Krokhmal et al., 2011; Uryasev and Rockafellar, 2013). In particular, it has been argued that the positive homogeneity property (A3) may be omitted in many situations; the corresponding risk measures that satisfy axioms (A1), (A2), and (A4) are called *convex measures of risk* (Ruszczyński and Shapiro, 2006).

Our interest in these two classes of risk measures stems from the following *infimal convolution* representation that facilitates their use in mathematical programming problems.

Theorem 1 (Krokhmal, 2007; Vinel and Krokhmal, 2014a) *Function $\rho(X)$ is a proper coherent (resp., convex) measure of risk if and only if it can be represented by the following infimal convolution of a l.s.c. function $\phi : \mathcal{X} \mapsto \mathbb{R}$ such that $\phi(0) = 0$, $\phi(\eta) > \eta$ for all real $\eta \neq 0$, and which satisfies (A1)–(A3) (resp., (A1)–(A2)):*

$$\rho(X) = \inf_{\eta} \; \eta + \phi(X - \eta). \tag{2}$$

Moreover, the infimum in (2) *is attained for all X, so* $\inf_\eta$ *may be replaced by* $\min_{\eta\in\mathbb{R}}$.

Representation (2) can be used for construction of coherent (convex) risk measures through an appropriate choice of function ϕ. The present work concerns risk measures of type (2) that can directly incorporate decision maker's risk preferences as given by the utility theory of von Neumann and Morgenstern (1944). This is desirable in view of the well-known fact (see, e.g., Schied and Follmer, 2002) that risk preferences expressed by coherent/convex measures of risk are generally not compatible with rational risk-averse preferences (i.e., those defined by a non-decreasing concave utility function u).

Given that we operate with stochastic cost/loss variables, let $v(t) = -u(-t)$ be the utility function adapted to loss variable X, or *deutility* function that quantifies dissatisfaction with cost or loss X. Then, $\mathrm{CE}(X) = v^{-1}\big(\mathsf{E}v(X)\big)$ represents the certainty equivalent (CE) of loss X, i.e., such a deterministic loss that a rational decision maker with deutility function v would be indifferent between $\mathrm{CE}(X)$ and stochastic loss profile X. The following argument can be used to construct risk measures of the form (2) that employ rational utility maximizer's preferences via certainty equivalents (Vinel and Krokhmal, 2015, see also Ben-Tal and Teboulle, 2007). Consider a decision maker who faces an uncertain future loss X, but who can allocate an amount η of resources now to cover the future loss. It will cost $v^{-1}\mathsf{E}v(X-\eta)_+$ to cover the remaining losses $(X-\eta)_+$, where $t_+ = \max\{0, t\}$ and an operator-like notation is used for v, i.e., $v^{-1}\mathsf{E}v(X-\eta)_+ = v^{-1}(\mathsf{E}v((X-\eta)_+))$. The total cost can then be optimized with an appropriate choice of η, such that the risk $\rho(X)$ of a future loss X reduces to

$$\rho(X) = \min_{\eta} \;\; \eta + \frac{1}{1-\alpha} v^{-1}\mathsf{E}v(X-\eta)_+, \quad \alpha \in (0,1), \tag{3}$$

where $(1-\alpha)^{-1} > 1$ is a penalty factor (a detailed discussion of representation (3) and related aspects is presented in Vinel and Krokhmal, 2014a).

Notably, expressing $\phi(X)$ in (2) via certainty equivalents necessarily requires that $(\cdot)_+$ appears in (3) in order for $\phi(X)$ to conform to the conditions of Theorem 1 (Vinel and Krokhmal, 2014a). The conditions on v that guarantee convexity of $\mathrm{CE}(X) = v^{-1}\mathsf{E}v(X)$, and, correspondingly, of $\phi(X)$, can be found, for example, in Ben-Tal and Teboulle (2007): v should be three times continuously differentiable, and $v'(t)/v''(t)$ be convex. In what follows, we implicitly assume that $\phi(X) = (1-\alpha)^{-1}v^{-1}\mathsf{E}v(X_+)$ is convex and satisfies the conditions of Theorem 1:

(U1) *Function $v(t)$ is continuously differentiable, increasing, convex, and, moreover, such that $v(0) = 0$ and the certainty equivalent $v^{-1}\mathsf{E}v(X)$ is convex in X.*

A key property of risk measures (3) is isotonicity with respect to second order stochastic dominance (SSD), provided that deutility function v is convex and nondecreasing:

(A5) *SSD isotonicity: $\rho(X) \leq \rho(Y)$ for all $X, Y \in \mathcal{X}$ such that $(-X) \succeq_{\mathrm{SSD}} (-Y)$.*

Recall that payoff profile Y_1 dominates Y_2 with respect to SSD, $Y_1 \succeq_{\mathrm{SSD}} Y_2$, if and only if $\mathsf{E}u(Y_1) \geq \mathsf{E}u(Y_2)$ holds for all non-decreasing concave utility functions u, or, in other words, if every rational risk-averse decision maker prefers Y_1 over Y_2. In this regard, (A5) implies that risk measures (3) "inherit" the risk preferences given by the utility u (equivalently, v). It is important to note that coherent and convex measures of risk are generally not SSD-isotonic (Krokhmal et al., 2011).

Another common property of risk measures (3) is that they are "tail" risk measures in the sense that the tail $\{X : X \geq \eta^*_\alpha(X)\}$ of the loss distribution is used to quantify risk, where the location of the "tail cutoff" point $\eta^*_\alpha(X)$, which is a minimizer in (3), can be adjusted according to risk preferences via the parameter α (see Krokhmal, 2007; Vinel and Krokhmal, 2014a).

Several practical and interesting risk measure families can be obtained from (3) by selecting a specific deutility function v. If $v(t) = t$, then (3) defines the well-known Conditional Value-at-Risk measure (Rockafellar and Uryasev, 2002, 2000):

$$\mathsf{CVaR}_\alpha(X) = \min_\eta \ \eta + (1-\alpha)^{-1}\mathsf{E}(X-\eta)_+, \quad \alpha \in (0,1). \tag{4}$$

If $v(t) = t^p$ for $t \geq 0$ and $p > 1$, then representation (3) yields a two-parametric family of *higher-moment coherent risk measures* (HMCR) (Krokhmal, 2007):

$$\mathsf{HMCR}_{p,\alpha}(X) = \min_\eta \ \eta + (1-\alpha)^{-1}\|(X-\eta)_+\|_p, \quad \alpha \in (0,1), \quad p \geq 1, \tag{5}$$

where $\|X\|_p = \big(\mathsf{E}|X|^p\big)^{1/p}$. If $v(t) = \lambda^t - 1$, $\lambda > 1$, then one obtains the family of log-exponential convex measures of risk (Vinel and Krokhmal, 2015):

$$\mathsf{LogExpCR}_{\lambda,\alpha}(X) = \min_\eta \ \eta + (1-\alpha)^{-1}\log_\lambda \mathsf{E}\lambda^{(X-\eta)_+}, \quad \alpha \in (0,1), \quad \lambda > 1. \tag{6}$$

Unlike the CVaR and HMCR measures that are coherent, the LogExpCR measure is *convex* but not coherent as it does not satisfy the positive homogeneity axiom (A3).

Perhaps one of the most widely used coherent measures of risk is defined by (4), which represents, roughly speaking, the conditional expectation of losses that may occur in the $(1-\alpha)\cdot 100\%$ of worst realizations of X. Clearly, CVaR measure is a special case of (5) when $p = 1$, $\mathsf{HMCR}_{1,\alpha}(X) = \mathsf{CVaR}_\alpha(X)$. When $p > 1$, HMCR measures quantify risk via higher tail moments $\|(X-\eta)_+\|_p$, and have been shown to be better suited for applications that involve heavy tailed loss distributions (Krokhmal, 2007). Likewise, the LogExpCR family (6) is designed for dealing with heavy-tailed distributions; moreover, in addition to being SSD-isotonic, LogExpCR measures are isotonic with respect to stochastic dominance of arbitrary order (kSD), see Vinel and Krokhmal (2015).

Next we discuss the implementation of the risk measures discussed above in mathematical programming problems.

2.1 Implementation in Stochastic Programming

Assume that loss X is a function of the decision variable $\mathbf{x}$, $X = X(\mathbf{x},\omega)$, where $\omega \in \Omega$. Then, for a compact and convex feasible set $C \subset \mathbb{R}^n$, consider a stochastic programming problem with a risk constraint in the form

$$\min\big\{g(\mathbf{x}) : \rho\big(X(\mathbf{x},\omega)\big) \leq h(\mathbf{x}),\ \mathbf{x} \in C\big\}. \tag{7}$$

Theorem 2 *Consider problem* (7) *where set $C \subset \mathbb{R}^n$ is compact and convex, and functions $g(\mathbf{x})$ and $h(\mathbf{x})$ are convex and concave on C, respectively. If, further, the cost or loss function $X(\mathbf{x},\omega)$ is convex in*

$\mathbf{x}$*, and ρ is a coherent or convex measure of risk with representation* (2)*, then problem* (7) *is equivalent to*

$$\min\big\{g(\mathbf{x}) : \eta + \phi\big(X(\mathbf{x},\omega) - \eta\big) \le h(\mathbf{x}),\ (\mathbf{x},\eta) \in C \times \mathbb{R}\big\}, \tag{8}$$

in the sense that (7) *and* (8) *achieve minima at the same values of the decision variable* $\mathbf{x}$ *and their optimal objective values coincide. Further, if the risk constraint in* (7) *is binding at optimality,* $(\mathbf{x}^*, \eta^*)$ *achieves the minimum of* (8) *if and only if* $\mathbf{x}^*$ *is an optimal solution of* (7) *and*

$$\eta^* \in \arg\min_\eta\ \eta + \phi\big(X(\mathbf{x}^*,\omega) - \eta\big).$$

Proof: See Krokhmal (2007). □

Remark 1 Note that the risk minimization problem

$$\min\big\{\rho\big(X(\mathbf{x},\omega)\big) : \mathbf{x} \in C\big\} \tag{9}$$

is obtained from (7) by introduction of a dummy variable x_{n+1} and letting $g(\mathbf{x}) = h(\mathbf{x}) = x_{n+1}$.

Let function ϕ in (8) have the form $\phi(X) = (1-\alpha)^{-1}v^{-1}\mathsf{E}v(X_+)$. Given a discrete set of scenarios $\{\omega_1,\ldots,\omega_N\} = \Omega$ that induce cost or loss outcomes $X(\mathbf{x},\omega_1),\ldots,X(\mathbf{x},\omega_N)$ for any given decision vector $\mathbf{x}$, it is easy to see that the risk constraint in (8) can be represented by the following set of inequalities:

$$\eta + (1-\alpha)^{-1}w_0 \le h(\mathbf{x}), \tag{10a}$$

$$w_0 \ge v^{-1}\Bigg(\sum_{j\in\mathcal{N}} \pi_j v(w_j)\Bigg), \tag{10b}$$

$$w_j \ge X(\mathbf{x},\omega_j) - \eta, \quad j \in \mathcal{N}, \tag{10c}$$

$$w_j \ge 0, \quad j \in \mathcal{N}, \tag{10d}$$

where $\mathcal{N}$ denotes the set of scenario indices, $\mathcal{N} = \{1,\ldots,N\}$, and $\pi_j = \mathsf{P}(\omega_j) > 0$ represent the corresponding scenario probabilities that satisfy $\pi_1 + \cdots + \pi_N = 1$.

In the above discussion it was shown that several types of risk measures emerge from different choices of the deutility function v. Here we note that the corresponding representations of constraint (10b) in the context of HMCR and LogExpCR measures lead to sufficiently "nice", i.e., convex, mathematical programming models. For HMCR measures inequality (10b) becomes

$$w_0 \ge \Big(\sum_{j\in\mathcal{N}} \pi_j w_j^p\Big)^{1/p}, \tag{11}$$

which is equivalent to a standard p-order cone under affine scaling. Noteworthy instances of (11) for which readily available mathematical programming solution methods exist include $p = 1, 2$. In the particular case of $p = 1$, which corresponds to CVaR, the problem reduces to a linear programming (LP) model. For instances when $p = 2$, a second-order cone programming (SOCP) model that is efficiently solvable using long-step self-dual interior point methods transpires. However, no similarly efficient solution methods exist for solving p-order conic constrained problems when $p \in (1,2)\cup(2,\infty)$ due to the fact that the p-cone is not self-dual in this case. Additional discussion and computational considerations for such instances are given in Section 4.1. Lastly, the following exponential inequality corresponds to constraint (10b) when ρ is a LogExpCR measure:

$$w_0 \geq \ln \sum\nolimits_{j\in\mathcal{N}} \pi_j e^{w_j}, \tag{12}$$

which is also convex and allows for the resulting optimization problem to be solved using appropriate (e.g., interior point) methods.

3 Scenario Decomposition Algorithm

Large-scale stochastic optimization models with CVaR measure (4) and the corresponding solution algorithms have received considerable attention in the literature. In this section we propose an efficient scenario decomposition algorithm for solving large-scale mathematical programming problems that use certainty equivalent-based risk measures (3), which contain CVaR as a special case.

The algorithm relies on solving a series of relaxation problems containing linear combinations of scenario-based constraints that are systematically decomposed until an optimal solution of the original problem is found or the problem is proven to be infeasible. Naturally, the core assumption behind such a scheme is that sequential solutions of smaller relaxation problems can be achieved within shorter computation times. By virtue of Section 2, when the distribution of loss function $X(\mathbf{x}, \omega)$ has a finite support (scenario set) $\Omega = \{\omega_1, \ldots, \omega_N\}$ with probabilities $\mathsf{P}(\omega_j) = \pi_j > 0$, the stochastic programming problem with risk constraint (8) admits the form

$$\begin{aligned}
\min \quad & g(\mathbf{x}) && \text{(13a)}\\
\text{s. t.} \quad & \mathbf{x} \in C, && \text{(13b)}\\
& \eta + (1-\alpha)^{-1} w_0 \leq h(\mathbf{x}), && \text{(13c)}\\
& w_0 \geq v^{-1}\Big(\sum_{j\in\mathcal{N}} \pi_j v(w_j)\Big), && \text{(13d)}\\
& w_j \geq X(\mathbf{x}, \omega_j) - \eta, \quad j \in \mathcal{N}, && \text{(13e)}\\
& w_j \geq 0, \quad j \in \mathcal{N}, && \text{(13f)}
\end{aligned}$$

where $\mathcal{N} = \{1, \ldots, N\}$. If we assume that function $g(\mathbf{x})$ and feasible set C are "nice" in the sense that problem $\min\{g(\mathbf{x}) : \mathbf{x} \in C\}$ admits efficient solution methods, then formulation (13) may present challenges that are two-fold. First, constraint (13d) may need a specialized solution approach, especially in the case of large N. Similarly, when N is large, computational difficulties may be associated with handling the large number of constraints (13e)–(13f). In this work we present an iterative procedure for dealing with a large number of scenario-based inequalities (13e)–(13f).

Since the original problem (13) with many constraints of the form (13e)–(13f) may be hard solve, a relaxation of (13) can be constructed by aggregating some of the scenario constraints. Let $\{\mathcal{S}_k : k \in \mathcal{K}\}$ denote a *partition* of the set $\mathcal{N}$ of scenario indices (which we will simply call scenario set), i.e.,

$$\bigcup_{k\in\mathcal{K}} \mathcal{S}_k = \mathcal{N}, \quad \mathcal{S}_i \cap \mathcal{S}_j = \emptyset \quad \text{for all} \quad i, j \in \mathcal{K},\ i \neq j.$$

The aggregation of scenario constraints by adding inequalities (13e) within sets $\mathcal{S}_k$ produces the following *master problem*:

$$
\begin{aligned}
\min \quad & g(\mathbf{x}) && \text{(14a)}\\
\text{s. t.} \quad & \mathbf{x} \in C, && \text{(14b)}\\
& \eta + (1-\alpha)^{-1} w_0 \le h(\mathbf{x}), && \text{(14c)}\\
& w_0 \ge v^{-1}\Big(\sum_{j\in\mathcal{N}} \pi_j v(w_j)\Big), && \text{(14d)}\\
& \sum_{j\in\mathcal{S}_k} w_j \ge \sum_{j\in\mathcal{S}_k} X(\mathbf{x},\omega_j) - |\mathcal{S}_k|\eta, \quad k\in\mathcal{K}, && \text{(14e)}\\
& w_j \ge 0, \quad j\in\mathcal{N}. && \text{(14f)}
\end{aligned}
$$

Clearly, any feasible solution of (13) is also feasible for (14), and the optimal value of (14) represents a lower bound for that of (13). Since the relaxed problem contains fewer scenario-based constraints (14e), it is potentially easier to solve. It would then be of interest to determine the conditions under which an optimal solution of (14) is also optimal for the original problem (13). Assuming that $\mathbf{x}^*$ is an optimal solution of (14), consider the problem

$$
\begin{aligned}
\min \quad & \eta + (1-\alpha)^{-1} w_0 && \text{(15a)}\\
\text{s. t.} \quad & w_0 \ge v^{-1}\Big(\sum_{j\in\mathcal{N}} \pi_j v(w_j)\Big), && \text{(15b)}\\
& w_j \ge X(\mathbf{x}^*,\omega_j) - \eta, \quad j\in\mathcal{N}, && \text{(15c)}\\
& w_j \ge 0, \quad j\in\mathcal{N}. && \text{(15d)}
\end{aligned}
$$

Proposition 1 *Consider problem* (13) *and its relaxation* (14) *obtained by aggregating scenario constraints* (13e) *over sets $\mathcal{S}_k$, $k\in\mathcal{K}$, that form a partition of $\mathcal{N} = \{1,\dots,N\}$. Assuming that* (13) *is feasible, consider problem* (15) *where $\mathbf{x}^*$ is an optimal solution of relaxation* (14)*. Let $(\eta^{**}, \mathbf{w}^{**})$ be an optimal solution of* (15)*. If the optimal value of* (15) *satisfies condition*

$$\eta^{**} + (1-\alpha)^{-1} w_0^{**} \le h(\mathbf{x}^*), \tag{16}$$

then $(\mathbf{x}^*, \eta^{**}, \mathbf{w}^{**})$ is an optimal solution of the original problem (13)*.*

Proof: Let $\mathbf{x}^\circ$ be an optimal solution of (13). Obviously, one has $g(\mathbf{x}^*) \le g(\mathbf{x}^\circ)$. The statement of the proposition then follows immediately by observing that inequality (16) guarantees the triple $(\mathbf{x}^*, \eta^{**}, \mathbf{w}^{**})$ to be feasible for problem (13). □

The statement of Proposition 1 allows one to solve the original problem (13) by constructing an appropriate partition of $\mathcal{N}$ and solving the corresponding master problem (14). Below we outline an iterative procedure that accomplishes this goal.

Step 0: The algorithm is initialized by including all scenarios in a single partition, $\mathcal{K} = \{0\}$, $\mathcal{S}_0 = \{1,\dots,N\}$.

Step 1: For a current partition $\{\mathcal{S}_k : k\in\mathcal{K}\}$, solve the master problem (14). If (14) is infeasible, then the original problem (13) is infeasible as well, and the algorithm terminates. Otherwise, let $\mathbf{x}^*$ be an optimal solution of the master (14).

Step 2: Given a solution $\mathbf{x}^*$ of the master, solve problem (15), and let $(\eta^{**}, \mathbf{w}^{**})$ denote the corresponding optimal solution. If condition (16) is satisfied, the algorithm terminates with $(\mathbf{x}^*, \eta^{**}, \mathbf{w}^{**})$ being an optimal solution of (13) due to Proposition 1. If, however, condition (16) is violated,

$$\eta^{**} + (1-\alpha)^{-1} w_0^{**} > h(\mathbf{x}^*),$$

then the algorithm proceeds to Step 3 to update the partition.

Step 3: Determine the set of scenario-based constraints in (15) that, for a given solution of the master $\mathbf{x}^*$, are binding at optimality:

$$\mathcal{J} = \{j \in \mathcal{N} : w_j^{**} = X(\mathbf{x}^*, \omega_j) - \eta^{**} > 0\} \tag{17}$$

Then, the elements of $\mathcal{J}$ are removed from the existing sets $\mathcal{S}_k$:

$$\mathcal{S}_k = \mathcal{S}_k \setminus \mathcal{J}, \quad k \in \mathcal{K},$$

and added to the partition as single-element sets:

$$\big\{\mathcal{S}_0, \ldots, \mathcal{S}_K\big\} \cup \big\{\mathcal{S}_{K+1}, \ldots, \mathcal{S}_{K+|\mathcal{J}|}\big\}, \text{ where } \mathcal{S}_{K+i} = \{j_i\} \text{ for each } j_i \in \mathcal{J},\ i = 1, \ldots, |\mathcal{J}|,$$

and the algorithm proceeds to Step 1.

Theorem 3 *Assume that in problem* (13) *functions* $g(\mathbf{x})$ *and* $X(\mathbf{x}, \omega)$ *are convex in* $\mathbf{x}$*,* $h(\mathbf{x})$ *is concave in* $\mathbf{x}$*,* v *satisfies assumption (U1), and the set* C *is convex and compact. Then, the described scenario decomposition algorithm either finds an optimal solution of problem* (13) *or declares its infeasibility after at most* N *iterations.*

Proof: Let us show that during an iteration of the algorithm the size of the partition of the set $\mathcal{N}$ of scenarios increases by at least one.

Let $\{\mathcal{S}_k : k \in \mathcal{K}\}$ be the current partition of $\mathcal{N}$, $(\mathbf{x}^*, \eta^*, \mathbf{w}^*)$ be the corresponding optimal solution of (14), and $(\eta^{**}, \mathbf{w}^{**})$ be an optimal solution of (15) for the given $\mathbf{x}^*$, such that the stopping condition (16) is not satisfied,

$$\eta^{**} + (1-\alpha)^{-1} w_0^{**} > h(\mathbf{x}^*). \tag{18}$$

Let $\bar{\mathcal{S}}^*$ denote the set of constraints (15c) that are binding at optimality,

$$\bar{\mathcal{S}}^* = \big\{j : w_j^{**} = X(\mathbf{x}^*, \omega_j) - \eta^{**} > 0,\ j \in \mathcal{N}\big\}.$$

Next, consider a problem obtained from (15) with a given $\mathbf{x}^*$ by aggregating the constraints (15c) that are non-binding at optimality:

$$\min \quad \eta + (1-\alpha)^{-1} w_0 \tag{19a}$$

$$\text{s. t.} \quad w_0 \geq v^{-1}\bigg(\sum_{j \in \mathcal{S}_0} \pi_j v(w_j)\bigg), \tag{19b}$$

$$w_j \geq X(\mathbf{x}^*, \omega_j) - \eta, \quad j \in \bar{\mathcal{S}}^*, \tag{19c}$$

$$\sum_{j \in \mathcal{S}^*} w_j \geq \sum_{j \in \mathcal{S}^*} X(\mathbf{x}^*, \omega_j) - |\mathcal{S}^*|\eta, \tag{19d}$$

$$w_j \geq 0, \quad j \in \mathcal{N}, \tag{19e}$$

where $\mathcal{S}^* = \mathcal{N} \setminus \bar{\mathcal{S}}^*$. Obviously, an optimal solution $(\eta^{**}, \mathbf{w}^{**})$ of (15) will also be optimal for (19).

Next, observe that at any stage of the algorithm, the partition $\{\mathcal{S}_k : k \in \mathcal{K}\}$ is such that there exists at most one set with $|\mathcal{S}_k| > 1$, namely set $\mathcal{S}_0$, and the rest of the sets in the partition satisfy $|\mathcal{S}_k| = 1$, $k \neq 0$. Let us denote

$$\bar{\mathcal{S}}_0 = \mathcal{N} \setminus \mathcal{S}_0 = \bigcup_{k \in \mathcal{K} \setminus \{0\}} \mathcal{S}_k.$$

Assume that $\bar{\mathcal{S}}^* \subseteq \bar{\mathcal{S}}_0$. By rewriting the master problem (14) as

$$\begin{aligned}
\min \quad & g(\mathbf{x}) && \text{(20a)}\\
\text{s. t.} \quad & \mathbf{x} \in C, && \text{(20b)}\\
& \eta + (1-\alpha)^{-1} w_0 \leq h(\mathbf{x}), && \text{(20c)}\\
& w_0 \geq v^{-1}\Big(\sum_{j \in \mathcal{N}} \pi_j v(w_j) \Big), && \text{(20d)}\\
& w_j \geq X(\mathbf{x}, \omega_j) - \eta, \quad j \in \bar{\mathcal{S}}_0, && \text{(20e)}\\
& \sum_{j \in \mathcal{S}_0} w_j \geq \sum_{j \in \mathcal{S}_0} X(\mathbf{x}, \omega_j) - |\mathcal{S}_0|\eta, && \text{(20f)}\\
& w_j \geq 0, \quad j \in \mathcal{N}, && \text{(20g)}
\end{aligned}$$

we observe that the components $\eta^*, \mathbf{w}^*$ of its optimal solution are feasible for (19). Indeed, from (20e) one has that

$$w_j^* \geq X(\mathbf{x}^*, \omega_j) - \eta^*, \quad j \in \bar{\mathcal{S}}^*,$$

which satisfies (19c), and also

$$w_j^* \geq X(\mathbf{x}^*, \omega_j) - \eta^*, \quad j \in \bar{\mathcal{S}}_0 \setminus \bar{\mathcal{S}}^* = \mathcal{S}^* \setminus \mathcal{S}_0.$$

Adding the last inequalities yields

$$\sum_{j \in \mathcal{S}^* \setminus \mathcal{S}_0} w_j^* \geq \sum_{j \in \mathcal{S}^* \setminus \mathcal{S}_0} X(\mathbf{x}^*, \omega_j) - |\mathcal{S}^* \setminus \mathcal{S}_0| \, \eta^*,$$

which can then be aggregated with (20f) to produce

$$\sum_{j \in \mathcal{S}^*} w_j^* \geq \sum_{j \in \mathcal{S}^*} X(\mathbf{x}^*, \omega_j) - |\mathcal{S}^*|\eta^*,$$

verifying the feasibility of $(\eta^*, \mathbf{w}^*)$ for (19). Since (20c) has to hold for $(\mathbf{x}^*, \eta^*, \mathbf{w}^*)$, we obtain that

$$\eta^{**} + (1-\alpha)^{-1} w^{**} \leq \eta^* + (1-\alpha)^{-1} w^* \leq h(\mathbf{x}^*),$$

which furnishes a contradiction with (18). Therefore, one has to have $\bar{\mathcal{S}}_0 \subset \bar{\mathcal{S}}^*$ for (18) to hold, meaning that at least one additional scenario from $\bar{\mathcal{S}}^*$ will be added to the partition during Step 3 of the algorithm. It is easy to see that the number of iterations cannot exceed the number N of scenarios. □

Remark 2 The fact that the proposed scenario decomposition method terminates within at most N iterations represents an important advantage over several existing cutting-plane methods that were developed in the literature for problems involving Conditional Value-at-Risk measure (Künzi-Bay and Mayer, 2006), integrated chance constraints (Klein Haneveld and van der Vlerk, 2006), and SSD constraints (Roman et al., 2006). In the mentioned works, the cutting-plane algorithms utilized supporting hyperplane representations for scenario constraints, which were themselves exponential in the size N of scenario sets. Although finite convergence of the cutting plane techniques was guaranteed by the polyhedral structure of the scenario constraints (in the case when $X(\mathbf{x}, \omega)$ is linear in $\mathbf{x}$), no estimate for the sufficient number of iterations was provided. A level-type regularization of cutting plane method for problems with SSD constraints, which allows for an estimate of the number of cuts due to Lemaréchal et al. (1995), is discussed in Fábián et al. (2011).

3.1 An Efficient Solution Method for Sub-Problem (15)

Although formulation (15) may be solved using appropriate mathematical programming techniques, an efficient alternative solution method can be employed by noting that (15) is equivalent to

$$\min\ \eta + \frac{1}{1-\alpha} v^{-1}\left(\sum_{j\in\mathcal{N}} \pi_j v\big(X(\mathbf{x}^*, \omega_j) - \eta\big)_+ \right), \tag{21}$$

which is a mathematical programming implementation of representation (3) under a finite scenario model where realizations $X(\mathbf{x}^*, \omega_j)$ represent scenario losses corresponding to an optimal decision $\mathbf{x}^*$ in the master problem (14). An optimal value of η in (15) and (21) can be computed directly using its properties dictated by representation (3).

Namely, let $X_j = X(\mathbf{x}^*, \omega_j)$ represent the optimal loss in scenario j for problem (14), and let $X_{(m)}$ be the m-th smallest outcome among $X_1, \ldots, X_N$, such that

$$X_{(1)} \le X_{(2)} \le \ldots \le X_{(N)}.$$

The following proposition enables evaluation of η^{**} as a "cutoff" point within the tail of the loss distribution.

Proposition 2 *Given a function $v(\cdot)$ that satisfies (U1) and an $\alpha \in (0,1)$, a sufficient condition for η^{**} to be an optimal solution in problems* (21) *and* (15) *has the form*

$$\frac{\sum_{j:X_j>\eta^{**}} \pi_j v'(X_j - \eta^{**})}{v'(v^{-1}(\sum_{j\in\mathcal{N}} \pi_j v(X - \eta^{**})_+))} + \alpha - 1 = 0, \tag{22}$$

where v' denotes the derivative of v.

Proof: The underlying assumption (U1) on v entails that $\phi(X) = (1-\alpha)^{-1} v^{-1} \mathsf{E} v(X)$ is convex, whence the objective function of (21)

$$\Phi_X(\eta) = \eta + \phi(X - \eta) = \eta + \frac{1}{1-\alpha} v^{-1}\Big(\sum_{j\in\mathcal{N}} \pi_j v(X_j - \eta)_+ \Big) \tag{23}$$

is convex on $\mathbb{R}$. Moreover, the condition $\phi(\eta) > \eta$ for $\eta \neq 0$ of Theorem 1 guarantees that the set of minimizers of $\Phi_X(\eta)$ is compact and convex in $\mathbb{R}$. Indeed, it is easy to see that $\Phi_X(\eta) = \eta$ for $\eta \geq X_{(N)}$ and $\Phi_X(\eta) \sim -\frac{\alpha\eta}{1-\alpha}$ for $\eta \ll -1$.

Now, consider the left derivative of $\Phi_X(\eta)$ at a given point $\eta = \eta^{**}$:

$$\begin{aligned}
&-(1-\alpha) + (1-\alpha)\frac{d^-}{d\eta}\Phi_X(\eta)\bigg|_{\eta=\eta^{**}} = \frac{d^-}{d\eta}\left\{v^{-1}\Big(\textstyle\sum_{j\in\mathcal{N}}\pi_j v(X_j-\eta)_+\Big)\right\}\bigg|_{\eta=\eta^{**}}\\
&= \lim_{\epsilon\to 0+}\frac{1}{-\epsilon}\left\{v^{-1}\bigg(\sum_{j:X_j\geq\eta^{**}}\pi_j v(X_j-\eta^{**}+\epsilon)\bigg) - v^{-1}\bigg(\sum_{j:X_j\geq\eta^{**}}\pi_j v(X_j-\eta^{**})\bigg)\right\}\\
&= \frac{d^-}{d\eta}\left\{v^{-1}\bigg(\sum_{j:X_j\geq\eta^{**}}\pi_j v(X_j-\eta)\bigg)\right\}\bigg|_{\eta=\eta^{**}} = \frac{d}{d\eta}\left\{v^{-1}\bigg(\sum_{j:X_j\geq\eta^{**}}\pi_j v(X_j-\eta)\bigg)\right\}\bigg|_{\eta=\eta^{**}},
\end{aligned}$$

where the last equality follows from the continuous differentiability of function $v^{-1}\big(\sum_{j:X_j\geq\eta^{**}}\pi_j v(X_j-\eta^{**})\big)$ at the point η^{**} due to the assumed properties of v. Analogously, the right derivative of $\Phi_X(\eta)$ at $\eta = \eta^{**}$ equals to

$$\frac{d^+}{d\eta}\Phi_X(\eta)\bigg|_{\eta=\eta^{**}} = 1 + \frac{1}{1-\alpha}\frac{d}{d\eta}\left\{v^{-1}\bigg(\sum_{j:X_j>\eta^{**}}\pi_j v(X_j-\eta)\bigg)\right\}\bigg|_{\eta=\eta^{**}},$$

where the strict inequality in summation is due to fact that $v(X_j-\eta^{**}-\epsilon)_+ = 0$ for all $\epsilon > 0$ if $\eta^{**} \leq X_j$.

Observe that $\Phi_X(\eta)$ may only be non-differentiable at points $\eta = X_j$. Indeed, for any $\eta^{**} \neq X_j$, $j \in \mathcal{N}$, the obtained expressions for left and right derivatives become equivalent, and equation (22) is obtained from the first order optimality conditions by computing the derivatives of the functions in braces and noting that $\sum_{j:X_j\geq\eta^{**}}\pi_j v(X_j-\eta^{**}) = \sum_{j:X_j>\eta^{**}}\pi_j v(X_j-\eta^{**}) = \sum_{j\in\mathcal{N}}\pi_j v(X_j-\eta^{**})_+$. □

Recall that the presented above scenario decomposition algorithm uses the subproblem (15) for determining an optimal value of η^{**}, as well as for identifying (during Step 3) the set $\mathcal{J}$ of scenarios that are binding at optimality, i.e., for which $X(\mathbf{x}^*, \omega_j) - \eta^{**} > 0$. This can be accomplished with the help of the derived optimality condition (22) as follows.

Step (i) Compute values $X_j = X(\mathbf{x}^*, \omega_j)$, where $\mathbf{x}^*$ is an optimal solution of (14), and sort them in ascending order: $X_{(1)} \leq \ldots \leq X_{(N)}$.

Step (ii) For $m = N, N-1, \ldots, 1$, compute values T_m as

$$\begin{aligned}
T_N &= 1-\alpha,\\
T_m &= 1-\alpha - \frac{\sum_{j=m+1}^{N}\pi_j v'\big(X_{(j)}-X_{(m)}\big)}{v'\Big(v^{-1}\Big(\sum_{j=m+1}^{N}\pi_j v\big(X_{(j)}-X_{(m)}\big)\Big)\Big)}, \quad m = N-1,\ldots,1,
\end{aligned} \tag{24}$$

until m^* is found such that

$$T_{m^*} \leq 0,\ T_{m^*+1} > 0. \tag{25}$$

Step (iii) If $T_{m^*} = 0$, then the solution η^{**} of (15), (21) is equal to $X_{(m^*)}$. Otherwise, η^{**} satisfies

$$\eta^{**} \in \big(X_{(m^*)}, X_{(m^*+1)}\big],$$

and its value can be found by using an appropriate numerical procedure, such as Newton's method. The set $\mathcal{J}$ in (17) is then obtained as

$$\mathcal{J} = \{j : X_j = X_{(k)},\ k = m^* + 1, \ldots, N\}.$$

Proposition 3 *Given an optimal solution* $\mathbf{x}^*$ *of the master problem* (14)*, the algorithm described in steps (i)–(iii) yields an optimal value* η^{**} *in* (15)*,* (21) *and the set* $\mathcal{J}$ *to be used during steps 2 and 3 of the scenario decomposition algorithm.*

Proof: First, observe that an optimal solution η^{**} of (15) and (21) satisfies $\eta^{**} \leq X_{(N)}$. Indeed, assume to the contrary that $\eta^{**} = X_{(N)} + \epsilon$ for some $\epsilon > 0$. The optimal value of (15) and (3) is then equal to $X_{(N)} + \epsilon$, and can be improved by selecting, e.g., $\epsilon = \epsilon/2$.

Next, observe that quantities T_m are equal, up to a factor $1-\alpha$, to the right derivatives of function $\Phi_X(\eta)$ (23) at $\eta = X_{(m)}$, i.e., $T_m = (1-\alpha)\frac{d^+}{d\eta}\Phi_X(\eta)\big|_{\eta=X_{(m)}}$. The value of $T_N = 1 - \alpha$ follows directly from the fact that $\Phi_X(\eta) = \eta$ for $\eta \geq X_{(N)}$. Then, if strict inequalities in (25) hold, two cases are possible. Namely, an optimal η^{**} is located inside the interval $\big(X_{(m^*)}, X_{(m^*+1)}\big)$ if $\frac{d^-}{d\eta}\Phi_X(X_{(m^*+1)}) > 0$. Alternatively, $\eta^{**} = X_{(m^*+1)}$ if $\frac{d^-}{d\eta}\Phi_X(X_{(m^*+1)}) \leq 0$. Thus, we have the second statement of step (iii).

If $T_{m^*} = 0$ in (25), observe that necessarily $\frac{d^-}{d\eta}\Phi_X(X_{m^*}) \leq 0$ since the left derivative of Φ_X at $X_{(m)}$ differs from the expression (24) by an extra summand $\pi_m v'(0)$ in the numerator. If $v'(0) = 0$ then $\frac{d^-}{d\eta}\Phi_X(X_{m^*}) = \frac{d^+}{d\eta}\Phi_X(X_{m^*}) = 0$ and $\eta^{**} = X_{(m^*)}$ is a minimum due to Proposition 2. If $v'(0) > 0$ then $\frac{d^-}{d\eta}\Phi_X(X_{m^*}) < 0$ and $\eta^{**} = X_{(m^*)}$ is again either a unique minimizer, or represents the left endpoint of the set of minimizers. This validates the first claim of step (iii).

Once the value of η^{**} is obtained during step (iii), the set $\mathcal{J}$ in (17) is constructed as the set of scenario indices corresponding to $X_{(m^*+1)}, X_{(m^*+2)}, \ldots, X_{(N)}$.

Note that it is not necessary to prove that there always exists $m^* \in \{1, \ldots, N-1\}$ such that $T_{m^*} \leq 0$ and $T_{m^*+1} > 0$. If indeed it were to happen that $T_m > 0$ for all $m = 1, \ldots, N$, this would imply that set $\mathcal{J}$ must contain all scenarios, i.e., $\mathcal{J} = \mathcal{N}$, making the exact value of η^{**} irrelevant in this case, since the original problem (13) would have to be solved at the next iteration of the scenario decomposition algorithm. □

Remark 3 We conclude this section by noting that the presented scenario decomposition approach is applicable, with appropriate modifications, to more general forms of downside risk measures $\rho(X) = \min_\eta\{\eta + \phi((X-\eta)_+)\}$. The focus of our discussion on the case when function ϕ has the form of a certainty equivalent, $\phi(X) = v^{-1}\mathsf{E}v(X_+)$, is dictated mainly by the fact that the resulting constraint (13d) encompasses a number of interesting and practically relevant special cases, such as second-order cone, p-order cone, and log-exponential constraints.

4 Computational Experiments: Portfolio Optimization with HMCR and LogExpCR Measures

Portfolio optimization problems are commonly used as an experimental platform in risk management and stochastic optimization. In this section we illustrate the computational performance of the proposed scenario decomposition algorithm on a portfolio optimization problem, where the investment risk is quantified using HMCR or LogExpCR measures.

A standard formulation of portfolio optimization problem entails determining the vector of portfolio weights $\mathbf{x} = (x_1, \dots, x_n)^\top$ of n assets so as to minimize the risk while maintaining a prescribed level of expected return. We adopt the traditional definition of portfolio losses X as negative portfolio returns, $X(\mathbf{x}, \omega) = -\mathbf{r}(\omega)^\top \mathbf{x}$, where $\mathbf{r}(\omega) = (r_1(\omega), \dots, r_n(\omega))^\top$ are random returns of the assets. Then, the portfolio selection model takes the general form

$$\begin{aligned} \min \quad & \rho\big(-\mathbf{r}(\omega)^\top \mathbf{x}\big) && \text{(26a)} \\ \text{s. t.} \quad & \mathbf{1}^\top \mathbf{x} = 1, && \text{(26b)} \\ & \mathsf{E}\big[\mathbf{r}(\omega)^\top \mathbf{x}\big] \geq \bar{r}, && \text{(26c)} \\ & \mathbf{x} \geq \mathbf{0}, && \text{(26d)} \end{aligned}$$

where $\mathbf{1} = (1, \dots, 1)^\top$, equality (26b) represents the budget constraint, (26b) ensures a minimum expected portfolio return level, $\bar{r}$, and (26d) corresponds to no-short-selling constraints.

The distribution of the random vector $\mathbf{r}(\omega)$ of assets' returns is given by a finite set of N equiprobable scenarios $\mathbf{r}_j = \mathbf{r}(\omega_j) = (r_{1j}, \dots, r_{nj})^\top$,

$$\pi_j = \mathsf{P}\big\{\mathbf{r} = (r_{1j}, \dots, r_{nj})^\top\big\} = 1/N, \quad j \in \mathcal{N} \equiv \{1, \dots, N\}. \tag{27}$$

4.1 Portfolio Optimization with Higher Moment Coherent Risk Measures

In the case when risk measure ρ in (26) is selected as a higher moment coherent risk measure, $\rho(X) = \mathsf{HMCR}_{p,\alpha}(X)$, the portfolio optimization problem (26) can be written in a stochastic programming form that is consistent with the general formulation (13) as

$$\begin{aligned} \min \quad & \eta + (1-\alpha)^{-1} w_0 && \text{(28a)} \\ \text{s. t.} \quad & w_0 \geq \|(w_1, \dots, w_N)\|_p, && \text{(28b)} \\ & \pi_j^{-1/p} w_j \geq -\mathbf{r}_j^\top \mathbf{x} - \eta, \quad j \in \mathcal{N}, && \text{(28c)} \\ & \mathbf{x} \in C, \quad \mathbf{w} \geq \mathbf{0}, && \text{(28d)} \end{aligned}$$

where C represents a polyhedral set comprising the expected return, budget, and no-short-selling constraints on the vector of portfolio weights $\mathbf{x}$:

$$C = \Big\{\mathbf{x} \in \mathbb{R}^n \ : \ \textstyle\sum_{j \in \mathcal{N}} \pi_j \mathbf{r}_j^\top \mathbf{x} \geq \bar{r}, \ \ \mathbf{1}^\top \mathbf{x} = 1, \ \ \mathbf{x} \geq \mathbf{0}\Big\}. \tag{29}$$

Due to the presence of p-order cone constraint (28b), formulation (28) constitutes a p-order cone programming problem (pOCP).

Solution methods for problem (28) are dictated by the specific value of parameter p in (28b). As has been mentioned, in the case of $p = 1$ formulation (28) reduces to a LP problem that corresponds to a choice of risk measure as the CVaR, a case that has received a considerable attention in the literature. In view of this, of particular interest are nonlinear instances of problem (28), which correspond to values of the parameter $p \in (1, +\infty)$.

Below we consider instances of (28) with $p = 2$ and $p = 3$. In the case of $p = 2$, problem (28) can be solved using SOCP self-dual interior point methods. In the case of $p = 3$ and, generally, $p \in (1, 2) \cup (2, \infty)$, the p-cone (28b) is not self-dual, and we employ two techniques for solving (28) and the corresponding master problem (14): (i) a SOCP-based approach that relies on the fact that for a rational p, a p-order cone can be equivalently represented via a sequence of second order cones, and (ii) an LP-based approach that allows for obtaining exact solutions of pOCP problems via cutting-plane methods.

Detailed discussions of the respective formulations of problems (28) are provided below. Throughout this section, we use abbreviations in brackets to denote the different formulations of the "complete" versions of (28) (i.e., with complete set of scenario constraints (28c)). For each "complete" formulation, we also consider the corresponding scenario decomposition approach, indicated by suffix "SD". Within the scenario decomposition approach, we present formulations of the master problem (denoted by subscript "MP"); the respective subproblems are then constructed accordingly. For example, the SOCP version of the complete problem (28) with $p = 2$ is denoted [SOCP], while the same problem solved by scenario decomposition is referred to as [SOCP-SD], with the master problem being denoted as [SOCP-SD]$_{\mathrm{MP}}$ (see below).

4.1.1 SOCP Formulation in $p = 2$ Case.

In case when $p = 2$, formulation (28) constitutes a standard SOCP problem that can be solved using a number of available SOCP solvers, such as CPLEX, MOSEK, GUROBI, etc. In order to solve it using the scenario decomposition algorithm presented in Section 3, the master problem (14) is formulated with respect to the original problem (28) with $p = 2$ as follows:

$$\begin{aligned} \min \quad & \eta + (1-\alpha)^{-1} w_0 \\ \text{s. t.} \quad & w_0 \geq \|(w_1, \ldots, w_N)\|_2, \\ & \sum_{j \in \mathcal{S}_k} \frac{\pi_j^{1/2}}{\pi^{(k)}} w_j \geq \left(\sum_{j \in \mathcal{S}_k} \frac{\pi_j}{\pi^{(k)}} \mathbf{r}_j^\top \right) \mathbf{x} - \eta, \quad k \in \mathcal{K}, \\ & \mathbf{w} \geq \mathbf{0}, \ \ \mathbf{x} \in C. \end{aligned} \qquad \text{[SOCP-SD]}_{\mathrm{MP}}$$

Note that in the case of $\mathsf{HMCR}_{2,\alpha}$ measure, the function $v(t) = t^2$ is positive homogeneous of degree two, which allows for eliminating the scenario probabilities π_j from constraint (14d) and representing the latter in the form of a second order cone in the full formulation (28) and in the master problem [SOCP-SD]$_{\mathrm{MP}}$. This affects constraints (14d), which then can be written in the form of the second constraint in [SOCP-SD]$_{\mathrm{MP}}$. The subproblem (15) is reformulated accordingly.

4.1.2 SOCP Reformulation of p-Order Cone Program.

One of the possible approaches for solving the pOCP problem (28) with $p = 3$ involves reformulating the p-cone constraint (28b) via a set of quadratic cone constraints. Such an exact reformulation is possible when the parameter p has a rational value, $p = q/s$. Then, a (q/s)-order cone constraint in the positive orthant $\mathbb{R}_+^{N+1}$

$$\left\{ \mathbf{w} \geq \mathbf{0} \ : \ w_0 \geq (w_1^{q/s} + \ldots + w_N^{q/s})^{s/q} \right\} \qquad (30)$$

may equivalently be represented as the following set in $\mathbb{R}_+^{N+1} \times \mathbb{R}_+^N$:

$$\big\{\mathbf{w}, \mathbf{u} \geq \mathbf{0} \ : \ w_0 \geq \|\mathbf{u}\|_1, \ \ w_j^q \leq u_j^s w_0^{q-s}, \ \ j \in \mathcal{N}\big\}. \tag{31}$$

Each of the N nonlinear inequalities in (31) can in turn be represented as a sequence of three-dimensional rotated second-order cones of the form $\xi_0^2 \leq \xi_1 \xi_2$, resulting in a SOCP reformulation of the rational-order cone (30) (Nesterov and Nemirovski, 1994; Alizadeh and Goldfarb, 2003; Krokhmal and Soberanis, 2010). Such a representation, however, is not unique and in general may comprise a varying number of rotated second order cones for a given $p = q/s$. In this case study we use the technique of Morenko et al. (2013), which allows for representing rational order p-cones with $p = q/s$ in ρ^{N+1} via $N\lceil \log_2 q \rceil$ second order cones. Namely, in the case of $p = 3$, when $q = 3$, $s = 1$, the 3-order cone (30) can equivalently be replaced with $\lceil \log_2 3 \rceil N = 2N$ quadratic cones

$$\big\{\mathbf{w}, \mathbf{u}, \mathbf{v} \geq \mathbf{0} \ : \ w_0 \geq \|\mathbf{u}\|_1, \ \ w_j^2 \leq w_0 v_j, \ \ v_j^2 \leq w_j u_j, \ \ j \in \mathcal{N}\big\}. \tag{32}$$

In accordance with the above, a p-order cone inequality in $\mathbb{R}^{N+1}$ can be represented by a set of 3D second order cone constraints and a linear inequality when p is a positive rational number. Thus, the [SpOCP] problem (28) takes the following form:

$$\begin{aligned} \min \quad & \eta + (1-\alpha)^{-1} w_0 \\ \text{s. t.} \quad & w_0 \geq \|\mathbf{u}\|_1, \\ & w_j^2 \leq w_0 v_j, \ v_j^2 \leq w_j u_j, \quad j \in \mathcal{N}, \\ & \pi_j^{-1/p} w_j \geq -\mathbf{r}_j^\top \mathbf{x} - \eta, \quad j \in \mathcal{N}, \\ & \mathbf{x} \in C, \ \ \mathbf{w}, \mathbf{v}, \mathbf{u} \geq \mathbf{0}. \end{aligned} \tag{SpOCP}$$

The corresponding master problem sub-problem [SpOCP-SD]$_{\text{MP}}$ in the scenario decomposition-based method is constructed by replacing constraints of the form (28c) in the last problem as follows:

$$\begin{aligned} \min \quad & \eta + (1-\alpha)^{-1} w_0 \\ \text{s. t.} \quad & w_0 \geq \|\mathbf{u}\|_1, \\ & w_j^2 \leq w_0 v_j, \ v_j^2 \leq w_j u_j, \quad j \in \mathcal{N}, \\ & \sum_{j \in \mathcal{S}_k} \frac{\pi_j^{1-1/p}}{\pi^{(k)}} w_j \geq \left(\sum_{j \in \mathcal{S}_k} \frac{\pi_j}{\pi^{(k)}} \mathbf{r}_j^\top \right) \mathbf{x} - \eta, \quad k \in \mathcal{K}, \\ & \mathbf{x} \in C, \ \ \mathbf{w}, \mathbf{v}, \mathbf{u} \geq \mathbf{0}. \end{aligned} \tag{[SpOCP-SD]$_{\text{MP}}$}$$

4.1.3 An Exact Solution Method for pOCP Programs Based on Polyhedral Approximations.

Computational methods for solving p-order cone programming problems that are based on polyhedral approximations (Krokhmal and Soberanis, 2010; Vinel and Krokhmal, 2014b) represent an alternative to interior-point approaches, and can be beneficial in situations when a pOCP problem needs to be solved repeatedly, with small variations in problem data or problem structure.

Thus, in addition to the SOCP-based approaches for solving the pOCP problem (28) discussed above, we also employ an exact polyhedral-based approach with $O(\varepsilon^{-1})$ iteration complexity that was proposed in

Vinel and Krokhmal (2014b). It consists in reformulating the p-order cone $w_0 \geq \|(w_1, \dots, w_N)\|_p$ via a set of three-dimensional p-cones

$$w_0 = w_{2N-1}, \quad w_{N+j} \geq \|(w_{2j-1}, w_{2j})\|_p, \quad j = 1, \dots, N-1, \tag{33}$$

and then iteratively building outer polyhedral approximations of the 3D p-cones until the solution of desired accuracy $\varepsilon > 0$ is obtained,

$$\|(w_1, \dots, w_N)\|_p \leq (1 + \varepsilon) w_0.$$

In the context of the lifted representation (33), the above ε-relaxation of p-cone inequality translates into $N - 1$ corresponding approximation inequalities for 3D p-cones:

$$\|(w^*_{2j-1}, w^*_{2j})\|_p \leq (1 + \epsilon) w^*_{N+j}, \quad j = 1, \dots, N-1, \tag{34}$$

where $\epsilon = (1 + \varepsilon)^{1/\lceil \log_2 N \rceil} - 1$. Then, for a given $\varepsilon > 0$, an ε-approximate solution of pOCP portfolio optimization problem (28) is obtained by iteratively solving the linear programming problem

$$\begin{aligned}
\min \quad & \eta + (1-\alpha)^{-1} w_0 \\
\text{s. t.} \quad & w_0 = w_{2N-1}, \\
& w_{N+j} \geq \alpha_p(\theta_{k_j}) w_{2j-1} + \beta_p(\theta_{k_j}) w_{2j}, \quad \theta_{k_j} \in \Theta_j, \quad j = 1, \dots, N-1, \\
& \pi_j^{-1/p} w_j \geq -\mathbf{r}_j^\top \mathbf{x} - \eta, \quad j \in \mathcal{N}, \\
& \mathbf{x} \in C, \ \mathbf{w} \geq \mathbf{0},
\end{aligned} \tag{LpOCP}$$

where coefficients α_p and β_p are defined as

$$\alpha_p(\theta) = \frac{\cos^{p-1}\theta}{(\cos^p \theta + \sin^p \theta)^{1-\frac{1}{p}}}, \quad \beta_p(\theta) = \frac{\sin^{p-1}\theta}{(\cos^p \theta + \sin^p \theta)^{1-\frac{1}{p}}}.$$

If, for a given solution $\mathbf{w}^* = (w^*_0, \dots, w^*_{2N-1})$ of [LpOCP], the approximation condition (34) is not satisfied for some $j = 1, \dots, N-1$,

$$\|(w^*_{2j-1}, w^*_{2j})\|_p > (1 + \epsilon) w^*_{N+j}, \tag{35}$$

then a cut of the form

$$w_{N+j} \geq \alpha_p(\theta^*_j) w_{2j-1} + \beta_p(\theta^*_j) w_{2j}, \quad \theta^*_j = \arctan \frac{w^*_{2j}}{w^*_{2j-1}}, \tag{36}$$

is added to [LpOCP]. The process is initialized with $\Theta_j = \{\theta_1\}$, $\theta_1 = \pi/4$, $j = 1, \dots, N-1$, and continues until no violations of condition (35) are found. In Vinel and Krokhmal (2014b) it was shown that this cutting-plane procedure generates an ε-approximate solution to pOCP problem (28) within $O(\varepsilon^{-1})$ iterations.

The described cutting plane scheme can be employed to solve the master problem corresponding to the pOCP problem (28). Namely, the cutting-plane formulation of this master problem is obtained by replacing the p-cone constraint (28b) with cutting planes similarly to [LpOCP], and the set of N scenario constraints (28c) with the aggregated constraints (compare to [SpOCP-SD]$_{\mathrm{MP}}$):

$$\begin{aligned}
\min\quad & \eta + (1-\alpha)^{-1} t \\
\text{s. t.}\quad & w_0 = w_{2N-1}, \\
& w_{N+j} \geq \alpha_p(\theta_{k_j}) w_{2j-1} + \beta_p(\theta_{k_j}) w_{2j}, \quad \theta_{k_j} \in \Theta_j, \quad j = 1, \ldots, N-1, \\
& \sum_{j \in \mathcal{S}_k} \frac{\pi_j^{1-1/p}}{\pi^{(k)}} w_j \geq \left(\sum_{j \in \mathcal{S}_k} \frac{\pi_j}{\pi^{(k)}} \mathbf{r}_j^\top \right) \mathbf{x} - \eta, \quad k \in \mathcal{K}, \\
& \mathbf{x} \in C, \ \mathbf{w} \geq \mathbf{0}.
\end{aligned} \qquad \text{[LpOCP-SD]}_{\mathrm{LB}}$$

4.2 Portfolio Optimization with Log Exponential Convex Risk Measures

In order to demonstrate the applicability of the proposed method when solving problems with measures of risk other than the HMCR class, we examine an analogous experimental framework for instances when $\rho(X) = \mathsf{LogExpCR}_{e,\alpha}(X)$. The portfolio optimization problem (26) may then be written as

$$\begin{aligned}
\min\quad & \eta + (1-\alpha)^{-1} w_0 \\
\text{s. t.}\quad & w_0 \geq \ln \sum_{j \in \mathcal{N}} \pi_j e^{w_j}, \\
& w_j \geq -\mathbf{r}_j^\top \mathbf{x} - \eta, \quad j \in \mathcal{N}, \\
& \mathbf{x} \in C, \ \mathbf{w} \geq \mathbf{0}.
\end{aligned} \qquad \text{[LogExpCP]}$$

Note that in contrast to pOCP and SOCP problems discussed in the preceding subsections, the above formulation is not a conic program. Since it involves a convex log-exponential constraint, we call this problem a log-exponential convex programming problem (LogExpCP) that can be solved with interior point methods.

The corresponding master problem for the scenario decomposition algorithm is obtained from [LogExpCP] by aggregating the scenario constraints in accordance to (14):

$$\begin{aligned}
\min\quad & \eta + (1-\alpha)^{-1} w_0 \\
\text{s. t.}\quad & w_0 \geq \ln \sum_{j \in \mathcal{N}} \pi_j e^{w_j}, \\
& \sum_{j \in \mathcal{S}_k} w_j \geq - \sum_{j \in \mathcal{S}_k} \mathbf{r}_j^\top \mathbf{x} - |\mathcal{S}_k| \eta, \quad k \in \mathcal{K}, \\
& \mathbf{x} \in C, \ \mathbf{w} \geq \mathbf{0}.
\end{aligned} \qquad \text{[LogExpCP-SD]}_{\mathrm{MP}}$$

In the next section we examine the computational performances within each implementation class of problem (28).

4.3 Computational Results

The portfolio optimization problems described in Section 4.1 and 4.2 were implemented in C++ using callable libraries of three solvers, CPLEX 12.5, GUROBI 5.02, and MOSEK 6. Computations ran on a six-core 2.30GHz PC with 128GB RAM in 64-bit Windows environment. In the context of benchmarking, each adopted formulation was tested against its scenario decomposition-based implementation. Moreover, it was of particular interest to examine the performance of the scenario decomposition algorithm using various risk measure configurations, thus, the following problem settings were solved: problems [SOCP]-[SOCP-SD] with risk measure as defined by (5) for $p = 2$; problems [SpOCP]-[SpOCP-SD] and [LpOCP]-[LpOCP-SD] with measure (5) for $p = 3$; and problems [LogExpCP]-[LogExpCP-SD] with risk measure (6). The value of parameter α in the employed risk measures was fixed at $\alpha = 0.9$ throughout.

The scenario data in our numerical experiments was generated as follows. First, a set of n stocks ($n = 50$, 100, 200) was selected at random from the S&P500 index. Then, a covariance matrix of daily returns as well as the expected returns were estimated for the specific set of n stocks using historical prices from January 1, 2006 to January 1, 2012. Finally, the desired number N of scenarios, ranging from 1,000 to 100,000, have been generated as N independent and identically distributed samples from a multivariate normal distribution with the obtained mean and covariance matrix.

On account of precision arithmetic errors associated with the numerical solvers, we introduced a tolerance level $\epsilon > 0$ to specify the permissible gap in the stopping criterion (16):

$$\eta^{**} + (1-\alpha)^{-1} w_0^{**} \leq h(\mathbf{x}^*) + \epsilon. \tag{37}$$

Specifically, the value $\epsilon = 10^{-5}$ was was chosen to match the reduced cost of the simplex method in CPLEX and GUROBI. In a similar manner, we adjust (24) around m^* for precision errors as

$$T_{m^*+1}(p) - \epsilon < 0 \quad \text{and} \quad T_{m^*}(p) + \epsilon > 0.$$

Empirical observations suggest the accumulation of numerical errors is exacerbated by the use of fractional values of scenarios in assets returns, r_{ij}. To alleviate the numerical accuracy issues, the data in respective problem instances of the scenario decomposition algorithm were appropriately scaled.

The results of our numerical experiments are summarized in Tables 1 – 5. Unless stated otherwise, the reported running time values are averaged over 20 instances. Table 1 presents the computational times observed during solving the full formulation, [SOCP], of problem (28) with HMCR measure and $p = 2$, and solving the same problem using the scenario decomposition algorithm, [SOCP-SD], with the three solvers, CPLEX, GUROBI, and MOSEK. Observe that the scenario decomposition method performs better for all instances and solvers, with the exception of the largest three scenario instances when using GUROBI with $n = 50$ assets. However, this trend is tampered as the number of assets increases.

Table 2 reports the running times observed during solving of the second-order cone reformulation of the pOCP version of problem (28) with $p = 3$, in the full formulation ([SpOCP]) and via the scenario decomposition algorithm ([SpOCP-SD]). The obtained results indicate that, although the scenario decomposition algorithm is slower on smaller problem instances, it outperforms direct solution methods as the numbers of scenarios N and assets n in the problem increase. Due to observed numerical instabilities, the CPLEX solver was not considered for this particular experiment.

Next, the same problem is solved using using the polyhedral approximation cutting-plane method described in Section 4.1. Table 3 shows the running times achieved by all three solvers for problems [LpOCP] and [LpOCP-SD] with $p = 3$. In this case, the scenario decomposition method resulted in order-of-magnitude improvements, which can be attributed to the "warm-start" capabilities of CPLEX and GUROBI's simplex solvers. Consistent with these conclusions is also the fact that the simplex-based solvers of CPLEX and GUROBI yield improved solution times on the full problem formulation

		CPLEX		GUROBI		MOSEK	
n	N	[SOCP]	[SOCP-SD]	[SOCP]	[SOCP-SD]	[SOCP]	[SOCP-SD]
50	1000	1.00	0.46	0.62	0.45	0.26	0.15
	2500	3.03	0.51	1.88	1.07	0.60	0.36
	5000	6.58	0.55	3.81	2.78	1.24	0.72
	10000	13.72	1.35	9.56	7.89	2.56	1.61
	25000	31.03	3.53	32.40	34.04	7.33	5.18
	50000	60.62	9.05	101.09	117.24	17.64	12.43
	100000	137.14	25.25	327.95	449.78	36.78	33.02
100	1000	2.46	0.86	1.73	0.42	0.61	0.18
	2500	6.14	0.99	4.87	1.17	1.50	0.47
	5000	13.69	1.10	11.13	3.55	3.25	1.15
	10000	27.06	2.21	21.94	9.63	6.69	3.03
	25000	72.95	8.85	71.34	37.48	20.41	6.88
	50000	157.25	20.88	185.56	129.37	44.01	16.61
	100000	319.90	58.29	464.12	467.35	79.75	41.58
200	1000	6.87	2.19	5.60	0.58	6.68	0.29
	2500	17.48	2.10	15.36	1.37	4.49	0.73
	5000	34.93	2.98	33.96	4.15	9.36	1.92
	10000	76.13	5.03	63.67	16.50	19.54	5.51
	25000	206.29	24.16	196.45	54.00	53.89	29.15
	50000	447.85	55.93	438.40	152.76	112.47	28.85
	100000	950.17	112.60	998.86	539.46	234.68	61.98

Table 1: Average computation times (in seconds) obtained by solving problems [SOCP] and [SOCP-SD] for $p = 2$ using CPLEX, GUROBI and MOSEK. All running times are averaged over 20 instances.

comparing to the SOCP-based reformulation [SpOCP], where barrier solvers were invoked. The discrepancy between [LpOCP] and [LpOCP-SD] solution times is especially prominent for MOSEK, but in this case it appears that MOSEK's interior-point LP solver was much less effective at solving the [LpOCP] formulation using the cutting plane method.

Finally, Table 4 displays the running times for the discussed implementation of problems [LogExpCR] and [LogExpCP-SD]. Of the three solvers considered in this case study, only MOSEK was capable of handling problems with constraints that involve sums of univariate exponential functions. Again, the scenario decomposition-based solution method appears to be preferable in comparison to solving the full formulation. Note, however, that computational times were not averaged over 20 instances in this case due to numerical difficulties associated with the solver for many instances of [LogExpCP].

It is also of interest to comment on the number of scenarios that had to be generated during the scenario decomposition procedure in order to yield an optimal solution. Table 5 lists the corresponding average number of scenarios partitioned for each problem type over all instances. Although these numbers may slightly differ among the three solvers, we only present results for MOSEK as it was the only solver used to solve all the problem in Sections 4.1 and 4.2. Observe that far fewer scenarios are required relative to the total set size N. In fact, as a percentage of the total number of scenarios, the number of scenarios that were generated during the algorithm in order to achieve optimality was between 0.7% and 11% of the total scenario set size.

		GUROBI		MOSEK	
n	N	[SpOCP]	[SpCOP-SD]	[SpOCP]	[SpCOP-SD]
50	1000	2.58	2.73	0.18	0.63
	2500	10.63	6.61	0.49	0.96
	5000	32.01	19.27	1.06	1.70
	10000	87.27	41.34	2.31	3.49
	25000	198.56	92.39	7.14	6.70
	50000	455.63	540.09	16.36	13.70
	100000	1217.96	2080.34	35.33	30.29
100	1000	7.16	3.14	0.30	0.75
	2500	29.47	8.44	0.85	1.37
	5000	90.25	19.74	1.88	2.32
	10000	277.72	44.31	4.52	3.91
	25000	642.63	92.11	12.66	8.66
	50000	1365.37	1716.37	28.64	15.10
	100000	—	—	65.48	28.29
200	1000	17.86	3.87	0.69	1.01
	2500	78.28	8.65	1.90	1.56
	5000	276.89	22.40	4.41	2.47
	10000	799.65	49.02	9.88	4.84
	25000	2118.11	107.14	29.99	9.60
	50000	—	—	64.52	17.41
	100000	—	—	139.87	34.99

Table 2: Average computation times (in seconds) obtained by solving problems [SpCOP] and [SpCOP-SD] for $p = 3$ using GUROBI and MOSEK. All running times are averaged over 20 instances and symbol "—" indicates that the time limit of 3600 seconds was exceeded.

5 Conclusions

In this work, we propose an efficient algorithm for solving large-scale convex stochastic programming problems that involve a class of risk functionals in the form of infimal convolutions of certainty equivalents. We exploit the property induced by such risk functionals that a significant portion of scenarios is not required to obtain an optimal solution. The developed scenario decomposition technique is contingent on the identification and separation of "non-redundant" scenarios by solving a series of smaller relaxation problems. It is shown that the number of iterations of the algorithm is bounded by the number of scenarios in the problem. Numerical experiments with portfolio optimization problems based on simulated return data following the covariance structure of randomly chosen S&P500 stocks demonstrate that significant reductions in solution times may be achieved by employing the proposed algorithm. Particularly, performance improvements were observed for the large-scale instances when using HMCR measures with $p = 2, 3$, and LogExpCR measures.

Acknowledgements This work was supported in part by the AFOSR grant FA9550-12-1-0142 and the U.S. Department of Air Force grant FA8651-12-2-0010. In addition, support by the AFRL Mathematical Modeling and Optimization Institute is gratefully acknowledged.

		CPLEX		GUROBI		MOSEK	
n	N	[LpOCP]	[LpOCP-SD]	[LpOCP]	[LpOCP-SD]	[LpOCP]	[LpOCP-SD]
50	1000	0.27	0.12	0.22	0.59	0.82	0.46
	2500	1.65	0.24	0.74	0.83	4.26	0.66
	5000	6.81	0.46	2.31	1.54	15.08	1.46
	10000	19.20	1.42	7.73	3.86	60.66	3.75
	25000	31.93	3.93	56.52	13.74	381.67	11.34
	50000	179.49	16.07	117.72	36.51	1412.81	25.47
	100000	903.36	62.79	474.68	112.72	—	54.45
100	1000	0.37	0.13	0.23	0.61	2.94	0.65
	2500	2.22	0.28	0.86	0.98	7.11	1.06
	5000	8.58	0.79	2.82	1.76	32.20	1.95
	10000	28.71	2.18	9.28	4.13	122.75	4.99
	25000	45.37	4.99	35.11	13.13	1138.99	15.34
	50000	200.12	18.80	122.21	39.78	2753.54	34.17
	100000	3336.26	82.79	1316.29	138.74	—	80.15
200	1000	0.61	0.20	0.33	0.89	15.68	1.06
	2500	3.13	0.44	1.30	1.17	20.64	1.37
	5000	13.25	1.01	3.72	2.11	70.49	2.97
	10000	47.97	3.31	13.20	4.72	322.36	8.12
	25000	195.28	6.98	94.45	14.77	2418.52	26.91
	50000	936.60	27.20	665.61	45.43	—	53.62
	100000	—	114.08	3301.44	160.92	—	123.89

Table 3: Average computation times (in seconds) obtained by solving problems [LpOCP] and [LpOCP-SD] for $p = 3$ using CPLEX, GUROBI and MOSEK. All running times are averaged over 20 instances and symbol "—" indicates that the time limit of 3600 seconds was exceeded.

References

Alizadeh, F. and Goldfarb, D. (2003) "Second-order cone programming," *Mathematical Programming*, **95** (1), 3–51.

Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. (1999) "Coherent Measures of Risk," *Mathematical Finance*, **9** (3), 203–228.

Ben-Tal, A. and Teboulle, M. (2007) "An Old-New Concept of Convex Risk Measures: An Optimized Certainty Equivalent," *Mathematical Finance*, **17** (3), 449–476.

Delbaen, F. (2002) "Coherent risk measures on general probability spaces," in: K. Sandmann and P. J. Schönbucher (Eds.) "Advances in Finance and Stochastics: Essays in Honour of Dieter Sondermann," 1–37, Springer.

Espinoza, D. and Moreno, E. (2012) "Fast sample average approximation for minimizing Conditional-Value-at-Risk," *Preprint Paper*.

Fábián, C. I., Mitra, G., Roman, D., and Zverovich, V. (2011) "An enhanced model for portfolio choice with SSD criteria: a constructive approach," *Quantitative Finance*, **11**, 1525–1534.

Klein Haneveld, W. K. and van der Vlerk, M. H. (2006) "Integrated chance constraints: reduced forms and an algorithm," *Computational Management Science*, **3**, 245–269.

		MOSEK		Instances Solved
n	N	[LogExpCP]	[LogExpCP-SD]	
50	1000	0.61	0.27	12
	2500	0.97	0.58	14
	5000	1.89	1.18	12
	10000	4.88	2.57	9
	25000	14.99	7.94	12
	50000	26.65	18.76	15
	100000	65.45	61.48	17
100	1000	0.57	0.25	17
	2500	1.65	0.53	16
	5000	3.69	1.14	10
	10000	9.18	2.53	15
	25000	24.61	13.83	13
	50000	50.66	39.72	19
	100000	148.54	59.02	16
200	1000	5.25	0.37	19
	2500	4.22	0.75	17
	5000	9.53	1.39	18
	10000	21.17	2.63	17
	25000	62.03	7.59	17
	50000	145.89	16.47	18
	100000	333.73	43.56	19

Table 4: Average computation times (in seconds) obtained by solving a specified number of instances for problems [LogExpCP] and [LogExpCP-SD] using MOSEK solver.

Krokhmal, P. (2007) “Higher Moment Coherent Risk Measures,” *Quantitative Finance*, **7** (4), 373–387.

Krokhmal, P. and Soberanis, P. (2010) “Risk optimization with p-order conic constraints: A linear programming approach,” *European Journal of Operational Research*, **301** (3), 653–671.

Krokhmal, P., Zabarankin, M., and Uryasev, S. (2011) “Modeling and optimization of risk,” *Surveys in Operations Research and Management Science*, **16** (2), 49–66.

Künzi-Bay, A. and Mayer, J. (2006) “Computational aspects of minimizing conditional value-at-risk,” *Computational Management Science*, **3** (1), 3–27.

Lemaréchal, C., Nemirovskii, A., and Nesterov, Y. (1995) “New variants of bundle methods,” *Mathematical Programming*, **69**, 111–147.

Lim, C., Sherali, H. D., and Uryasev, S. (2010) “Portfolio optimization by minimizing conditional value-at-risk via nondifferentiable optimization,” *Computational Optimization and Applications*, **46** (3), 391–415.

Morenko, Y., Vinel, A., Yu, Z., and Krokhmal, P. (2013) “On p-cone linear discrimination,” *European Journal of Operational Research*, **231**, 784–789.

Nesterov, Y. E. and Nemirovski, A. (1994) *Interior Point Polynomial Algorithms in Convex Programming*, volume 13 of *Studies in Applied Mathematics*, SIAM, Philadelphia, PA.

		MOSEK			
n	N	[SOCP-SD]	[SpOCP-SD]	[LpOCP-SD]	[LogExpCP-SD]
50	1000	80.3	24.8	21.3	61.8
	2500	180.8	47.8	47.0	77.8
	5000	349.3	80.3	79.0	104.6
	10000	711.6	133.4	128.3	154.3
	25000	1834.9	232.0	318.3	178.2
	50000	3582.1	445.4	675.0	841.7
	100000	6945.1	774.1	1346.4	1447.5
100	1000	87.2	32.0	27.0	81.4
	2500	191.2	73.6	74.1	107.8
	5000	367.6	107.4	102.4	192.2
	10000	711.1	148.9	156.9	229.7
	25000	1808.6	278.1	348.6	1869.1
	50000	3802.9	457.8	729.7	2418.6
	100000	7323.3	831.3	1395.8	923.4
200	1000	108.2	39.5	36.4	100.7
	2500	201.7	72.7	73.0	154.5
	5000	395.6	116.3	119.6	198.1
	10000	744.0	184.9	171.2	304.6
	25000	1805.5	308.3	347.0	464.2
	50000	3607.8	512.2	697.6	788.1
	100000	7198.9	865.0	1384.3	1153.5

Table 5: Average number of partitioned scenarios from solving the scenario decomposition-based problems listed in Section 4.1 and 4.2.

Rockafellar, R. T. and Uryasev, S. (2000) “Optimization of Conditional Value-at-Risk,” *Journal of Risk*, **2**, 21–41.

Rockafellar, R. T. and Uryasev, S. (2002) “Conditional Value-at-Risk for General Loss Distributions,” *Journal of Banking and Finance*, **26** (7), 1443–1471.

Roman, D., Darby-Dowman, K., and Mitra, G. (2006) “Portfolio construction based on stochastic dominance and target return distributions,” *Mathematical Programming*, **108**, 541–569.

Ruszczyński, A. and Shapiro, A. (2006) “Optimization of Convex Risk Functions,” *Mathematics of Operations Research*, **31** (3), 433–452.

Schied, A., A.ed and Follmer, H. (2002) “Robust preferences and convex measures of risk,” in: K. Sandmann and P. J. Schönbucher (Eds.) “Advances in Finance and Stochastics: Essays in Honour of Dieter Sondermann,” 39–56, Springer.

Shapiro, A., Dentcheva, D., and Ruszczyński (2009) *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, Philadelphia.

Subramanian, D. and Huang, P. (2008) “A Novel Algorithm for Stochastic Linear Programs with Conditional-value-at-risk (CVaR) Constraints,” *IBM Research Report, RC24752*.

Subramanian, D. and Huang, P. (2009) “An Efficient Decomposition Algorithm for Static, Stochastic, Linear and Mixed-Integer Linear Programs with Conditional-Value-at-Risk Constraints,” *IBM Research Report, RC24752*.

Uryasev, S. and Rockafellar, R. T. (2013) “The fundamental risk quadrangle in risk management, optimization and statistical estimation,” *Surveys in Operations Research and Management Science*, **18**, 33–53.

Vinel, A. and Krokhmal, P. (2014a) “On valid inequalities for mixed integer p-order cone programming,” *Journal of Optimization Theory and Applications*, **160** (2), 439–456.

Vinel, A. and Krokhmal, P. (2014b) “Polyhedral approximations in p-order cone programming,” *Optimization Methods & Software*, **29**, 1210–1237.

Vinel, A. and Krokhmal, P. (2015) “Certainty equivalent measures of risk,” *Annals of Operations Research*, DOI:10.1007/s10479–015–1801–0.

von Neumann, J. and Morgenstern, O. (1944) *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, 1953rd edition.

Noname manuscript No.
(will be inserted by the editor)

A stochastic PDE-constrained optimization approach to vibration control of an electrically conductive composite plate subjected to mechanical and electromagnetic loads

D. Chernikov · P. Krokhmal · O. I. Zhupanska · C. L. Pasiliao

Received: date / Accepted: date

Abstract A new two-stage stochastic partial differential equation (PDE)-constrained optimization methodology is developed for the active vibration control of structures in the presence of uncertainties in mechanical loads. The methodology relies on the two-stage stochastic optimization formulation with an embedded first-order black-box PDE-constrained optimization procedure. The PDE-constrained optimization procedure utilizes a first-order active-set algorithm with a conjugate gradient method. The objective function is determined through solution of the governing PDEs and its gradient is computed using automatic differentiation with hyper-dual numbers. The developed optimization methodology is applied to the problem of post-impact vibration control (via applied electromagnetic field) of an electrically conductive carbon fiber reinforced composite plate subjected to an uncertain, or stochastic, impact load. The corresponding governing PDEs consist of a nonlinear coupled system of equations of motion and Maxwell's equations. The conducted computational study shows that the obtained two-stage optimization solution allows for a significant suppression of vibrations caused by the randomized impact load in all impact load scenarios. Also, the effectiveness of the developed methodology is illustrated in the case of a deterministic impact load, where the two-stage strategy enables one to practically eliminate post-impact vibrations.

Keywords PDE-constrained optimization · two-stage stochastic optimization · electro-magneto-mechanical coupling · composite materials

1 Introduction

In electrically conductive solids, mechanical and electromagnetic fields interact through the Lorentz ponderomotive force that is exerted by the electromagnetic field. Analysis of this field interaction requires simultaneous solution of Maxwell's equations for electromagnetic field (Maugin, 1988) and equations of motion of continuous media that involve the Lorentz force as a body force, whereby the system of governing equations becomes coupled and nonlinear. This field coupling leads to many interesting effects observed in the mechanical behavior of the electrically conductive solids subjected to electromagnetic load, including changes in the stress state (Moon, 1984; Zhupanska and Sierakowski, 2007, 2011; Higuchi et al, 2007), vibration behavior (Barakati and Zhupanska, 2012a; Rudnicki, 2002), and unusual stability behavior (Hasanyan and Piliposyan, 2001; Hasanyan et al, 2006; Eringen, 1989). Electro-magneto-mechanical coupling can potentially lead to the development of structures amenable to active control by the electromagnetic field. Interactions between mechanical, electromagnetic, and thermal fields provide a basis for the multifunctional materials and structures.

Composite materials are often considered to be materials of choice for multifunctional applications (Gibson, 2010) due to their multiphase nature and inherent tailorability.

Dmitry Chernikov
Department of Mechanical and Industrial Engineering
University of Iowa, Iowa City, IA 52241
E-mail: dmitry-chernikov@uiowa.edu

Pavlo Krokhmal
Department of Mechanical and Industrial Engineering
University of Iowa, Iowa City, IA 52241
Tel.: +1-319-335-5680
Fax: +1-319-335-5669
E-mail: krokhmal@engineering.uiowa.edu

Olesya I. Zhupanska
Department of Mechanical and Industrial Engineering
University of Iowa, Iowa City, IA 52241
E-mail: ozhupans@engineering.uiowa.edu

Crystal L. Pasiliao
Air Force Research Lab, Eglin AFB, FL 32542
E-mail: crystal.pasiliao@eglin.af.mil

As a result, the recent years witnessed a growing interest in electro-magneto-mechanical interactions in composites. Most of the studies have been focused on the mechanics, while less attention was paid to the optimization of multifunctional composites and structures. The present work makes contribution to the latter subject.

The present work is closely related to the recent studies on the electro-magneto-mechanical coupling in electrically conductive anisotropic composites (Zhupanska and Sierakowski, 2007, 2011; Barakati and Zhupanska, 2012a,b, 2013, 2014), where the effects of the steady, slowly varying, and pulsed electromagnetic fields on the mechanical response of single-layer and laminated anisotropic composite plates were examined. The interacting effects of the applied electric current, external magnetic field, and mechanical load were studied. It has been shown that the characteristics of the electromagnetic field (waveform, duration of application, intensity) can significantly reduce the stressed and deformed states of the electrically conductive plate and decrease the amplitude of vibrations. In particular, to achieve the maximum reduction in the plate deflection and stress, the application of the mechanical load must be coordinated with application of the electric current and its waveform. Moreover, an increase in the magnetic induction tends to reduce the amplitude of vibrations of the plate with a trend towards a more rapid decay at the stronger magnetic fields. An increase in the electric current density tends to decrease the amplitude of the plate vibrations. Furthermore, the effect of the electric current density becomes more pronounced as the magnetic field intensity increases. It has been concluded that concurrent application of a pulsed electromagnetic load could effectively mitigate the effects of the impact load and post-impact vibrations.

1.1 Active vibration control of a composite plate via an electromagnetic field: A conceptual application

The results of the previously discussed studies provided motivation for the present work on a stochastic partial differential equation (PDE)-constrained optimization approach to active control of the mechanical response of the electrically conductive composites, using an electromagnetic field. As a specific application, we consider the problem of vibration control – via application of an electromagnetic field – in an electrically conductive carbon fiber reinforced polymer (CFRP) composite plate subjected to a mechanical impact load with uncertain parameters (magnitude, duration, etc). We hypothesize that electromagnetically activated CFRP structural elements could provide additional protection against certain types of foreign object impacts, assuming that an appropriate sensor technology can be employed for applying an electromagnetic field to the composite structure at the moment of impact so as to increase the impact resistance and dampen post-impact vibrations.

The practical viability of this hypothetical scenario depends on a number of factors, among which are the availability of (i) composite materials with necessary mechanical and electromagnetic properties, (ii) adequate sensors to trigger application of an electromagnetic field, and (iii) ability to adjust and control characteristics of the applied electromagnetic field (i.e., waveform, duration of application, intensity) depending on the target composite material characteristics and applied impact load. Physics-based models of electro-magneto-mechanical coupling in electrically conductive composites can provide theoretical underpinnings for the development of the electromagnetically activated impact resistant structural elements, while PDE-constrained stochastic optimization can provide a path to the active control of these structural elements in the presence of uncertainties.

In Section 2 we outline the physical model of the field coupling phenomenon that is exploited in this work. Since the general model is prohibitively complex, a high-fidelity approximation of the governing equations in the case of thin composite plates is discussed. In Section 2.2 we introduce the actual boundary-value problem corresponding to the impact of a thin CFRP composite plate in a deterministic setting, i.e. when the impact load is known with certainty. This problem forms the basis for the stochastic PDE-constrained optimization problem that is introduced in Section 3. Numerical solution and optimization procedures are discussed in Section 4, and in Section 5 we present the results of computational studies.

2 Mechanics of electro-magneto-mechanical interactions in electrically conductive anisotropic composite plates

In this section we first outline the governing equations for anisotropic electrically conductive solids subjected to mechanical and electromagnetic loads. Then, we discuss a 2D plate approximation, as well as the resulting 2D nonlinear hyperbolic-parabolic system of PDEs that constitute the mathematical framework for solving problems of the dynamic mechanical response of the anisotropic electrically conductive plates subjected to mechanical and electromagnetic loads. See Zhupanska and Sierakowski (2007); Barakati and Zhupanska (2012a) for details.

2.1 Governing equations

The behavior and interaction of the mechanical and electromagnetic fields in electrically conductive solids can be determined from simultaneously solving the equations of motion that include the Lorentz ponderomotive force and Maxwell's equations:

$$\boldsymbol{\nabla}\cdot\mathbf{T}+\rho(\mathbf{F}+\mathbf{F}^L)=\rho\frac{\partial^2\mathbf{u}}{\partial t^2}, \tag{1}$$

$$\begin{aligned}&\operatorname{div}\mathbf{D}=\rho_e, \qquad \operatorname{div}\mathbf{B}=0,\\ &\operatorname{rot}\mathbf{E}=-\frac{\partial\mathbf{B}}{\partial t}, \quad \operatorname{rot}\mathbf{H}=\mathbf{j}+\frac{\partial\mathbf{D}}{\partial t}.\end{aligned} \tag{2}$$

Here $\mathbf{T}$ is the stress tensor, $\mathbf{u}$ is the displacement vector, ρ is density, $\mathbf{F}$ is the body force per unit mass, $\mathbf{F}^L$ is the Lorentz force per unit mass, and ∇ is the gradient operator. In Maxwell's equations (2), $\mathbf{D}$ represents the electric displacement vector, $\mathbf{B}$ is the magnetic induction, $\mathbf{E}$ is the electric field, $\mathbf{H}$ is the magnetic field, $\mathbf{j}$ is the current density vector, ρ_e is the electric charge density (which vanishes in electric conductors), and t is time.

Interaction between mechanical and electromagnetic fields in the electrically conductive materials is due to the Lorentz force, $\mathbf{F}^L$, that enters equations of motion (1) as a body force. It has been shown in Zhupanska and Sierakowski (2007) that the Lorentz force in the electrically conductive anisotropic solids takes the form:

$$\begin{aligned}\rho\mathbf{F}^L=\rho_e\left(\mathbf{E}+\frac{\partial\mathbf{u}}{\partial t}\times\mathbf{B}\right)+\left(\boldsymbol{\sigma}\left(\mathbf{E}+\frac{\partial\mathbf{u}}{\partial t}\times\mathbf{B}\right)\right)\times\mathbf{B}\\ +\Big(\big((\boldsymbol{\epsilon}-\varepsilon_0\mathbf{I})\mathbf{E}\big)\times\mathbf{B}\Big)_\alpha\nabla\left(\frac{\partial\mathbf{u}}{\partial t}\right)_\alpha+\mathbf{J}^*\times\mathbf{B},\end{aligned} \tag{3}$$

where $\boldsymbol{\sigma}$ is the electrical conductivity tensor, $\boldsymbol{\epsilon}$ is the electrical permittivity tensor, ε_0 is the electrical permittivity in the vacuum, ∇ is the gradient operator, and Einstein's summation convention is adopted with respect to index α. Therefore, the system of equations (1)–(2) is the system of nonlinear hyperbolic PDEs that represent the governing equations of electro-magneto-mechanical coupling in electrically conductive solids. The nonlinearity is due to the presence of the Lorentz force, which contains nonlinear terms with respect to the components of the mechanical and electromagnetic fields. In the most general dynamic case, the problem of solving the system of governing equations (1)–(2) for solids of even the simplest 3D geometries is insurmountable. In many situations, however, solution of equations (1)–(2) can be facilitated through appropriate *physics-based hypotheses*, or simplifications that allow one to reduce mathematical complexity of the model while preserving its physical fidelity by exploiting particular features of problem's geometry, etc.

With respect to the present work, a 2D approximation for the thin electrically conductive plates subjected to mechanical and electromagnetic loads is used. This approximation was developed in Zhupanska and Sierakowski (2007) and utilizes Kirchhoff hypothesis of non-deformable normals and electromagnetic hypotheses.

Next, we briefly outline the procedure to derive 2D approximation of the governing equations. More details can be found in Zhupanska and Sierakowski (2007); Barakati and Zhupanska (2012a). As for the mechanical part of the governing equations (1), the linear plate theory formulation based on the so-called Kirchhoff hypothesis of non-deformable normals is used. Equations of motion with respect to stress and moment resultants are obtained by integration of (1) across the thickness of the plate. In contrast to the problems with purely mechanical load, application of the Kirchhoff hypothesis and integration of the 3D equations of motion through the thickness of the plate does not produce 2D equations of motion. This is due to the presence of the terms with the Lorentz force components, which remain three-dimensional. Therefore, to obtain a 2D approximation to the equations of motion, one needs to derive a 2D approximation for the electromagnetic field and the Lorentz force for the case of thin plates. This is achieved by introducing additional hypotheses regarding the behavior of the electromagnetic field components, which imply that tangential components of the electric field vector and the normal component of the magnetic field vector do not change across the thickness of the plate and the variation of the tangential components of the magnetic field across the thickness of the plate is linear. A 2D approximation of Maxwell's equations (2) is obtained by representing functions $\mathbf{H}$, $\mathbf{E}$, and $\mathbf{J}$ via series expansions with respect to the coordinate z, integrating Maxwell's 3D equations across the thickness of the plate and invoking a quasistatic approximation for Maxwell's equations. The 2D expression for the Lorentz force is obtained (3) using the Kirchhoff hypothesis for the plate displacements and the set of the discussed electromagnetic hypotheses. The 2D equations of motion are then obtained by integrating the terms with the Lorentz force across the thickness of the plate in the equations of motion with respect to the stress and moment resultants.

Finally, 2D equations of motion and 2D Maxwell's equations constitute the system of governing equations for a mechanically and electrically conductive plate subjected to mechanical and electromagnetic loads and correspond to the linear plate theory. This system of equations is a nonlinear mixed system of parabolic and hyperbolic PDEs.

2.2 Impact problem: A deterministic formulation

In this section we present the boundary-value problem for a thin anisotropic composite plate subject to a deterministic mechanical impact load and electromagnetic field within the mathematical framework presented in the previous subsection. Such a deterministic formulation was considered in Barakati and Zhupanska (2012a) and forms the basis for the stochastic model with uncertain impact loads and the corresponding stochastic optimization formulations will be introduced in Section 3.

Consider a thin unidirectional fiber-reinforced (x-direction is the fiber direction) electrically conductive composite plate of width a and thickness h subjected to the transverse impact load:

$$\mathbf{p}(x,y,t) = \left[0, 0, p_z(x,y,t)\right], \tag{4}$$

time-dependent electric current of density:

$$\mathbf{J}(t) = \left[J_x(t), 0, 0\right], \tag{5}$$

and immersed in the constant magnetic field with the induction:

$$\mathbf{B}^* = \left[0, B_y^*, 0\right]. \tag{6}$$

It is assumed that the intensity of the current is such that the associated thermal effects are negligible.

The plate is transversely isotropic, where y–z is the plane of isotropy and the x–y plane coincides with the middle plane of the plate. The plate is assumed to be long in the fiber direction, which is also the direction of the applied current (x-direction), simply supported along the long sides, arbitrarily supported along the short sides (see Figure 1), and initially is at rest.

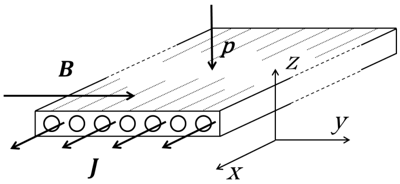


Fig. 1 Composite plate subjected to impact and electromagnetic loads

The corresponding mechanical and electromagnetic boundary conditions are:

$$\tau_{zz}\big|_{z=\frac{h}{2}} = -p_z(y,t), \tag{7a}$$

$$u_y\big|_{y=\pm\frac{a}{2}} = u_z\big|_{y=\pm\frac{a}{2}} = M_{yy}\big|_{y=\pm\frac{a}{2}} = 0, \tag{7b}$$

$$\left(E_x - \frac{\partial w}{\partial t}B_y^* + \frac{\partial v}{\partial t}B_z\right)\Big|_{y=-\frac{a}{2}} = 0, \quad E_x\Big|_{y=\frac{a}{2}} = 0. \tag{7c}$$

The applied transverse impact load (4) causes vibrations in the plate, which can potentially be mitigated by application of the external electromagnetic field consisting of the electric current of density (5) and magnetic induction (6). We are interested in the optimal characteristics of the electromagnetic field to maximally reduce mechanical vibrations caused by the impact load.

The formulated problem (4)-(7) for a long transversely isotropic plate admits the assumption of independence of the components of mechanical and electromagnetic fields of the coordinate x, which using the procedure described in Section 2 reduces the governing equations (1) and (2) to the form:

$$\begin{aligned}
\frac{1}{h}\frac{\partial N_{yy}}{\partial y} &= \rho\frac{\partial^2 v}{\partial t^2} + \sigma_x B_z^2\frac{\partial v}{\partial t} - \sigma_x B_y^* B_z\frac{\partial w}{\partial t} + \frac{\varepsilon_x - \varepsilon_0}{B_{22}}E_x B_z \\
&\quad \times \frac{\partial N_{yy}}{\partial t} - (\varepsilon_x - \varepsilon_0)E_x B_y^*\frac{\partial W}{\partial t} + B_z J_x(t) + \sigma_x E_x B_z, \\
\frac{1}{h}\frac{\partial N_{yz}}{\partial y} &= \rho\frac{\partial^2 w}{\partial t^2} + \frac{p_z(y,t)}{h} - \sigma_x B_y^* B_z\frac{\partial v}{\partial t} + \sigma_x\left(B_y^*\right)^2\frac{\partial w}{\partial t} \\
&\quad - (\varepsilon_x - \varepsilon_0)E_x B_z\frac{\partial W}{\partial t} - B_y^* J_x(t) - \sigma_x E_x B_y^*, \\
\frac{\partial M_{yy}}{\partial y} &= -\frac{\rho h^3}{12}\frac{\partial^2 W}{\partial t^2} + N_{yz} - \frac{1}{12}\sigma_x h^3 B_z^2\frac{\partial W}{\partial t} \\
&\quad + \frac{\varepsilon_x - \varepsilon_0}{B_{22}}E_x B_z\frac{\partial M_{yy}}{\partial t}, \\
\frac{\partial v}{\partial y} &= \frac{1}{hB_{22}}N_{yy}, \quad \frac{\partial^2 w}{\partial y^2} = -\frac{12}{h^3 B_{22}}M_{yy}, \quad \frac{\partial w}{\partial y} = W, \\
\frac{\partial B_z}{\partial y} &= \sigma_x\mu\left(E_x + \frac{\partial v}{\partial t}B_z - \frac{\partial w}{\partial t}B_y^*\right), \quad \frac{\partial E_x}{\partial y} = \frac{\partial B_z}{\partial t}.
\end{aligned} \tag{8}$$

Here v and w are the middle plane displacement components in y- and z-directions, respectively; $N_{yy} = \int_{-h/2}^{h/2}\tau_{yy}dz$ and $N_{yz} = \int_{-h/2}^{h/2}\tau_{yz}dz$ are the stress resultants; $M_{yy} = \int_{-h/2}^{h/2}\tau_{yy}zdz$ is the moment resultant; E_x is the x-component of the electric field; B_z is the z-component of the magnetic induction; σ_x and ε_x are the electrical conductivity and permittivity in x-direction, respectively; μ is the magnetic permeability; and $B_{22} = E_2/(1 - \nu_{12}\nu_{21})$, where E_2 is Young's modulus along the y-direction, ν_{12}, and ν_{21} are the corresponding Poisson ratios.

The system of the nonlinear PDEs (8) represents the governing equations in the context of this work. The formulated deterministic boundary-value problem (7)–(8) for low-velocity impact of a thin composite plate in the presence of an electromagnetic field forms the basis for the stochastic PDE-constrained optimization model of optimal vibration mitigation that is presented in Section 3.

3 A two-stage stochastic PDE-constrained optimization framework

In this section we first introduce a deterministic PDE-constrained optimization problem for vibration reduction in composite plates using an electromagnetic field, which is followed by the more general two-stage stochastic PDE-constrained optimization framework for control of composite structures in the presence of uncertainties in mechanical loads.

3.1 A PDE-constrained optimization formulation

The existence of field coupling effects between mechanical and electromagnetic fields in electrically conductive solids presents an opportunity for controlling and/or optimizing the mechanical response of the corresponding structures via application of an electromagnetic field. Assuming that the design or performance criterion of the structural element can be expressed through some function F to be minimized, the problem of optimization or control of the mechanical field via electromagnetic field generally reduces to a (nonlinear) PDE-constrained optimization problem of the form:

$$\min_{\boldsymbol{\theta}} \quad F_{[0,T]}(\mathbf{g},\boldsymbol{\theta},\boldsymbol{\xi}) \tag{9a}$$

$$\text{s. t.} \quad \frac{\partial \mathbf{g}}{\partial y} = \boldsymbol{\Phi}\left(\mathbf{g}, \frac{\partial \mathbf{g}}{\partial t}, \frac{\partial^2 \mathbf{g}}{\partial t^2}, \boldsymbol{\theta}, \boldsymbol{\xi}\right), \quad t \in [0,T], \tag{9b}$$

$$\mathbf{G}\left(\mathbf{g}, \frac{\partial \mathbf{g}}{\partial t}\right)\bigg|_{y=\pm\frac{a}{2}} = \mathbf{0}, \quad t \in [0,T], \tag{9c}$$

$$\underline{\boldsymbol{\theta}} \leq \boldsymbol{\theta} \leq \overline{\boldsymbol{\theta}}, \tag{9d}$$

where vector $\mathbf{g}$ represents the components of the mechanical field, i.e., displacements and stress and moment resultants in the governing equations (8), vector $\boldsymbol{\theta}$ contains the parameters of the electromagnetic field, vector $\boldsymbol{\xi}$ denotes the parameters of the mechanical load, $F_{[0,T]}(\mathbf{g},\boldsymbol{\theta},\boldsymbol{\xi})$ is the design/performance fitness function of the structure that is observed during time interval $[0,T]$, constraints (9b) and (9c) represent the system of governing PDEs (8) with boundary conditions (7), respectively, and $\underline{\boldsymbol{\theta}}$ and $\overline{\boldsymbol{\theta}}$ are the lower and upper bounds for the vector of control variables $\boldsymbol{\theta}$. Note that for the sake of simplicity, we omit the explicit dependency of $\mathbf{g}$ on the time variable t.

As the purpose of our optimization problem is to minimize the post-impact vibrations of the plate, the optimization criterion F in (9) is defined as the average squared middle plane displacement of the plate:

$$F_{[0,T]}(\mathbf{g},\boldsymbol{\theta},\boldsymbol{\xi}) = \frac{1}{T}\int_0^T \big(w_c(\boldsymbol{\theta},\boldsymbol{\xi},t)\big)^2 dt, \tag{10}$$

where $w_c = w|_{y=0}$ is the middle plane displacement at the center of the plate.

3.2 A two-stage stochastic programming formulation

It can be readily seen that the optimal parameters of the electromagnetic field as a solution of problem (9) depend heavily on the parameters of the applied impact load. Since the impact load can rarely be predicted or estimated with sufficient accuracy, in this subsection we discuss a stochastic extension of the general problem (9) under the assumption that the impact load is uncertain, or random.

To deal with the uncertainty in the parameters of the impact load, we resort to the *two-stage stochastic optimization* framework. In general, the discipline of stochastic optimization is concerned with determining optimal decision policies in situations when the decision making process is influenced by uncertainties in problem data (Prékopa, 1995; Birge and Louveaux, 1997; Kall and Mayer, 2005; Shapiro et al, 2009). One of the main assumptions within this framework is that the uncertain parameters can be described probabilistically as random variables from some probability space $(\Omega,\mathscr{F},\mathsf{P})$, where Ω is the set of random events, $\mathscr{F}$ is the sigma-algebra, and P is the probability measure. In other words, while the values of the uncertain parameters cannot be predicted with high degree of certainty, their probability distributions are believed to be known. The second assumption that is prevalent in most of stochastic optimization literature is that the probability distributions in question are *finite* ($|\Omega| < \infty$), and uncertainty in any given parameter ξ can be described by a finite set of possible realizations $\xi(\omega_1),\ldots,\xi(\omega_N)$, or *"scenarios"*, with each realization (scenario) $\omega_i \in \Omega$ having a prescribed non-zero probability $\mathsf{P}(\omega_i) > 0$.

The two-stage stochastic optimization framework models the situation when the decision-making process under uncertainty involves two decisions, or actions: the initial, or *first stage* decision/action, and a subsequent corrective, or *recourse*, or *second stage* decision/action. Namely, the first-stage action is selected *under uncertainty,* i.e., before the actual realizations of the uncertain factors can be observed. After the first-stage decision has been made, it is assumed that one can observe the actual *realized* values of the problem's uncertain parameters as well as their effect on the outcome of that decision (e.g., a person must place a bet in a horse race before its start; then the outcome of the race and the bet determine the winnings, if any).

Clearly, in most cases the first-stage action will not be optimally suited for any given realization of uncertainty. The second-stage, or recourse decision/action is made *after* the particular realization of uncertainties was observed, and its purpose is to correct the consequences of the first-stage action with respect to the actual observed outcome of uncertainty. It is important to emphasize that the second-stage decision is dependent on the observed realization of uncertainties and the first-stage decision; in turn, the first-stage decision must take into account the probability distribution of uncertainties and the corresponding second-stage actions (for example, a poorly chosen first-stage action may not allow for any *feasible* corrective actions).

Mathematically, a two-stage stochastic optimization problem can be written in the form:

$$\begin{aligned} \min \quad & \mathsf{E}_\omega\big(f_1(\mathbf{x},\omega)+f_2(\mathbf{x},\mathbf{y}(\omega),\omega)\big) \\ \text{s. t.} \quad & \mathbf{h}_1(\mathbf{x},\omega)\le \mathbf{0}, \quad \forall\omega\in\Omega, \\ & \mathbf{h}_2(\mathbf{x},\mathbf{y}(\omega),\omega)\le\mathbf{0}, \quad \forall\omega\in\Omega. \end{aligned} \tag{11}$$

Here, $\mathbf{x}$ denotes the vector of first-stage decisions and $\mathbf{y}=\mathbf{y}(\omega)$ denotes second-stage decision; note that we explicitly indicate its dependence on the random element ω from the set Ω of all possible random events. Function $f_1(\mathbf{x},\omega)$ denotes the first-stage design/decision criterion, and $f_2(\mathbf{x},\mathbf{y}(\omega),\omega)$ denotes the corresponding criterion for the second-stage action. Similarly, $\mathbf{h}_1(\mathbf{x},\omega)\le\mathbf{0}$ represents the first-stage constraints to be satisfied by the first-stage decision $\mathbf{x}$, and the next constraint stipulates that the second-stage constraints to be satisfied by the second-stage decision $\mathbf{y}(\omega)$ may depend explicitly on first-stage decision $\mathbf{x}$ and the observed realization of ω. An optimal solution of (11) delivers the best, on average, value of the first- and second-stage design criteria.

With respect to the problem of impact of a composite plate that was discussed in Section 2.2, we consider that the vector $\boldsymbol{\xi}$ of parameters that describe the mechanical impact load $p_z(t)=p_z(t;\boldsymbol{\xi})$ is random, $\boldsymbol{\xi}=\boldsymbol{\xi}(\omega)$, with a known distribution. Probability space Ω is finite and describes a finite number of scenarios, $\Omega=\{\omega_1,\dots,\omega_N\}$, where each scenario ω_i corresponds to a specific vector of parameters $\boldsymbol{\xi}(\omega_i)$ of the impact load, and the probabilities $\mathsf{P}(\omega_i)$ of random elements $\omega_i\in\Omega$ are known. The discrete scenarios may represent, for example, different types of foreign objects that may strike the composite plate.

It is assumed that the actual realization of the parameters of impact load, $\hat{\boldsymbol{\xi}}=\boldsymbol{\xi}(\omega_k)$ for some $\omega_k\in\Omega$, becomes known (observable) after a certain time T_0 (for example, an appropriate sensor technology can be employed to estimate the impact load during the impact event). The decision on the choice of control parameters $\boldsymbol{\theta}$ must be made at or prior to $t=0$, before the actual realization $\hat{\boldsymbol{\xi}}$ of the mechanical load can be observed. After time T_0, we have an opportunity for a corrective (recourse) action, which consists in adjusting the electromagnetic field so as to address the mismatch between the first-stage decision and the actual observation of uncertain parameters in the best way possible.

Specifically, during the first stage one applies an electromagnetic field with pre-computed parameters $\boldsymbol{\theta}$ so as to minimize the expected vibrations during the time period $t\in[0,T_0]$. It is assumed that during this time interval the profile of the mechanical load can be observed and identified, which allows for a subsequent correction $\boldsymbol{\theta}'=\boldsymbol{\theta}'(\omega)$ of the original selection $\boldsymbol{\theta}$, where we again explicitly indicate that the second-stage action $\boldsymbol{\theta}'$ depends on the observed realization $\omega\in\Omega$. Then, the two-stage stochastic PDE-constrained optimization problem that minimizes the plate's expected deflections can be formulated as:

$$\begin{aligned} \min_{\boldsymbol{\theta},\boldsymbol{\theta}'} \quad & \mathsf{E}_\omega\Big(F_{[0,T_0]}\big(\mathbf{g}(\omega),\boldsymbol{\theta},\boldsymbol{\xi}(\omega)\big) \\ & \qquad + F_{[T_0,T_1]}\big(\mathbf{g}'(\omega),\boldsymbol{\theta}'(\omega),\boldsymbol{\xi}(\omega)\big)\Big) \end{aligned} \tag{12}$$

s. t.

$$\frac{\partial\mathbf{g}(\omega)}{\partial y}=\boldsymbol{\Phi}\left(\mathbf{g},\frac{\partial\mathbf{g}}{\partial t},\frac{\partial^2\mathbf{g}}{\partial t^2},\boldsymbol{\theta},\boldsymbol{\xi}(\omega)\right),\ t\in[0,T_0],\ \forall\omega\in\Omega,$$

$$\mathbf{G}\left(\mathbf{g}(\omega),\frac{\partial\mathbf{g}(\omega)}{\partial t}\right)\bigg|_{y=\pm\frac{a}{2}}=\mathbf{0},\ t\in[0,T_0],\ \forall\omega\in\Omega,$$

$$\frac{\partial\mathbf{g}'(\omega)}{\partial y}=\boldsymbol{\Phi}\left(\mathbf{g}',\frac{\partial\mathbf{g}'}{\partial t},\frac{\partial^2\mathbf{g}'}{\partial t^2},\boldsymbol{\theta}'(\omega),\boldsymbol{\xi}(\omega)\right),$$
$$t\in[T_0,T_1],\ \forall\omega\in\Omega,$$

$$\mathbf{G}\left(\mathbf{g}'(\omega),\frac{\partial\mathbf{g}'(\omega)}{\partial t}\right)\bigg|_{y=\pm\frac{a}{2}}=\mathbf{0},\ t\in[T_0,T_1],\ \forall\omega\in\Omega,$$

$$\mathbf{g}\big|_{t=T_0}=\mathbf{g}'\big|_{t=T_0},\quad \frac{\partial\mathbf{g}}{\partial t}\bigg|_{t=T_0}=\frac{\partial\mathbf{g}'}{\partial t}\bigg|_{t=T_0},\ \forall\omega\in\Omega,$$

$$\underline{\boldsymbol{\theta}}\le\boldsymbol{\theta},\boldsymbol{\theta}'(\omega)\le\overline{\boldsymbol{\theta}},\quad\forall\omega\in\Omega.$$

Note the explicit dependence of vectors $\mathbf{g}(\omega)$, $\mathbf{g}'(\omega)$, $\boldsymbol{\theta}'(\omega)$, and $\boldsymbol{\xi}(\omega)$ on the random element $\omega\in\Omega$. The first term in the objective function of problem (12) corresponds to the first stage, when the parameters of the problem $\boldsymbol{\xi}(\omega)$ are uncertain with a known discrete distribution. During this stage, an electromagnetic field characterized by vector of parameters $\boldsymbol{\theta}$ is applied to minimize the expected value of $F_{[0,T_0]}\big(\mathbf{g}(\omega),\boldsymbol{\theta},\boldsymbol{\xi}(\omega)\big)$, the average squared middle plane displacement at the center of the plate during time interval $[0,T_0]$. The first two constraints in (12) stipulate that the governing PDEs (8) and boundary conditions (7) must hold at $t\in[0,T_0]$ for any of the possible impact load scenarios.

The second term in the objective of (12) represents the average squared middle plane displacement at the center of the plate during the second stage, from $t=T_0$ to $t=T_1$, which depends explicitly on the second-stage action $\boldsymbol{\theta}'(\omega)$ and implicitly on the preceding first stage action $\boldsymbol{\theta}$, by means of the continuity conditions that are given as the fifth line of constraints in (12). The values of vector $\mathbf{g}$ during time interval $[T_0,T_1]$ are denoted as $\mathbf{g}'$, and the third and fourth constraints of (12) require that the governing equations and boundary conditions hold during $[T_0,T_1]$ for all scenarios $\omega\in\Omega$. The fifth line of constraints (12) represents the continuity conditions at $t=T_0$ for the first-and second-stage mechanical fields $\mathbf{g}$ and $\mathbf{g}'$.

The two-stage stochastic PDE-constrained optimization problem (12) formalizes the proposed approach to control of mechanical structures under uncertainties with respect to the considered problem of impact of a composite plate. Clearly, the proposed framework allows for obvious generalizations.

In the remainder of the paper we discuss the numerical solution procedures for problem (12) as well as physical viability of its solutions.

4 Numerical solution and optimization methods

In this section we discuss the basic steps of solution procedure for the two-stage stochastic PDE-constrained problem (12) in the case of an impacted composite plate as presented in Section 2.2.

4.1 Numerical solution of the governing system of PDEs

Presence of a system of nonlinear PDEs as constraints in problem (12) necessitates effective solution methods for the respective PDEs in order to solve (12). With respect to the specific boundary value problem for the plate subjected to impact and electromagnetic loads, we employ the methods proposed in Zhupanska and Sierakowski (2007); Barakati and Zhupanska (2012a). For the sake of completeness of the exposition, we outline the key points of the corresponding solution procedure below.

The system of nonlinear governing PDEs (8) that enters the two-stage PDE-constrained problem (12) can be rewritten in the form:

$$\frac{\partial \mathbf{g}}{\partial y} = \boldsymbol{\Phi}\left(\mathbf{g}, \frac{\partial \mathbf{g}}{\partial t}, \frac{\partial^2 \mathbf{g}}{\partial t^2}, y, t, \boldsymbol{\theta}, \boldsymbol{\xi}\right), \tag{13}$$

where $\mathbf{g} = \mathbf{g}(x,y,t,\boldsymbol{\theta})$ is a vector of variables $\mathbf{g} = [v,w,W,N_{yy},N_{yz},M_{yy},E_x,B_z]$, $\boldsymbol{\Phi}$ is a nonlinear function from (8), and $\boldsymbol{\theta}$ is the optimization variable, i.e., the vector containing the parameters of the electromagnetic field (to be defined in Section 5).

A numerical solution procedure for this systems consists of a sequential application of a finite difference time integration, quasilinearization of the resulting system of the nonlinear ordinary differential equations (ODEs), and a finite difference spatial integration of the obtained two-point boundary value problem. The first step is to discretize (13) with respect to time t by applying Newmark finite difference time integration scheme (Newmark, 1959). This reduces (13) to the nonlinear two-point boundary problem for the system of ODEs:

$$\frac{d\mathbf{g}}{dy} = \boldsymbol{\Phi}_1(\mathbf{g}, y, \boldsymbol{\theta}, \boldsymbol{\xi}), \tag{14}$$

This system is solved at discrete moments of time with timestep Δt by using a quasilinearization method of Bellman and Kalaba (1965). This method allows for substituting the solution of (14) with a sequential solution of a linearized system with linearized boundary conditions:

$$\begin{aligned}
&\frac{d}{dy}\mathbf{g}^{k+1} = \boldsymbol{\Phi}_1(\mathbf{g}^k, y, \boldsymbol{\theta}, \boldsymbol{\xi}) + \mathbf{A}(\mathbf{g}^k, y, \boldsymbol{\theta}, \boldsymbol{\xi})(\mathbf{g}^{k+1} - \mathbf{g}^k),\\
&\left\{A_{ij}(\mathbf{g}^k, y, \boldsymbol{\theta}, \boldsymbol{\xi})\right\} = \left\{\frac{\partial \Phi_{1_i}(\mathbf{g}^k, y, \boldsymbol{\theta}, \boldsymbol{\xi})}{\partial g_j}\right\},\\
&\mathbf{D}_1(\mathbf{g}^k)\,\mathbf{g}^{k+1}(y_0) = \mathbf{d}_1(\mathbf{g}^k),\\
&\mathbf{D}_2(\mathbf{g}^k)\,\mathbf{g}^{k+1}(y_N) = \mathbf{d}_2(\mathbf{g}^k),
\end{aligned} \tag{15}$$

where $\mathbf{g}^{k+1}$ and $\mathbf{g}^k$ are the solutions on the current and previous iteration steps. A good choice for the initial guess $\mathbf{g}^0$ is a solution from the previous time step. Points y_0 and y_N correspond to the edges of the plate, matrices $\mathbf{D}_i(\mathbf{g}^k)$ and vectors $\mathbf{d}_i(\mathbf{g}^k)$, $i = 1,2$, are derived from the boundary conditions at $y = y_0$ and $y = y_N$. The sequence $\{\mathbf{g}^{k+1}\}$ of the solutions of the system (15) quickly converges to the solution of the nonlinear system and the stopping criterion for the iterative procedure is:

$$\max_i \left| g_i^{k+1}/g_i^k - 1 \right| \le \delta, \tag{16}$$

where $\delta > 0$ is the prescribed accuracy.

To solve the system of linear ODEs in (15) we employ the superposition method (Atkinson et al, 2009). If M is the dimensionality of the system and there are $M/2$ boundary conditions on both the left ($y = y_0$) and right ($y = y_N$) ends, then we may represent the solution of the system of the linear ODEs by a linear combination of $M/2$ linearly independent general solutions of the homogeneous system and one particular solution of the inhomogeneous system:

$$\mathbf{g}^{k+1}(y) = \sum_{j=1}^{M/2} c_j \mathbf{G}^j(y) + \mathbf{G}^{\frac{M}{2}+1}(y), \tag{17}$$

where c_j are the linear coefficients. The values of $\mathbf{G}^j$, $j = 1,\dots,\frac{M}{2}+1$ are obtained on the left end from the boundary conditions and then are propagated to the right end with the aid of the fourth-order Runge-Kutta method. At the right end the linear coefficients c_j can be found from the boundary conditions by solving a system of linear algebraic equations. In order to guarantee that vectors $\mathbf{G}^j$ are independent, and therefore coefficients c_j are uniquely determined at the right end, an orthonormalization procedure is employed after each iteration of the Runge-Kutta method. The corresponding transformation matrices are then used to restore the coefficients c_j.

4.2 PDE-constrained optimization framework

The existing approaches to PDE-constrained optimization problems can generally be categorized into two groups (Herzog and Kunisch, 2010). The first group of methods fits under the umbrella of "black box" optimization. This framework implies that one is able to obtain certain information about the objective function, which usually includes the value of the function for any given feasible point, its gradient and, perhaps, its higher order derivatives (depending on which optimization algorithm is employed) at that point. This information is then used to further direct the search for an optimal solution. It must be emphasized, however, that PDE constraints are embedded in the computation of the objective and its gradient and thus need to be satisfied at every step of the algorithm, which potentially makes this approach computationally expensive. For example, in the present work the value of the objective function is obtained by numerically solving a system of nonlinear PDEs using the procedure described in Section 4.1.

An alternative "discretize-then-optimize" approach consists in, first, discretizing the system of PDEs and replacing the PDE constraints in the problem with the resulting discretizations, often in the form of linear constraints. This generally leads to improved computational efficiency, as the system of governing PDEs is not required to be solved at every step. On the other hand, this method is not applicable to every type of PDE-constrained problem; for example, in our case the governing system of nonlinear PDEs cannot be solved by straightforward discretization.

The specifics of our particular problem dictates the use of black-box first-order optimization procedure, which can be summarized as follows: (i) compute the objective function by solving the governing system of PDEs numerically; (ii) compute first-order information, i.e., the gradient of the objective function at the current feasible point; (iii) apply a first-order optimization algorithm.

The value of objective function in (12) depends on the solution of a system of PDEs, which makes analytical computation of its gradient impractical. To this end, numerical differentiation techniques, such as complex differentiation (Squire and Trapp, 1998) or some version of automatic differentiation (Rall, 1986) can be employed.

4.3 Numerical differentiation

The proposed solution approach for two-stage stochastic PDE-constrained optimization problem (12) is based on first-order methods and requires computation of the gradient of the objective function at a given feasible point. Specifically, we are interested in the full derivatives of $F_{[0,T]}\big(\mathbf{g}(\omega),\boldsymbol{\theta},\boldsymbol{\xi}(\omega)\big)$ with respect to $\boldsymbol{\theta}$ and $F_{[T_0,T_1]}\big(\mathbf{g}'(\omega),\boldsymbol{\theta}'(\omega),\boldsymbol{\xi}(\omega)\big)$ with respect to $\big[\boldsymbol{\theta},\,\boldsymbol{\theta}'(\omega)\big]$. The function F itself has a quite simple structure, however, $\mathbf{g}$ and $\mathbf{g}'$ are implicitly dependent on parameters $\boldsymbol{\theta}$ and/or $\boldsymbol{\theta}'(\omega)$, as they are coupled through the system of governing equations (8). Next in this subsection we will not distinguish between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'(\omega)$ and refer to them as a single vector of parameters $\boldsymbol{\theta}$ that is used as an input to the system of governing equations.

There exists a number of methods for numerically computing a derivative of a function, among which are finite-difference method, adjoint method, complex differentiation, automatic (algorithmic) differentiation. In our work, we use the method which is closely related to both complex and automatic differentiation.

Complex differentiation method (Squire and Trapp, 1998; Martins et al, 2003, 2001) is applicable in case of an analytic function of a real variable. Instead of taking a small step in the direction of the real axis, as is customary in finite difference methods, a small increment is considered in the direction of the imaginary axis:

$$f(x+is) = f(x) + isf'(x) - \frac{s^2}{2!}f''(x) - i\frac{s^3}{3!}f'''(x) + O(s^4).$$

If s is small enough, by computing $f(x+is)$ one can obtain approximations to the values of $f(x)$ and $f'(x)$:

$$f(x) = \operatorname{Re} f(x+is) + O(s^2),\ f'(x) = \frac{\operatorname{Im} f(x+is)}{s} + O(s^2).$$

As it can be readily seen, the complex differentiation method offers a significant improvement in accuracy comparing to the traditional finite-difference approach at a relatively low computational overhead, as there is no a subtraction cancelation error. In practice, it allows for fast and stable computation of derivatives at almost machine precision. However, in the multivariate case, $f = f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^m$, one would have to evaluate $f(\mathbf{x}+is\mathbf{e}_k)$, where $\mathbf{e}_k$ is the k-th orthant in $\mathbb{R}^m$, for each $k = 1,\ldots,m$, in order to compute the gradient of f at the point $\mathbf{x}$. This obviously increases significantly the computational effort for evaluation of the gradient of $f(\mathbf{x})$. Alternative methods for numerical differentiation of multivariate functions that are based on the the same principle employ various generalizations of complex numbers.

Existing generalizations of complex numbers rely on different definitions of the imaginary unit. One of such generalizations is represented by *dual numbers* (Kantor and Solodovnikov, 1989; Piponi, 2004) of the form $a+\eta b$, where η is the *dual unit*, $\eta \neq 0$, $\eta^2 = 0$. Similarly, *hyper-dual numbers* have the form $a = a_0 + \eta_1 a_1 + \cdots + \eta_m a_m$ with m imaginary dual parts η_i such that $\eta_i\eta_j = 0$ for all i, j. The arithmetic operations with hyper-dual numbers are defined as follows:

$$\begin{aligned}
a+b &= a_0+b_0+\eta_1(a_1+b_1)+\cdots+\eta_m(a_m+b_m),\\
ab &= a_0b_0+\eta_1(a_1b_0+a_0b_1)+\cdots+\eta_m(a_0b_m+a_mb_0),\\
a/b &= \big(a_0b_0+\eta_1(a_1b_0-a_0b_1)+\ldots\\
&\qquad +\eta_m(a_mb_0-a_0b_m)\big)/b_0^2.
\end{aligned} \tag{18}$$

Then, given a multivariate function $f(x_1,\ldots,x_m)$, each of its m arguments can be represented as a hyper-dual number with m imaginary parts. More precisely, let variable x_i at a given point x_i^0 be represented by a hyper-dual number whose real part is equal to x_i^0 and all imaginary parts are set to zero, with the exception of the i-th imaginary part which is set to 1. It can be shown that upon application of the above hyper-dual arithmetic rules (18) for computation of the (hyper-dual) value of f, one obtains that the real part of the result is equal to $f(x_1^0,\ldots,x_m^0)$, and the i-th imaginary part is equal to $(\partial f/\partial x_i)\big|_{\mathbf{x}=\mathbf{x}^0}$. As an illustration, consider

$$f(x_1,x_2) = \frac{(x_1+x_2)x_1}{x_2}.$$

To find $\frac{\partial}{\partial x_1} f\left(x_1^0,x_2^0\right)$ and $\frac{\partial}{\partial x_2} f(x_1^0,x_2^0)$, let $x_1 = x_1^0+\eta_1 1+\eta_2 0$, $x_2 = x_2^0+\eta_1 0+\eta_2 1$. Then

$$\begin{aligned} f(x_1,y_1) &= \frac{x_1^0\left(x_1^0+x_2^0\right)}{x_2^0} + \eta_1\frac{2x_1^0+x_2^0}{x_2^0} - \eta_2\left(\frac{x_1^0}{x_2^0}\right)^2 \\ &= f(x_1^0,x_2^0)+\eta_1\frac{\partial}{\partial x_1}f(x_1^0,x_2^0)+\eta_2\frac{\partial}{\partial x_2}f(x_1^0,x_2^0). \end{aligned}$$

The described technique is, in fact, a forward mode of automatic differentiation (Rall, 1986), when derivative information is propagated forward with the computations according the the differentiation chain rule. There are different variations of this framework; more discussion of automatic differentiation with hyper-dual numbers can be found in Rall (1986); Piponi (2004); Fike and Alonso (2011).

In our case we need the full derivative of $F_{[0,T_0]}\left(\mathbf{g}(\boldsymbol{\theta}),\boldsymbol{\theta},\boldsymbol{\xi}\right)$ with respect to $\boldsymbol{\theta}$. The structure of F itself is quite simple and $\partial F/\partial\boldsymbol{\theta}$ can be found analytically. The biggest difficulty is to find the derivative of $\mathbf{g}$ with respect to $\boldsymbol{\theta}$. To do this, the governing system of equations (8) is solved numerically using hyper-dual numbers. The imaginary dual parts of all the input parameters except $\boldsymbol{\theta}$ are set to zero, while the i-th component of vector $\boldsymbol{\theta}$ has the form $\theta_i = \theta_i^0+\eta_i$, where θ_i^0 is the corresponding numerical value of the input parameter. The imaginary dual parts of the resulting hyper-dual values of the components of vector $\mathbf{g}$ then represent the sought partial derivatives of $\mathbf{g}$.

4.4 Optimization methods

Having computed the value and gradient of the objective function, we are now in a position to apply a first-order optimization scheme. In this study, we employed the active set method due to Hager and Zhang (2006). The algorithm consists of the nonmonotone gradient projection scheme and regular unconstrained conjugate gradient method and switches between them under certain conditions. We will outline the ideas of both of these methods and how they are connected. More details on the active set algorithm including convergence analysis can be found in Hager and Zhang (2006, 2005).

Nonmonotone gradient projection algorithm (NGPA) can be applied to the so-called "box-constrained" optimization problems of the form:

$$\min_{\mathbf{x}}\ \{f(\mathbf{x}) : \mathbf{l}\le\mathbf{x}\le\mathbf{u}\}.$$

Let us denote the feasible set of this problem as $\Theta = \{\mathbf{x}\in\mathbb{R}^n : \mathbf{l}\le\mathbf{x}\le\mathbf{u}\}$, and define $P(\mathbf{x})$ as the projection of a point in $\mathbb{R}^n$ on Θ:

$$P(\mathbf{x}) = \arg\min_{\mathbf{y}\in\Theta} \|\mathbf{x}-\mathbf{y}\|.$$

If $\mathbf{x}_k\in\Theta$ is the current iterate, we compute $\mathbf{x}_k' = \mathbf{x}_k-\alpha_k\mathbf{q}_k$, where $\mathbf{q}_k$ is the gradient of the objective function f at $\mathbf{x}_k$ and α_k is the corresponding step length. The point $\mathbf{x}_k'$ can be infeasible, so its projection $P(\mathbf{x}_k')$ on the feasible set is computed. By using a nonmonotone line search in the direction of the vector $\mathbf{d}_k = P(\mathbf{x}_k')-\mathbf{x}_k$, a new iterate $\mathbf{x}_{k+1}$ is found.

For unconstrained optimization problems, a conjugate gradient method can be used. Its main principle is that every step is made in the direction of steepest descent which is corrected by previous direction multiplied by some β:

$$\mathbf{x}_{k+1} = \mathbf{x}_k+\delta_k\mathbf{d}_k,\quad \mathbf{d}_{k+1} = -\mathbf{q}_{k+1}+\bar{\beta}_k^N\mathbf{d}_k,\quad \mathbf{d}_0 = -\mathbf{q}_0,$$

where δ_i is the step length chosen by inexact line search. In our work, the following conjugate gradient method by Hager and Zhang (2005) is used:

$$\bar{\beta}_k^N = \max\{\beta_k^N,\eta_k\},\ \ \beta_k^N = \frac{1}{\mathbf{d}_k^\top\mathbf{x}_k}\left(\mathbf{x}_k-2\mathbf{d}_k\frac{\|\mathbf{x}_k\|^2}{\mathbf{d}_k^\top\mathbf{x}_k}\right)^\top\mathbf{g}_{k+1},$$
$$\eta_k = \frac{-1}{\|\mathbf{d}_k\|\min\{\eta,\|\mathbf{q}_k\|\}}.$$

The nonmonotone gradient projection algorithm is globally convergent and in theory can deal with box-constrained optimization quite well. However, in practice its speed of convergence can be slow near a local minimizer. At the same time, the conjugate gradient method often has superlinear convergence for unconstrained optimization problems. The active set algorithm takes advantage of both these methods by using NGPA to determine active constraints (faces of the feasible set Θ, containing current iterate $\mathbf{x}_k$). Then, the conjugate gradient method is used to optimize over that face.

4.5 Solution procedure

Now, knowing all the main components of the solution procedure, we can assemble them together to show how the problem is solved. Since the system of governing equations is solved numerically by discretization, we modify appropriately expression (10) of the optimization criterion. Namely, assuming that the discretization time step Δt is sufficiently small, the integral in (10) can be approximated by:

$$F_{[0,T]}(\mathbf{g},\boldsymbol{\theta},\boldsymbol{\xi}) \simeq \frac{1}{T/\Delta t}\sum_{k=1}^{T/\Delta t}\left(w_c(\boldsymbol{\theta},\boldsymbol{\xi},t_k)\right)^2, \tag{19}$$

where the values of w_c are taken at time instants $t_k = k\Delta t$. Note also that the constant factor Δt in the above summation can be disregarded in the optimization problem since it is present as a constant scaling factor in the objectives of optimization problems (9) and (12).

The system of governing equations is solved using hyper-dual arithmetic to obtain the derivative of w_c with respect to $\boldsymbol{\theta}$ for each time step t_k. Knowing all the derivatives $\frac{\partial w_c}{\partial \theta_i}$, an approximation of the derivatives $\frac{\partial}{\partial \theta_i}F_{[0,T]}(\mathbf{g},\boldsymbol{\theta},\boldsymbol{\xi})$ can be found using the standard chain rule in (19). Given that the distribution of stochastic factors $\boldsymbol{\xi} = \boldsymbol{\xi}(\omega)$ in the two-stage stochastic PDE-constrained optimization problem (12) is assumed to be discrete with a finite support, and therefore can be modeled by a finite scenario set $\Omega = \{\omega_1,\ldots,\omega_N\}$, where $\mathsf{P}(\omega_i) > 0$ and $\sum_{i=1}^{N}\mathsf{P}(\omega_i) = 1$, problem (12) can be presented in the following form

$$\min_{\boldsymbol{\theta},\boldsymbol{\theta}'(\omega_i)} \sum_{i=1}^{N}\mathsf{P}(\omega_i)\Bigg(F_{[0,T_0]}\Big(\mathbf{g}(\omega_i),\boldsymbol{\theta},\boldsymbol{\xi}(\omega_i)\Big) \tag{20a}$$

$$+F_{[T_0,T_1]}\Big(\mathbf{g}'(\omega_i),\boldsymbol{\theta}'(\omega_i),\boldsymbol{\xi}(\omega_i)\Big)\Bigg)$$

s. t.

$$\frac{\partial \mathbf{g}(\omega_i)}{\partial y} = \boldsymbol{\Phi}\left(\mathbf{g}(\omega_i),\frac{\partial \mathbf{g}}{\partial t},\frac{\partial^2 \mathbf{g}}{\partial t^2},\boldsymbol{\theta}^{(1)},\boldsymbol{\xi}(\omega_i)\right),$$

$$t\in[0,T_0],\ \ \forall i\in\{1,\ldots,N\}, \tag{20b}$$

$$\mathbf{G}\left(\mathbf{g}(\omega_i),\frac{\partial \mathbf{g}(\omega_i)}{\partial t}\right)\Bigg|_{y=\pm\frac{a}{2}} = \mathbf{0},$$

$$t\in[0,T_0],\ \ \forall i\in\{1,\ldots,N\}, \tag{20c}$$

$$\frac{\partial \mathbf{g}'(\omega_i)}{\partial y} = \boldsymbol{\Phi}\left(\mathbf{g}'(\omega_i),\frac{\partial \mathbf{g}'}{\partial t},\frac{\partial^2 \mathbf{g}'}{\partial t^2},\boldsymbol{\theta}^{(2)}(\omega_i),\boldsymbol{\xi}(\omega_i)\right),$$

$$t\in[T_0,T_1],\ \ \forall i\in\{1,\ldots,N\}, \tag{20d}$$

$$\mathbf{G}\left(\mathbf{g}'(\omega_i),\frac{\partial \mathbf{g}'(\omega_i)}{\partial t}\right)\Bigg|_{y=\pm\frac{a}{2}} = \mathbf{0},$$

$$t\in[T_0,T_1],\ \ \forall i\in\{1,\ldots,N\}, \tag{20e}$$

$$\mathbf{g}(\omega_i)\big|_{t=T_0} = \mathbf{g}'(\omega_i)\big|_{t=T_0},\quad \frac{\partial \mathbf{g}(\omega_i)}{\partial t}\Bigg|_{t=T_0} = \frac{\partial \mathbf{g}'(\omega_i)}{\partial t}\Bigg|_{t=T_0},$$

$$\forall i\in\{1,\ldots,N\}, \tag{20f}$$

$$\underline{\boldsymbol{\theta}} \le \boldsymbol{\theta},\ \boldsymbol{\theta}'(\omega_i) \le \bar{\boldsymbol{\theta}},\quad \forall i\in\{1,\ldots,N\}.$$

To find the first group of summands of the objective (20a), $F_{[0,T_0]}\big(\mathbf{g}(\omega_i),\boldsymbol{\theta},\boldsymbol{\xi}(\omega_i)\big)$, and their partial derivatives w.r.t. $\boldsymbol{\theta}$, the boundary-value problem (20b)–(20c) is solved numerically for each $\omega_i \in \Omega$ using hyper-dual numbers. For the second group of components of the objective, $F_{[T_0,T_1]}\big(\mathbf{g}'(\omega_i),\boldsymbol{\theta}'(\omega_i),\boldsymbol{\xi}(\omega_i)\big)$, the boundary-value problem (20d)–(20e) must be solved and the continuity conditions (20f) must be satisfied. Note that $\mathbf{g}'$ implicitly depends on $\boldsymbol{\theta}$, and thus in the gradient of $\mathbf{g}'$ there are twice as many components as in the gradient of $\mathbf{g}$. In practice, to take into account this implicit dependence and continuity conditions in computing the value and gradient of $F_{[T_0,T_1]}\Big(\mathbf{g}'(\omega_i),\boldsymbol{\theta}'(\omega_i),\boldsymbol{\xi}(\omega_i)\Big)$, system (20b)–(20e) is solved for $t\in[0,T_1]$, using hyper-dual numbers for each $\omega_i\in\Omega$, with control parameters being switched from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}'(\omega_i)$ at time T_0. Then, the value of F and its derivatives, are computed according to (19) with first $T_0/\Delta t$ terms being ignored.

In order to perform optimization step of the active set algorithm, two systems of PDEs (20b, 20d) with boundary conditions (20c, 20e) in the constraints are solved in hyper-dual numbers for each $\omega_i\in\Omega$ per above. This enables one to compute the value and gradient of objective function (20a). The outlined computational procedure was implemented in C++ programming language.

5 Numerical results

In this section we report optimization results for a single-layer, transversely isotropic (x-axis is the axis of material symmetry and y-z is the plane of isotropy) carbon fiber reinforced composite plate of width $a = 0.1524$ m and thickness $h = 0.0021$ m. Elastic properties of the composite plate are as follows: Young's modulus in the fiber direction is $E_1 = 102.97$ GPa, Young's modulus in the transverse direction is $E_2 = 7.55$ GPa, Poisson's ratios are $\nu_{21} = \nu_{13} = 0.3$, density of the composite is $\rho = 1594$ kg/m^3, electrical conductivity in the fiber direction is $\sigma_1 = 39000$ S/m and electrical permittivity in the fiber direction is $\varepsilon_1 = 2.5015\times10^{-10}$ F/m. The plate is subjected to a transverse impact load (4) at the initial time moment, $t = 0$. Simultaneously, an electromagnetic load is applied and consists of the time-dependent electric current applied in the fiber direction (5) and constant in-plane magnetic field applied in the direction perpendicular to the electric current (6) (see Figure 1). Application of the electromagnetic load is expected to mitigate the effects of the mechanical impact by maximally reducing the post-impact vibrations of the plate.

The (randomized) applied impact load $\mathbf{p}$ (4) has the following profile, where the maximum impact pressure p_0 and the impact duration τ_p are uncertain parameters, which is

Scenario, ω	Probability, $\mathsf{P}(\omega)$	$p_0(\omega)$, MPa	$\tau_p(\omega)$, ms
ω_1	1/3	7.5	8.0
ω_2	1/3	10.0	10.0
ω_3	1/3	20.0	12.0

Table 1 Scenario realizations of the maximum impact pressure p_0 and impact duration τ_p of the mechanical impact load (21).

indicated by their dependence on a random event $\omega \in \Omega$:

$$p_x(y,t) = 0, \quad p_y(y,t) = 0, \tag{21}$$

$$p_z(y,t) = \begin{cases} -p_0(\omega)\sqrt{1-\left(\frac{y}{b}\right)^2}\sin\frac{\pi t}{\tau_p(\omega)}, \\ \qquad\qquad |y| \le b, \quad 0 \le t \le \tau_p(\omega), \\ 0, \qquad b < |y| \le \frac{a}{2}, \quad t > \tau_p(\omega). \end{cases}$$

Here $b = 0.01h$ is the width of the impact zone.

In such a way, the vector $\boldsymbol{\xi}(\omega)$ of uncertain parameters in the two-stage stochastic PDE-constrained optimization problem (12) contains p_0 and τ_p:

$$\boldsymbol{\xi}(\omega) = \left[p_0(\omega), \tau_p(\omega)\right].$$

It is assumed that the set Ω of random events contains three equiprobable elements, or scenarios:

$$\Omega = \{\omega_1, \omega_2, \omega_3\}, \quad \text{where} \quad \mathsf{P}(\omega_i) = 1/3, \;\; i = 1,2,3.$$

In the context of the conceptual application described in Section 1.1, this corresponds to the composite plate being hit at random by, e.g., three possible types of foreign objects or projectiles. Table 1 presents the numerical values of the possible realizations of the maximum impact pressure and impact duration of the impact load (21). The small size of the scenario set is chosen specifically for the illustrative purposes of our computational experiments; in practice, realistic descriptions of uncertainties require larger scenario sets.

The duration T_0 of the first stage was set at $T_0 = 10$ ms, which is equal to the average duration of impact in the considered scenarios. This reflects our assumptions that an appropriate sensory technology will allow for estimating the parameters of impact load during the impact event (see Sections 1.1 and 3.2). The total duration of computational time was set at $T_1 = 50$ ms. According to the two-stage stochastic framework described in Section 3.2, the electromagnetic field in the configuration prescribed by the first-stage solution is applied at $t = 0$. At $t = T_0$, the parameters of the electromagnetic field are changed as dictated by the second-stage solution; in such a way, the durations of the first and second stages are 10 ms and 40 ms, respectively.

The parameters of the magnetic field (6) applied to the plate are as follows:

$$B_x = 0, \quad B_y = B_y^* = 1.0 \text{ T}, \quad B_z = 0, \tag{22}$$

and the density $\mathbf{J}(t)$ of the time-dependent electric current (5) applied in the fiber direction is

$$J_x(t) = J_0 e^{-t/\tau_e} \sin\frac{\pi t}{\tau_s}, \quad J_y(t) = J_z(t) = 0, \tag{23}$$

where J_0, τ_e, and τ_s are the parameters determining the electric current waveform, i.e., the maximum current density, fall and rise times. The quantities J_0, τ_e, and τ_s constitute the vector $\boldsymbol{\theta}$ of decision variables, or control parameters:

$$\boldsymbol{\theta} = [J_0, \tau_e, \tau_s].$$

It is worth noting that the magnitude of magnetic field B_y is not formally included in the vector $\boldsymbol{\theta}$, and is fixed at the given value of 1 T in accordance to (22). This is due to our observation that when B_y was allowed to vary within a prescribed bounds (the so-called "box constraints") $0 \le B_y \le \overline{B}$, at optimality the decision variable B_y always assumed the maximum possible value, $B_y = \overline{B}$. Hence, for simplicity the value of B_y was fixed as in (22). The rest of the decision variables were box-constrained as follows:

$$|J_0| \le 10^8 \text{ A/m}^2, \quad 10^{-5} \text{ s} \le \tau_s, \tau_e \le 10^9 \text{ s}, \tag{24}$$

where the prescribed range of allowable current density values was chosen so as to eliminate Joule heating considerations (more precisely, to ensure that the thermal effects associated with application of electric current are negligible, see Barakati and Zhupanska (2012b) for an in-depth discussion of this issue). The box constraints on the fall and rise times τ_e and τ_s are selected in order to ensure realistic current profiles (as in the case of the lower bound) as well as to avoid numerical difficulties with convergence of the described above optimization procedures (as in the case of the upper bound).

During the optimization procedure, the initial values for both first and second stage solution vectors $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\theta}^{(1)}(\omega)$, $\omega \in \Omega$, were chosen as follows: $J_0 = 1.0 \times 10^6$ A/m^2, $\tau_s = 4.8$ ms, $\tau_e = 4.8$ ms.

The optimal solution of the two-stage stochastic PDE-constrained optimization problem (12) (or (20)) obtained during the described above solution process is presented in Table 2, which contains the parameters (J_0, τ_s, τ_e) of the waveform (23) of electric current as the components of the first-stage solution vector $\boldsymbol{\theta}^{(0)}$ and second-stage vectors $\boldsymbol{\theta}^{(1)}(\omega_i)$, $i = 1,2,3$. The corresponding optimal waveform profiles of the electric current (23) are shown in Figure 2. Again, we emphasize the structure of the obtained two-stage stochastic solution: during the time interval $[0, T_0]$, i.e., from $t = 0$ until $t = 10$ ms, the optimal first-stage electric current ($J_0 = 1.81 \times 10^6$ A/m^2, $\tau_s = 10.7$ ms, $\tau_e = 36.2$ ms) is applied in order to minimize the expected plate deflection due to an uncertain impact load. According to the assumptions of our model, the parameters of the actual realization of the randomized impact load (i.e., the actual observed scenario)

Table 2 Optimal parameters of the electric current (23) obtained after solving the two-stage stochastic PDE-constrained optimization problem (12).

Parameter	First stage	Second stage, scenario		
		ω_1	ω_2	ω_3
J_0, $10^6 \times$ A/m^2	1.81	100.0	0.93	100.0
τ_s, ms	10.7	4.9	7.4	2.2
τ_e, ms	36.2	2.5	0.01	4.1

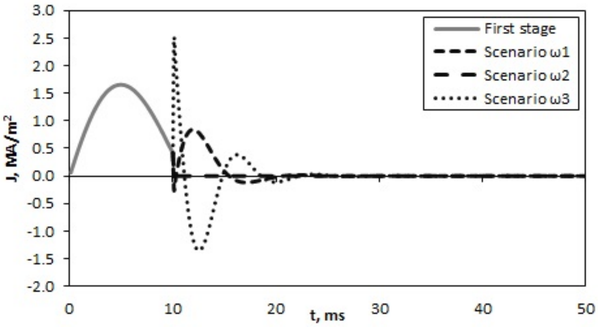


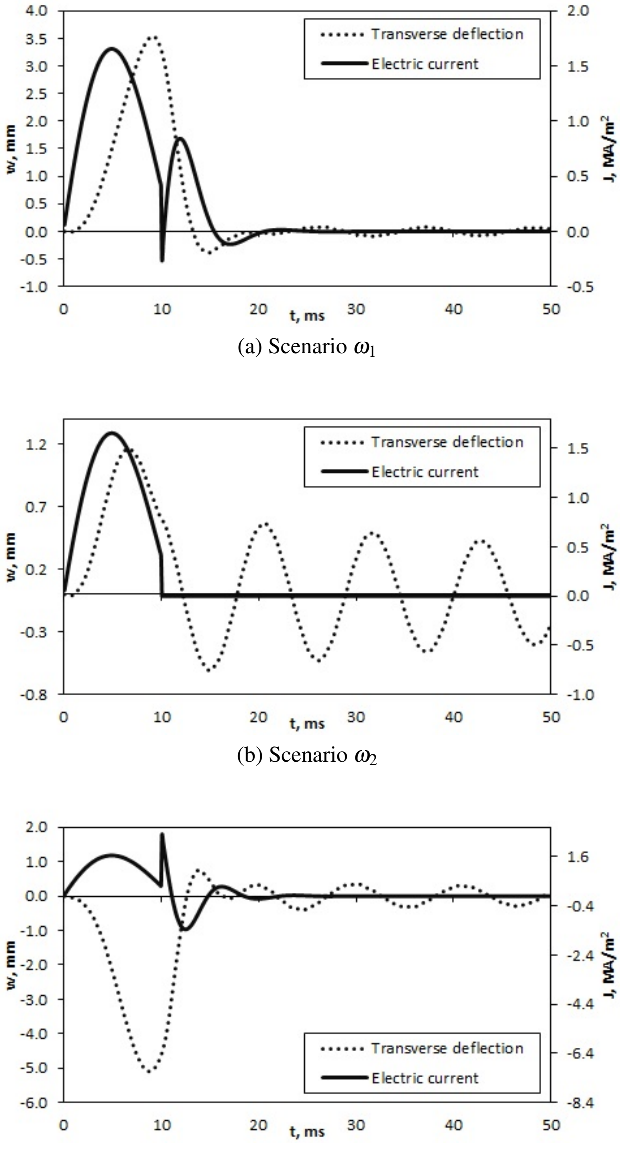
Fig. 2 Optimal electric current waveforms as specified in Table 2.

become known by time $t = T_0 = 10$ ms, and, depending on the observed scenario, the parameters of the electric current are "switched" at $t = T_0$ to the corresponding second-stage solution values so as to minimize post-impact vibrations of the plate. For example, if it is determined that the impact was "light", i.e., an impact load corresponding to scenario ω_1 was observed during $[0, T_0]$, then at $t = T_0$ the parameters of the electric current are changed to $J_0 = 10^8$ A/m^2, $\tau_s = 4.9$ ms, $\tau_e = 2.5$ ms.

The resulting vibrations of the plate during the time interval $[0, T_1]$ (i.e., from 0 to 50 ms) are displayed for each scenario, along with the corresponding current profile, in Figure 3. Note that in all three subfigures of Figure 3, the profile of the electric current between $t = 0$ and $t = 10$ ms is the same and represents the first-stage solution (due to the differences in the maximum impact pressure across the scenarios, the subfigures use different scales on the vertical axes). It is also of interest to note that in scenarios ω_1 and ω_2 the electromagnetic load applied during the first stage is such that it causes the plate to deflect in the direction opposite to the direction of impact. This observation is also in accord with the formulated model: the first stage solution minimizes the plate deflection "on average"; in addition, the magnitude of maximum impact presure in scenario ω_3 is two to almost three times higher than those in scenarios ω_1 and ω_2.

Figure 4 presents, for each of the three scenarios, the comparisons of the plate's transverse deflection with and without application of the (optimal) electromagnetic field. It is clear that the constructed two-stage stochastic optimization solution allows for significant suppression of vibrations caused by uncertain impact load in all three scenarios. It can be seen from Figure 4 that, while the developed two-stage model and the corresponding optimal parameters of the electromagnetic field result in substantial dampening of post-impact vibrations, the vibrations are not suppressed completely. This is a natural consequence of the fact that the impact load is uncertain, and therefore it is impossible to provide the "best" response to each of the possible scenarios.

Next we illustrate the effectiveness of the developed framework in the situation when the impact load is known beforehand, i.e., when it can be regarded deterministic. One can expect that in this case the parameters of the electromag-

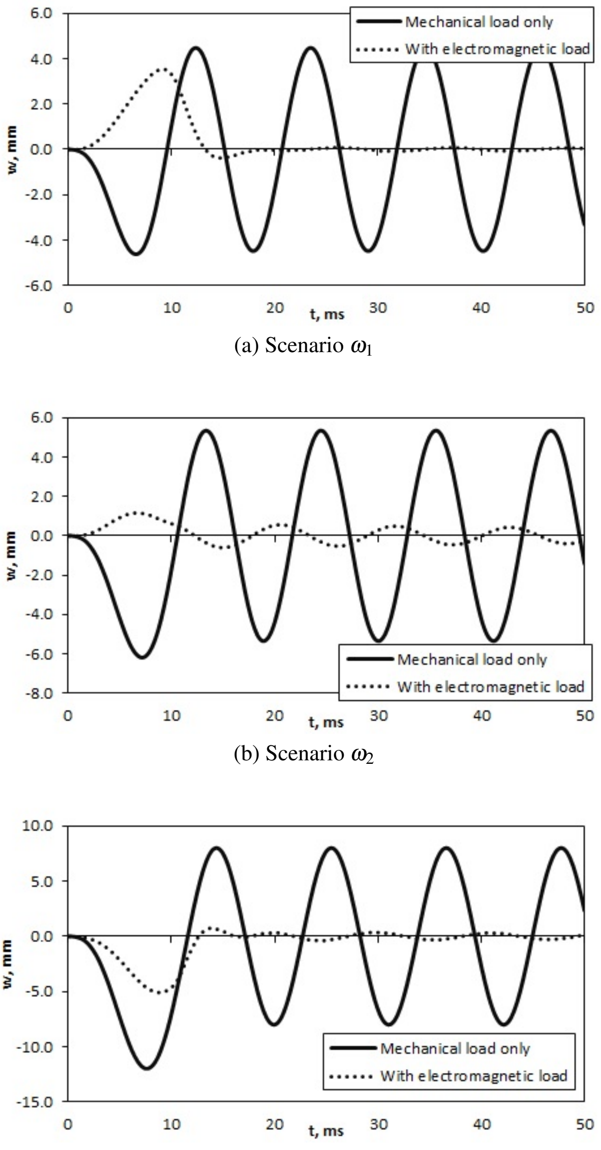


(a) Scenario ω_1

(b) Scenario ω_2

(c) Scenario ω_3

Fig. 3 Transverse deflection of the plate and optimal electric current waveforms corresponding to different impact load scenarios.



(a) Scenario ω_1

(b) Scenario ω_2

(c) Scenario ω_3

Fig. 4 Transverse deflection in the center of the plate vs. time for different scenarios.

Table 3 Optimal parameters of the electric current in the deterministic case when impact load has the same parameters as in scenario ω_2 of the stochastic case.

Parameter	First stage	Second stage, ω_2
J_0, $10^6 \times$ A/m^2	1.46412	0.924654
τ_s, ms	10.0972	7.18217
τ_e, ms	124.662	0.01

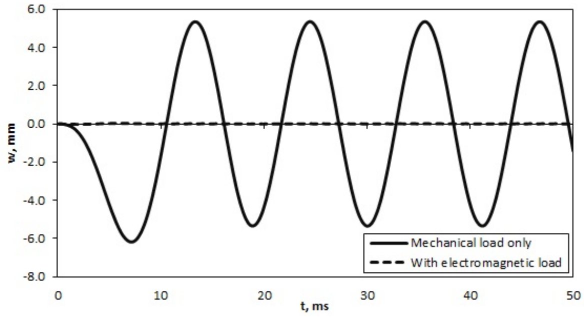


Fig. 5 Transverse deflection of the plate in the deterministic case that is based on scenario ω_2.

netic field can be tuned to achieve a much better mitigation of post-impact effects comparing to the stochastic case.

In particular, we assume that the deterministic impact load has the same parameters as the load of scenario ω_2, in accordance to expression (21) and Table 1. It is then convenient to consider that the stochastic problem is solved under the assumption that all scenarios, except ω_2, are impossible, i.e.,

$$\mathsf{P}(\omega_1) = 0, \quad \mathsf{P}(\omega_2) = 1, \quad \mathsf{P}(\omega_3) = 0.$$

This implies that in the scenario-based formulation (20) of the two-stage stochastic PDE-constrained optimization problem (12) the terms in the objective function that correspond to scenarios ω_1 and ω_3 are eliminated, and, in addition, the constraints that enforce satisfaction of the PDE equations and boundary conditions in scenarios ω_1, ω_3 are also disregarded.

Note, however, that the two-stage structure of the solution of (20) is still preserved, which means that at $t = T_0$ the parameters of the applied electric current are allowed to change. In other words, electric currents of two different waveforms determined by $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}(\omega_2)$ are applied during time intervals $[0, T_0]$ and $[T_0, T_1]$, respectively. During the time interval $[0, T_0]$, electric current with parameters given by the first stage solution $\boldsymbol{\theta}^{(1)}$ is used to optimally mitigate the impact itself, while during $[T_0, T_1]$ the electric current with parameters $\boldsymbol{\theta}^{(2)}(\omega_2)$ then suppresses the post-impact effects.

With exception of modifications just described, the rest of the parameters of the problem are the same as before. The obtained solution of this deterministic problem is given in Table 3. Figure 5 shows the transverse middle plane deflection, w_c, at the center of the plate, $y = 0$, as a function of time for the cases when only the mechanical load is present, and when the optimal electromagnetic field is applied as well. It is easy to see that in a deterministic setting the proposed framework is capable of practically eliminating the vibrations.

6 Conclusions

In this work, a two-stage stochastic PDE-constrained optimization methodology is developed for the active vibration control of structures in the presence of uncertainties in mechanical loads. The solution methodology includes a black-box first-order optimization procedure embedded in the two-stage stochastic optimization formulation. The black-box first-order optimization procedure consists of solving a system of governing PDEs and automatic differentiation with hyper-dual numbers for computing the objective function and its gradient, respectively; and applying a first-order active-set algorithm with a conjugate gradient method for solving the optimization problem.

The developed optimization methodology is applied to the problem of post-impact vibration control (via applied electromagnetic filed) of an electrically conductive carbon fiber reinforced composite plate subjected to an uncertain, or stochastic, impact load. The system of governing PDEs describing such problem consists of nonlinear equations of motion and Maxwell's equations. The randomized impact load applied to the plate is comprised of three equiprobable scenarios with different parameters of maximum impact pressure and impact duration. Simultaneously, according to the two-stage stochastic optimization framework, an electromagnetic load in the configuration prescribed by the first-stage optimization solution is applied at the initial moment of time and is changed at the end of the first stage as dictated by the second-stage optimization solution. The electromagnetic load is comprised of a time-dependent electric current applied in the fiber direction and a constant in-plane magnetic field applied in the direction perpendicular to the electric current. Electric current waveform characteristics (i.e., the maximum current density, fall and rise times) constitute the vector of optimization variables, or control parameters. The optimal solution of the two-stage stochastic PDE-constrained optimization problem represents a sequence of actions, where the first-stage electric current waveform is applied at the moment of impact without knowing the actual impact load parameters; the second-stage electric current waveform represents a corrective action, which is applied when the parameters of the actual impact load have been observed/identified. The results show that the constructed two-stage optimization solution allows for a significant suppression of vibrations caused by the randomized impact load in all impact load scenarios. Lastly, the effectiveness of the developed methodology is illustrated in the case of a deterministic impact load, where the two-stage strategy enables one to practically eliminate post-impact vibrations.

Acknowledgements Olesya Zhupanska and Dmitry Chernikov would like to acknowledge the support of DARPA, N66001-11-1-4133 (Disclaimer: Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of DARPA). Dmitry Chernikov is grateful for the support from the AFRL Mathematical Modeling and Optimization Institute during Summer 2013 and Summer 2014. Pavlo Krokhmal would like to acknowledge AFOSR grant FA9550-12-1-0142 and the U.S. Dept of Air Force grant FA8651-12-2-0010.

References

Atkinson KE, Han W, Stewart DE (2009) Numerical Solutions of Ordinary Differential Equations. John Wiley and Sons, Inc., New Jersey

Barakati A, Zhupanska OI (2012a) Analysis of the effects of a pulsed electromagnetic field on the dynamic response of electrically conductive composites. Applied Mathematical Modelling 36:6072–6089

Barakati A, Zhupanska OI (2012b) Thermal and mechanical response of a carbon fiber reinforced composite to a transverse impact and in-plane pulsed electromagnetic loads. Journal of Engineering Materials and Technology 134(3):031,004

Barakati A, Zhupanska OI (2013) Influence of the electric current waveform on the dynamic response of the electrified composites. International Journal of Mechanics and Materials in Design 9(1):11–20

Barakati A, Zhupanska OI (2014) Mechanical response of electrically conductive laminated composite plates in the presence of an electromagnetic field. Composite Structures 113:298–307

Bellman RE, Kalaba RE (1965) Modern Analytic and Computational Methods in Science and Mathematics. American Elsevier Publishing Company, Inc., New York

Birge JR, Louveaux F (1997) Introduction to Stochastic Programming. Springer, New York

Eringen AC (1989) Theory of electromagnetic elastic plates. International Journal of Engineering Science 27:363–375

Fike JA, Alonso JJ (2011) The development of hyper-dual numbers for exact second-derivative calculations. In: 49th AIAA Aerospace Sciences Meeting, vol 886

Gibson R (2010) A review of recent research on mechanics of multifunctional composite materials and structures. Composite Structures 92(12):2793–2810

Hager WW, Zhang H (2005) A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM Journal on Optimization 16(1):170–192

Hager WW, Zhang H (2006) A new active set algorithm for box constrained optimization. SIAM Journal on Optimization 17(2):526–557

Hasanyan D, Librescu L, Ambur D (2006) Buckling and postbuckling of magnetoelastic flat plates carrying an electric current. International Journal of Solids and Structures 43:4971–4996

Hasanyan DJ, Piliposyan GT (2001) Modelling and stability of magnetosoft ferromagnetic plates in a magnetic field. Proceedings of the Royal Society A 457:2063–2077

Herzog R, Kunisch K (2010) Algorithms for pde-constrained optimization. GAMM-Mitteilungen 33(2):163–176

Higuchi M, Kawamura R, Tanigawa Y (2007) Magneto-thermo-elastic stresses induced by a transient magnetic field in a conducting solid circular cylinder. International Journal of Solids and Structures 44:5316–5335

Kall P, Mayer J (2005) Stochastic Linear Programming: Models, Theory, and Computation. Springer

Kantor IL, Solodovnikov AS (1989) Hypercomplex Numbers: An Elementary Introduction to Algebras. Springer-Verlag, New York

Martins JRRA, Sturdza P, Alonso JJ (2001) The connection between the complex-step derivative approximation and algorithmic differentiation. In: 39th Aerospace Sciences Meeting and Exhibit, vol 921, p 2001

Martins JRRA, Sturdza P, Alonso JJ (2003) The complex-step derivative approximation. ACM Transactions on Mathematical Software (TOMS) 29(3):245–262

Maugin GA (1988) Continuum Mechanics of Electromagnetic Solids. North-Holland, Amsterdam

Moon FC (1984) Magnetosolid Mechanics. Wiley, New York

Newmark NM (1959) A method of computation for structural dynamics. Journal of the Engineering Mechanics Division Proceedings of the ASCE 85:67–97

Piponi D (2004) Automatic differentiation, c++ templates, and photogrammetry. Journal of Graphics Tools 9(4):41–55

Prékopa A (1995) Stochastic Programming. Kluwer Academic Publishers

Rall LB (1986) The arithmetic of differentiation. Mathematics Magazine 59(5):pp. 275–282

Rudnicki M (2002) Eigenvalue solutions for free motion of electroconductive plate in magnetic field. International Journal of Engineering Science 40:93–107

Shapiro A, Dentcheva D, Ruszczyński A (2009) Lectures on Stochastic Programming: Modeling and Theory. SIAM, Philadelphia, PA

Squire W, Trapp G (1998) Using complex variables to estimate derivatives of real functions. SIAM Review 40(1):110–112

Zhupanska OI, Sierakowski RL (2007) Effects of an electromagnetic field on the mechanical response of composites. Journal of Composite Materials 41(5):633–652

Zhupanska OI, Sierakowski RL (2011) Electro-thermomechanical coupling in carbon fiber polymer matrix composites. Acta Mechanica 218(3–4):219–232

On polyhedral approximations in p-order cone programming

Alexander Vinel Pavlo A. Krokhmal*

Department of Mechanical and Industrial Engineering
University of Iowa, 3131 Seamans Center, Iowa City, IA 52242, USA

Abstract

This paper discusses the use of polyhedral approximations in solving of p-order cone programming (pOCP) problems, or linear problems with p-order cone constraints, and their mixed-integer extensions. In particular, it is shown that the cutting-plane technique proposed in Krokhmal and Soberanis (2010) for a special type of polyhedral approximations of pOCP problems, which allows for generation of cuts in a constant time not dependent on the accuracy of approximation, is applicable to a larger family of polyhedral approximations. We also show that it can further be extended to form an exact solution method for pOCP problems with $O(\varepsilon^{-1})$ iteration complexity. Moreover, it is demonstrated that an analogous constant-time cut generating algorithm exists for recursively constructed lifted polyhedral approximations of second-order cones due to Ben-Tal and Nemirovski (2001). It is also shown that the developed polyhedral approximations and the corresponding cutting plane solution methods can be efficiently used for obtaining exact solutions of mixed-integer pOCP problems.

Keywords: p-order cone programming, second-order cone programming, polyhedral approximation, cutting plane methods, mixed-integer p-order cone programming, stochastic programming, portfolio optimization.

1 Introduction

In this paper we consider solving linear programming problems with p-order cone constraints

$$
\begin{aligned}
\min\quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} && \text{(1a)}\\
\text{s. t.}\quad & \mathbf{A}\mathbf{x} \le \mathbf{b}, && \text{(1b)}\\
& \big\|\mathbf{C}^{(k)}\mathbf{x} + \mathbf{e}^{(k)}\big\|_{p_k} \le \mathbf{h}^{(k)\mathsf{T}}\mathbf{x} + f^{(k)}, \quad k = 1, \dots, K, && \text{(1c)}\\
& \mathbf{x} \in \mathbb{R}^n,
\end{aligned}
$$

where $\|\cdot\|_p$ denotes the p-norm in $\mathbb{R}^N$:

$$
\|\mathbf{a}\|_p = \begin{cases} \big(|a_1|^p + \dots + |a_N|^p\big)^{1/p}, & p \in [1, \infty),\\ \max\big\{|a_1|, \dots, |a_N|\big\}, & p = \infty. \end{cases} \tag{2}
$$

We call formulation (1) a p-order cone programming problem (pOCP) by analogy with second-order cone programming (SOCP), which constitutes a special case of (1) when $p_k = 2$ for all $k = 1, \dots, K$.

Our motivation for considering problems of the form (1) stems from risk-averse optimization under uncertainty and stochastic programming, where use of certain classes of risk measures leads to problems with p-order cone constraints; see Section 4.1 for details.

[*] Corresponding author. E-mail: krokhmal@engineering.uiowa.edu

The available literature on solving problem (1) with "general" values of $p_k \in (1, \infty)$, i.e., not restricted to well-studied special cases of $p_k = 1, 2$, or ∞, is relatively limited. Interior-point approaches to p-order cone programming have been considered by Xue and Ye [25] with respect to minimization of sum of p-norms; a self-concordant barrier for p-cone has also been introduced by Nesterov [19]. Glineur and Terlaky [12] proposed an interior-point algorithm along with the corresponding barrier functions for a related problem of l_p-norm optimization (see also [21]). A polyhedral approximation approach to pOCP problems was considered by Krokhmal and Soberanis [15]. In the case when p is a rational number, the existing primal-dual methods of second-order cone programming can be employed for solving p-order cone optimization problems using a reduction of p-order cone constraints to a system of linear and second-order cone constraints proposed by Nesterov and Nemirovski [20] and Ben-Tal and Nemirovski [8], see also Morenko et al. [18].

This paper represents a continuation of the work of Krokhmal and Soberanis [15] on polyhedral approximation approaches to solving pOCP problems. The contribution of this work to the literature consists of the following: it is shown that the cutting plane method developed in [15] for solving a special type of polyhedral approximations of pOCP problems, which allows for generation of cuts in a constant time not dependent on the accuracy of approximation, is applicable to a larger family of polyhedral approximations. Further, it is demonstrated that this constant-time cut generation procedure can be modified so as constitute an exact solution method with $O(\varepsilon^{-1})$ iteration complexity. Next, we present a constant-time cut generation scheme for lifted polyhedral approximations of SOCP problems due to Ben-Tal and Nemirovski [9]. The noteworthy aspect of this result is that Ben-Tal and Nemirovski's lifted polyhedral approximation is constructed recursively, with the length of recursion controlling the accuracy of approximation, yet the cuts can be generated in a constant time that does not depend on the accuracy/recursion length. Finally, we illustrate that the polyhedral approximation approach and the corresponding cutting plane solution methods can be efficiently employed for obtaining exact solutions of mixed-integer extensions of pOCP problems (see below).

The paper is organized as follows: in Section 2 we discuss the general properties of polyhedral approximations of p-cones, Section 3.1 summarizes the general cutting plane method for polyhedral approximations of pOCP problems. In Sections 3.2 and 3.3 we explore fast constant-time cut generating techniques for gradient-based and lifted polyhedral approximations of pOCP and SOCP problems, respectively. The developed solution techniques are then illustrated on pOCP and SOCP problems of type (1), and are also employed for solving mixed-integer p-order cone programming (MIpOCP) problems

$$\begin{aligned}
\min \quad & \mathbf{c}^\mathsf{T}\mathbf{x} + \mathbf{d}^\mathsf{T}\mathbf{z} && \text{(3a)}\\
\text{s. t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} \le \mathbf{b}, && \text{(3b)}\\
& \left\|\mathbf{C}^{(k)}\mathbf{x} + \mathbf{D}^{(k)}\mathbf{z} + \mathbf{e}^{(k)}\right\|_{p_k} \le \mathbf{h}^{(k)\mathsf{T}}\mathbf{x} + \mathbf{g}^{(k)\mathsf{T}}\mathbf{z} + f^{(k)}, \quad k = 1, \ldots, K, && \text{(3c)}\\
& \mathbf{x} \in \mathbb{R}^n, \ \ \mathbf{z} \in \mathbb{Z}^m, && \text{(3d)}
\end{aligned}$$

which arise in the context of portfolio optimization with certain risk measures. The corresponding discussion is presented in Section 4.

2 Polyhedral approximations of p-order cones

In contrast to the Euclidean ($p = 2$) norm, which admits a representation via scalar product, $\|\mathbf{a}\|_2 = \left(\mathbf{a}^\mathsf{T}\mathbf{a}\right)^{1/2}$, the general $p \neq 2$ norm $\|\cdot\|_p$ explicitly requires the absolute value operator $|\cdot|$ in (2). Thus, in what follows it suffices to consider p-cones in the positive orthant of $\mathbb{R}^{N+1}$,

$$\mathcal{K}_p^{(N+1)} = \left\{\boldsymbol{\xi} \in \mathbb{R}_+^{N+1} \;\middle|\; \xi_0 \ge \|(\xi_1, \ldots, \xi_N)\|_p\right\}, \tag{4}$$

since in the context of problems (1) and (3) the absolute values of p-norm operands can be expressed using linear constraints. Then, by a polyhedral approximation of $\mathcal{K}_p^{(N+1)}$ we understand a polyhedral cone in $\mathbb{R}_+^{N+1+\kappa_m}$, where

$\kappa_m \geq 0$ may be generally non-zero,

$$\mathcal{H}^{(N+1)}_{p,m} = \left\{ \begin{pmatrix} \boldsymbol{\xi} \\ \mathbf{u} \end{pmatrix} \in \mathbb{R}^{N+1+\kappa_m}_{+} \;\middle|\; \mathbf{H}^{(N+1)}_{p,m} \begin{pmatrix} \boldsymbol{\xi} \\ \mathbf{u} \end{pmatrix} \geq \mathbf{0} \right\}, \tag{5}$$

having the properties that:

(H1) any $(\xi_0, \ldots, \xi_N)^\mathsf{T} \in \mathcal{K}^{(N+1)}_p$ can be extended to some $(\xi_0, \ldots, \xi_N, u_1, \ldots, u_{\kappa_m})^\mathsf{T} \in \mathcal{H}^{(N+1)}_{p,m}$;

(H2) for some prescribed $\varepsilon = \varepsilon(m) > 0$, any $(\xi_0, \ldots, u_{\kappa_m})^\mathsf{T} \in \mathcal{H}^{(N+1)}_{p,m}$ satisfies $\|(\xi_1, \ldots, \xi_N)\|_p \leq (1+\varepsilon)\xi_0$.

Here m is the parameter of the construction that controls the approximation accuracy ε. Replacing each of the p-order cone constraints in problem (1) by their polyhedral approximations of the form (5), we obtain a linear programming approximation of the pOCP problem (1):

$$\min \left\{ \mathbf{c}^\mathsf{T}\mathbf{x} \;\middle|\; \mathbf{A}\mathbf{x} \leq \mathbf{b},\; \mathbf{H}^{(N_k+1)}_{p_k,m_k} \begin{pmatrix} \mathbf{h}^{(k)\mathsf{T}}\mathbf{x} + f^{(k)} \\ \mathbf{C}^{(k)}\mathbf{x} + \mathbf{e}^{(k)} \\ \mathbf{u}^{(k)} \end{pmatrix} \geq \mathbf{0}, \quad \mathbf{u}^{(k)} \geq \mathbf{0}, \quad k = 1, \ldots, K \right\}. \tag{6}$$

Observe that the projection of the feasible region of (6) on the space of variables $\mathbf{x}$ lies in between the feasible set of pOCP (1) and that of its "ε-relaxation",

$$\min \left\{ \mathbf{c}^\mathsf{T}\mathbf{x} \;\middle|\; \mathbf{A}\mathbf{x} \leq \mathbf{b},\; \left\| \mathbf{C}^{(k)}\mathbf{x} + \mathbf{e}^{(k)} \right\|_{p_k} \leq (1+\varepsilon)\big(\mathbf{h}^{(k)\mathsf{T}}\mathbf{x} + f^{(k)}\big),\; k = 1, \ldots, K \right\}. \tag{7}$$

Thus, problem (6) represents an ε-approximation of pOCP (1), given that the feasible regions of problems (1) and (7) are "close". Conditions under which the feasible sets of (1) and (7) are indeed $O(\varepsilon)$-close have been given by Ben-Tal and Nemirovski [9, Proposition 4.1] for the case of $p = 2$, and their argumentation carries over to the case of $p \neq 2$ practically without modifications. Specifically, if we denote by (pOCP) and (pOCP$_\varepsilon$) the initial problem (1) and its ε-relaxation (7), respectively, the following holds.

Proposition 1 (Ben-Tal and Nemirovski [9]) *Assume that* (pOCP) *is: (i) strictly feasible, i.e., there exist* $\bar{\mathbf{x}}$ *and* $r > 0$ *such that*

$$\mathbf{A}\bar{\mathbf{x}} \leq \mathbf{b}, \quad \left\| \mathbf{C}^{(k)}\bar{\mathbf{x}} + \mathbf{e}^{(k)} \right\|_{p_k} \leq \mathbf{h}^{(k)\mathsf{T}}\bar{\mathbf{x}} + f^{(k)} - r, \quad k = 1, \ldots, K, \tag{8a}$$

and (ii) "semibounded", i.e., there exists $R > 0$ *such that*

$$\mathbf{A}\mathbf{x} \leq \mathbf{b},\; \left\| \mathbf{C}^{(k)}\mathbf{x} + \mathbf{e}^{(k)} \right\|_{p_k} \leq \mathbf{h}^{(k)\mathsf{T}}\mathbf{x} + f^{(k)},\; k = 1, \ldots, K \;\Rightarrow\; \mathbf{h}^{(k)\mathsf{T}}\mathbf{x} + f^{(k)} \leq R,\; k = 1, \ldots, K. \tag{8b}$$

Then for every $\varepsilon > 0$ *such that* $\gamma(\varepsilon) = R\varepsilon/r < 1$*, one has*

$$\gamma(\varepsilon)\bar{\mathbf{x}} + (1 - \gamma(\varepsilon))\,\mathrm{Feas}\,(\mathrm{pOCP}_\varepsilon) \subset \mathrm{Feas}\,(\mathrm{pOCP}) \subset \mathrm{Feas}\,(\mathrm{pOCP}_\varepsilon), \tag{8c}$$

where Feas (P) *denotes the feasible set of a problem* (P).

Remark 1 As noted in [9], the second inclusion in (8c) holds trivially, whereas the first inclusion rules out (under the stated conditions) the situations in which, for example, the pOCP problem is infeasible but every its ε-relaxation is feasible.

In constructing polyhedral approximations (5) of p-order cones we follow the "lift-and-approximate" approach of Ben-Tal and Nemirovski [9], who developed efficient, in terms of dimensionality, polyhedral approximations for quadratic cones. The first step in the construction procedure consists in a lifted representation, dubbed by the authors "tower of variables", of a p-cone in $\mathbb{R}^{N+1}_{+}$, as a nested sequence of $N-1$ three-dimensional p-cones.

The original construction relied on the assumption that $N = 2^d$ for some integer $d \geq 1$, which was by no means restrictive, but allowed for a simple structure of the lifted set, which could be visualized as a symmetric binary tree of three-dimensional cone inequalities that are partitioned into $d = \log_2 N$ "levels", with 2^{d-l} inequalities at a level l. Below we present a slightly different notation/representation of the "tower-of-variables" lifting technique that does not explicitly use the binary tree structure, and which simplifies its practical implementation in the case of general $N \neq 2^d$. Namely, given the $(N+1)$-dimensional p-cone, consider the set defined by intersection of $N-1$ three-dimensional p-cones in $\mathbb{R}_+^{N+1} \times \mathbb{R}_+^{N-1}$:

$$\xi_0 = \xi_{2N-1}, \quad \xi_{N+j} \geq \|(\xi_{2j-1}, \xi_{2j})\|_p, \quad j = 1, \ldots, N-1. \tag{9}$$

Proposition 2 *Projection of set* (9) *onto the space of variables* $(\xi_0, \ldots, \xi_N)$ *coincides with the set* (4). *In other words, any* $\boldsymbol{\xi} \in \mathbb{R}_+^{N+1}$ *that satisfies* (4) *can be extended to* $\boldsymbol{\xi} \in \mathbb{R}_+^{N+1} \times \mathbb{R}_+^{N-1}$ *that satisfies* (9), *and any* $\boldsymbol{\xi} \in \mathbb{R}_+^{2N}$ *satisfying* (9) *is such that its first* $N+1$ *components satisfy* (4).

Proof: Follows immediately by expanding the recursion in (9). □

Remark 2 The chain inequalities (9) can similarly be organized into a binary tree, where the variable on the left-hand side of p-cone inequality represents a parent node, and the two variables on the right-hand side are its child nodes. Such a binary tree, however, will have a rather non-symmetric structure. If, for example, $N = 5$, then $\xi_9 = \xi_0$ is the root, or level $3 = \lceil \log_2 5 \rceil$ node, ξ_8, ξ_7 are level-2 nodes, $\xi_3, \ldots, \xi_6$ are level-1 nodes, and ξ_1, ξ_2 are level-0 nodes. If, on the other hand, $N = 2^d$, then the binary tree becomes symmetric and coincides with that in [9], where level 0 contains the nodes $\xi_1, \ldots, \xi_N$.

The second step of the procedure is to construct a polyhedral approximation

$$\mathcal{H}_{p,m}^{(3)} = \left\{ \begin{pmatrix} \boldsymbol{\xi} \\ \mathbf{u} \end{pmatrix} \in \mathbb{R}_+^{3+\kappa_m} \;\middle|\; \mathbf{H}_{p,m}^{(3)} \begin{pmatrix} \boldsymbol{\xi} \\ \mathbf{u} \end{pmatrix} \geq \mathbf{0} \right\} \tag{10}$$

for each of the three-dimensional p-cones in (9). Observe that if approximation (10) of each of the three-dimensional p-cones (9) contains $O(\nu)$ facets, $\nu = \nu(m)$, the total number of facets in the approximation of the original $(N+1)$-dimensional p-cone is $O(\nu N)$, i.e., it is linear in the dimensionality N of the original p-cone.

Proposition 3 *Consider cone* (4) *and its lifted representation* (9). *If each of the three-dimensional cones in* (9) *is approximated by* (10) *with an accuracy* $\epsilon > 0$, *the resulting approximation accuracy* ε *of the original cone* (4) *satisfies*

$$\varepsilon \leq (1+\epsilon)^{\lceil \log_2 N \rceil} - 1.$$

Proof: The vector $\boldsymbol{\xi} \in \mathbb{R}_+^{2N}$ must satisfy $\xi_0 = \xi_{2N-1}$, $(1+\epsilon)\xi_{N+j} \geq \|(\xi_{2j-1}, \xi_{2j})\|_p$, $j = 1, \ldots, N-1$. Expanding the recursion, we obtain

$$\begin{aligned} \xi_0^p = \xi_{2N-1}^p &\geq \frac{\xi_{2N-3}^p}{(1+\epsilon)^p} + \frac{\xi_{2N-2}^p}{(1+\epsilon)^p} \geq \frac{\xi_{2N-7}^p}{(1+\epsilon)^{2p}} + \frac{\xi_{2N-6}^p}{(1+\epsilon)^{2p}} + \frac{\xi_{2N-5}^p}{(1+\epsilon)^{2p}} + \frac{\xi_{2N-4}^p}{(1+\epsilon)^{2p}} \geq \cdots \\ &\geq \frac{\xi_1^p}{(1+\epsilon)^{pk_1}} + \ldots + \frac{\xi_N^p}{(1+\epsilon)^{pk_N}}, \end{aligned}$$

where k_i is the number of "levels" in the "tower of variables" on the way from ξ_{2N-1} to ξ_i. It is straightforward to check that $k_i \in \{\lceil \log_2 N \rceil - 1, \lceil \log_2 N \rceil\}$, whence $(1+\epsilon)^{\lceil \log_2 N \rceil}\xi_0 \geq \|(\xi_1, \ldots, \xi_N)\|_p$. □

When $p = 1$ or $p = \infty$, the cone $\mathcal{K}_p^{(3)}$ is already polyhedral; in the case of $p = 2$, the problem of constructing a polyhedral approximation of the second-order cone $\mathcal{K}_2^{(3)}$ was also addressed by Ben-Tal and Nemirovski [9], who proposed the following *lifted* polyhedral approximation of $\mathcal{K}_2^{(3)}$,

$$
\begin{aligned}
& u_0 \geq \xi_1, && && \text{(11a)}\\
& v_0 \geq \xi_2, && && \text{(11b)}\\
& u_i = \cos\left(\tfrac{\pi}{2^{i+1}}\right) u_{i-1} + \sin\left(\tfrac{\pi}{2^{i+1}}\right) v_{i-1}, && i = 1, \dots, m, && \text{(11c)}\\
& v_i \geq \left| -\sin\left(\tfrac{\pi}{2^{i+1}}\right) u_{i-1} + \cos\left(\tfrac{\pi}{2^{i+1}}\right) v_{i-1} \right|, && i = 1, \dots, m, && \text{(11d)}\\
& u_m \leq \xi_0,\ v_m \leq \tan\left(\tfrac{\pi}{2^{m+1}}\right) u_m, && && \text{(11e)}\\
& 0 \leq u_i,\ v_i, && i = 0, \dots, m. && \text{(11f)}
\end{aligned}
$$

Remarkably, the accuracy of the polyhedral approximation (11) is exponentially small in m: $\epsilon(m) = O\big(4^{-m}\big)$. The construction is based on an elegant geometric argument that utilizes a well-known elementary fact that rotation of a vector in $\mathbb{R}^2$ is an affine transformation that preserves the Euclidean norm (2-norm) and that the parameters of this affine transform depend only on the angle of rotation. An approach to constructing a framework of polyhedral relations that generalizes inductive constructions of extended formulations via projections, such as the polyhedral approximation (11) has been introduced by Kaibel and Pashkovich [13].

Unfortunately, the lifted polyhedral approximation (11) of the second-order cone $\mathcal{K}_2^{(3)}$ does not seem to be extandable to general p-order cones $\mathcal{K}_p^{(3)}$ with $p \in (1, 2) \cup (2, \infty)$. Therefore, we employ a "gradient" approximation of $\mathcal{K}_p^{(3)}$ using circumscribed planes. Given the parameter of construction $m \in \mathbb{N}$, let us call function $\varphi_m : [0, m] \mapsto [0, \pi/2]$ an *approximation function* if it is continuous and strictly increasing on $[0, m]$, and, moreover, satisfies

$$
\Delta\varphi_m = \max_{i=0,\dots,m-1} \{\varphi_m(i+1) - \varphi_m(i)\} \to 0, \quad m \to \infty.
$$

Then, for the following parametrization of the p-cone surface in $\mathbb{R}^3_+$

$$
\xi_1 = \xi_0 \frac{\cos\theta}{(\cos^p\theta + \sin^p\theta)^{1/p}}, \quad \xi_2 = \xi_0 \frac{\sin\theta}{(\cos^p\theta + \sin^p\theta)^{1/p}}, \quad \xi_0 \geq 0, \quad \theta \in [0, \tfrac{\pi}{2}], \tag{12}
$$

where θ is the polar angle, any given approximation function φ_m generates a *gradient approximation* of $\mathcal{K}_p^{(3)}$

$$
\mathcal{H}_{p,m}^{(3)}(\varphi_m) = \left\{ \boldsymbol{\xi} \in \mathbb{R}^3_+ \mid \xi_0 \geq \alpha_{p,i}[\varphi_m]\, \xi_1 + \beta_{p,i}[\varphi_m]\, \xi_2, \ i = 0, \dots, m \right\}, \tag{13a}
$$

where

$$
\begin{pmatrix} \alpha_{p,i}[\varphi_m] \\ \beta_{p,i}[\varphi_m] \end{pmatrix} = \big(\cos^p \varphi_m(i) + \sin^p \varphi_m(i)\big)^{1/p-1} \begin{pmatrix} \cos^{p-1} \varphi_m(i) \\ \sin^{p-1} \varphi_m(i) \end{pmatrix}, \quad i = 0, \dots, m. \tag{13b}
$$

The values $\varphi_m(i)$ in (13) represent the polar angles at which the planes $\xi_0 = \alpha_{p,i}\xi_1 + \beta_{p,i}\xi_2$ are tangent to the p-cone $\mathcal{K}_p^{(3)}$. In such a way, the properties of the polyhedral approximation (13) of the p-cone $\mathcal{K}_p^{(3)}$ are determined by the values of φ_m at integer values $\{0, \dots, m\}$ of its argument; nevertheless, the computability properties of $\varphi_m(t)$ for arbitrary values $t \in [0, m]$ are also of major importance, as will be shown in the next section. The following proposition establishes the quality of the gradient polyhedral approximation (13), and is a generalization of a similar result established for a special choice of φ_m in [15].

Proposition 4 *For large enough values of $m \in \mathbb{N}$, the polyhedral set $\mathcal{H}_{p,m}^{(3)}(\varphi_m)$ defined by the gradient approximation* (13) *with approximation function φ_m satisfies properties (H1)–(H2). Specifically, if the approximation function is such that for some $r > 0$*

$$
\Delta\varphi_m = O(m^{-r}), \quad m \gg 1,
$$

then for any $\boldsymbol{\xi} \in \mathcal{K}_p^{(3)}$ one has $\boldsymbol{\xi} \in \mathcal{H}_{p,m}^{(3)}$, and any $\boldsymbol{\xi} \in \mathcal{H}_{p,m}^{(3)}$ satisfies $\|(\xi_1, \xi_2)\|_p \leq (1 + \epsilon(m))\xi_0$, where the approximation accuracy $\epsilon(m)$ is polynomially small in m:

$$
\epsilon(m) = O\big(m^{-r\min\{p,2\}}\big), \quad m \gg 1.
$$

Remark 3 One possible choice of φ_m is $\varphi_m(t) = \frac{\pi}{2m}t$, which yields a "uniform" gradient approximation of the p-cone, i.e., a gradient approximation (13) where the circumscribed planes are spaced "uniformly" with respect to the polar angle θ, and are tangent to the p-cone at the polar angles $\theta_i = \frac{\pi i}{2m}$. If $p = 2$, the uniform approximation can be seen as "optimal", since it has the same accuracy at each sector $[\frac{\pi i}{2m}, \frac{\pi(i+1)}{2m}]$, and thus requires the smallest number of facets to achieve a given approximation accuracy. In the case of $p \neq 2$, however, the accuracy of the uniform gradient approximation varies from sector to sector. Thus, it may be of interest to construct an approximation function φ_m that results in a constant accuracy at each sector $[\varphi_m(i), \varphi_m(i + 1)]$ of p-cone, thereby minimizing the number of facets needed to achieve the desired accuracy. On the other hand, if the structure of the problem is such that an optimal solution is known to be located in a certain part of the cone, it might be beneficial to construct an approximation that is more accurate within this particular region and less accurate outside of it. These considerations provide an intuition on how a careful choice of φ_m may reduce the size of the problem in question. In this work, however, we do not discuss the question of constructing an "optimal" approximation, instead focusing on the issues related to solving the polyhedral approximations of pOCP problems.

For $p = 2$ and a given approximation accuracy, the lifted polyhedral approximation (11) due to Ben-Tal and Nemirovski [9] is superior to the gradient polyhedral approximation (13) in terms of dimensionality. However, computational studies [11, 15] indicated that solving polyhedral approximations, either lifted or gradient, of SOCP problems was computationally inefficient comparing to "native" SOCP solution techniques, such as self-dual interior-point methods.

At the same time, the computational efficiency of the polyhedral approximation approach can be substantially improved by employing decomposition methods that exploit the specific structure of polyhedral approximations in (13), whereby the polyhedral approximation approach becomes competitive with SOCP-based solution methods for pOCP problems with $p \neq 2$. This was demonstrated for a special case of the uniform gradient polyhedral approximation [15]. In the next section we show that analogous computational efficiencies can be achieved for more general gradient polyhedral approximations of pOCP problems, as well as for the lifted polyhedral approximation of SOCP problems.

3 Cutting plane methods for polyhedral approximations of SOCP and pOCP problems

Computationally efficient methods for solving polyhedral approximations (5) of SOCP and pOCP problems can be constructed by taking advantage of (i) the special structure of the problem induced by the "tower-of-variables" representation of high-dimensional cones as an intersection of three-dimensional ones in a lifted space, and (ii) the special structures of polyhedral approximations of three-dimensional quadratic or p-order cones.

With respect to (i), a cutting plane method that, given a polyhedral approximation for 3D cones, utilizes the structure of the "tower-of-variables" reformulation in the approximating problems (5), was proposed in [15]. This method is briefly described in Section 3.1 below, since it is necessary in the context of (ii), namely, for exploiting the special properties of gradient and lifted polyhedral approximations of 3D cones for fast cut generation. In particular, the discussion that follows in Sections 3.2 and 3.3 demonstrates that, despite the differences in construction and properties, the lifted Ben-Tal-Nemirovski's approximation (11) of quadratic cones and the gradient approximation (13) of p-cones offer the same computational efficiency for cut generation.

3.1 A cutting plane procedure for polyhedral approximations of pOCP problems

The cutting plane algorithm described here is applicable to reformulations of pOCP problems obtained using the "tower-of-variables" lifting technique (9). Assuming for simplicity that problem (1) contains only one p-cone constraint ($K = 1$) of dimension $N + 1$, the corresponding reformulation of (1) is obtained by lifting the p-cone constraint using the “tower-of-variables” method as

$$\begin{aligned}
\min \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} && && \text{(14a)}\\
\text{s. t.} \quad & \mathbf{A}\mathbf{x} && \le \mathbf{b} && \text{(14b)}\\
& w_{N+j} && \ge \|(w_{2j-1}, w_{2j})\|_p, \quad j = 1,\dots,N-1, && \text{(14c)}\\
& w_j && \ge |(\mathbf{C}\mathbf{x}+\mathbf{e})_j|, \qquad j = 1,\dots,N, && \text{(14d)}\\
& w_{2N-1} && = \mathbf{h}^{\mathsf{T}}\mathbf{x} + f, && \text{(14e)}
\end{aligned}$$

where $\mathbf{w} \in \mathbb{R}^{2N-1}$. Each of the three-dimensional p-order cones (14c) is subsequently replaced by its polyhedral approximation (10), which yields the following polyhedral approximation of pOCP (1):

$$\begin{aligned}
\min \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} && \text{(15a)}\\
\text{s. t.} \quad & \mathbf{H}^{(3)}_{p,m}\begin{pmatrix}\mathbf{w}_j\\ \mathbf{u}_j\end{pmatrix} \ge \mathbf{0}, \quad j = 1,\dots,N-1, && \text{(15b)}\\
& \mathbf{u}_j \in \mathbb{R}^{\kappa_m}_{+}, && \text{(15c)}\\
& \text{(14b), (14d), (14e),} && \text{(15d)}
\end{aligned}$$

where the vectors $\mathbf{w}_j$ stand for the triplets $\mathbf{w}_j = (w_{N+j}, w_{2j-1}, w_{2j})^{\mathsf{T}}$. Constructed in such a way polyhedral approximation of the pOCP problem (1) possesses a special structure that can be exploited for solving the LP problem (15) efficiently. In particular, the following cutting plane representation for (15) was presented [15]:

$$\begin{aligned}
\min \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} && \text{(16a)}\\
\text{s. t.} \quad & w_{N+j} \ge (0,\dots,0,w_{2j-1},w_{2j})\,\hat{\boldsymbol{\pi}}_i, \quad i \in \mathcal{P}_{p,m},\ j = 1,\dots,N-1, && \text{(16b)}\\
& \text{(14b), (14d), (14e),} && \text{(16c)}
\end{aligned}$$

where $\mathcal{P}_{p,m}$ is the set of vertices $\hat{\boldsymbol{\pi}}_i$ of the polyhedron

$$\left\{ \boldsymbol{\pi} \ge \mathbf{0} \;\middle|\; \mathbf{H}^{\mathsf{T}}_{p,m}\boldsymbol{\pi} \le \begin{pmatrix}1\\ \mathbf{0}\end{pmatrix} \right\}, \tag{17}$$

and the matrix $\mathbf{H}_{p,m}$ is obtained by augmenting the approximation matrix $\mathbf{H}^{(3)}_{p,m}$ with two extra rows $(0,1,0\cdots0)$, $(0,0,1,0\cdots0)$, where 1’s correspond to the variables w_{2j-1} and w_{2j}:

$$\mathbf{H}_{p,m} = \begin{pmatrix}\mathbf{H}^{(3)}_{p,m}\\ 0\,1\,0\cdots0\\ 0\,0\,1\cdots0\end{pmatrix}.$$

Constraints (16b) are then generated via an iterative procedure. Assuming that problem (16) is bounded, consider the master problem in the form

$$\begin{aligned}
\min \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} && \text{(18a)}\\
\text{s. t.} \quad & w_{N+j} \ge \varsigma_{j,i}\, w_{2j-1} + \tau_{j,i}\, w_{2j}, \quad i = 1,\dots,C_j,\ j = 1,\dots,N-1, && \text{(18b)}\\
& \text{(14b), (14d), (14e),} && \text{(18c)}
\end{aligned}$$

where $\varsigma_{j,i}$ and $\tau_{j,i}$ stand for the components $\hat{\pi}_{\nu-1}$ and $\hat{\pi}_{\nu}$ of the vector $\hat{\boldsymbol{\pi}} \in \mathbb{R}^{\nu}$, and C_j is the number of constraints generated during preceding iterations. Let $(\mathbf{x}^*,\, \mathbf{w}^*) \in \mathbb{R}^{n+2N-1}$ be an optimal solution of the master (note that if (18) is infeasible, then (16) is infeasible too, and the procedure stops). For each $j = 1,\dots,N-1$, the following LP problem is solved:

$$\zeta_j^* := \max\left\{ \left(0,\dots,0,w^*_{2j-1},w^*_{2j}\right)\boldsymbol{\pi} \;\middle|\; \mathbf{H}^{\mathsf{T}}_{p,m}\boldsymbol{\pi} \le \begin{pmatrix}1\\ \mathbf{0}\end{pmatrix},\ \boldsymbol{\pi} \ge \mathbf{0} \right\}, \tag{19}$$

and it is checked whether the condition

$$w_{N+j}^* \geq \zeta_j^* = w_{2j-1}^* \pi_{\nu-1}^{*(j)} + w_{2j}^* \pi_\nu^{*(j)} \tag{20}$$

holds, where $\boldsymbol{\pi}^{*(j)}$ is an optimal solution of (19). If it does not, a new constraint (18b) is added for the variable w_{N+j} by incrementing the corresponding counter of constraints in (18b): $C_j := C_j + 1$, and setting $\varsigma_{j,i'} = \pi_{\nu-1}^{*(j)}$, $\tau_{j,i'} = \pi_\nu^{*(j)}$ for $i' = C_j$. Upon checking condition (20) for all variables w_{N+j}, $j = 1, \ldots, N-1$, in (18), the master problem (18) is augmented with new constraints and is solved again. If (20) holds for all variables w_{N+j}, and thus no new cuts are generated during an iteration, the current solution $\mathbf{x}^*$, $\mathbf{w}^*$ of the master problem is optimal for the original LP approximation problem (16). In such a way, the described cutting plane procedure obtains an optimal solution, if it exists, of the original LP approximation problem (16) after a finite number of iterations, with, perhaps, some anticycling scheme employed.

3.2 Fast cut generation for gradient approximations of p-order cones

The cutting-plane scheme of Section 3.1 exploits the properties of the "tower-of-variables" representation (9) of high-dimensional p-cones as a nested sequence of 3D p-cones to facilitate solving (large-scale) polyhedral approximations (5). In this section we show that if the gradient polyhedral approximation (13) is used for approximating three-dimensional p-cones in (15), the structure of this approximation can be utilized to achieve significant computational savings, provided that the approximation function φ_m of the gradient polyhedral approximation satisfies a certain computability condition.

Proposition 5 *Consider a polyhedral approximation* (6) *of pOCP problem* (1)*, obtained by reformulating each of the K p-cones in* (1) *using the "tower-of-variables" representation* (9) *and then applying the gradient polyhedral approximation* (13) *with parameter of construction m and approximation function φ_m. Then, if φ_m^{-1} is computable in $O(1)$ time, during an iteration of the cutting plane scheme of Section 3.1 new cuts can be generated in $O\big(\sum_k N_k\big)$ time that is independent of m, where $N_k + 1$ is the dimension of k^{th} p-cone in* (1)*.*

Similarly to Proposition 4, this result strengthens the statement in [15]. We still provide its proof here, since it is necessary for formalizing a subsequent observation in Proposition 6.

Proof of Proposition 5: When the gradient polyhedral approximation (13) is used, the cut-generating problem (19) can be formulated as

$$\max\left\{ \sum_{i=0}^{m} (\alpha_{p,i}\xi_1^* + \beta_{p,i}\xi_2^*)\,\pi_i - \sum_{i=1}^{2} \xi_i^* s_i \;\middle|\; \sum_{i=0}^{m} \pi_i \leq 1,\;\; \pi_0, \ldots, \pi_m \geq 0,\;\; s_1, s_2 \geq 0 \right\}, \tag{21}$$

where the constants ξ_1^* and ξ_2^* stand for the corresponding elements of the current optimal solution $\mathbf{w}^*$ of the master problem: $\xi_1^* = w_{2j-1}^*$, $\xi_2^* = w_{2j}^*$. Disregarding the trivial case of $\xi_1^* = \xi_2^* = 0$, we assume that at least one of these parameters is positive: $\xi_1^* + \xi_2^* > 0$. It is clear that solving (21) amounts to finding a maximum element of the set $\{\alpha_{p,i}\xi_1^* + \beta_{p,i}\xi_2^*\}_{i=0,\ldots,m}$. Namely, if one has

$$i^* \in \arg\max_{i=0,\ldots,m} \{\alpha_{p,i}\,\xi_1^* + \beta_{p,i}\,\xi_2^*\}, \tag{22a}$$

then an optimal solution $\boldsymbol{\pi}^*$ of (21) is given by

$$\pi_i^* = 0,\; i \in \{0, \ldots, m\} \setminus i^*; \quad \pi_{i^*}^* = 1; \quad s_1 = \alpha_{p,i^*}; \quad s_2 = \beta_{p,i^*}. \tag{22b}$$

For fixed $\xi_1^*, \xi_2^* \geq 0$ and $p > 1$, consider the function

$$g(t) = \xi_1^* \frac{\cos^{p-1} t}{(\cos^p t + \sin^p t)^{1-1/p}} + \xi_2^* \frac{\sin^{p-1} t}{(\cos^p t + \sin^p t)^{1-1/p}}, \quad t \in [0, \tfrac{\pi}{2}],$$

with the derivative

$$g'(t) = (p-1)\frac{\sin^{p-1} t \, \cos^{p-1} t}{(\cos^p t + \sin^p t)^{2-1/p}} \left(\frac{-\xi_1^*}{\cos t} + \frac{\xi_2^*}{\sin t} \right).$$

Obviously, for $t \in [0, \frac{\pi}{2}]$ function $g(t)$ is either strictly monotone (when one of ξ_1^*, ξ_2^* is zero) or has a unique global maximum at $t^* = \arctan(\xi_2^*/\xi_1^*)$. Then, for a continuous and strictly increasing approximating function $\varphi_m : [0, m] \mapsto [0, \frac{\pi}{2}]$, the function $g(\varphi_m(\cdot))$ is also either monotone on $[0, m]$ or has a unique maximum at $\varphi_m^{-1}(\arctan(\xi_2^*/\xi_1^*))$. Consequently, if the inverse φ_m^{-1} of the approximating function is computable in $O(1)$ time, the index i^* of a maximum element of the sequence

$$g(\varphi_m(i)) = \xi_1^* \, \alpha_{p,i} + \xi_2^* \, \beta_{p,i}, \quad i = 0, \ldots, m,$$

which defines an optimal solution (22) of cut-generating problem (21), can be determined in $O(1)$ time as

$$i^* \in \arg\max \left\{ \varphi_m^{-1}(0), \lfloor \varphi_m^{-1}(t^*) \rfloor, \lfloor \varphi_m^{-1}(t^*) \rfloor + 1, \varphi_m^{-1}(\tfrac{\pi}{2}) \right\}, \quad \text{where} \quad t^* = \arctan(\xi_2^*/\xi_1^*). \tag{23}$$

Given that each p-cone constraint of order p_k and dimensionality $N_k + 1$ requires $N_k - 1$ such operations, generation of new cuts in problem (18) that employs a gradient polyhedral approximation requires $O\big(\sum_k N_k\big)$ time. □

Remark 4 An example of the approximation function φ_m whose inverse $\varphi_m^{-1}(t)$ is not computable in a constant time for any given $t \in [0, \frac{\pi}{2}]$ can be furnished as $\hat{\varphi}_m(z) = (\hat{\theta}_{i+1} - \hat{\theta}_i)(z - i) + \hat{\theta}_i$ for $i \le z \le i+1, i = 0, \ldots, m$, where $0 \le \hat{\theta}_0 < \hat{\theta}_1 < \ldots < \hat{\theta}_m \le \frac{\pi}{2}$. In other words, it is a piecewise linear function corresponding to some arbitrarily prescribed polar angles $\hat{\theta}_i$, $i = 0, \ldots, m$, that determine locations of the facets of the polyhedral approximation. It is easy to see that evaluation of $\hat{\varphi}_m^{-1}(t')$ for any given t' requires determining k such that $t' \in [\hat{\theta}_k, \hat{\theta}_{k+1}]$, which cannot be generally done in a constant time that is independent of m.

In the case when $\xi_1^*, \xi_2^* > 0$, the index i^* of the cut that may have to be added to the master is given by $\lfloor \varphi_m^{-1}(t^*) \rfloor$ or $\lfloor \varphi_m^{-1}(t^*) \rfloor + 1$. Note that as m increases (and the quality of approximation becomes finer), for any fixed $\xi_1^*, \xi_2^* > 0$ the facets corresponding to $\lfloor \varphi_m^{-1}(t^*) \rfloor$, $\lfloor \varphi_m^{-1}(t^*) \rfloor + 1$ converge to a plane tangent to the cone at the point determined by the polar angle $\theta^* = \arctan(\xi_2^*/\xi_1^*)$, so that the corresponding cut takes the form

$$w_{N+j} \ge w_{2j-1} \frac{\cos^{p-1} \theta^*}{(\cos^p \theta^* + \sin^p \theta^*)^{1-1/p}} + w_{2j} \frac{\sin^{p-1} \theta^*}{(\cos^p \theta^* + \sin^p \theta^*)^{1-1/p}}, \quad \theta^* = \arctan \frac{w_{2j}^*}{w_{2j-1}^*}. \tag{24}$$

In this case, one does not need to solve the cut-generating LP (19) and check condition (20) in order to add the corresponding cut. Namely, for a current solution $\mathbf{w}^*$ of the master, cut (24) is added to the master if the condition

$$\left\| \left(w_{2j-1}^*, w_{2j}^* \right) \right\|_p \le (1 + \epsilon) w_{N+j}^* \tag{25}$$

is not satisfied for the respective $j = 1, \ldots, N-1$. The following proposition formalizes this procedure.

Proposition 6 *Given an instance of pOCP problem* (1) *that satisfies the conditions of Proposition 1, consider a cutting plane scheme for constructing an approximate solution of its lifted reformulation* (14)*, where the master problem has the form* (18)*, and for a given solution* $\mathbf{x}^*, \mathbf{w}^*$ *of the master, cuts of the form* (24) *are added if condition* (25) *is not satisfied for a specific* j*. Assuming that* (18) *is bounded, this cutting plane procedure terminates after a finite number of iterations for any given* $\varepsilon > 0$*, with, perhaps, some anti-cycling scheme applied. In particular, the algorithm is guaranteed to generate at most* $O(\varepsilon^{-1})$ *cutting planes, and in the special case of* $p = 2$ *the described cutting plane algorithm is guaranteed to stop after at most* $O(\varepsilon^{-0.5})$ *iterations.*

Proof: Given $\varepsilon > 0$, let ϵ be the corresponding approximation accuracy of 3D p-cones in (14) due to Proposition 3:

$$\epsilon = (1 + \varepsilon)^{1/\lceil \log_2 N \rceil} - 1 = \lceil \log_2 N \rceil^{-1} \varepsilon + O\big(\varepsilon^2\big), \tag{26}$$

and w^*_{N+j}, w^*_{2j-1}, and w^*_{2j} be the elements of the current solution of the master. We will show that there exists some δ_ϵ such that if θ^*_j is located at an angular distance closer than δ_ϵ from an existing cut, then (24) implies (25), i.e., no new cut can be added within δ_ϵ from an existing one. By (24), for any existing cut at polar angle θ_k the solution of the master should satisfy

$$\begin{aligned} w^*_{N+j} \ge{}& w^*_{2j-1}\frac{\cos^{p-1}\theta_k}{(\cos^p\theta_k+\sin^p\theta_k)^{1-\frac{1}{p}}} + w^*_{2j}\frac{\sin^{p-1}\theta_k}{(\cos^p\theta_k+\sin^p\theta_k)^{1-\frac{1}{p}}} = \left\|(w^*_{2j-1},w^*_{2j})\right\|_p \\ &\times\left(\frac{\cos\theta^*_j}{(\cos^p\theta^*_j+\sin^p\theta^*_j)^{\frac{1}{p}}}\frac{\cos^{p-1}\theta_k}{(\cos^p\theta_k+\sin^p\theta_k)^{1-\frac{1}{p}}} + \frac{\sin\theta^*_j}{(\cos^p\theta^*_j+\sin^p\theta^*_j)^{\frac{1}{p}}}\frac{\sin^{p-1}\theta_k}{(\cos^p\theta_k+\sin^p\theta_k)^{1-\frac{1}{p}}}\right), \end{aligned}$$

where $\theta^*_j = \arctan\dfrac{w^*_{2j}}{w^*_{2j-1}}$. Let $\theta^*_j = \theta_k + \delta$, in which case

$$\begin{aligned} w^*_{N+j} &\ge \left\|(w^*_{2j-1},w^*_{2j})\right\|_p \frac{\cos\delta\,(\cos^p\theta_k+\sin^p\theta_k)+\sin\delta\,(\sin^{p-1}\theta_k\cos\theta_k-\cos^{p-1}\theta_k\sin\theta_k)}{(\cos^p(\theta_k+\delta)+\sin^p(\theta_k+\delta))^{\frac{1}{p}}(\cos^p\theta_k+\sin^p\theta_k)^{1-\frac{1}{p}}} \\ &= \left\|(w^*_{2j-1},w^*_{2j})\right\|_p\big(A(\theta_k,\delta)\cos\delta+B(\theta_k,\delta)\sin\delta\big), \end{aligned} \tag{27}$$

where we denote

$$\begin{aligned} A(\theta_k,\delta) &= \frac{(\cos^p\theta_k+\sin^p\theta_k)^{\frac{1}{p}}}{(\cos^p(\theta_k+\delta)+\sin^p(\theta_k+\delta))^{\frac{1}{p}}}, \\ B(\theta_k,\delta) &= \frac{\sin^{p-1}\theta_k\cos\theta_k-\cos^{p-1}\theta_k\sin\theta_k}{(\cos^p(\theta_k+\delta)+\sin^p(\theta_k+\delta))^{\frac{1}{p}}(\cos^p\theta_k+\sin^p\theta_k)^{1-\frac{1}{p}}}. \end{aligned}$$

As $|\delta|$ approaches zero, the right-hand side in (27) converges uniformly to $\left\|(w^*_{2j-1},w^*_{2j})\right\|_p$. Namely, let $K_0 = \min_\theta\|(\cos\theta,\sin\theta)\|_p = \text{const} > 0$, then

$$\begin{aligned} &\big|A(\theta_k,\delta)\cos\delta+B(\theta_k,\delta)\sin\delta-1\big| \le |B(\theta_k,\delta)|\sin|\delta| + A(\theta_k,\delta)(1-\cos\delta)+|A(\theta_k,\delta)-1| \\ &\le \frac{2}{K_0^p}\sin|\delta| + \frac{1}{K_0}(1-\cos\delta)+\frac{1}{K_0}\Big|\big(\cos^p\theta_k+\sin^p\theta_k\big)^{\frac{1}{p}} - \big(\cos^p(\theta_k+\delta)+\sin^p(\theta_k+\delta)\big)^{\frac{1}{p}}\Big| \\ &\le \frac{2}{K_0^p}\sin|\delta| + \frac{1}{K_0}(1-\cos\delta)+\frac{2}{K_0^p}|\delta| \\ &\le \frac{2\pi}{K_0^p}\left|\frac{\delta}{\pi/2}\right| + \frac{\pi^2}{4K_0}\left|\frac{\delta}{\pi/2}\right|^2 \le \left(\frac{2\pi}{K_0^p}+\frac{\pi^2}{4K_0}\right)\left|\frac{\delta}{\pi/2}\right| =: K_1|\delta|, \end{aligned}$$

where Lagrange's mean value theorem for the function $f(t) = \|(\sin t,\cos t)\|_p$ was utilized, along with the well known facts that $\sin|t| \le |t|$ and $1-\cos t \le t^2$.

Then, for any $\epsilon > 0$ there exists $\delta_\epsilon = \frac{1}{K_1}\frac{\epsilon}{1+\epsilon}$ such that for any θ_k and any $|\delta| \le \delta_\epsilon$ condition (24) implies (25) by $w^*_{N+j} \ge (1-K_1|\delta|)\|(w^*_{2j-1},w^*_{2j})\|_p \ge \frac{1}{1+\epsilon}\|(w^*_{2j-1},w^*_{2j})\|_p$. Hence, no two cuts can be located closer than at an angular distance of δ_ϵ, whereby no more than $\left\lceil\frac{\pi}{2\delta_\epsilon}\right\rceil + 1 = O\big(\epsilon^{-1}\big)$ cuts can be generated. A stronger result holds for $p = 2$, indeed, observe that in this case (27) can be rewritten as

$$\begin{aligned} w^*_{N+j} &\ge w^*_{2j-1}\cos\theta_k + w^*_{2j}\sin\theta_k \\ &= \left\|\big(w^*_{2j-1},w^*_{2j-1}\big)\right\|_2(\cos\theta^*_j\cos\theta_k+\sin\theta^*_j\sin\theta_k) = \left\|\big(w^*_{2j-1},w^*_{2j-1}\big)\right\|_2\cos\delta. \end{aligned} \tag{28}$$

Again, in order for (28) to imply (25), one has to require that $\cos\delta \ge \frac{1}{1+\epsilon}$, or $\cos\delta_\epsilon = \frac{1}{1+\epsilon}$, which implies $\delta_\epsilon = O(\epsilon^{0.5})$. The statement of the proposition then follows immediately from (26). □

Remark 5 The cutting plane procedure outlined in Proposition 6 represents an *exact* solution algorithm for the lifted pOCP problem (14), and, correspondingly, the original pOCP problem (1), in the sense that it does not rely on any particular form of polyhedral approximation once an approximate solution $\mathbf{x}_{\varepsilon_1}$ is obtained with a given accuracy $\varepsilon = \varepsilon_1$, an (improved) solution $\mathbf{x}_{\varepsilon_2}$ can subsequently be constructed by setting new accuracy $\varepsilon = \varepsilon_2 < \varepsilon_1$ and resuming the cutting plane algorithm (i.e., the algorithm does not have to be restarted). In contrast, the cutting plane method of Section 3.1 in this case would require updating the algorithm itself, namely changing the LP problem (19) that is used to generate new cuts. The $O(\varepsilon^{-1})$ iteration complexity of the described method in the case of general $p \neq 2$ is comparable to $O(\varepsilon^{-1})$ iteration complexity of first-order methods for SOCP ([6, 16], see also [5, 17]), while in the $p = 2$ case it improves to $O(\varepsilon^{-0.5})$. Of course, the computational cost per iteration increases, and in the worst case the last iterations would require solving LPs with $O(\varepsilon^{-1})$ (respectively, $O(\varepsilon^{-0.5})$) constraints. In practice, however, the described method terminates within a relatively small number of iterations (see Section 4.2).

3.3 Fast cut generation for lifted polyhedral approximation of second-order cones

In this section we demonstrate that a result analogous to Proposition 5 can be formulated in the case of the lifted approximation (11) due to Ben-Tal and Nemirovski [9], i.e., such an approximation also allows for efficient application of the cut-generation technique.

In accordance with the cutting plane method of Section 3.1, consider the master problem (18) that corresponds to a polyhedral approximation of the SOCP ($p = 2$) version of problem (14), where Ben-Tal and Nemirovski's lifted polyhedral approximation (11) of three-dimensional quadratic cones in the "tower-of-variables" is used. In this case, the coefficients $\varsigma_{j,i}$, $\tau_{j,i}$ in (18b) are found as the simplex multipliers of the first two constraints of the LP problem

$$\begin{aligned}
z_j^* = \min \quad & z && && \text{(29a)}\\
\text{s. t.} \quad & u_0 \geq w_{2j-1}^*, && && \text{(29b)}\\
& v_0 \geq w_{2j}^*, && && \text{(29c)}\\
& u_i = \cos\Big(\frac{\pi}{2^{i+1}}\Big)u_{i-1} + \sin\Big(\frac{\pi}{2^{i+1}}\Big)v_{i-1}, && i = 1,\dots,m, && \text{(29d)}\\
& v_i \geq \Big| -\sin\Big(\frac{\pi}{2^{i+1}}\Big)u_{i-1} + \cos\Big(\frac{\pi}{2^{i+1}}\Big)v_{i-1}\Big|, && i = 1,\dots,m, && \text{(29e)}\\
& u_m \leq z, && && \text{(29f)}\\
& v_m \leq \tan\Big(\frac{\pi}{2^{m+1}}\Big)u_m, && && \text{(29g)}\\
& \mathbf{u}, \mathbf{v}, z \geq 0,
\end{aligned}$$

where w_{2j-1}^*, w_{2j}^* are the components of the optimal solution of the master problem obtained during the most recent iteration. If the optimal value of (29) satisfies $w_{N+j}^* < z_j^*$, then a new cut of the form (18b) is added to the master.

It is important to note that, unlike the gradient polyhedral approximation (13) of p-cones, the lifted approximation (11) of quadratic cones due to Ben-Tal and Nemirovski is constructed *recursively*, where the parameter m represents the recursion counter and controls approximation accuracy. Intuitively, the process of constructing this lifted approximation of a 3D quadratic cone can be visualized as a sequence of "rotations" and "reflections" in $\mathbb{R}^2$. Given a vector (u_0, v_0) in the positive quadrant of the plane, during the first iteration of the recursion it is rotated clockwise by $\pi/4$ around the origin and, if the rotation puts it into the lower half-plane, it is reflected symmetrically about the horizontal axis, resulting in vector (u_1, v_1) that is again in the positive quadrant. During the second iteration, vector (u_1, v_1) is rotated clockwise by $\pi/8$ and reflected symmetrically about the horizontal axis if it falls into the lower half-plane due to the rotation. The resulting vector is designated (u_2, v_2), and so on.

In view of this, as the first step of constructing a $O(1)$ solution algorithm for the dual of (29), we formally show that an optimal solution of (29) can be obtained in $O(m)$ time by applying the above recursion procedure to vector

(w^*_{2j-1}, w^*_{2j}).

To this end, let us denote by (r_i, α_i) the polar coordinates of the pair (u_i, v_i) in (29):

$$r_i = r_i(u_i, v_i) = \|(u_i, v_i)\|_2, \quad \alpha_i = \alpha_i(u_i, v_i) = \arg(u_i, v_i) = \arctan(v_i/u_i).$$

In what follows, we will use notations (u_i, v_i) and (r_i, α_i) interchangeably. Since one can always put $z = u_m$ in (29), the discussion of feasibility and optimality in (29) reduces to that for the pair of vectors $(\mathbf{u}, \mathbf{v}) = (u_0, \dots, u_m; v_0, \dots, v_m)$. First, let us make two observations.

Observation 1 *If* $(\mathbf{u}, \mathbf{v})$ *is feasible for* (29)*, then* $\alpha_i \le \dfrac{\pi}{2^{i+1}}$ *for* $i = 0, \dots, m$.

Proof: Indeed, if for some i_0 one has $\alpha_{i_0} > \dfrac{\pi}{2^{i_0+1}}$, then by (29d)–(29e) $\alpha_{i_0+1} > \dfrac{\pi}{2^{i_0+2}}$, which, by continuation, yields a contradiction with (29g) that requires $\alpha_m \le \dfrac{\pi}{2^{m+1}}$. □

Observation 2 *Given a feasible* $(\mathbf{u}, \mathbf{v})$ *and* $i_0 \in \{1, \dots, m\}$*, a feasible* $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ *can be constructed that satisfies* $(u_i, v_i) = (\tilde{u}_i, \tilde{v}_i)$ *for* $i \le i_0 - 1$ *and* $(\tilde{r}_i, \tilde{\alpha}_i) = \left(\tilde{r}_{i-1}, \left|\tilde{\alpha}_{i-1} - \frac{\pi}{2^{i+1}}\right|\right)$ *for* $i \ge i_0$.

Proof: For this, we only need to verify that (29g) is satisfied for $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})$. Due to Observation 1, one has $\alpha_{i_0-1} \le \dfrac{\pi}{2^{i_0}}$. Thus, by construction $\tilde{\alpha}_{i_0} \le \dfrac{\pi}{2^{i_0+1}}, \tilde{\alpha}_{i_0+1} \le \dfrac{\pi}{2^{i_0+2}}, \dots, \tilde{\alpha}_m \le \dfrac{\pi}{2^{m+1}}$, which is equivalent to (29g). □

With this in mind we can construct an optimal solution to the problem under consideration.

Lemma 1 *An optimal solution for the problem* (29) *can be obtained by setting constraints* (29b)–(29f) *to equalities, or, in other words,* $r^*_0 = \|(w^*_{2j-1}, w^*_{2j})\|$, $\alpha^*_0 = \arg(u_0, v_0)$*, and* $r^*_i = r^*_{i-1}$, $\alpha^*_i = \left|\alpha^*_{i-1} - \frac{\pi}{2^{i+1}}\right|$ *for* $i = 1, \dots, m$.

Proof: For a feasible $(\mathbf{u}, \mathbf{v})$, let k be the largest of those $i \in \{1, \dots, m\}$ for which (29e) is a strict inequality i.e., k is such that constraint (29e) is non-binding for $i = k$ and binding for $i = k+1, \dots, m$. Following Observation 2 with $i_0 = k$, define a feasible $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ which satisfies

$$\begin{aligned}
(\tilde{u}_i, \tilde{v}_i) &= (u_i, v_i), \quad i = 0, \dots, k-1, \\
(\tilde{r}_k, \tilde{\alpha}_k) &= \left(r_{k-1}, \left|\alpha_{k-1} - \frac{\pi}{2^{k+1}}\right|\right), \\
(\tilde{r}_i, \tilde{\alpha}_i) &= \left(\tilde{r}_{i-1}, \left|\alpha_{i-1} - \frac{\pi}{2^{i+1}}\right|\right), \quad i = k+1, \dots, m.
\end{aligned} \tag{30}$$

From the definition of k and (30) it follows that $\alpha_k = \tilde{\alpha}_k + \Delta$, where $\Delta > 0$ due to (29e). By construction, one has

$$r_k = r_{k-1} \frac{\cos \tilde{\alpha}_k}{\cos(\tilde{\alpha}_k + \Delta)} > \tilde{r}_k. \tag{31}$$

Now let us demonstrate that $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ yields at least as good objective value as $(\mathbf{u}, \mathbf{v})$, or in other words, $\tilde{u}_m \le u_m$. Note that the definition of k and (30) immediately imply that

$$u_m = r_m \cos \alpha_m = r_k \cos \alpha_m, \quad \tilde{u}_m = \tilde{r}_m \cos \tilde{\alpha}_m = \tilde{r}_k \cos \tilde{\alpha}_m, \tag{32}$$

and

$$\begin{aligned}
\tilde{\alpha}_m &= \left|\frac{\pi}{2^{m+1}} - \left|\frac{\pi}{2^m} - \dots - \left|\frac{\pi}{2^{k+2}} - \tilde{\alpha}_k\right| \dots\right|\right|, \\
\alpha_m &= \left|\frac{\pi}{2^{m+1}} - \left|\frac{\pi}{2^m} - \dots - \left|\frac{\pi}{2^{k+2}} - \alpha_k\right| \dots\right|\right|.
\end{aligned} \tag{33}$$

Let us consider three cases:

(a) Assume that $\alpha_k = \tilde{\alpha}_k + \Delta \leq \frac{\pi}{2^{m+1}}$. In this case equalities (33) yield the following expressions for α_m and $\tilde{\alpha}_m$:

$$\tilde{\alpha}_m = \frac{\pi}{2^{m+1}} - \tilde{\alpha}_k, \quad \alpha_m = \frac{\pi}{2^{m+1}} - (\tilde{\alpha}_k + \Delta) < \tilde{\alpha}_m,$$

which upon substitution in (32) provide that $u_m \geq \tilde{u}_m$.

(b) Now consider the case of $\tilde{\alpha}_k > \tilde{\alpha}_m$. Successive application of the inequality $||a| - |b|| \leq |a - b|$ to the expressions in (33) yields that $|\alpha_m - \tilde{\alpha}_m| \leq \Delta$, and consequently $\alpha_m \leq \tilde{\alpha}_m + \Delta$. Thus, from (32) one has $u_m = r_k \cos\alpha_m \geq r_k \cos(\tilde{\alpha}_m + \Delta)$. Upon substituting expression (31) for r_k into the last inequality, we obtain

$$u_m \geq r_{k-1} \frac{\cos\tilde{\alpha}_k}{\cos(\tilde{\alpha}_k + \Delta)} \cos(\tilde{\alpha}_m + \Delta) =: f(\Delta).$$

Noting that $f'(\Delta) = r_{k-1}\cos(\tilde{\alpha}_k) \frac{\sin(\tilde{\alpha}_k - \tilde{\alpha}_m)}{\cos^2(\tilde{\alpha}_m + \Delta)} > 0$ for $\tilde{\alpha}_k > \tilde{\alpha}_m$ and $f(0) = \tilde{u}_m$, we can conclude that $u_m \geq f(\Delta) \geq f(0) = \tilde{u}_m$.

(c) Finally, suppose that both conditions of (a) and (b) are not satisfied i.e., $\tilde{\alpha}_k \leq \tilde{\alpha}_m$ and $\tilde{\alpha}_k + \Delta > \frac{\pi}{2^{m+1}}$. Consider, the ratio of u_m and $\tilde{u}_m$ as given by (32), where expressions (31) and (30) are used for r_k and $\tilde{r}_k$, respectively:

$$\frac{u_m}{\tilde{u}_m} = \frac{\cos\tilde{\alpha}_k \, \cos\alpha_m}{\cos\tilde{\alpha}_m \, \cos(\tilde{\alpha}_k + \Delta)}.$$

The above assumption and Observation 1 imply that $\tilde{\alpha}_k \leq \tilde{\alpha}_m$ and $\alpha_m \leq \frac{\pi}{2^{m+1}} < \tilde{\alpha}_k + \Delta$, whence the last equality readily yields $u_m/\tilde{u}_m \geq 1$.

In (a)–(c) we have shown that for feasible $(\mathbf{u}, \mathbf{v})$ such that constraint (29e) is binding for $i = k+1, \ldots, m$, we can construct a feasible solution with at least as good objective and constraint (29e) binding for $i = k, \ldots, m$. Using this claim inductively, we can conclude that for any feasible $(\mathbf{u}, \mathbf{v})$ one can construct a feasible solution for which all constraints in (29e) are satisfied as equalities and which has objective at least as good as $(\mathbf{u}, \mathbf{v})$.

Finally, note that a similar argument can be constructed if (29b) or (29c) are not active. Indeed, the case when $v_0 > w^*_{2j}$ is completely analogous to the case when (29e) is not active. Similarly, if $u_0 > w^*_{2j-1}$, which essentially increases the value of r_0 and reduces the value of α_0 by some δ, let us denote as r'_0, α'_m and u'_m the new values of r_0, α_m, and u_m corresponding to this case. Then we can observe that $u'_m = r'_0 \cos\alpha'_m > r'_0 \frac{\sin\alpha_0}{\sin(\alpha_0 - \delta)} \cos(\alpha_m - \delta)$. Hence, $\frac{u_m}{u'_m} = \frac{\cos\alpha_m \sin(\alpha_0 - \delta)}{\cos(\alpha_m - \delta)\sin\alpha_0} = \frac{\cos\delta - \cot\alpha_0 \sin\delta}{\cos\delta + \tan\alpha_m \sin\delta} < 1$.

Thus, we can observe that the solution, constructed by setting constraints (29b)–(29f) to equalities yields at least as good objective value as any other feasible solution. □

By virtue of Lemma 1, the problem of finding optimal of (29) is reduced to the following: given $\alpha_0 \in [0, \frac{\pi}{2}]$ and $m \geq 1$, determine α_m from the recurrent relations

$$\alpha_i = \left|\alpha_{i-1} - \frac{\pi}{2^{i+1}}\right|, \quad i = 1, \ldots, m. \tag{34}$$

Clearly, this can be done in $O(m)$ time. Below we show that determining α_m from recursion (34) requires $O(1)$ time.

For now, let us assume that $\alpha_0 \neq \frac{i\pi}{2^{m+1}}$. For $k = 1, \ldots, 2^m$, define set $A^{(m)}_k = \left(\frac{(k-1)\pi}{2^{m+1}}, \frac{k\pi}{2^{m+1}}\right)$. Note that by Observation 1, $\alpha_m \in A^{(m)}_1$ for any α_0.

Lemma 2 *If* $\alpha_0 \in A_k^{(m)}$ *and* α_m *is given by* (34)*, then*

$$\alpha_m = \begin{cases} \alpha_0 - \dfrac{(k-1)\pi}{2^{m+1}}, & \textit{if } k \textit{ is even} \\ \dfrac{k\pi}{2^{m+1}} - \alpha_0, & \textit{if } k \textit{ is odd.} \end{cases} \tag{35}$$

Proof: First, note that, by construction, the recursive relation (34) corresponds to the process of rotations and reflections i.e., if we treat α_i as a polar angle, then α_{i+1} is obtained by rotating α_i clockwise by $\dfrac{\pi}{2^{i+1}}$ and then, if the result is in the lower half-plane, reflecting with respect to the horizontal axis. In accordance to (34), a reflection is performed whenever $\alpha_{i-1} - \dfrac{\pi}{2^{i+1}} < 0$, therefore for a given α_0 we can define the number of reflections $\xi^{(m)}(\alpha_0)$ as

$$\xi^{(m)}(\alpha_0) = \left|\left\{i \ : \ \alpha_{i-1} - \frac{\pi}{2^{i+1}} < 0\right\}\right|.$$

Next, note that if $\alpha_0, \beta_0 \in A_k^{(m)}$, then $\xi^{(m)}(\alpha_0) = \xi^{(m)}(\beta_0)$ and, moreover, for any i there exists k_i such that $\alpha_i, \beta_i \in A_{k_i}^{(m)}$. Indeed, by the definition of set $A_k^{(m)}$ we have that $\operatorname{sign}\left(\alpha_0 - \frac{\pi}{4}\right) = \operatorname{sign}\left(\beta_0 - \frac{\pi}{4}\right)$ and thus $\alpha_1, \beta_1 \in A_{k_1}^{(m)}$, where $k_1 = k - 2^{m-1}$ if $k \geq 2^{m-1} + 1$ (no reflection) or $k_1 = 2^{m-1} - k + 1$ if $k \leq 2^{m-1}$ (one reflection). Successively repeating this argument we observe that it holds for any i.

Hence, we can define $\xi_k^{(m)}$ as the number of reflections due to (34) for $\alpha_0 \in A_k^{(m)}$, or $\xi_k^{(m)} = \xi^{(m)}(\alpha_0)$ for any $\alpha_0 \in A_k^{(m)}$. Let us show that if $\alpha_0 \in A_k^{(m)}$, then

$$\alpha_m = \begin{cases} \alpha_0 - \dfrac{(k-1)\pi}{2^{m+1}}, & \text{if } \xi_k^{(m)} \text{ is even,} \\ \dfrac{k\pi}{2^{m+1}} - \alpha_0, & \text{if } \xi_k^{(m)} \text{ is odd.} \end{cases}$$

Using the identity $|a| = a \operatorname{sign} a$, the recursive representation (34) can be written as

$$\alpha_m = \delta_m\Big(\cdots\Big(\delta_2\Big(\delta_1\Big(\alpha_0 - \frac{\pi}{4}\Big) - \frac{\pi}{8}\Big)\cdots - \frac{\pi}{2^{m+1}}\Big) = \alpha_0 \prod_{i=1}^{m} \delta_i - \delta, \tag{36}$$

where

$$\delta_i = \operatorname{sign}\Big(\alpha_{i-1} - \frac{\pi}{2^{i+1}}\Big) \quad \text{and} \quad \delta = \sum_{j=1}^{m} \frac{\pi}{2^{j+1}} \prod_{i=j}^{m} \delta_i.$$

According to the arguments given above, $\prod_{i=1}^{m} \delta_i$ and δ should be the same for all $\alpha_0 \in A_k^{(m)}$. Also note that $\prod_{i=1}^{m} \delta_i = \pm 1$, and for all α_0 we should have $\alpha_m \in \left[0, \dfrac{\pi}{2^{m+1}}\right]$. Suppose that $\prod_{i=1}^{m} \delta_i = 1$, i.e., $\alpha_m = \alpha_0 - \delta$, which is a linear translation of the interval $\left[\dfrac{(k-1)\pi}{2^{m+1}}, \dfrac{k\pi}{2^{m+1}}\right]$. Since the result of the translation should be contained in $\left[0, \dfrac{\pi}{2^{m+1}}\right]$, we have that $\delta = \dfrac{(k-1)\pi}{2^{m+1}}$. Similarly, one can conclude that

$$\delta = \begin{cases} \dfrac{(k-1)\pi}{2^{m+1}}, & \text{if } \prod\limits_{i=1}^{m} \delta_i = 1, \\ -\dfrac{k\pi}{2^{m+1}}, & \text{if } \prod\limits_{i=1}^{m} \delta_i = -1. \end{cases} \tag{37}$$

Now, let us show that

$$|\xi_j^{(m)} - \xi_{j-1}^{(m)}| = 1, \tag{38}$$

or, in other words, parity of $\xi_j^{(m)}$ alternates with j. In order to see this, consider the following inductive argument.

i. Observe that $\xi_1^{(1)} = 1, \xi_2^{(1)} = 0$, i.e., the claim holds for $m = 1$. Indeed, the claim immediately follows from the fact that $\alpha_1 = \left|\alpha_0 - \dfrac{\pi}{4}\right|$ for $m = 1$.

ii. Let $m \geq 2$ and $k \leq 2^{m-1}$, then

$$\xi_k^{(m)} = \xi_{2^m-k+1}^{(m)} + 1. \tag{39}$$

Indeed, $\alpha_0 \in A_k^{(m)}$ with $k \leq 2^{m-1}$ implies that $\alpha_0 < \dfrac{k\pi}{2^{m+1}} \leq \dfrac{\pi}{4}$, and hence $\alpha_1 = \dfrac{\pi}{4} - \alpha_0$, or, equivalently, $\alpha_1 \in A_{2^{m-1}-k+1}^{(m)}$ with one reflection performed. Similarly, for $\alpha_0 \in A_{2^m-k+1}^{(m)}$ with $k \leq 2^{m-1}$ we have that $\alpha_0 > \dfrac{(2^m-k)\pi}{2^{m+1}} \geq \dfrac{\pi}{4}$, whence $\alpha_1 = \alpha_0 - \dfrac{\pi}{4}$ i.e., $\alpha_1 \in A_{2^m-k+1-2^{m-1}}^{(m)} = A_{2^{m-1}-k+1}^{(m)}$, requiring no reflections. Note that both cases $\alpha_0 \in A_{2^m-k+1}^{(m)}$ and $\alpha_0 \in A_k^{(m)}$ result in $\alpha_1 \in A_{2^{m-1}-k+1}^{(m)}$ with the latter requiring one reflection, which means that $\xi_k^{(m)} = \xi_{2^m-k+1}^{(m)} + 1$.

iii. Let $m \geq 2$ and $k \geq 2^{m-1} + 1$, then

$$\xi_k^{(m)} = \xi_{k-2^{m-1}}^{(m-1)}. \tag{40}$$

Similarly to the above, for $k \geq 2^{m-1} + 1$ and $\alpha_0 \in A_k^{(m)}$ it holds that $\alpha_0 > \dfrac{(k-1)\pi}{2^{m+1}} \geq \dfrac{\pi}{4}$, meaning that $\alpha_1 = \alpha_0 - \dfrac{\pi}{4} \in A_{k-2^{m-1}}^{(m)}$ with no reflections. Rewriting (34) as $2\alpha_{i+1} = \left|2\alpha_i - \dfrac{\pi}{2^{i+1}}\right|$, let $\beta_i = 2\alpha_{i+1}$, whence $\beta_0 = 2\alpha_1$ and $\beta_i = \left|\beta_{i-1} - \dfrac{\pi}{2^{i+1}}\right|$, $i = 1, \ldots, m-1$. Then, observing that $\beta_0 \in A_{k-2^{m-1}}^{(m-1)}$, it is easy to see that for $k \geq 2^{m-1} + 1$, the problem of finding β_{m-1} given $\beta_0 \in A_{k-2^{m-1}}^{(m-1)}$ is equivalent to the problem of determining α_m from $\alpha_0 \in A_k^{(m)}$ and, therefore, $\xi_k^{(m)} = \xi_{k-2^{m-1}}^{(m-1)}$.

iv. Now, assume that (38) holds for some $m \geq 1$ and let us show that it also holds for $m + 1$. To this end, consider the value of $\left|\xi_j^{(m+1)} - \xi_{j-1}^{(m+1)}\right|$: if $j > 2^m + 1$ (i.e., (iii) can be used for both j and $j-1$), then from (40) we have that $\left|\xi_j^{(m+1)} - \xi_{j-1}^{(m+1)}\right| = \left|\xi_{j-2^m}^{(m)} - \xi_{j-1-2^m}^{(m)}\right| = 1$. If $j \leq 2^m$ (i.e., (ii) can be used for both j and $j-1$), then from (39) it follows that $\left|\xi_j^{(m+1)} - \xi_{j-1}^{(m+1)}\right| = \left|\xi_{2^{m+1}-j+1}^{(m+1)} - \xi_{2^{m+1}-j+2}^{(m+1)}\right|$. By substituting $j' = 2^{m+1} - j + 2$ we have that $\left|\xi_j^{(m+1)} - \xi_{j-1}^{(m+1)}\right| = \left|\xi_{j'}^{(m+1)} - \xi_{j'-1}^{(m+1)}\right|$, where $j' > 2^m + 1$, which reduces to the previous case. Otherwise, if $j = 2^m + 1$, then from (39) one has $\left|\xi_j^{(m+1)} - \xi_{j-1}^{(m+1)}\right| = \left|\xi_j^{(m+1)} - (\xi_j^{(m+1)} + 1)\right| = 1$. Thus, inductively we observe that (38) holds for any m.

Finally, from (i) and (40) we observe that $\xi_{2^m}^{(m)} = 0$ for all m, thus (38) entails that $\xi_k^{(m)}$ is even iff k is even. □

Lemma 3 *If* $\alpha_0 = \dfrac{k\pi}{2^{m+1}}$, *then the recursive relations* (34) *yield*

$$\alpha_m = \begin{cases} 0, & \textit{if } k \textit{ is odd} \\ \dfrac{\pi}{2^{m+1}}, & \textit{if } k \textit{ is even.} \end{cases} \tag{41}$$

Proof: It is straightforward to see that for $\alpha_0 = \dfrac{\pi}{2}$ recursion (34) yields $\alpha_m = \dfrac{\pi}{2^{m+1}}$. Also observe that α_m defined by the recursion (34) is continuous with respect to α_0. Let $\alpha_0 = \dfrac{k\pi}{2^{m+1}}$, $k < 2^m$ and consider a strictly monotone sequence $\alpha_0^+(n) \downarrow \alpha_0$ with the corresponding sequence $\alpha_m^+(n)$ obtained by the recursion (34). For sufficiently large n we have that $\alpha_0^+(n) \in A_{k+1}^{(m)}$. If k is odd, then by Lemma 2 we have that $\alpha_m^+(n) =$

$\alpha_0^+(n) - \dfrac{k\pi}{2^{m+1}} \to 0$, i.e., by continuity of α_m with respect to α_0, such α_0 yields $\alpha_m = 0$. And if k is even, then $\alpha_m^+(n) = \dfrac{(k+1)\pi}{2^{m+1}} - \alpha_0^+(n) \to \dfrac{\pi}{2^{m+1}}$, i.e., $\alpha_m = \dfrac{\pi}{2^{m+1}}$. □

Based on Lemmas 1 – 3 the following corollary can be formulated.

Corollary 1 *An optimal solution of problem* (29) *can be constructed in a constant* $O(1)$ *time that does not depend on the accuracy of approximation induced by m. Particularly, if* $\alpha_0 = \arg(w_1, w_2)$ *and* $r_0 = \|(w_1, w_2)\|_2$*, then optimal value of* u_m *can be found as* $u_m = r_0 \cos \alpha_m$*, where*

$$\alpha_m = \begin{cases} \alpha_0 - \dfrac{(k-1)\pi}{2^{m+1}}, & \alpha_0 \in \left(\dfrac{(k-1)\pi}{2^{m+1}}, \dfrac{k\pi}{2^{m+1}}\right] \text{ and } k \text{ is even}, \\ \dfrac{k\pi}{2^{m+1}} - \alpha_0, & \alpha_0 \in \left(\dfrac{(k-1)\pi}{2^{m+1}}, \dfrac{k\pi}{2^{m+1}}\right] \text{ and } k \text{ is odd}, \\ \dfrac{\pi}{2^{m+1}}, & \alpha_0 = 0. \end{cases} \tag{42}$$

Now, let us consider the simplex multipliers of (29) that yield new cuts. By Lemma 1 we can equivalently rewrite the problem as

$$\begin{aligned} \min \quad & u_m, && && \text{(43a)} \\ \text{s. t.} \quad & u_0 = w^*_{2j-1}, && && \text{(43b)} \\ & v_0 = w^*_{2j}, && && \text{(43c)} \\ & u_i = \cos\left(\frac{\pi}{2^{i+1}}\right) u_{i-1} + \sin\left(\frac{\pi}{2^{i+1}}\right) v_{i-1}, && i = 1, \dots, m, && \text{(43d)} \\ & v_i = \delta_i \Big(- \sin\left(\frac{\pi}{2^{i+1}}\right) u_{i-1} + \cos\left(\frac{\pi}{2^{i+1}}\right) v_{i-1} \Big), && i = 1, \dots, m, && \text{(43e)} \\ & \mathbf{u}, \mathbf{v} \geq \mathbf{0}, \end{aligned}$$

where

$$\delta_i = \operatorname{sign}\left(- \sin\left(\frac{\pi}{2^{i+1}}\right) u_{i-1} + \cos\left(\frac{\pi}{2^{i+1}}\right) v_{i-1} \right).$$

Note that for given w_1, w_2 these δ_i are constants and coincide with δ_i defined in (36). It is easy to see that, by construction, (43) has only one feasible point, which is an optimal solution for the initial problem (29). Again, we assume that $\delta_i \neq 0$.

Denote by y_i the simplex multipliers for constraints (43b) and (43d), and by t_i the simplex multipliers for constraints (43c) and (43e), the dual problem can be formulated as

$$\begin{aligned} \max \quad & w^*_{2j-1} y_0 + w^*_{2j} t_0 && && \text{(44a)} \\ \text{s. t.} \quad & y_{i-1} - \cos\left(\frac{\pi}{2^{i+1}}\right) y_i + \delta_i \sin\left(\frac{\pi}{2^{i+1}}\right) t_i \leq 0, && i = 1, \dots, m, && \text{(44b)} \\ & t_{i-1} - \sin\left(\frac{\pi}{2^{i+1}}\right) y_i - \delta_i \cos\left(\frac{\pi}{2^{i+1}}\right) t_i \leq 0, && i = 1, \dots, m, && \text{(44c)} \\ & y_m \leq 1, && && \text{(44d)} \\ & t_m \leq 0. && && \text{(44e)} \end{aligned}$$

Lemma 4 *An optimal solution of* (44) *can be found by setting all the constraints to equalities, in which case*

$$\begin{aligned} & y_m = 1, \quad t_m = 0, \\ & y_{i-1} = \cos\left(\frac{\pi}{2^{i+1}} + \delta_i \left(\frac{\pi}{2^{i+2}} + \ldots + \delta_{m-1} \frac{\pi}{2^{m+1}} \right) \ldots \right), \qquad i = 1, \dots, m, \\ & t_{i-1} = \sin\left(\frac{\pi}{2^{i+1}} + \delta_i \left(\frac{\pi}{2^{i+2}} + \ldots + \delta_{m-1} \frac{\pi}{2^{m+1}} \right) \ldots \right), \qquad i = 1, \dots, m. \end{aligned} \tag{45}$$

Proof: Indeed, let $y_m = 1$, $t_m = 0$ and let us set all the constraints to equalities. Then $y_{m-1} = \cos\frac{\pi}{2^{m+1}}$, $t_{m-1} = \sin\frac{\pi}{2^{m+1}}$. Further, from the elementary trigonometry we obtain that

$$\begin{aligned} y_{m-2} &= y_{m-1}\cos\frac{\pi}{2^m} - \delta_{m-1}t_{m-1}\sin\frac{\pi}{2^m} = \cos\left(\frac{\pi}{2^m} + \delta_{m-1}\frac{\pi}{2^{m+1}}\right), \\ t_{m-2} &= y_{m-1}\sin\frac{\pi}{2^m} + \delta_{m-1}t_{m-1}\cos\frac{\pi}{2^m} = \sin\left(\frac{\pi}{2^m} + \delta_{m-1}\frac{\pi}{2^{m+1}}\right). \end{aligned}$$

Inductively we can see that in this case (45) holds. Finally, by comparing primal (43) and dual (44) we observe that by complementary slackness, (45) gives an optimal solution for the dual. □

Recall that in order to construct a new cut we need the values of simplex multipliers for constraints (29b) and (29c) i.e., y_0 and t_0. By Lemma 4, one has $y_0 = \cos\gamma$ and $t_0 = \sin\gamma$, where

$$\gamma = \frac{\pi}{4} + \delta_1\left(\frac{\pi}{8} + \delta_2\left(\frac{\pi}{16} + \ldots + \delta_{m-1}\frac{\pi}{2^{m+1}}\right)\ldots\right).$$

Also note that by duality, $w^*_{2j-1}y_0 + w^*_{2j}t_0 = z^*$, hence $|\gamma - \alpha_0| = \arccos\frac{z^*}{\left\|\left(w^*_{2j-1}, w^*_{2j}\right)\right\|_2}$. Now, by comparing this with Lemma 4 and Corollary 1 it follows that

$$\gamma = \begin{cases} \alpha_0 - \arccos\dfrac{z^*}{\left\|\left(w^*_{2j-1}, w^*_{2j}\right)\right\|_2}, & \alpha_0 \in \left(\dfrac{(k-1)\pi}{2^{m+1}}, \dfrac{k\pi}{2^{m+1}}\right) \text{ and } k \text{ is even}, \\ \alpha_0 + \arccos\dfrac{z^*}{\left\|\left(w^*_{2j-1}, w^*_{2j}\right)\right\|_2}, & \alpha_0 \in \left(\dfrac{(k-1)\pi}{2^{m+1}}, \dfrac{k\pi}{2^{m+1}}\right) \text{ and } k \text{ is odd}. \end{cases} \tag{46}$$

Finally, observe that if $\delta_i = 0$ for some i, then both expressions in (46) can be converted into a part of a feasible solution of the dual (44) and since they yield the same optimal objective value, any can be taken for cut construction. In such a way, we have shown that the following proposition holds.

Proposition 7 *Consider the SOCP version of problem* (1) *with K second-order ($p_k = 2$) cone constraints of dimension N_k+1, and its polyhedral approximation* (6) *obtained by reformulating each second-order cone constraint using the "tower-of-variables" representation* (9) *and applying Ben-Tal-Nemirovski's lifted polyhedral approximation* (11) *with parameter of approximation m to the resulting $N_k - 1$ three-dimensional second-order cones. Then, during an iteration of the cutting plane scheme of Section 3.1, new cuts can be generated in a* constant $O\left(\sum_k N_k\right)$ *time that does not depend on m.*

Remark 6 While the statement of Proposition 7 parallels that of Proposition 5 for gradient polyhedral approximations of p-cones, its significance with respect to Ben-Tal-Nemirovski's lifted polyhedral approximation of quadratic cones is substantially different, due to the fact that Ben-Tal-Nemirovski's approximation is essentially *recursive* in construction. In this sense, Proposition 7 and Lemma 2 provide a "shortcut" method for computing this recursion in a constant time that does not depend on the recursion's depth.

Remark 7 It is well documented [11, 15] that methods based on polyhedral approximations do not generally outperform self-dual interior-point SOCP methods. As such, the new approximate solution method for SOCP problems introduced by Proposition 7 is not expected to be generally superior to interior-point or first-order solution approaches for SOCP [5, 6, 16, 17]. Nevertheless, the proposed cutting-plane procedure for lifted polyhedral approximations of SOCP problems can provide computational advantages in situations that require repetitive solving of a SOCP instance with slight variations in data. In this context, the resulting approximating problem is an LP of a moderate size, and an extensive body of literature on solving such problems can be utilized, including warm-start procedures. As an illustration of this, in the next section we study mixed-integer pOCP (MIpOCP) problems (3). The branch-and-bound framework discussed there relies on repetitive solution of the polyhedral approximation of a continuous relaxation of MIpOCP problem instead of its exact nonlinear formulation, and can benefit significantly from warm start capabilities of the solvers.

4 Numerical experiments

Our interest in solving optimization problems with p-order cone constraints stems from recent developments in risk averse decision making under uncertainty and stochastic optimization. Namely, mathematical programming problems with p-order cone constraints arise naturally in the context of stochastic optimization models whose objective or constraints involve so-called *coherent risk measures* [2] of a special kind. In this case study we focus on stochastic programming models of portfolio optimization with a certain class of coherent risk measures.

4.1 Portfolio optimization with higher moment coherent risk measures

Higher moment coherent risk measures Given a probability space $(\Omega, \mathcal{F}, \mathsf{P})$, let a random outcome X, which represents a cost or a loss, be an element of the linear space $\mathcal{L}_p(\Omega, \mathcal{F}, \mathsf{P})$ of $\mathcal{F}$-measurable functions $X : \Omega \mapsto \mathbb{R}$, where $p \geq 1$. Then, a risk measure $\rho(X)$ can be defined as a mapping $\rho : \mathcal{L}_p \mapsto \mathbb{R}$. In particular, the higher moment coherent risk (HMCR) measures [14], which we focus on in this study, have been defined as optimal values of the following (convex) stochastic programming problem

$$\mathsf{HMCR}_{p,\alpha}(X) = \min_{\eta\in\mathbb{R}} \; \eta + (1-\alpha)^{-1} \big\| [X-\eta]_+ \big\|_p, \quad \alpha \in (0,1), \quad p \geq 1, \tag{47}$$

where $[X]_+ = \max\{0, X\}$ and $\|X\|_p = (\mathsf{E}|X|^p)^{1/p}$. By definition, HMCR measures quantify risk in terms of higher tail moments of loss distribution, which are commonly associated with "risk". HMCR measures possess a number of notable properties, including coherence [2], and isotonicity with respect to the second-order stochastic dominance (SSD), which allows for consistence with the utility theory of von Neumann and Morgenstern [24]. Risk measures (47) are also amenable to efficient incorporation in stochastic programming problems, where outcome X is regarded as a function on the decision vector $\mathbf{x}$ and random event $\omega \in \Omega$: $X = X(\mathbf{x}, \omega)$. Namely, if, traditionally to stochastic programming, it is assumed that the set Ω is discrete and consists of N scenarios, $\Omega = \{\omega_1, \ldots, \omega_N\}$, with the corresponding probabilities $\varpi_1, \ldots, \varpi_N$, then expressions involving HMCR measures, e.g., $\mathsf{HMCR}_{p,\alpha}(X(\mathbf{x}, \omega)) \leq u$, can be implemented via $(N+1)$-dimensional p-order cone constraints. For a detailed discussion of the properties of HMCR measures, see [14].

pOCP portfolio optimization model In the context of portfolio optimization problems, it is customary to define the cost/loss outcome X as the negative rate of return of the portfolio, $X(\mathbf{x}, \omega) = -\mathbf{r}(\omega)^\mathsf{T}\mathbf{x}$, where $\mathbf{x}$ stands for the vector of portfolio weights, and $\mathbf{r} = \mathbf{r}(\omega)$ is the uncertain vector of assets' returns. Then, one may formulate the problem of minimizing the portfolio risk as given by the HMCR measure, subject to the expected return constraint and the budget constraint as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^n_+} \Big\{ \mathsf{HMCR}_{\alpha,p}(-\mathbf{r}^\mathsf{T}\mathbf{x}) \;\Big|\; \mathsf{E}(\mathbf{r}^\mathsf{T}\mathbf{x}) \geq \bar{r}, \; \mathbf{1}^\mathsf{T}\mathbf{x} \leq 1 \Big\}, \tag{48}$$

where $\bar{r}$ is the prescribed level of expected return, $\mathbf{x} \in \mathbb{R}^n_+$ denotes the no-short-selling requirement, and $\mathbf{1} = (1, \ldots, 1)^\mathsf{T}$. If $\mathbf{r}(\omega)$ is discretely distributed, $\mathsf{P}\{\mathbf{r}(\omega) = \mathbf{r}_j\} = \varpi_j$, $j = 1, \ldots, N$, then (48) reduces to pOCP problem with a single p-order cone constraint:

$$\begin{aligned}
\min \quad & \eta + (1-\alpha)^{-1} t \\
\text{s. t.} \quad & t \geq \|\mathbf{w}\|_p, \\
& \mathrm{Diag}\big(\varpi_1^{-1/p}, \ldots, \varpi_N^{-1/p}\big)\, \mathbf{w} + (\mathbf{r}_1, \ldots, \mathbf{r}_N)^\mathsf{T}\mathbf{x} + \mathbf{1}\eta \;\geq\; \mathbf{0}, \\
& \mathbf{x}^\mathsf{T}(\varpi_1\mathbf{r}_1 + \ldots + \varpi_N\mathbf{r}_N) \geq \bar{r}, \\
& \mathbf{1}^\mathsf{T}\mathbf{x} \leq 1, \\
& \mathbf{x} \geq \mathbf{0}, \;\; \mathbf{w} \geq \mathbf{0},
\end{aligned} \tag{49}$$

where $\mathrm{Diag}(a_1, \ldots, a_k)$ denotes the square $k \times k$ matrix whose diagonal elements are equal to $a_1, \ldots, a_k$ and off-diagonal elements are zero.

MIpOCP portfolio optimization models In addition to the convex portfolio optimization model (48), we consider two mixed-integer extensions of (48). One of them is a cardinality-constrained portfolio optimization problem, which allows for no more than M assets in the portfolio, where M is a given constant:

$$\min_{\mathbf{x}\in\mathbb{R}^n_+,\ \mathbf{z}\in\{0,1\}^n} \left\{ \mathsf{HMCR}_{\alpha,p}(-\mathbf{r}^\mathsf{T}\mathbf{x}) \,\middle|\, \mathsf{E}(\mathbf{r}^\mathsf{T}\mathbf{x}) \ge \bar{r},\ \ \mathbf{1}^\mathsf{T}\mathbf{x} \le 1,\ \ \mathbf{x} \le \mathbf{z},\ \ \mathbf{1}^\mathsf{T}\mathbf{z} \le M \right\}, \tag{50}$$

Similarly to (48), formulation (50) represents a 0–1 MIpOCP problem with a single conic constraint. In addition, we consider portfolio optimization with lot-buying constraints, which reflect a common real-life trading policy that assets can only be bought in *lots* of shares (for instance, in multiples of 1,000 shares). In this case, the portfolio allocation problem can be formulated as MIpOCP with a p-order cone constraint,

$$\min_{\mathbf{x}\in\mathbb{R}^n_+,\ \mathbf{z}\in\mathbb{Z}^n_+} \left\{ \mathsf{HMCR}_{\alpha,p}(-\mathbf{r}^\mathsf{T}\mathbf{x}) \,\middle|\, \mathsf{E}(\mathbf{r}^\mathsf{T}\mathbf{x}) \ge \bar{r},\quad \mathbf{1}^\mathsf{T}\mathbf{x} \le 1,\quad \mathbf{x} = \frac{L}{C}\,\mathrm{Diag}(\mathbf{p})\,\mathbf{z} \right\}, \tag{51}$$

where L is the size of the lot, C is the investment capital (in dollars), and vector $\mathbf{p} \in \mathbb{R}^n$ represents the prices of assets.

The following proposition ensures that the introduced portfolio optimization problems with HMCR measures (48)–(51) are amenable to the polyhedral approximation solution approach discussed in the previous sections.

Proposition 8 *If pOCP problem* (49) *is feasible, then it satisfies the approximation conditions* (8) *of Proposition 1. Moreover, the same applies to continuous relaxations of MIpOCP problems* (50) *and* (51).

Proof: Evidently, the strict feasibility condition (8a) can always be satisfied by selecting sufficiently large t and η in (49). To see that (49) is "semibounded" in the sense (8b), note that the only unrestricted variable in the problem is η, but due to the properties of the optimal solution of (47) (see [14]) it can be bounded as $|\eta| \le \max_{j,\mathbf{x}}\{|\mathbf{r}_j^\mathsf{T}\mathbf{x}|\} \le \max_j \|\mathbf{r}_j\|_\infty$. The same arguments apply to relaxations of (50) and (51). □

Implementation and Scenario Data We used the LP and Barrier MIP solvers of IBM ILOG CPLEX 12.2 to obtain solutions to the formulated portfolio optimization problems. All problems were coded in C++ and computations ran on a 3GHz PC with 4GB RAM in Windows XP 32bit environment. The additional details of numerical experiments are discussed in the corresponding subsections below.

In both continuous and discrete portfolio optimization problems, we used historical data for n stocks chosen at random from the S&P500 index. Namely, returns over N consequent 10-day periods starting at a (common) randomized date were used to construct the set of N equiprobable scenarios ($\varpi_j = N^{-1}$, $j = 1, \ldots, N$) for the stochastic vector $\mathbf{r}$. The values of parameters L, C, K, α, and $\bar{r}$ were set as follows: $L = 100$, $C = 100{,}000$, $M = 5, \alpha = 0.9, \bar{r} = 0.005$.

4.2 Cutting plane techniques for the lifted and gradient approximations of SOCP problems

The pOCP formulation (49) of portfolio selection model (48) was used to evaluate the performance of polyhedral approximation-based solution methods discussed in Section 3. Particularly, we were interested in comparing the cutting plane methods for solving gradient ($p = 2$) and lifted polyhedral approximations of SOCP problems that were presented in Sections 3.2 and 3.3, respectively. Recall that the gradient polyhedral approximation, while being applicable to cones of arbitrary order $p \in (1, \infty)$, in the case of $p = 2$ is inferior to Ben-Tal and Nemirovski's lifted polyhedral approximation of second-order cones. At the same time, the results of Sections 3.2 and 3.3 demonstrate that, in the context of the cutting plane scheme of Section 3.1, both types of polyhedral approximations are amenable to generation of cutting planes in a *constant time* that does not depend on the accuracy of approximation. Thus, it was of interest to compare the cutting plane techniques for gradient and lifted approximations of the SOCP ($p = 2$) version of portfolio optimization problem (49).

In particular, four types of solution methods were studied. First, the complete LP formulation of Ben-Tal-Nemirovski's lifted polyhedral approximation of problem (49) with $p = 2$ was solved using CPLEX 12.2 LP solver (referred to as "LP-lifted" below). Second, this polyhedral approximation LP was solved using the cutting plane method of Section 3.1 combined with the fast cut generation technique of Section 3.3 (referred to as "CG-lifted").

Third, the SOCP version of (49) was solved using the "exact" cutting plane method of Proposition 6 (recall that this cutting plane method derives from the corresponding scheme for gradient polyhedral approximation, but does not require a polyhedral approximation problem to be formulated). This method is referred to as "CG-exact".

Lastly, we solved a gradient polyhedral approximation of the SOCP version of (49) using the cutting plane method of Section 3.1 with the fast cut-generation scheme of Section 3.2. The gradient polyhedral approximation was, however, "optimized" in this case to reduce the number of approximating facets as described below, and is referred to as "CG-grad-opt".

Recall that Proposition 3 furnishes an expression for the approximation accuracy ε of $(N+1)$-dimensional p-cone provided that each of the three-dimensional p-cones is approximated with the same accuracy ϵ. It can be shown (see [11]) that in the case of the lifted approximation technique [9] applied to second-order cones, the size of polyhedral approximation can be reduced without sacrificing its accuracy ε by properly selecting the accuracies ϵ_i of 3D cone approximations at each level i of the "tower-of-variables". This approach can also be utilized in the case of lifting procedure (9) for p-cones,

$$\xi_0 = \xi_{2N-1}, \quad \xi_{N+j} \geq \|(\xi_{2j-1}, \xi_{2j})\|_p, \quad j = 1, \ldots, N-1.$$

Particularly, by introducing approximation accuracies for 3D p-cones at each "level" as $\epsilon_1, \epsilon_2, ..., \epsilon_\ell$, where $\ell = \lceil \log_2 N \rceil$, one can observe that

$$\xi_0^p = \xi_{2N-1}^p \geq \frac{\xi_{2N-3}^p}{(1+\epsilon_1)^p} + \frac{\xi_{2N-2}^p}{(1+\epsilon_1)^p} \geq \frac{\xi_{2N-7}^p}{(1+\epsilon_1)^p(1+\epsilon_2)^p} + \frac{\xi_{2N-6}^p}{(1+\epsilon_1)^p(1+\epsilon_2)^p}$$
$$+ \frac{\xi_{2N-5}^p}{(1+\epsilon_1)^p(1+\epsilon_2)^p} + \frac{\xi_{2N-4}^p}{(1+\epsilon_1)^p(1+\epsilon_2)^p} \geq \ldots \geq \frac{\xi_1^p}{\prod_{i=1}^{k_1}(1+\epsilon_i)^p} + \ldots + \frac{\xi_N^p}{\prod_{i=1}^{k_N}(1+\epsilon_i)^p},$$

where once again $k_i \in \{\lceil \log_2 N \rceil - 1, \lceil \log_2 N \rceil\}$ is the number of "levels" in the "tower of variables" on the way from ξ_{2N-1} to ξ_i. Then, the total number of approximation facets can be reduced by solving the following problem:

$$\min_{m_i \in \mathbb{N}_+} \left\{ \sum_{i=1}^{\ell} q_i m_i \,\middle|\, 1 + \varepsilon \geq \prod_{i=1}^{\ell} \Big(1 + \epsilon_i(m_i)\Big) \right\}, \tag{52}$$

where, for a given i, m_i is the number of facets in polyhedral approximation of a 3D p-cone at "level" i, $\epsilon_i = \epsilon_i(m_i)$ is the main term of the corresponding approximation accuracy, and q_i is the number of 3D p-cones thusly approximated. The objective of (52) represents the total number of approximation facets, while the constraint ensures that the desired approximation accuracy ε of the multidimensional p-cone is achieved. A feasible solution to (52) can be obtained analytically by solving its continuous relaxation with relaxed constraint $\sum_{i=1}^{\ell} \epsilon_i(m_i) \leq \ln(1+\varepsilon)$, and then taking $m_i = \lceil m_i^* \rceil$, where m_i^* is the solution of the relaxed problem. This procedure resulted in, on average, a 30% reduction in the number of approximating facets for the uniform gradient polyhedral approximation.

The results are summarized in Table 1, where for each combination of the number of assets n, number of scenarios N, and approximation accuracy ε, the running times are averaged over 20 instances. It has been noted that for the linear programming problems resulting from the lifted approximation, CPLEX Dual Simplex solver performed better on smaller problem instances, while CPLEX Barrier solver was superior on larger instances. Thus, we used the Barrier solver for all instances except for the two smaller problem sizes (the first six rows in Table 1). At the same time, for the cut-generation approaches we used CPLEX Dual Simplex solver (selected by default).

n, N	ε	LP-lifted	CG-lifted	CG-exact	CG-grad-opt
50, 500	10^{-2}	0.43	0.12	0.11	0.10
	10^{-4}	0.63	0.18	0.17	0.14
	10^{-8}	2.77	0.31	0.32	0.32
150, 1500	10^{-2}	1.83	0.96	0.98	0.89
	10^{-4}	3.85	1.24	1.18	1.09
	10^{-8}	16.29	1.67	1.65	1.64
150, 3000	10^{-2}	37.24	1.66	1.29	1.98
	10^{-4}	96.39	5.80	5.03	5.52
	10^{-8}	296.20	15.11	15.63	15.55
200, 5000	10^{-2}	151.91	9.31	10.20	7.46
	10^{-4}	230.21	23.49	22.76	22.87
	10^{-8}	791.41	48.30	47.48	47.08
200, 10000	10^{-2}	320.80	17.93	18.52	17.26
	10^{-4}	624.63	45.96	46.56	45.09
	10^{-8}	$--$	97.13	96.23	96.97
200, 20000	10^{-2}	677.14	31.56	31.15	30.21
	10^{-4}	898.74	85.95	86.43	84.12
	10^{-8}	$***$	195.99	196.20	195.36

Table 1: Average running time (in seconds) for solving portfolio optimization problem (48)–(49) with $p = 2$, where n, N, and ε denote the number of assets, the number of scenarios (dimension of the cone), and the approximation accuracy of the cone constraint, respectively. "LP-lifted" corresponds to solving the full LP resulting from the lifted polyhedral approximation due to [9], "CG-lifted" – solving this LP using cut generation technique of Section 3.3, "CG-exact" – solving SOCP problem using the "exact" cutting plane method of Proposition 6, and "CG-grad-opt" – solving LP resulting from gradient polyhedral approximation with reduced number of facets due to (52) using cut generation of Section 3.2. All running times are averaged over 20 instances. Symbol "$--$" indicates cases when computations exceeded 1 hour time limit, while "$***$" indicates cases for which the solver returned "Out of memory" error.

It follows from Table 1 that the cutting plane technique of Sections 3.1 and 3.3 for solving Ben-Tal-Nemirovski's lifted approximations of SOCP problems ("CG-lifted") provides significant computational improvements over solving the "complete" LP formulation of such approximations ("LP-lifted"). This is consistent with the corresponding findings reported in [15] for uniform gradient polyhedral approximations of pOCP problems. It is also worth noting that the performance of the cutting plane method of Section 3.1 in combination with fast cut generation of Section 3.3 ("CG-lifted") is on par with that of the "exact" cutting plane method of Proposition 6 ("CG-exact"). However, the cutting plane method of Section 3.1 and Section 3.3 for gradient polyhedral approximations with reduced number of facets ("CG-grad-opt") generally works slightly faster than the other two cutting plane methods, though the observed improvement is insignificant. Finally, we note that relatively few iterations of the cutting plane methods were required to reach optimality in the corresponding problems; for instance, in the case of the exact solution method ("CG-exact"), an ε-optimal solution was obtained after an average of 11 to 12 iterations, for $\varepsilon = 10^{-8}$. Interestingly, the number of iterations has exhibited rather little dependence on the problem size: for example, instances with $N = 5{,}000$, $N = 10{,}000$, and $N = 20{,}000$ required an average of 11.2, 11.4, and 11.5 iterations, respectively, to be solved within a 10^{-8} accuracy.

4.3 Polyhedral approximations and cutting plane techniques for rational-order mixed-integer pOCP problems

The approaches to constructing and solving polyhedral approximations of pOCP problems (1) described above, can also be efficiently applied to mixed-integer extensions of pOCP (MIpOCP) (3); in particular, we are considering rational-order MIpOCP problems, i.e., instances (3) where all p_k are rational: $p_k = r_k/s_k$.

The existing literature on mixed-integer programming problems with conic constraints is relatively limited, with the majority of research in this area being focused on solving mixed-integer problems with self-dual cone constraints, particularly second-order cone and semidefinite cone constraints. Mixed-integer second order conic programming problems of type (3) with $p = 2$ have recently been studied in [3, 4, 10, 22] and some others. Particularly, Çezik and Iyengar [10] discuss application of Chvátal-Gomory and disjunctive cuts for 0-1 conic programming. Vielma et al. [22] proposed a branch-and-bound algorithm for mixed-integer second-order cone programming (MISOCP) problems that allows for significant computational savings by employing Ben-Tal–Nemirovski's lifted polyhedral approximation of the SOCP relaxation at each node of the branch-and-bound tree instead of solving the nonlinear SOCP relaxation itself, which is only invoked when an integer-valued solution of the polyhedral approximation is found, and is used to declare incumbent or branch further. Atamtürk and Narayanan [3, 4] developed mixed-integer rounding cuts for MISOCP problems, as well as lifted cuts for general mixed-integer cone programming problems, which were then applied to derive lifted cuts for 0-1 MISOCP problems. These techniques were extended to the case of general MIpOCP problems with $p \neq 2$ in [23]. In another recent work by Belotti et al. [7], nonlinear disjunctive conic cuts for MISOCP problems were proposed.

In this study of MIpOCP problems (3), we follow the approach of Vielma et al. [22], i.e., instead of solving a nonlinear pOCP relaxation of (3) at each node i of the branch-and-bound tree,

$$\begin{aligned} \min \quad & \mathbf{c}^\mathsf{T}\mathbf{x} + \mathbf{d}^\mathsf{T}\mathbf{z} \\ \text{s. t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} \le \mathbf{b}, \\ & \left\| \mathbf{C}^{(k)}\mathbf{x} + \mathbf{D}^{(k)}\mathbf{z} + \mathbf{e}^{(k)} \right\|_{p_k} \le \mathbf{h}^{(k)\mathsf{T}}\mathbf{x} + \mathbf{g}^{(k)\mathsf{T}}\mathbf{z} + f^{(k)}, \quad k = 1, \dots, K, \\ & \mathbf{x} \in \mathbb{R}^n, \quad \underline{\mathbf{z}}^{(i)} \le \mathbf{z} \le \overline{\mathbf{z}}^{(i)}, \end{aligned} \tag{53}$$

we solve its polyhedral approximation

$$\begin{aligned} \min \quad & \mathbf{c}^\mathsf{T}\mathbf{x} + \mathbf{d}^\mathsf{T}\mathbf{z} \\ \text{s. t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} \le \mathbf{b}, \\ & \mathbf{H}_{p_k, m_k}^{(N_k+1)} \begin{pmatrix} \mathbf{C}^{(k)}\mathbf{x} + \mathbf{D}^{(k)}\mathbf{z} + \mathbf{e}^{(k)} \\ \mathbf{h}^{(k)\mathsf{T}}\mathbf{x} + \mathbf{g}^{(k)\mathsf{T}}\mathbf{z} + f^{(k)} \\ \mathbf{w}^{(k)} \end{pmatrix} \ge \mathbf{0}, \quad k = 1, \dots, K, \\ & \mathbf{x} \in \mathbb{R}^n, \quad \underline{\mathbf{z}}^{(i)} \le \mathbf{z} \le \overline{\mathbf{z}}^{(i)}, \end{aligned} \tag{54}$$

where $\underline{\mathbf{z}}^{(i)}$, $\overline{\mathbf{z}}^{(i)}$ are the lower and upper bounds on the relaxed values of variables $\mathbf{z}$, and the approximation matrix $\mathbf{H}_{p_k, m_k}^{(N_k+1)}$ is constructed using lifting procedure (9) and applying gradient polyhedral approximation (13) to the resulting 3D p-cones. In particular, we employ the fast cutting plane scheme for polyhedral gradient approximation presented in Section 3.2 to solve the LP problem (54) at each node of the tree.

Only when an integer-valued solution of (54) is found, in order to check its feasibility with respect to the exact nonlinear formulation (3) and declare incumbent or branch further, the exact pOCP relaxation (53) of MIpOCP must be solved with bounds on the relaxed values of variables $\mathbf{z}$ determined by the integer-valued solution in question (see [22] for details). To solve the pOCP relaxation (53) exactly, we reformulate (53) in the SOCP form by representing p-order cone constraints via a set of second-order cones. Such a representation is available for rational-order cones (see, e.g., [1, 8, 20]), but it is generally non-unique and requires $O(N \log r)$ three-dimensional rotated quadratic cones to represent $(N + 1)$-dimensional p-cone with $p = r/s$ [15]. We use the "economical" SOCP representation of rational-order cones due to Morenko et al. [18], which allows for replacing an (r/s)-cone in $\mathbb{R}^{N+1}$ with exactly $\lceil \log_2 r \rceil N$ quadratic cones; in application to (53) with $p_k = r_k/s_k$ it yields a SOCP problem of the form

$$\begin{aligned}
\min \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} + \mathbf{d}^{\mathsf{T}}\mathbf{z} \\
\text{s. t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} \le \mathbf{b}, \\
& \begin{pmatrix} \mathbf{C}^{(k)}\mathbf{x} + \mathbf{D}^{(k)}\mathbf{z} + \mathbf{e}^{(k)} \\ \mathbf{h}^{(k)\mathsf{T}}\mathbf{x} + \mathbf{g}^{(k)\mathsf{T}}\mathbf{z} + f^{(k)} \\ \mathbf{w}^{(k)} \end{pmatrix} \in \mathcal{S}^{N_k}_{r_k/s_k}, \quad k = 1, \dots, K, \\
& \mathbf{x} \in \mathbb{R}^n, \ \ \underline{\mathbf{z}} \le \mathbf{z} \le \overline{\mathbf{z}},
\end{aligned} \tag{55}$$

where $\mathcal{S}^{N_k}_{r_k/s_k}$ is a set of $N_k \lceil \log_2 r_k \rceil$ "rotated" quadratic three-dimensional cones of the form $\xi_0^2 \le \xi_1 \xi_2$ that is equivalent to the original $(N_k + 1)$-dimensional p_k-cone.

In summary, the proposed branch-and-bound method for MIpOCP problems relies primarily on a polyhedral approximation (54) of the problem's continuous relaxation that is solved using the fast cutting plane generation technique. Additionally, a SOCP solver is called to obtain an exact solution of the SOCP reformulation (55) of the MIpOCP relaxation when a new incumbent solution is found. Alternatively, the exact cutting-plane algorithm described in Proposition 6 can be used to solve the MIpOCP relaxation (53) for each new incumbent solution. In our computational experiments, the choice of one or the other exact solution method did not have a noticeable effect on the overall performance, since the bulk of the computational time is spent at the non-integer nodes of the branch-and-bound tree, and calls to an exact solver were made only occasionally.

The described polyhedral approximation-based approach to solving MIpOCP problems was coded in C++ using CPLEX Concert Technology. In particular, the cutting plane scheme for solving the polyhedral approximation (54) of the relaxation (53) of the MIpOCP problem was implemented using CPLEX's callback functionality, and the SOCP reformulation (55) of (53) was solved using CPLEX Barrier solver.

The computational performance of this algorithm (referred to as BnB/CP below) was compared to that of the standard CPLEX 12.2 MIP Barrier solver, which was employed to solve MIpOCP problems in the SOCP reformulation:

$$\begin{aligned}
\min \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} + \mathbf{d}^{\mathsf{T}}\mathbf{z} \\
\text{s. t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} \le \mathbf{b}, \\
& \begin{pmatrix} \mathbf{C}^{(k)}\mathbf{x} + \mathbf{D}^{(k)}\mathbf{z} + \mathbf{e}^{(k)} \\ \mathbf{h}^{(k)\mathsf{T}}\mathbf{x} + \mathbf{g}^{(k)\mathsf{T}}\mathbf{z} + f^{(k)} \\ \mathbf{w}^{(k)} \end{pmatrix} \in \mathcal{S}^{N_k}_{r_k/s_k}, \quad k = 1, \dots, K, \\
& \mathbf{x} \in \mathbb{R}^n, \ \ \mathbf{z} \in \mathbb{R}^m,
\end{aligned}$$

where, as before, $\mathcal{S}^{N_k}_{r_k/s_k}$ denotes the set of second-order cones equivalent to a $(N_k + 1)$-dimensional (r_k/s_k)-cone constructed in accordance with [18].

Namely, the BnB/CP algorithm and CPLEX MIP Barrier solver were applied to MIpOCP problems with $p = 3.0$ in the form of portfolio optimization with cardinality constraints (50) and lot-buying constraints (51) of various sizes (number of integer variables $n = 50, 100, 200$, dimensionality of p-cone $N = 250, \dots, 1500$). The results are summarized in Tables 2 and 3, respectively, where the running times are averaged over 20 instances. Observe that in the case of cardinality-constrained portfolio optimization problems, the proposed BnB/CP method is inferior to the standard CPLEX MIP Barrier solver on smaller instances, and outperforms it on larger instances. This trend is confirmed by the numerical experiments on portfolio optimization problems with lot-buying constraints, which are generally harder to solve than the cardinality-constrained problems. In this latter case, the BnB/CP method dominates the standard CPLEX MIP Barrier solver on all problem instances. Moreover, it is important to point out that CPLEX 12.2 employs its own polyhedral approximations of second-order cones for solving MISOCP problems, and the results presented in Tables 2 and 3 demonstrate the contribution of the proposed fast cutting plane techniques for solving the polyhedral approximations of conic programming problems.

Note that the chosen value of the parameter $p = 3.0$ in (50) and (51) provided for conditions in which the SOCP reformulation approach would be most competitive with the proposed BnB method. In accordance with the above, the value of $p = 3$ allows for the smallest number, $\lceil \log_2 3 \rceil N = 2N$, of quadratic cones in the SOCP reformulation when $p = r/s \neq 2$. Larger number of quadratic cones in MISOCP reformulations of rational-order MIpOCP generally lead to longer solution times, while the size of polyhedral approximations used in the proposed BnB method does not depend on p, resulting in relatively constant solution times.

	$n = 50$		$n = 100$		$n = 200$	
N	Barrier MIP	BnB/CP	Barrier MIP	BnB/CP	Barrier MIP	BnB/CP
250	**8.43**	11.96	**13.12**	14.56	**21.45**	32.90
500	**11.67**	15.43	37.68	**36.79**	**60.11**	65.87
1000	**12.77**	19.58	38.18	**35.40**	89.36	**75.81**
1500	**33.80**	47.01	107.27	**92.63**	284.44	**190.46**

Table 2: Average running times (in seconds) for BnB/CP implementation of portfolio optimization problem with cardinality constraint (50) and $p = 3.0$, benchmarked against IBM ILOG CPLEX 12.2 MIP Barrier solver applied to SOCP reformulation of (50). Better running times are highlighted in bold.

	$n = 50$		$n = 100$		$n = 200$	
N	Barrier MIP	BnB/CP	Barrier MIP	BnB/CP	Barrier MIP	BnB/CP
250	38.46	**27.91**	114.77	**82.92**	1020.84	**743.22**
500	99.41	**55.17**	339.63	**254.41**	2163.89	**1196.76**
1000	586.51	**506.10**	2666.62	**2395.59**	1.99%	**1.18%**

Table 3: Average running times (in seconds) for BnB/CP implementation of portfolio optimization problem with lot-buying constraints (51) and $p = 3.0$, benchmarked against IBM ILOG CPLEX 12.2 MIP Barrier solver applied to SOCP reformulation of (51). Better running times are highlighted in bold, and XX% denotes the integrality gap after 1 hour.

5 Conclusions

In this paper we discussed the use of polyhedral approximations in the context of solving linear and mixed-integer programming problems with p-order cone constraints. In particular, we showed that the fast cutting-plane method for solving pOCP problems originally proposed by Krokhmal and Soberanis [15] for a special case of gradient approximation of p-cones, which allows for cut generation in a constant time independent of the approximation accuracy, can be extended to a broader class of polyhedral approximations. Moreover, a variation of this approach is proposed that constitutes an exact pOCP solution method with $O(\varepsilon^{-1})$ iteration complexity. In addition, we show that generation of cutting planes in a time that is independent of the approximation accuracy is available for the lifted polyhedral approximation of second-order cones due to Ben-Tal and Nemirovski [9], which is itself recursively constructed, with the number of recursion steps being dependent on the desired accuracy. Finally, it is demonstrated that the developed cutting plane techniques can be effectively applied to obtain exact solutions of mixed-integer p-order cone programming problems.

6 Acknowledgements

This work was supported in part by AFOSR grant FA9550-12-1-0142 and NSF grant EPS1101284.

References

[1] F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Math. Program. 95 (2003), pp. 3–51, Available at `http://dx.doi.org/10.1007/s10107-002-0339-5`.

[2] P. Artzner, F. Delbaen, J.M. Eber, and D. Heath, *Coherent measures of risk*, Math. Finance 9 (1999), pp. 203–228, Available at `http://dx.doi.org/10.1111/1467-9965.00068`.

[3] A. Atamtürk and V. Narayanan, *Conic mixed-integer rounding cuts*, Math. Program. 122 (2010), pp. 1–20, Available at `http://dx.doi.org/10.1007/s10107-008-0239-4`.

[4] A. Atamtürk and V. Narayanan, *Lifting for conic mixed-integer programming*, Math. Program. 126 (2011), pp. 351–363, Available at `http://dx.doi.org/10.1007/s10107-009-0282-9`.

[5] N.S. Aybat and G. Iyengar, *Unified approach for minimizing composite norms*, Math. Program. (2012), Available at `http://arxiv.org/abs/1005.4733`.

[6] N.S. Aybat and G. Iyengar, *An augmented lagrangian method for conic convex programming*, Working paper (2013), Available at `http://arxiv.org/abs/1302.6322`.

[7] P. Belotti, J. Goez, T. Polik I.and Ralphs, and T. Terlaky, *A conic representation of the convex hull of disjunctive sets and conic cuts for integer second order cone optimization*, Working paper (2012).

[8] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPS/SIAM Series on Optimization, Vol. 2, SIAM, Philadelphia, PA, 2001.

[9] A. Ben-Tal and A. Nemirovski, *On polyhedral approximations of the second-order cone*, Math. Oper. Res. 26 (2001), pp. 193–205, Available at `http://dx.doi.org/10.1287/moor.26.2.193.10561`.

[10] M.T. Çezik and G. Iyengar, *Cuts for mixed 0-1 conic programming*, Math. Program. 104 (2005), pp. 179–202, Available at `http://dx.doi.org/10.1007/s10107-005-0578-3`.

[11] F. Glineur, *Computational experiments with a linear approximation of second order cone optimization*, Tech. Rep. 0001, Service de Mathématique et de Recherche Opérationnelle, Faculté Polytechnique de Mons, Mons, Belgium, 2000.

[12] F. Glineur and T. Terlaky, *Conic formulation for l_p-norm optimization*, J. Optim. Theory Appl. 122 (2004), pp. 285–307, Available at `http://dx.doi.org/10.1023/B:JOTA.0000042522.65261.51`.

[13] V. Kaibel and K. Pashkovich, *Constructing extended formulations from reflection relations*, in *Integer programming and combinatorial optimization*, Lecture Notes in Comput. Sci., Vol. 6655, Springer, Heidelberg, 2011, pp. 287–300, Available at `http://dx.doi.org/10.1007/978-3-642-20807-2_23`.

[14] P.A. Krokhmal, *Higher moment coherent risk measures*, Quant. Finance 7 (2007), pp. 373–387, Available at `http://dx.doi.org/10.1080/14697680701458307`.

[15] P.A. Krokhmal and P. Soberanis, *Risk optimization with p-order conic constraints: A linear programming approach*, European J. Oper. Res. 201 (2010), pp. 653–671, Available at `http://dx.doi.org/10.1016/j.ejor.2009.03.053`.

[16] G. Lan, Z. Lu, and R.D.C. Monteiro, *Primal-dual first-order methods with $\mathcal{O}(1/\epsilon)$ iteration-complexity for cone programming*, Math. Program. 126 (2011), pp. 1–29, Available at `http://dx.doi.org/10.1007/s10107-008-0261-6`.

[17] G. Lan and R.D.C. Monteiro, *Iteration-complexity of first-order penalty methods for convex programming*, Math. Program. 138 (2013), pp. 115–139, Available at `http://dx.doi.org/10.1007/s10107-012-0588-x`.

[18] Y. Morenko, A. Vinel, Z. Yu, and P. Krokhmal, *On p-norm linear discrimination*, European J. Oper. Res. 231 (2013), pp. 784–789.

[19] Y. Nesterov, *Towards non-symmetric conic optimization*, Optim. Methods Softw. 27 (2012), pp. 893–917.

[20] Y.E. Nesterov and A. Nemirovski, *Interior Point Polynomial Algorithms in Convex Programming*, Studies in Applied Mathematics, Vol. 13, SIAM, Philadelphia, PA, 1994.

[21] T. Terlaky, *On l_p programming*, European J. Oper. Res. 22 (1985), pp. 70–100, Available at `http://dx.doi.org/10.1016/0377-2217(85)90116-X`.

[22] J.P. Vielma, S. Ahmed, and G.L. Nemhauser, *A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs*, INFORMS J. Comput. 20 (2008), pp. 438–450, Available at `http://dx.doi.org/10.1287/ijoc.1070.0256`.

[23] A. Vinel and P. Krokhmal, *On valid inequalities for mixed integer p-order cone programming*, Journal of Optimization Theory and Applications (2013), Available at `http://dx.doi.org/10.1007/s10957-013-0315-7`.

[24] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, 1953rd ed., Princeton University Press, Princeton, NJ, 1944.

[25] G. Xue and Y. Ye, *An efficient algorithm for minimizing a sum of p-norms*, SIAM J. Optim. 10 (2000), pp. 551–579, Available at `http://dx.doi.org/10.1137/S1052623497327088`.

On risk-averse maximum weighted subgraph problems

Maciej Rysz, Mohammad Mirghorbani, Pavlo Krokhmal, Eduardo L. Pasiliao

December 15, 2013

Abstract

In this work, we consider a class of risk-averse maximum weighted subgraph problems (R-MWSP). Namely, assuming that each vertex of the graph is associated with a stochastic weight, such that the joint distribution is known, the goal is to obtain a subgraph of minimum risk satisfying a given hereditary property. We employ a stochastic programming framework that is based on the formalism of modern theory of risk measures in order to find minimum-risk hereditary structures in graphs with stochastic vertex weights. The introduced form of risk function for measuring the risk of subgraphs ensures that optimal solutions of R-MWS problems represent maximal subgraphs. A graph-based branch-and-bound algorithm for solving the proposed problems is developed and illustrated on a special case of risk-averse maximum weighted clique problem. Numerical experiments on randomly generated Erdös-Rényi graphs demonstrate the computational performance of the developed branch-and-bound algorithm.

Keywords: Risk-averse maximum weighted subgraph problem, risk-averse maximum clique problem, maximum weight clique problem, stochastic weights, coherent risk measures

1 Introduction and motivation

For decades, network problems with topologically exogenous information have occupied a prominent place in the graph theory and network science literature. A popular class of problems of this type involves finding a subset of minimum or maximum weight and conforming to a prescribed structural property in a graph whose vertices are characterized by deterministic weights [4, 5, 14, 22, 25]. Several influential studies have established a foundation for exact combinatorial solution algorithms for such problems [6, 11, 26]. Most notably, Carraghan and Pardalos [11] developed a backtracking branch-and-bound method for efficiently solving the maximum clique problem by exploiting the hereditary property [30] of complete subgraphs. Many extensions of their work improved upon the process of reducing the search space by using vertex coloring schemes for branching and for obtaining upper bounds on the maximum achievable subgraph order (see, e.g., [10, 17, 29]). Analogous weight-based procedures have also been used when seeking a maximum weight subgraph in the presence of deterministic vertex weights [4, 21, 25].

Significant emphasis has also been placed on network problems with uncertain exogenous information evidenced in various forms that influences the overall topology, flow distribution and costs, etc. Particularly common are considerations of stochastic factors in context of network flow and vehicle routing problems where uncertainties are attributed to arc capacities or node demands [3, 9, 15, 16]. Also, a number of studies examined the effects of probabilistic arc failures in networks [1, 31] and introduced risk-based approaches to minimize the corresponding flow losses [8, 28]. The problem of finding a subset of vertices of maximum cardinality that form a clique with a specified probability, given that edges in the graph can fail with some probabilities, is studied in [23]; a similar approach in application to certain clique relaxations is pursued in [34]. Although uncertainties in most of the aforementioned cases influence decisions related to directed network flows, far less emphasis has been placed on examining decision making regarding optimal subgraph topologies and resource allocation in settings where uncertainties are induced by stochastic factors associated with network vertices.

In this work, we employ a stochastic programming framework that is based on formalism of risk measures [18], and in particular, coherent risk measures [2, 12], in order to find minimum-risk structures in graphs with stochastic vertex weights. Namely, we consider a class of *risk-averse maximum weighted*[1] *subgraph problems* (R-MWSP) that represent a stochastic extension of the so-called maximum weight subgraph problems considered in the literature in the context of hereditary graph-theoretical properties. We propose a graph-based branch-and-bound algorithm for solving problems in the R-MWSP class, which is generally applicable to maximum weight subgraph problems where a subgraph's weight is given by a super-additive function whose evaluation requires solving an optimization problem. As an illustrative example of the proposed concepts, we consider a risk-averse maximum weighted clique problem.

The remainder of the paper is organized as follows. In Section 2 we introduce the general formulation of R-MWS problems and discuss their properties. Section 3 presents solution methods for R-MWSP, including a mathematical programming formulation and a graph-based (combinatorial) branch-and-bound method. Finally, Section 4 considers a numerical case study on solving risk-averse maximum weighted clique problems, where risk is quantified using a class of nonlinear coherent risk measures, in randomly generated graphs with various densities.

2 Risk-averse stochastic maximum vertex problem

Let $G = (V, E)$ be an undirected graph where each vertex $i \in V$ has a positive weight $w_i > 0$. For any subset S of its vertices, let $G[S]$ denote the subgraph of G *induced by* S, i.e., a graph such that any of its vertices i, j are connected by an edge if and only if (i, j) is an edge in G.

Property Π is said to be *hereditary with respect to induced subgraphs* (*hereditary* for short) if for any graph satisfying Π the removal of a vertex preserves Π in the resulting induced subgraph. Examples of hereditary properties include *"complete"*; *"independent"*, or *"stable"*; *"degree constrained"*; *"planar"*, etc. Given a hereditary property Π, it may be of interest to find a subgraph of G that satisfies Π and has the largest additive weight, which is known as the *maximum weight subgraph problem*, or the *maximum weight* Π *problem*:

$$\max_{S \subseteq V} \Big\{ \sum_{i \in S} w_i \, : \, G[S] \text{ satisfies } \Pi \Big\}. \tag{1}$$

A subgraph of G that satisfies Π and whose order cannot be further increased without violating Π is known as a *maximal* Π*-subgraph*; the largest such subgraph represents the *maximum* Π*-subgraph*. Obviously, an optimal solution of the maximum weight Π problem (1) is necessarily a maximal Π-subgraph, but may not be its maximum Π-subgraph.

[1] The rationale for the chosen terminology is explained in Remark 1.

Finding subgraphs of maximum weight with hereditary properties represents a large and important class of graph theoretical problems. A seminal result regarding maximum subgraph problems with hereditary properties was established by Yannakakis [33]. Particularly, property Π is called *nontrivial* if it is satisfied by a single-vertex graph and not satisfied by every graph, and is called *interesting* if the order of graphs satisfying Π is unbounded. Then, the following holds:

Theorem 1 (Yannakakis [33]) *If property Π is hereditary with respect to induced subgraphs, nontrivial, and interesting, then the maximum Π problem*

$$\max_{S\subseteq V} \big\{|S| : G[S] \textit{ satisfies } \Pi\big\}$$

is NP-complete.

It is straightforward that the statement of this theorem extends to the version of the maximum weight Π problem (1). Some of the most well known instances of (1) include the maximum weight clique problem (MWCP) and maximum weight independent set problem.

Now we pose the question that served as motivation for the present endeavor: *What if the vertex weights w_i are uncertain?* In this case, extending the deterministic formulation (1) into the stochastic domain is not straightforward and requires additional considerations. Indeed, minimization of the random quantity that is represented by the sum of random weights in (1) is ill-posed in the context of decision making under uncertainty that requires a deterministic optimal solution. Therefore, the sum of stochastic weights in the objective has to be replaced with a statistical functional that utilizes the distributional information about the weights' uncertainties. The traditional stochastic optimization approach, for example, involves seeking the best "expected outcome", which in this setting would translate into maximizing the expected weight of an induced subgraph $G[S]$. It is easy to see, however, that maximization of the expected subgraph weight trivially reduces to the deterministic maximum weight Π formulation with expected vertex weights: $\mathsf{E}\big(\sum_{i\in S} w_i\big) = \sum_{i\in S} \mathsf{E} w_i$.

In this work, we pursue a *risk-averse* approach and consider the problem of finding the subgraph of G that satisfies property Π and has the lowest risk. Namely, let X_i denote a stochastic variable that is associated with vertex $i \in V$ and assume that the joint distribution of vector $\mathbf{X}_G = (X_1, \ldots, X_{|V|})$ is known. Assuming that the random quantities X_i, $i \in V$, represent costs or losses, consider the problem of finding the *minimum-risk* subgraph in G with property Π, or the risk-averse maximum weighted Π problem:

$$\min_{S\subseteq V} \big\{\mathscr{R}(S;\mathbf{X}_G) : G[S] \text{ satisfies } \Pi\big\}. \tag{2}$$

In formulation (2), the functional $\mathscr{R}(S;\mathbf{X}_G)$ quantifies the risk of the induced subgraph $G[S]$ given the distributional information $\mathbf{X}_G$, and is undefined as yet.

In order to formally define the risk $\mathscr{R}(S;\mathbf{X}_G)$ of a subgraph $G[S]$ in (2), we invoke the concept of *risk measure* that is well known in stochastic optimization literature [18]. Namely, given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the set of random events, $\mathcal{F}$ is the σ-algebra, and $\mathbb{P}$ is the probability measure, a risk measure is defined as a mapping $\rho : \mathcal{X} \mapsto \mathbb{R}$, where $\mathcal{X}$ is a linear space of $\mathcal{F}$-measurable functions $X : \Omega \mapsto \mathbb{R}$. This basic definition is typically augmented by additional properties, such as convexity, monotonicity, etc. (see below) that are dictated by applications.

Then, given a risk measure ρ that we additionally assume to be lower semi-continuous (l.s.c.), the risk $\mathscr{R}(S; \mathbf{X}_G)$ of a subgraph of G induced on a set of vertices $S \subseteq V(G)$ with uncertain vertex weights X_i can be defined as an optimal value of the following stochastic programming problem:

$$\mathscr{R}(S; \mathbf{X}_G) = \min \left\{ \rho\Big(\sum_{i \in S} u_i X_i \Big) : \sum_{i \in S} u_i = 1,\ u_i \geq 0,\ i \in S \right\}. \tag{3}$$

Recall that function $f : \mathcal{X} \mapsto \overline{\mathbb{R}}$ is l.s.c. if and only if the sets $\{X \in \mathcal{X} : f(X) \leq a\}$ are closed for all $a \in \mathbb{R}$. Obviously, lower semi-continuity of risk measure ρ is necessary for the minimization problem in (3) to be well-posed. In the sequel, it will be implicitly assumed that the risk measure ρ in (3) is l.s.c.

The rationale behind definition (3) of subgraph risk function $\mathscr{R}(\cdot)$ is that, similarly to many "nice" risk measures, such as those discussed below, it allows for risk reduction through diversification:

Proposition 1 *Given a graph $G = (V, E)$ with stochastic weights X_i, $i \in V$, and a l.s.c. risk measure ρ, the subgraph risk function $\mathscr{R}$ defined by* (3) *satisfies*

$$\mathscr{R}(S_2; \mathbf{X}_G) \leq \mathscr{R}(S_1; \mathbf{X}_G) \quad \textit{for all} \quad S_1 \subseteq S_2. \tag{4}$$

Proof: For $S_1 \subseteq S_2$, denote

$$\mathbf{u}^{(k)} \in \arg\min \left\{ \rho\Big(\sum_{i \in S_k} u_i X_i \Big) : \sum_{i \in S_k} u_i = 1;\ \ u_i \geq 0,\ \ i \in S_k \right\}, \quad k = 1, 2.$$

Then, one immediately has

$$\mathscr{R}(S_2; \mathbf{X}_G) = \rho\Big(\sum_{i \in S_2} u_i^{(2)} X_i \Big) \leq \rho\Big(\sum_{i \in S_1} u_i^{(1)} X_i + \sum_{j \in S_2 \setminus S_1} 0 \cdot X_j \Big) = \mathscr{R}(S_1; \mathbf{X}_G),$$

due to lower semicontinuity of risk measure ρ. □

Note that the power of definition (3) via solution of a stochastic programming problem is evidenced in the fact that the property (4) of risk reduction via diversification property holds for any l.s.c. risk measure $\rho : \mathcal{X} \mapsto \mathbb{R}$. Secondly, property (4) implies the following important observation regarding the optimal solution of the risk-averse maximum weighted Π problem (2):

Corollary 1 *There exists an optimal solution of the risk-averse maximum weighted Π problem* (2) *with $\mathscr{R}(S; \mathbf{X}_G)$ defined by* (3) *that is a maximal Π-subgraph in G.*

Remark 1 The introduced problem (2) of finding minimum-risk subgraphs with risk defined by (3) is strongly related to the class of maximum-weight subgraph problems (1), in the sense that both are concerned with weighted graphs, and their optimal solutions can be represented by maximal subgraphs; however, in contrast to (1), an optimal solution of (2)–(3) is not a subgraph of maximum "weight". To emphasize the similarities and differences with (1), we call the risk-minimization problem (2) a "risk-averse maximum *weighted* subgraph problem".

In this respect, it is worth mentioning that the presented framework differs from other recent studies that also utilized formally defined risk measures for quantifying the risk in graphs, but relied on explicit maximization of the subgraph's cardinality or weight while requiring that its risk be bounded (see, e.g., [23, 34]):

$$\min_{S\subseteq V} \big\{|S| : \mathrm{Risk}(S) \leq c_0,\ G[S] \text{ satisfies } \Pi\big\}.$$

Indeed, the proposed definition (3) of risk function $\mathscr{R}$ in the R-MWS problem (2) implies that maximization of a solution's cardinality is a consequence of risk minimization via diversification.

Further properties of $\mathscr{R}(S;\mathbf{X_G})$ depend on those of the risk measure ρ in (3). In this work we assume ρ to belong to a family of *coherent* measures of risk. According to [2], risk measure ρ is called coherent if it satisfies the following four properties (axioms):

(A1) *monotonicity:* $\rho(X) \leq \rho(Y)$ for all $X, Y \in \mathcal{X}$ such that $X \leq Y$;

(A2) *subadditivity:* $\rho(X + Y) \leq \rho(X) + \rho(Y)$ for all $X, Y \in \mathcal{X}$;

(A3) *positive homogeneity:* $\rho(\lambda X) = \lambda\rho(X)$ for all $X \in \mathcal{X}$ and $\lambda > 0$;

(A4) *transitional invariance:* $\rho(X + a) = \rho(X) + a$ for all $X \in \mathcal{X}$ and $a \in \mathbb{R}$.

An intuitive interpretation of the above axioms is as follows. Axiom (A1) guarantees that lower losses yield lower risk. The sub-additivity axiom (A2) is important in the context of *risk reduction via diversification*. It is also of fundamental significance from the optimization viewpoint, since it yields, together with the positive homogeneity axiom (A3), the all-important convexity property:

$$\rho(\lambda X + (1-\lambda)Y) \leq \lambda\rho(X) + (1-\lambda)\rho(Y) \quad \text{for all} \quad X, Y \in \mathcal{X},\ \ \lambda \in [0, 1].$$

The positive homogeneity property (A3) postulates that losses and risk scale correspondingly. Axiom (A4) ensures that a constant change in X will translate equivalently in risk $\rho(X)$.

The next proposition states that when the risk measure ρ in (3) is coherent, or at least possesses properties (A1), (A3), (A4), then the corresponding subgraph risk function $\mathscr{R}(S;\mathbf{X_G})$ satisfies properties analogous to (A1), (A3), (A4) with respect to the stochastic weights vector $\mathbf{X_G}$.

Proposition 2 *Let $G = (V, E)$ be an undirected graph, and $\mathbf{X_G} = (X_1, \ldots, X_{|V|})$, and $\mathbf{Y_G} = (Y_1, \ldots, Y_{|V|})$ be vectors of stochastic weights whose components are defined on the same linear space $\mathcal{X}$. If the risk measure ρ in* (3) *is l.s.c. and satisfies axioms (A1), (A3), and (A4) of coherency, then for any induced subgraph $G[S]$ the subgraph risk function $\mathscr{R}$ defined in* (3) *satisfies the following properties:*

(G1) $\mathscr{R}(S;\mathbf{X_G}) \leq \mathscr{R}(S;\mathbf{Y_G})$ *for all* $\mathbf{X_G} \leq \mathbf{Y_G}$*;*

(G2) $\mathscr{R}(S;\lambda\mathbf{X_G}) = \lambda\mathscr{R}(S;\mathbf{X_G})$ *for all* $\mathbf{X_G}$ *and* $\lambda > 0$*;*

(G3) $\mathscr{R}(S;\mathbf{X_G} + a\mathbf{1}) = \mathscr{R}(S;\mathbf{X_G}) + a$ *for all* $a \in \mathbb{R}$*;*

where $\mathbf{1}$ *is the vector of ones, and the vector inequality* $\mathbf{X_G} \leq \mathbf{Y_G}$ *is interpreted component-wise.*

Proof: Consider, for example, property (G1). Denoting, as before,

$$\mathbf{u}^Z \in \arg\min \Big\{ \rho\Big(\sum_{i \in S} u_i Z_i \Big) : \sum_{i \in S} u_i = 1; \;\; u_i \geq 0, \;\; i \in S \Big\},$$

we have

$$\mathscr{R}(S; \mathbf{X}_G) = \rho\Big(\sum_{i \in S} u_i^X X_i \Big) \leq \rho\Big(\sum_{i \in S} u_i^Y X_i \Big).$$

On the other hand, from $X_i \leq Y_i$ it follows that

$$\sum_{i \in S} u_i^Y X_i \leq \sum_{i \in S} u_i^Y Y_i,$$

whence

$$\rho\Big(\sum_{i \in S} u_i^Y X_i \Big) \leq \rho\Big(\sum_{i \in S} u_i^Y Y_i \Big) = \mathscr{R}(S; \mathbf{Y}_G).$$

Properties (G2) and (G3) are verified similarly. □

Observe that $\mathscr{R}(S; \mathbf{X}_G)$ does not obey the sub-additivity with respect to the stochastic weights, i.e., in general

$$\mathscr{R}(S; \mathbf{X}_G + \mathbf{Y}_G) \not\leq \mathscr{R}(S; \mathbf{X}_G) + \mathscr{R}(S; \mathbf{Y}_G).$$

With respect to the traditional risk measures $\rho : \mathcal{X} \mapsto \mathbb{R}$, the failure to satisfy the sub-additivity requirement (or, if positive homogeneity also does not hold, the convexity requirement) implies that such a risk measure is ill fitting for risk reduction via diversification. In other words, it is possible that diversification can result in an increased risk exposure, as measured by a non-subadditive (correspondingly, nonconvex) risk measure ρ.

In the context of proposed risk function $\mathscr{R}$ for subgraphs, risk reduction via diversification is already ascertained by (4), which, with respect to the problem of finding a Π-subgraph with the smallest risk, ensures that adding new vertices to the existing feasible solution that satisfies a hereditary property Π is always beneficial, provided that Π is not violated by the addition of new vertices. Yet, under an additional assumption that the stochastic vertex weights have non-negative support, i.e., $\mathbf{X}_G \geq \mathbf{0}$, the subgraph risk function $\mathscr{R}(S; \mathbf{X}_G)$ can be shown to be "set-subadditive". Namely, one has

Proposition 3 *Let the stochastic vertex weights X_i, $i \in V$, of graph $G = (V, E)$ satisfy $X_i \geq 0$, $i \in V$. Then, for any $S_1, S_2 \subseteq V$ the subgraph risk function $\mathscr{R}(S; \mathbf{X}_G)$ defined by* (3) *satisfies*

$$\mathscr{R}(S_1 \cup S_2; \mathbf{X}_G) \leq \mathscr{R}(S_1; \mathbf{X}_G) + \mathscr{R}(S_2; \mathbf{X}_G), \tag{5}$$

provided that the risk measure ρ in (3) *is l.s.c. and satisfies (A1) and (A2).*

Proof: If ρ satisfies axioms (A1) and (A2), then $\rho(X) \geq 0$ for any $X \geq 0$. Immediately, one has $\mathscr{R}(S_1 \cup S_2; \mathbf{X}_G) \leq \mathscr{R}(S_1; \mathbf{X}_G) \leq \mathscr{R}(S_1; \mathbf{X}_G) + \mathscr{R}(S_2; \mathbf{X}_G)$. □

Naturally, in the context of risk-averse maximum weighted Π problems where Π is hereditary, one should also require that S_1, S_2, and $S_1 \cup S_2$ satisfy Π.

Note that the assumption of nonnegative support for vertex weights X_i is analogous to the standard assumption of positive vertex weights in hereditary maximum weight subgraph problems such as the maximum clique and independent set problems [5, 27].

3 Solution approaches for risk-averse maximum weighted subgraph problems

In this section we consider a mathematical programming formulation for the R-MWS Π problem (2), where the risk $\mathscr{R}(S)$ of induced subgraph $G[S]$ is defined as in (3), and propose a graph-based, or combinatorial branch-and-bound algorithm that represents an extension of the well-known branch-and-bound schemes for the maximum clique problem [11, 25, 26].

3.1 A mathematical programming formulation

Given a graph-theoretic property Π, let binary decision variables x_i indicate whether node $i \in V$ belongs to a subset S, such that the induced subgraph $G[S]$ satisfies Π:

$$x_i = \begin{cases} 1, & i \in S \text{ such that } G[S] \text{ satisfies } \Pi \\ 0, & \text{otherwise.} \end{cases}$$

Further, let $\mathbf{\Pi}_G(\mathbf{x}) \leq \mathbf{0}$ denote the *structural constraints* such that for any $\tilde{\mathbf{x}} \in \{0,1\}^{|V|}$, $\mathbf{\Pi}_G(\tilde{\mathbf{x}}) \leq \mathbf{0}$ if and only if $G[\tilde{S}]$ satisfies Π, where $\tilde{S} = \{i \in V : \tilde{x}_i = 1\}$. Then, the following proposition, which we give without proof, formalizes a mathematical programming representation for the risk-averse maximum weighted Π problem (2) with risk $\mathscr{R}(S; \mathbf{X}_G)$ defined by (3) if the property Π is hereditary on induced subgraphs:

Proposition 4 *Let $G = (V, E)$ be an undirected graph with stochastic vertex weights X_i, $i \in V$, and Π be a property hereditary on induced subgraphs. Then, the R-MWS Π problem* (2) *with risk defined by* (3) *can equivalently be represented as a mixed 0–1 programming problem*

$$\begin{aligned} \min \quad & \rho\big(\mathbf{u}^\top \mathbf{X}_G\big) \\ \text{s. t.} \quad & \mathbf{u}^\top \mathbf{1} = 1 \\ & \mathbf{u} \leq \mathbf{x} \\ & \mathbf{\Pi}_G(\mathbf{x}) \leq \mathbf{0} \\ & \mathbf{x} \in \{0,1\}^{|V|}, \quad \mathbf{u} \in \mathbb{R}_+^{|V|}. \end{aligned} \tag{6}$$

When the property Π in (6) denotes graph completeness, one can choose, for example, the well-known *edge formulation* of the maximum clique problem (see, e.g., [27]) to represent the structural constraints in (6) as

$$\big\{\mathbf{x} \in \{0,1\}^{|V|} : \mathbf{\Pi}_G(\mathbf{x}) \leq \mathbf{0}\big\} = \big\{\mathbf{x} \in \{0,1\}^{|V|} : x_i + x_j \leq 1 \ \text{ for all } \ (i,j) \in \overline{E}\big\},$$

where $\overline{E}$ represents the complement edges of graph G, whereby the mathematical programming formulation of the R-MWS clique problem (2)–(3) takes the form

$$\begin{aligned} \min \quad & \rho\Big(\sum_{i\in V} u_i\, X_i\Big) \\ \text{s. t.} \quad & \sum_{i\in V} u_i = 1 \\ & u_i \le x_i, \quad i \in V \\ & x_i + x_j \le 1, \quad (i,j) \in \overline{E} \\ & x_i \in \{0,1\}, \;\; u_i \ge 0, \quad i \in V. \end{aligned} \tag{7}$$

Formulations (6)-(7) allow for handling risk measures ρ whose representations come in the form of mathematical programming problems, and can be solved with appropriate (nonlinear) mixed integer programming solvers.

A combinatorial branch-and-bound algorithm that allows for exploiting the structure of problems (6)-(7) imposed by the underlying graph G is described next.

3.2 A graph-based branch-and-bound algorithm

The combinatorial branch-and-bound (BnB) algorithm works by navigating between "levels" of the BnB tree until a subgraph of G that satisfies property Π and is guaranteed to be of lowest risk as measured by (3) is found. The algorithm starts at level $\ell = 0$ with a partial solution $Q := \emptyset$, incumbent solution $Q^* := \emptyset$, and a global upper bound $L^* := +\infty$ on risk of Q^*. Throughout the algorithm, the partial solution Q contains the vertices in V such that $G[Q]$ has property Π, and set Q^* induces, per Corollary 1, a maximal Π-subgraph whose risk equals L^* in G hitherto.

Within the current branch of the BnB tree, "level" ℓ is associated with the *candidate set* C_ℓ of vertices such that any single vertex of C_ℓ can be added to the current partial solution Q without violating property Π. Branching is performed by removing a *branching* vertex q from C_ℓ and adding it to the partial solution Q. The algorithm is initialized with $C_0 := V$, and, as soon as the partial solution Q is updated after branching at level ℓ, the corresponding candidate set at level $\ell + 1$ is constructed by removing all vertices from C_ℓ whose inclusion in Q would break the property Π, i.e.,

$$C_{\ell+1} := \{i \in C_\ell : G[i \cup Q] \text{ satisfies } \Pi\}. \tag{8}$$

As a result, immediately after branching at level ℓ the cardinality of partial solution set Q is equal to $|Q| = \ell + 1$.

The bounding step of the BnB algorithm involves evaluating the quality of the solution that can be obtained by exploring further the subgraph induced by vertices in $Q \cup C_{\ell+1}$. Observe that an exact approach of directly finding the Π-subgraph with the lowest possible risk that is contained in $G[Q \cup$

$C_{\ell+1}]$ entails solving the following restriction of problem (6):

$$\begin{aligned} \mathscr{R}(Q \cup C_{\ell+1}; \mathbf{X}_G) = \min \quad & \rho\big(\mathbf{u}^\top \mathbf{X}_G\big) \\ \text{s. t.} \quad & \mathbf{u}^\top \mathbf{1} = 1 \\ & \mathbf{u} \le \mathbf{x}, \\ & \mathbf{\Pi}_G(\mathbf{x}) \le \mathbf{0}, \\ & \mathbf{x} \in \{0,1\}^{|V|}, \ \ \mathbf{u} \in \mathbb{R}_+^{|V|}, \\ & x_i = 0, \quad i \in V \setminus (Q \cup C_{\ell+1}). \end{aligned} \tag{9}$$

As (9) is a (nonlinear) mixed 0–1 problem, solving it at every node of the BnB tree is impractical. Instead, a lower bound on the value of $\mathscr{R}(Q \cup C_{\ell+1}; \mathbf{X}_G)$ given by (9) can be computed. However, in contrast to the traditional mixed integer programming approach of constructing a lower bound by relaxing the integrality constraints, we formulate a lower bound problem by completely eliminating the 0-1 variables x_i along with the structural constraints:

$$\begin{aligned} \mathscr{R}(Q \cup C_{\ell+1}; \mathbf{X}_G) \ge \mathcal{L}(Q \cup C_{\ell+1}) := \min \quad & \rho\Big(\sum_{i\in V} u_i \, X_i\Big) \\ \text{s. t.} \quad & \sum_{i\in V} u_i = 1 \\ & u_i = 0, \quad i \in V \setminus (Q \cup C_{\ell+1}) \\ & u_i \ge 0, \quad i \in Q \cup C_{\ell+1}. \end{aligned} \tag{10}$$

Observe that the structural constraints $\mathbf{\Pi}_G(\mathbf{x}) \le \mathbf{0}$ in problem (9) are satisfied by variables $\{x_i : i \in Q\}$ (since $G[Q]$ satisfies Π), as well as by variables $\{x_i : i \in Q \cup j_0\}$ for each $j_0 \in C_{\ell+1}$ (since $G[Q \cup j_0]$ for each vertex j_0 in $C_{\ell+1}$ also satisfies Π, per definition (8) of the candidate set $C_{\ell+1}$). Hence, the corresponding structural constraints are redundant in (9). On the other hand, the structural constraints are not necessarily satisfied by variables $\{x_i : i \in C_{\ell+1}\}$ and $\{x_i : i \in Q \cup C_{\ell+1}\}$, since $G[C_{\ell+1}]$ and $G[Q \cup C_{\ell+1}]$ do not necessarily satisfy Π. Thus, (10) is a relaxation of (9), and, by virtue of Proposition 1, the solution to (10) provides a lower bound on the minimum risk achievable in any Π-subgraph induced on the union of Q with any subset of $C_{\ell+1}$, i.e.,

$$\mathcal{L}(Q \cup C_{\ell+1}) \le \mathscr{R}(Q \cup C_{\ell+1}; \mathbf{X}_G) \le \mathscr{R}(Q \cup S; \mathbf{X}_G) \ \text{ for any } \ S \subseteq C_{\ell+1}.$$

Observe that if $\ell' = \ell + 1$ represents the next level in the BnB tree, and Q' is the corresponding partial solution, then due to the definition (8) of candidate set one has

$$(Q' \cup C_{\ell'+1}) \subseteq (Q \cup C_{\ell+1}),$$

whence the risk $\mathscr{R}(Q \cup C_{\ell+1}; \mathbf{X}_G)$ does not decrease as ℓ increases (or, in other words, as new vertices are added to the partial solution Q and the algorithm proceeds to deeper levels ℓ of the BnB tree). We next show that this observation is an effective bounding criterion to obtain a Π-subgraph of lowest risk in G.

Depending on the computed value of $\mathcal{L}(Q \cup C_{\ell+1})$, the algorithm branches further or prunes/backtracks as follows. If $\mathcal{L}(Q \cup C_{\ell+1}) \ge L^*$, then the vertex q is removed from Q and the corresponding branch of the BnB tree is fathomed due to the fact that there exists no possibility of achieving a reduction in risk by sequential branching/refinement. Further, if $C_\ell \neq \emptyset$, another branching vertex is selected and removed from C_ℓ and added to Q. Otherwise, if $C_\ell = \emptyset$, the algorithm backtracks to level $\ell - 1$.

In the case of $\mathcal{L}(Q \cup C_{\ell+1}) < L^*$ and $C_{\ell+1} \neq \emptyset$, the algorithm proceeds to select a branching vertex q at the next level $\ell + 1$. If $\mathcal{L}(Q \cup C_{\ell+1}) < L^*$ and $C_{\ell+1} = \emptyset$, the subgraph induced by the partial solution Q represents a maximal Π-subgraph in G and is declared as the new incumbent solution, $Q^* := Q$, the global upper bound on risk is updated $L^* := \mathcal{L}(Q \cup C_{\ell+1})$, and the algorithm backtracks to level $\ell - 1$.

With regard to the branching rule, the observed computational performance suggests that branching on a vertex q with the smallest value of $\rho(X_q)$ or $\mathsf{E}X_q$ is most effective. To this end, vertices in the set $C_0 = V$ are pre-sorted during the initialization phase of the algorithm in descending order with respect to their risks $\rho(X_i)$ or expected values $\mathsf{E}X_i$, and then the last vertex in C_ℓ is selected for branching.

The outlined branch-and-bound procedure for R-MWS problems is formalized as Algorithm 1.

Algorithm 1: Graph-based branch-and-bound method for R-MWSP

```
Initialize: $\ell := 0$; $C_0 := V$; $Q := \emptyset$; $Q^* := \emptyset$; $L^* := \infty$;
while (not STOP) do
    if $C_\ell \neq \emptyset$ then
        select a vertex $q \in C_\ell$;
        $C_\ell := C_\ell \setminus q$;
        $Q := Q \cup q$;
        $C_{\ell+1} := \{i \in C_\ell : i \cup Q$ satisfies $\Pi\}$;
        solve $\mathcal{L}(Q \cup C_{\ell+1})$;
        if $\mathcal{L}(Q \cup C_{\ell+1}) < L^*$ then
            if $C_{\ell+1} \neq \emptyset$ then
                $\ell := \ell + 1$;
            else
                $Q^* := Q$;
                $L^* := \mathcal{L}(Q \cup C_{\ell+1})$;
                $Q := Q \setminus q$;
        else
            $Q := Q \setminus q$;
            if $\ell = 0$ then
                STOP
    else
        $\ell := \ell - 1$;
        if $\ell = -1$ then
            STOP
        $Q := Q \setminus q$;
return $Q^*$;
```

Depending on the particular form of risk measure ρ, evaluation of the lower bound by solving the relaxed problem (10) can be relatively expensive and be a major contributor to the overall computational cost of the proposed algorithm. Then, certain efficiencies in computing the lower bound value via (10) can be implemented by taking into account the properties of the subgraph risk function $\mathscr{R}$. Specifically, if at any point $(Q \cup C_{\ell+1}) \subseteq (Q' \cup C')$, where Q' and C' are a partial solution and a candidate set for which the lower bound value $\mathcal{L}(Q' \cup C')$ is known to exceed the current global upper bound, $\mathcal{L}(Q' \cup C') \geq L^*$, then $\mathcal{L}(Q \cup C_{\ell+1}) \geq \mathcal{L}(Q' \cup C') \geq L^*$ due to Proposition 1. The vertex q under consideration is then removed from Q and the corresponding subproblem is fathomed. In practice, however, retaining the list of sets $(Q' \cup C')$ with $\mathcal{L}(Q' \cup C') \geq L^*$ and checking whether the current $Q \cup C_{\ell+1}$ is a subset of some $Q' \cup C'$ has proven computationally expensive for even moderately sized problems, and is most notably exacerbated in graph topologies that contain a large number of maximal Π-subgraphs (for example, when the graph density increases in the context of risk averse maximum weighted clique problem). Therefore, a more modest approach is considered where only the vertices from incumbent solutions Q^* are retained and tested against unfathomed sets $(Q \cup C_{\ell+1})$.

4 Case study: Risk-averse stochastic maximum weighted clique problem with higher moment coherent risk measures

In this section we discuss the computational framework and conduct numerical experiments demonstrating the computational performance of the proposed BnB algorithm when solving the risk-averse maximum weighted clique problem (7). We use a family of higher-moment coherent risk (HMCR) measures that were introduced in [19] as optimal values to the stochastic programming problem of the form

$$\mathrm{HMCR}_{\alpha,p}(X) = \min_{\eta\in\mathbb{R}} \; \eta + (1-\alpha)^{-1} \left\|(X-\eta)^+\right\|_p, \quad \alpha \in (0,1), \quad p \geq 1, \tag{11}$$

where $X^+ = \max\{0, X\}$ and $\|X\|_p = \left(\mathsf{E}|X|^p\right)^{1/p}$. The HMCR measures are nonlinear measures of risk that quantify the risk of loss distribution X via its tail moments, and are particularly suitable for measuring risk in heavy-tailed data. HMCR measures possess a number of important properties, such as coherence, isotonicity with respect to the second-order stochastic dominance, which implies consistency with the expected utility theory, and so on. A popular case of (11), also known as the Conditional Value-at-Risk (CVaR) or Expected Shortfall risk measure, arises when $p = 1$:

$$\mathrm{CVaR}_\alpha(X) = \min_{\eta\in\mathbb{R}} \; \eta + (1-\alpha)^{-1}\mathsf{E}(X-\eta)^+, \quad \alpha \in (0,1). \tag{12}$$

Mathematical programming models containing HMCR measures in the objective or constraints can be formulated using p-order cone constraints. Traditionally to stochastic programming, the set of random events Ω is considered to be discrete, $\Omega = \{\omega_1, \ldots, \omega_N\}$, with the corresponding probabilities $\mathbb{P}(\omega_k) = \pi_k > 0$, such that $\pi_1 + \cdots + \pi_N = 1$. Then, the mathematical programming formulation (7) with risk measure $\rho(X)$ selected as $\mathrm{HMCR}_{p,\alpha}(X)$ takes the form of a mixed integer p-order cone programming

(MIpOCP) problem:

$$\begin{aligned}
\min \quad & \eta + (1-\alpha)^{-1} t \\
\text{s. t.} \quad & t \geq \|(y_1, \ldots, y_N)\|_p \\
& \pi_k^{-1/p} y_k \geq \sum_{i \in V} u_i X_{ik} - \eta, \quad k = 1, \ldots, N \\
& \sum_{i \in V} u_i = 1 \\
& u_i \leq x_i, \ \ i \in V \\
& x_i + x_j \leq 1, \ \ (i,j) \in \overline{E} \\
& x_i \in \{0,1\}, \ u_i \geq 0, \ i \in V; \ \ y_k \geq 0, \ k = 1, \ldots, N,
\end{aligned} \tag{13}$$

where X_{ik} is the realization of the stochastic weight of vertex $i \in V$ under scenario k, $k = 1, \ldots, N$, and the scenario probabilities $\mathbb{P}(X_1 = X_{1k}, \ldots, X_N = X_{Nk}) = \pi_k$. Similarly, the lower bound problem (10) for the combinatorial branch-and-bound algorithm described in the previous section takes the form

$$\begin{aligned}
\mathcal{L}(Q \cup C_{\ell+1}) = \min \quad & \eta + (1-\alpha)^{-1} t \\
\text{s. t.} \quad & t \geq \|y_1, \ldots, y_N\|_p \\
& \pi_k^{-1/p} y_k \geq \sum_{i \in V} u_i X_{ik} - \eta, \quad k = 1, \ldots, N \\
& \sum_{i \in V} u_i = 1 \\
& u_i \geq 0, \ \ i \in Q \cup C_{\ell+1} \\
& u_i = 0, \ \ i \in V \setminus (Q \cup C_{\ell+1}) \\
& y_k \geq 0, \ k = 1, \ldots, N.
\end{aligned} \tag{14}$$

In cases when $p = 1$ or 2, problems (13) and (14) reduce to linear programming (LP) and second order cone programming (SOCP) models, respectively. Both represent well established subjects in optimization, for which a range of efficient solvers exist. However, no efficient long-step self-dual interior point methods exist for solving p-order conic constrained problems when $p \in (1,2) \cup (2,\infty)$ due to the fact that the p-cone is not self-dual in this case. Below we discuss solution methods based on polyhedral approximations of p-order cones and representation of rational-order p-cones via second order cones.

Both these approaches rely on "lifting" a p-order cone into a higher dimensional space by representing it as an intersection of a (large) number of three-dimensional (3D) cones.

In order to construct a polyhedral approximation of p-cone $t \geq \|(y_1, \ldots, y_N)\|_p$, it first can be equivalently represented as a chain of 3D p-cone inequalities of the form [7, 32]:

$$t = y_{2N-1}, \quad y_{N+j} \geq \|(y_{2j-1}, y_{2j})\|_p, \quad j = 1, \ldots, N-1. \tag{15}$$

Then, each 3D p-cone in (15) is replaced with its (outer) gradient polyhedral approximation in the form of $m+1$ circumscribed planes:

$$y_{N+j} \geq y_{2j-1} \frac{\cos^{p-1}\theta_\nu}{(\cos^p\theta_\nu + \sin^p\theta_\nu)^{1-\frac{1}{p}}} + y_{2j} \frac{\sin^{p-1}\theta_\nu}{(\cos^p\theta_\nu + \sin^p\theta_\nu)^{1-\frac{1}{p}}}, \quad \theta_\nu = \frac{\pi\nu}{2m}, \ \nu = 0, \ldots, m. \tag{16}$$

The resulting approximating LP problem can be solved by an efficient cutting plane algorithm that admits generation of cutting planes in a constant time that does not depend on the accuracy of approximation [20, 32].

Alternatively, an exact solution of a p-order cone programming problem can be obtained by means of reformulating it as a SOCP problem in the case when the parameter p is a rational number. For example, in the case of $p = 3$, the p-order cone $t \geq \|(y_1, \ldots, y_N)\|_3$ can be represented via $2N$ rotated 3D quadratic cones [24]:

$$t = z_1 + \ldots + z_N, \quad y_j^2 \leq t v_j, \quad v_j^2 \leq z_j v_j, \quad j = 1, \ldots, N. \tag{17}$$

Both the described polyhedral approximation approach and SOCP reformulation approach have been employed in our implementation of the combinatorial branch-and-bound algorithm of Section 3.2 in the cases when the lower bound problem (14) is nonlinear, i.e., when $p > 1$.

Specifically, a polyhedral approximation of the lower bound problem (14) was solved at each node of the BnB tree instead of the exact the nonlinear problem (14) itself. This allows for a significant reduction in the computational cost of the BnB method, since the warm-start capabilities of LP simplex solvers can be utilized during repeated solving of the approximating LP problem.

The exact solution method that is based on the SOCP reformulation is employed for solving (14) once an incumbent solution is found, and the corresponding optimal value is used to update the global upper bound L^*. Due to the fact that the described polyhedral approximation is an outer approximation, one has

$$\mathcal{L}_{\text{LP}}(Q \cup C_{\ell+1}) \leq \mathcal{L}(Q \cup C_{\ell+1}), \tag{18}$$

where $\mathcal{L}_{\text{LP}}(Q \cup C_{\ell+1})$ is the optimal value given by the polyhedral (LP) approximation of the lower bound problem. This implies that for any $Q \cup C_{\ell'+1}$ containing an incumbent solution Q^*, the following holds

$$\mathcal{L}_{\text{LP}}(Q \cup C_{\ell'+1}) \leq \mathcal{L}_{\text{LP}}(Q^*) \leq \mathcal{L}(Q^*) = L^*,$$

which guarantees the correctness of the BnB algorithm relying on polyhedral approximations. Note, however, that inequality (18) also implies that the use of polyhedral approximations instead of the exact nonlinear formulation of the lower bound problem (14) allows for delayed pruning of "non-promising" branches of the BnB tree in situations when

$$\mathcal{L}_{\text{LP}}(Q \cup C_{\ell+1}) < L^* \leq \mathcal{L}(Q \cup C_{\ell+1}).$$

Still, in our experience, the computational savings due to the use of polyhedral approximations during the BnB procedure greatly outweigh the costs of possible delayed pruning.

Note also that in the special case of $p = 1$, when $\rho(X) = \text{CVaR}_\alpha(X)$, the lower bound problem (14) is an LP problem and thus requires no polyhedral approximation or SOCP reformulation.

4.1 Setup of the numerical experiments and results

The numerical studies of the risk-averse maximum weighted clique problem were conducted on randomly generated Erdös-Rényi graphs [13] of orders $|V| = 50, 100, 150, 200$ and average densities $d = 0.2, 0.5$, and 0.8. The stochastic weights of graphs' vertices were generated as i.i.d. samples from the uniform $U[0,1]$ distribution. In particular, we generated scenario sets with $N = 50, 100, 200, 500, 1000$ scenarios for each combination of graph order and density. The risk measure ρ has been selected as an HMCR measure (11) with $p = 1, 2, 3$ and $\alpha = 0.9$.

The combinatorial branch-and-bound algorithm of Section 3.1 with the additional specializations described above has been coded in C++, and we used the CPLEX Simplex and Barrier solvers for solving the polyhedral approximations and SOCP reformulations of the p-order cone programming lower bound problem (14), respectively. In the case of $p = 1$, the CPLEX Simplex solver was used to solve the lower bound problem directly.

The performance of the developed BnB method was compared with that of the mathematical programming formulation (13) of the risk-averse maximum weighted clique problem. The MIpOCP problem (13) was solved with CPLEX MIP solver in the case of $p = 1$, and CPLEX MIP Barrier solver was applied to the SOCP version of (13) in the case of $p = 2$ or SOCP reformulation of (13) in the case of $p = 3$.

The computations were ran on an Intel Xeon 3.30GHz PC with 128GB RAM, and version 12.5 of the CPLEX solver in Windows 7 64-bit environment was used.

Table 1 summarizes the computational times, averaged over five instances, corresponding to the aforementioned problem configurations with a fixed number of scenarios of $N = 100$. Observe that the BnB algorithm provides one to two orders of magnitude advantage in running time over the CPLEX MIP solver for all configurations, except that of $p = 1$ and $d = 0.8$. For the consecutive set of experiments, Table 2 demonstrates the effect of variations in the scenario size N for different graph orders and values of p while maintaining a constant average graph density of $d = 0.5$. The specified edge probability was chosen due to the fact that the size of the mathematical programming (13) formulation is density dependent. Mainly, the number of structural constraints $x_i + x_j \leq 1, (i, j) \in \overline{E}$ in (13) increases as d decreases. The opposite relationship holds true for the BnB algorithm, as the search space expands with the number of edges. Thus, a "fair" comparison between the two solution methods can be made on graphs with density $d = 0.5$.

It follows from Tables 1 and 2 that the computational advantages of the combinatorial BnB algorithm over the direct solution approach become more pronounced (up to two orders of magnitude) with increase in p, i.e., as full formulation (13) and the lower bound problem (14) become more difficult. Also of interest is the fact that the BnB method often yields better solution times for problems with $p = 3$ than $p = 2$. This is a consequence of a known property of the employed cutting-plane algorithm for solving polyhedral approximations of p-order cone programming problems, which becomes more effective as p increases [20].

5 Conclusions

In this study, we have considered a class R-MWS problems which entail finding a network subgraph of minimum risk satisfying some hereditary structural property. We employ the HMCR measures as a rigorous framework for quantifying the distributional information of the stochastic vertex weights. By means of diversification properties of the introduced optimization-based risk function for measuring risk of subgraphs, it was shown that the inclusion of additional vertices in a partial solution promotes the minimization of risk; hence, optimal solutions to R-MWS problems are maximal subgraphs. A combinatorial branch-and-bound algorithm utilizing the risk- and graph-related aspects of the problem structure was de-

		$d = 0.2$		$d = 0.5$		$d = 0.8$	
p	$\|V\|$	BnB	CPLEX	BnB	CPLEX	BnB	CPLEX
1	50	0.08	1.10	0.37	1.31	3.04	1.90
	100	0.24	6.43	4.02	28.06	206.46	121.27
	150	0.74	38.37	26.86	220.17	4065.16	2434.66
	200	1.67	118.13	73.73	1074.93	—	—
2	50	0.40	18.54	1.66	45.67	14.50	156.26
	100	1.38	110.67	19.37	412.90	956.93	2555.77
	150	3.37	629.38	124.99	2293.96	6154.76	—
	200	3.68	2822.38	166.44	—	—	—
3	50	1.35	54.58	2.38	91.98	14.15	273.10
	100	2.43	215.97	17.66	625.52	716.22	4644.90
	150	4.41	927.03	102.28	3560.27	—	—
	200	7.24	3031.77	412.74	—	—	—

Table 1: Average computation times* (in seconds) obtained by solving problem (13) using the proposed BnB algorithm and CPLEX with risk measure (11) and scenarios $N = 100$. All running times are averaged over 5 instances and symbol "—" indicates that the time limit of 7200 seconds was exceeded.

		$\|V\| = 50$		$\|V\| = 100$		$\|V\| = 150$		$\|V\| = 200$	
p	N	BnB	CPLEX	BnB	CPLEX	BnB	CPLEX	BnB	CPLEX
1	50	0.19	1.15	1.40	11.88	4.43	43.49	13.09	130.45
	100	0.37	1.31	4.02	28.06	26.86	220.17	73.73	1074.93
	200	0.87	3.01	14.64	71.93	84.83	443.74	329.76	2550.12
	500	4.70	10.40	72.90	219.40	429.80	1794.60	2118.60	—
	1000	14.87	28.82	259.48	702.97	1909.48	6094.66	—	—
2	50	0.80	22.96	4.10	167.30	12.89	961.32	37.67	3668.54
	100	1.66	45.67	19.37	412.90	124.99	2293.96	166.44	—
	200	6.57	109.72	131.44	907.95	797.04	5961.69	900.50	—
	500	61.10	552.80	970.10	—	3221.70	—	—	—
	1000	194.59	965.69	3669.37	—	—	—	—	—
3	50	1.22	34.85	3.96	245.79	11.99	1040.01	34.30	3847.40
	100	2.38	91.98	17.66	625.52	102.28	3560.27	412.74	—
	200	5.21	261.83	60.59	2388.44	333.61	—	1424.27	—
	500	20.10	1299.60	248.70	—	1751.90	—	—	—
	1000	58.00	3277.93	768.53	—	5634.04	—	—	—

Table 2: Average computation times (in seconds) obtained by solving problem (13) using the proposed BnB algorithm and CPLEX with risk measure (11) and edge density $d = 0.5$. All running times are averaged over 5 instances and symbol "—" indicates that the time limit of 7200 seconds was exceeded.

veloped and tested on a special case of the risk-averse maximal clique problem. Numerical experiments on randomly generated Erdös-Rényi graphs demonstrate that the proposed algorithm may significantly reduce solution times relative to an equivalent mathematical programming counterpart. Notably, improvements were observed for all the tested graph configurations when using the HMCR measures with $p = 2, 3$, and for graphs with edge probabilities of less than 0.8 when using an HMCR measure with $p = 1$.

6 Acknowledgements

This work was supported in part by the U.S. Dept. of Air Force grant FA8651-12-2-0010, AFOSR grant FA9550-12-1-0142, and NSF grant EPS1101284. The authors are grateful for the support from the AFRL Mathematical Modeling and Optimization Institute.

References

[1] Y. P. Aneja, R. Chandrasekaran, and K. P. K. Nair. Maximizing residual flow under an arc destruction. *Networks*, 38(4):194–198, 2001.

[2] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[3] A. Atamtrk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.

[4] L. Babel. A fast algorithm for the maximum weight clique problem. *Computing*, 52(1):31–38, 1994.

[5] E. Balas and J. Xue. Minimum weighted coloring of triangulated graphs, with application to maximum weight vertex packing and clique finding in arbitrary graphs. *SIAM J. Comput.*, 20(2):209–221, Mar. 1991.

[6] E. Balas and C. S. Yu. Finding a maximum clique in an arbitrary graph. *SIAM J. Comput.*, 15(4):1054–1068, Nov. 1986.

[7] A. Ben-Tal and A. Nemirovski. On polyhedral approximations of the second-order cone. *dim*, 50:1, 1999.

[8] V. L. Boginski, C. W. Commander, and T. Turko. Polynomial-time identification of robust network flows under uncertain arc failures. *Optimization Letters*, 3(3):461–473, 2009.

[9] A. M. Campbell and B. W. Thomas. Probabilistic traveling salesman problem with deadlines. *Transportation Science*, 42(1):1–21, 2008.

[10] R. Carmo and A. Zge. Branch and bound algorithms for the maximum clique problem under a unified framework. *Journal of the Brazilian Computer Society*, 18(2):137–151, 2012.

[11] R. Carraghan and P. M. Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9(6):375 – 382, 1990.

[12] F. Delbaen. Coherent risk measures on general probability spaces. In K. Sandmann and P. Schnbucher, editors, *Advances in Finance and Stochastics*, pages 1–37. Springer Berlin Heidelberg, 2002.

[13] P. Erdös and A. Rényi. On the evolution of random graphs. 5:17–61, 1960.

[14] R. G. Gallager, P. A. Humblet, and P. M. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Trans. Program. Lang. Syst.*, 5(1):66–77, Jan. 1983.

[15] G. D. Glockner and G. L. Nemhauser. A dynamic network flow problem with uncertain arc capacities: Formulation and problem structure. *Operations Research*, 48(2):233–242, 2000.

[16] A. Gupta, V. Nagarajan, and R. Ravi. Technical noteapproximation algorithms for vrp with stochastic demands. *Operations Research*, 60(1):123–127, 2012.

[17] J. Konc and D. Janezic. An improved branch and bound algorithm for the maximum clique problem. *proteins*, 4:5, 2007.

[18] P. Krokhmal, M. Zabarankin, and S. Uryasev. Modeling and optimization of risk. *Surveys in Operations Researh and Management Science*, 16(2):49–66, 2011.

[19] P. A. Krokhmal. Higher moment coherent risk measures. *Quantitative Finance*, 7:373–387, 2007.

[20] P. A. Krokhmal and P. Soberanis. Risk optimization with p-order conic constraints: A linear programming approach. *European Journal of Operational Research*, 201(3):653–671, 2010.

[21] D. Kumlander. A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search.

[22] D. Kumlander. On importance of a special sorting in the maximum-weight clique algorithm based on colour classes. In H. Le Thi, P. Bouvry, and T. Pham Dinh, editors, *Modelling, Computation and Optimization in Information Systems and Management Sciences*, volume 14 of *Communications in Computer and Information Science*, pages 165–174. Springer Berlin Heidelberg, 2008.

[23] Z. Miao, B. Balasundaram, and E. Pasiliao. An exact algorithm for the maximum probabilistic clique problem. *Working paper*.

[24] Y. Morenko, A. Vinel, Z. Yu, and P. Krokhmal. On p-norm linear discrimination. *European Journal of Operational Research*, 231(3):784–789, 2013.

[25] P. R. J. Östergård. A new algorithm for the maximum-weight clique problem. *Nordic J. of Computing*, 8(4):424–436, Dec. 2001.

[26] P. R. J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1–3):197–207, 2002. Special Issue devoted to the 6th Twente Workshop on Graphs and Combinatorial Optimization.

[27] P. M. Pardalos and J. Xue. The maximum clique problem. *Journal of Global Optimization*, 4:301–328, 1994.

[28] A. Sorokin, V. Boginski, A. Nahapetyan, and P. Pardalos. Computational risk management techniques for fixed charge network flow problems with uncertain arc failures. *Journal of Combinatorial Optimization*, 25(1):99–122, 2013.

[29] E. Tomita, Y. Sutani, T. Higashi, S. Takahashi, and M. Wakatsuki. A simple and faster branch-and-bound algorithm for finding a maximum clique. In M. Rahman and S. Fujita, editors, *WALCOM: Algorithms and Computation*, volume 5942 of *Lecture Notes in Computer Science*, pages 191–203. Springer Berlin Heidelberg, 2010.

[30] S. Trukhanov, C. Balasubramaniam, B. Balasundaram, and S. Butenko. Algorithms for detecting optimal hereditary structures in graphs, with application to clique relaxations. *Computational Optimization and Applications*, 56(1):113–130, 2013.

[31] B. Verweij, S. Ahmed, A. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications*, 24(2-3):289–333, 2003.

[32] A. Vinel and P. Krokhmal. On polyhedral approximations in p-order cone programming. *Working paper*, 2013.

[33] M. Yannakakis. Node-and edge-deletion np-complete problems. In *STOC'78: Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 253–264, New York, 1978. ACM Press.

[34] O. Yezerska, S. Butenko, and V. Boginski. Detecting robust cliques in the graphs subject to uncertain edge failures. *Working paper*.

Journal of Optimization Theory and Applications manuscript No.
(will be inserted by the editor)

On Valid Inequalities for Mixed Integer *p*-Order Cone Programming

Alexander Vinel · **Pavlo Krokhmal**

Communicated by Panos M. Pardalos

Abstract We discuss two families of valid inequalities for linear mixed integer programming problems with cone constraints of arbitrary order, which arise in the context of stochastic optimization with downside risk measures. In particular, we extend the results of Atamtürk and Narayanan (Math. Program., 2010, 2011), who developed mixed integer rounding cuts and lifted cuts for mixed integer programming problems with second order cone constraints. Numerical experiments conducted on randomly generated problems and portfolio optimization problems with historical data demonstrate the effectiveness of the proposed methods.

Keywords valid inequalities · nonlinear cuts · mixed integer *p*-order cone programming · stochastic optimization · risk measures

Mathematics Subject Classification (2000) 90C11 · 90C15 · 90C30

1 Introduction

In this work we consider mixed integer programming problems with linear objective and *p*-order cone constraints, which represent an extension of mixed integer second order cone programming (MISOCP) problems and subsequently are referred to as mixed integer *p*-order cone programmig (MIpOCP) problems. Specifically, we focus on a class of MIpOCP instances that arise in stochastic optimization problems with risk-based objective functions or constraints.

There exists a substantial literature on solution approaches for mixed integer conic programming problems. In many cases, the proposed methods attempt to extend some of the techniques developed for mixed integer linear programming. One of such research directions

This work was supported in part by AFOSR grant FA9550-12-1-0142 and NSF grant EPS1101284

A. Vinel
Department of Mechanical and Industrial Engineering, University of Iowa, Iowa City, IA 52242, USA
E-mail: alexander-vinel@uiowa.edu

P. Krokhmal (✉)
Department of Mechanical and Industrial Engineering, University of Iowa, Iowa City, IA 52242, USA
E-mail: krokhmal@engineering.uiowa.edu

concerns construction of branch-and-bound schemes based on outer polyhedral approximations of cones. This potentially allows for computational savings in traversing the branch-and-bound tree due to the “warm start” capabilities of linear programming solvers. In particular, Vielma et al. [1] proposed a branch-and-bound method for MISOCP that employed lifted polyhedral approximations of second order cones due to Ben-Tal and Nemirovski [2]. Vinel and Krokhmal [3] discuss further development of this approach in the case of MIpOCP. Drewes [4] presented subgradient-based linear outer approximations for the second order cone constraints in mixed integer programs. With respect to mixed integer nonlinear programming, a similar idea has been exploited by Bonami et al. [5] and Tawarmalani and Sahinidis [6].

Two approaches to generation of valid inequalities for MISOCP problems have been proposed by Atamtürk and Narayanan [7,8]. In the first paper the authors introduced a reformulation of a second order cone constraint using a set of two-dimensional second order cones and then derived valid inequalities for the resulting mixed integer sets. The obtained cuts were termed by the authors conic mixed integer rounding cuts. In [8], a general lifting procedure for deriving nonlinear conic valid inequalities was proposed and applied to 0-1 MISOCP problems.

In a recent work of Belotti et al. [9], disjunctive conic cuts for MISOCP problems are introduced. For the case of general convex sets, the authors are able to describe the convex hull of the intersection of a convex set and a linear disjunction. And in the particular case of the feasible set of the continuous relaxation of a MISOCP problem they derive a closed-form expression for such a convex hull, thus obtaining a new nonlinear conic cut.

Among other approaches to solving mixed integer cone programming problems one can mention the split closure of a strictly convex body [10], lift-and-project algorithm [11], Chvátal-Gomory and disjunctive cuts for 0-1 conic programming [12].

It is worth noting that the vast majority of the existing literature on mixed integer cone programming problems addresses the case of self-dual cones, and particularly second order cones, with relatively little attention paid to problems involving cones that are not self-dual, as in the case of MIpOCP with $p \in]1,2[\cup]2,\infty[$. In this work, we consider derivation of valid inequalities for mixed integer problems with p-order cone constraints following the techniques [7,8] proposed for MISOCP. We derive closed form expressions for two families of valid inequalities for MIpOCP problems: mixed integer rounding conic cuts and lifted conic cuts. We also propose to use outer polyhedral approximations as a practical way of employing nonlinear lifted cuts within branch-and-cut framework. With such an approach, we are able to obtain promising computational results on a number of portfolio optimization problems with real-life data.

The paper is organized as follows. In Section 2 we present mixed integer rounding cuts for p-cone constrained mixed integer sets. Section 3 discusses (nonlinear) lifted cuts for 0-1 and mixed integer p-order cone programming problems. Computational studies of the developed techniques on randomly generated MIpOCP problems as well as portfolio optimization problems with real-life data are discussed in Section 4, followed by concluding remarks in Section 5.

2 Conic Mixed Integer Rounding Cuts for p-Order Cones

In this section we present a class of mixed integer rounding cuts for MIpOCP problems arising in the context of risk-averse stochastic optimization. A mixed integer p-order cone programming problem has the form

$$\begin{aligned}\min\quad & (\mathbf{c}_x^\top \mathbf{x} + \mathbf{c}_y^\top \mathbf{y})\\ \text{s. t.}\quad & \mathbf{D}_x\mathbf{x} + \mathbf{D}_y\mathbf{y} \le \mathbf{d}\\ & \|\mathbf{A}_j\mathbf{x} + \mathbf{G}_j\mathbf{y} - \mathbf{b}_j\|_{p_j} \le \mathbf{e}_j^\top \mathbf{x} + \mathbf{f}_j^\top \mathbf{y} - h_j, \quad j = 1,\dots,k\\ & \mathbf{x} \in \mathbb{Z}_+^n,\ \mathbf{y} \in \mathbb{R}_+^q,\end{aligned} \tag{1}$$

where $p_j \in]1,\infty[$, and $\|\cdot\|_p$ is the usual p-norm in the Euclidean space of an appropriate dimension: $\|\mathbf{r}\|_p = (|r_1|^p + \ldots + |r_N|^p)^{1/p}$.

MIpOCP problems (1) can be obtained from stochastic programming models that involve specific families of risk measures in objectives or constraints. Namely, given a probability space $(\Omega, \mathscr{F}, \mu)$, let the cost or loss function Y be an element of the linear space $\mathscr{L}_p(\Omega, \mathscr{F}, \mu)$ of $\mathscr{F}$-measurable functions $Y : \Omega \mapsto \mathbb{R}$, where $p \ge 1$. Then, the *higher-moment coherent risk measures* $\mathsf{HMCR}_{p,\alpha}(Y)$ are defined as the optimal values of the following convex stochastic optimization problem [13]

$$\mathsf{HMCR}_{p,\alpha}(Y) = \min_{\eta\in\mathbb{R}} \Big\{ \eta + (1-\alpha)^{-1} \big\| [Y-\eta]_+ \big\|_p \Big\}, \quad \alpha \in]0,1[,\ \ p \ge 1, \tag{2}$$

where $[Y]_+ = \max\{0, Y\}$ and $\|Y\|_p = (\mathsf{E}|Y|^p)^{1/p}$. A related family of *semi-moment coherent risk measures,* or risk measures of semi-$\mathscr{L}_p$ type [14], is given as

$$\mathsf{SMCR}_{p,\beta}(Y) = \mathsf{E}Y + \beta \big\| [Y - \mathsf{E}Y]_+ \big\|_p, \quad \beta \in [0,1],\ \ p \ge 1. \tag{3}$$

In the case when the set Ω is finite, $\Omega = \{\omega_1, \ldots, \omega_m\}$, and the cost function $Y = Y(\mathbf{u}, \omega)$ is a piecewise linear convex function of the decision vector $\mathbf{u}$, terms with HMCR or SMCR measures in the objective function and/or constraints can be implemented via linear inequalities involving $Y(\mathbf{u}, \omega_i)$ and p-order cone constraints $t \ge \|(w_1, \ldots, w_m)\|_p$, thus leading to MIpOCP problem of the form

$$\begin{aligned}\min\quad & (\mathbf{c}_x^\top \mathbf{x} + \mathbf{c}_y^\top \mathbf{y})\\ \text{s. t.}\quad & \mathbf{D}_x\mathbf{x} + \mathbf{D}_y\mathbf{y} \le \mathbf{d}\\ & \|[\mathbf{A}_j\mathbf{x} + \mathbf{G}_j\mathbf{y} - \mathbf{b}_j]_+\|_{p_j} \le \mathbf{e}_j^\top \mathbf{x} + \mathbf{f}_j^\top \mathbf{y} - h_j, \quad j = 1,\dots,k\\ & \mathbf{x} \in \mathbb{Z}_+^n,\ \mathbf{y} \in \mathbb{R}_+^q,\end{aligned} \tag{4}$$

Formulation (4) differs from (1) by the presence of operator $[\cdot]_+$, which explicitly accounts for the problem structure induced by downside risk measures such as (2)–(3). For simplicity, we consider the case of a single p-cone constraint in (4), $k = 1$. Following the approach of [7] for constructing mixed integer rounding cuts for problems of type (1) with $p = 2$, we rewrite the p-cone constraint in (4) as

$$\begin{aligned}& t_0 \le \mathbf{e}^\top \mathbf{x} + \mathbf{f}^\top \mathbf{y} - h\\ & t_i \ge [\mathbf{a}_i^\top \mathbf{x} + \mathbf{g}_i^\top \mathbf{y} - b_i]_+, \quad i = 1,\dots,m\\ & t_0 \ge \|(t_1, \ldots, t_m)\|_p,\end{aligned}$$

where $\mathbf{a}_i$ and $\mathbf{g}_i$ denote the i-th rows of matrices $\mathbf{A}$ and $\mathbf{G}$, respectively. Then, the task of deriving valid inequalities for the original p-cone mixed integer set in (4) can be reduced to obtaining valid inequalities for the polyhedral mixed integer set

$$T = \big\{ \mathbf{x} \in \mathbb{Z}_+^n,\ \mathbf{y} \in \mathbb{R}_+^p,\ t \in \mathbb{R}\ :\ [\mathbf{a}^\top \mathbf{x} + \mathbf{g}^\top \mathbf{y} - b]_+ \le t \big\},$$

or, without loss of generality, the set

$$\widetilde{T} = \big\{(y^+, y^-, t, \mathbf{x}) \in \mathbb{R}^3_+ \times \mathbb{Z}^n_+ \;:\; [\mathbf{a}^\top \mathbf{x} + y^+ - y^- - b]_+ \leq t\big\}. \tag{5}$$

The following two propositions provide an expression for a family of such inequalities.

Proposition 2.1 *For $\alpha \neq 0$, the inequality*

$$\sum_{j=1}^{n} \phi_{f_{|\alpha|}}\left(\frac{a_j}{|\alpha|}\right) x_j - \phi_{f_{|\alpha|}}\left(\frac{b}{|\alpha|}\right) \leq \frac{t + y^-}{|\alpha|}, \tag{6}$$

where $f_\alpha = \frac{b}{|\alpha|} - \left\lfloor \frac{b}{|\alpha|} \right\rfloor$ and

$$\phi_f(a) = \begin{cases} (1-f)n, & n \leq a < n+f \\ (1-f)n + (a-n) - f, & n+f \leq a < n+1 \end{cases}$$

is valid for $\widetilde{T}$.

Proposition 2.2 *Inequalities* (6) *with $\alpha = a_j$, $j = 1, \ldots, n$, are sufficient to cut off all fractional extreme points of the relaxation of $\widetilde{T}$.*

Proofs of Propositions 2.1 and 2.2 are furnished in the Appendix. It is worth noting, however, that since (5) is a polyhedral mixed integer set, the derived valid inequalities can also be obtained using the general theory of mixed integer rounding (MIR) inequalities; see, for example, [15]. An advantage of the direct derivation is that it provides a natural way of dealing with continuous variables y^+, y^-, t. Propositions 2.1 and 2.2 justify the usage of inequalities of type (6) as cuts in a branch-and-cut procedure; following [7], we refer to these inequalities as conic MIR cuts. The results of numerical experiments on utilization of conic MIR cuts (6) in MIpOCP problems are presented in Section 4.

3 Lifted Conic Cuts for p-Order Cones

3.1 General Framework

Lifting for conic mixed integer programming was studied in [8], where a general approach for constructing valid nonlinear conic inequalities for mixed inter conic programming problems was proposed. Namely, consider a general mixed integer conic set

$$S^n(\mathbf{b}) = \left\{(\mathbf{x}^0, \ldots, \mathbf{x}^n) \in X^0 \times \cdots \times X^n \;:\; \mathbf{b} - \sum_{i=0}^{n} \mathbf{A}^i \mathbf{x}^i \in \mathscr{C}\right\}, \tag{7}$$

where $\mathbf{A}^i \in \mathbb{R}^{m \times n_i}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathscr{C}$ is a proper cone (a closed, convex, pointed cone with a nonempty interior), and each $X^i \subset \mathbb{R}^{n_i}$ is a mixed integer set. Similarly, $S^0(\mathbf{b}), \ldots, S^{n-1}(\mathbf{b})$ are restrictions of the set $S^n(\mathbf{b})$. Further, it is assumed that the following conic inequality

$$\mathbf{h} - \mathbf{F}^0 \mathbf{x}^0 \in \mathscr{K},$$

where $\mathscr{K}$ is a proper cone, is known to be valid for the restriction $S^0(\mathbf{b})$. The approach proposed in [8] is to iteratively find a sequence $\mathbf{F}^1, \ldots, \mathbf{F}^n$, such that

$$\mathbf{h} - \sum_{j=0}^{i} \mathbf{F}^j \mathbf{x}^j \in \mathscr{K} \tag{8}$$

is valid for the respective restriction $S^i(\mathbf{b})$ for all i. Such a procedure is called *lifting* and the resulting inequality that is valid for the initial mixed integer set $S^n(\mathbf{b})$ is called *lifted inequality*. In order to determine the values of $\mathbf{F}^1,\ldots,\mathbf{F}^n$, the *lifting set* is introduced for $\mathbf{v}\in\mathbb{R}^m$ as

$$\Phi_i(\mathbf{v}) = \left\{ \mathbf{d}\in\mathbb{R}^s \;:\; \mathbf{h} - \sum_{j=0}^{i} \mathbf{F}^j\mathbf{x}^j - \mathbf{d} \in \mathscr{K} \;\text{ for all }\; (\mathbf{x}^0,\ldots,\mathbf{x}^i)^\top \in S^i(\mathbf{b}-\mathbf{v}) \right\}.$$

Then, a necessary and sufficient condition for (8) to be valid can be formulated, which essentially provides a description of the set of valid inequalities.

Proposition 3.1 [8] *Inequality* (8) *is valid for* $S^i(\mathbf{b})$ *if and only if* $\mathbf{F}^i\mathbf{t}\in\Phi_i(\mathbf{A}^i\mathbf{t})$ *for all* $\mathbf{t}\in X^i$ *and* $i=0,\ldots,n$.

The condition established by Proposition 3.1 is still too general to be used for derivation of conic cuts. For example, it can be seen that in this way the resulting inequalities are *sequence-dependent*, i.e., a change in the order in which variables $\mathbf{x}^i$ are introduced will change the sets $\Phi_i(\mathbf{v})$. The following theorem provides a "sequence-independent" approach to construction of lifting procedure.

Theorem 3.1 [8] *If* $\Upsilon(\mathbf{v})\subseteq\Phi_0(\mathbf{v})$ *for all* $\mathbf{v}\in\mathbb{R}^m$ *and* Υ *is superadditive, then* (8) *is a lifted valid inequality for* $S^n(\mathbf{b})$ *whenever* $\mathbf{F}^i\mathbf{t}\in\Upsilon(\mathbf{A}^i\mathbf{t})$ *for all* $\mathbf{t}\in X^i$ *and* $i=0,\ldots,n$.

Then, the following procedure can be formulated for derivation of lifted conic inequalities:

Step 1. Compute $\Phi_0(\mathbf{v})$.
Step 2. If $\Phi_0(\mathbf{v})$ is not superadditive, find a superadditive $\Upsilon(\mathbf{v})\subset\Phi_0(\mathbf{v})$.
Step 3. For each i find $\mathbf{F}^i$ such that $\mathbf{F}^i\mathbf{t}\in\Upsilon(\mathbf{A}^i\mathbf{t})$ is satisfied for all $\mathbf{t}\in X^i$.

In [8] this process was employed to obtain nonlinear lifted conic cuts for 0-1 MISOCP problems; however, no computational results were reported. Below we apply this procedure to derive nonlinear lifted conic cuts for 0-1 and mixed integer p-order cone programming problems with risk-based constraints, and also discuss polyhedral approximations of these cuts that are used in numerical implementation.

3.2 Lifting Procedure for 0-1 p-Order Cone Programming Problems

In the case of 0-1 p-order cone programming problem, consider the following conic set

$$S_p^n(b) = \left\{ (\mathbf{x},\eta_+,\eta_-,y,t)\in\{0,1\}^n\times\mathbb{R}_+^4 \;:\; \Big[\sum_{i=1}^n a_ix_i + \eta_+ - \eta_- - b\Big]_+^p + y^p \le t^p \right\},$$

where $p\in]1,\infty[$. The set $S_p^n(b)$ represents a relaxation of a high dimensional 0-1 mixed integer p-order conic set: all but one dimensions of the p-cone are aggregated into the term y^p. By complementing the binary variables, if necessary, we can assume that all $a_i\ge 0$. The restriction S_p^0 of this set can be taken as

$$S_p^0(b) = \big\{ (x,y,t)\in\{0,1\}\times\mathbb{R}_+^2 \;:\; [x-b]_+^p + y^p \le t^p \big\}.$$

Notice that $S_p^0(b)$ has one extreme point $(b,0,0)$, which is fractional when $b\in]0,1[$. Thus, in the only interesting case we have $\lfloor b\rfloor = 0$. Using the results of the previous section, the initial valid inequality can be selected as $\big|(1-f)(x-\lfloor b\rfloor)\big|^p + y^p \le t^p$, where $f=b-\lfloor b\rfloor$ (the fact that this inequality is valid can be verified directly by examining the possible values of x,y,t). Now, by definition, in order to compute $\Phi_0(v)$ we need to find such d that inequality

$$\big|(1-f)(x-\lfloor b\rfloor)+d\big|^p+y^p\le t^p \tag{9}$$

is satisfied for all x,y,t such that $[x-b+v]_+^p+y^p\le t^p$.

Recalling that $\lfloor b\rfloor=0$ and, therefore, $f=b$, we obtain that (9) can be rewritten as $|(1-b)x+d|^p+y^p\le t^p$ for all x,y,t such that $[x-b+v]_+^p+y^p\le t^p$. Given that $x\in\{0,1\}$, for $x=0$ we have $|d|\le[v-b]_+$, and for $x=1$ we have $|1-b+d|\le[1-b+v]_+$. Thus, if $v\ge b$ then $|d|\le v-b$, and if $v<b$ then $d=0$, meaning that $|d|\le[v-b]_+$, whereby $\Phi_0(v)=\{d\,:\,|d|\le[v-b]_+\}$, which is superaddive. Finally, the following proposition holds.

Proposition 3.2 *Conic inequality*

$$\Big|(1-f)(x-\lfloor b\rfloor)+\sum_{i=1}^n\alpha_i x_i\Big|^p+y^p\le t^p \tag{10}$$

with $\alpha_i=[a_i-b]_+$ *is valid for the set* $S_p^n(b)$.

Proof Since $\Phi_0(v)$ is superadditive, by Theorem 3.1 we only need to verify that the chosen values of α_i satisfy $\alpha_i x\in\Phi_0(a_i x)$ for $x\in\{0,1\}$, which follows readily from the expression for $\Phi_0(v)$. □

3.3 Lifting Procedure for MIpOCP Problems

Similarly, in the case of MIpOCP problem we consider the set

$$\hat S_p^n(b)=\left\{(\mathbf{x},\eta_+,\eta_-,y,t)\in\mathbb{Z}_+^n\times\mathbb{R}_+^4\;:\;\Big[\sum_{i=1}^n a_i x_i+\eta_+-\eta_--b\Big]_+^p+y^p\le t^p\right\},$$

where $p\in]1,\infty[$. Once again, the set $\hat S_p^n(b)$ represents a relaxation of a high dimensional mixed integer p-order cone constraint. Let us also assume that values x_i are bounded, e.g., $x_i\in\{0,\dots,M\}$ for all i. Again, let us assume without loss of generality that $a_i>0$. The restriction of $\hat S_p^n(b)$ can be selected as

$$\hat S_p^0(b)=\big\{(x,y,t)\in\mathbb{Z}_+\times\mathbb{R}_+^2\;:\;[x-b]_+^p+y^p\le t^p\big\}, \tag{11}$$

but in this case let us choose a weaker initial valid inequality, $\big[(1-f)(x-\lfloor b\rfloor)\big]_+^p+y^p\le t^p$. The problem of computing $\Phi_0(v)$ is then reduced to the problem of finding values of d such that

$$\big[(1-f)x-\lfloor b\rfloor(1-f)+d\big]_+\le\big[x-b+v\big]_+. \tag{12}$$

Recall that we are only interested in a superadditive subset $\Upsilon(v)$ of such set. One of the possible choices is $\Upsilon(v)=\big\{d\ge0\;:\;d\le[v-b+\lfloor b\rfloor(1-f)]_+\big\}$. Indeed, $0\in\Upsilon(v)$ by definition, and (12) is a consequence of inequality $(1-f)x-\lfloor b\rfloor(1-f)+d\le x-b+v$, which yields the above expression for $\Upsilon(v)$. Lastly, the following proposition holds.

Proposition 3.3 *Conic inequality*

$$\Big[(1-f)(x-\lfloor b\rfloor)+\sum_{i=1}^{n}\alpha_i x_i\Big]_+^p + y^p \le t^p \tag{13}$$

with $\alpha_i = \left[\dfrac{a_i - b + \lfloor b\rfloor(1-f)}{M}\right]_+$ *is valid for* $\hat{S}_p^n(b)$.

Proof Indeed, in accordance to Section 3.1 it suffices to show that for such a choice of α_i we have $\alpha_i x \in \Upsilon(a_i x)$ for all x. For $x \neq 0$ we have

$$\Upsilon(a_i x) = \big\{d \ge 0 \,:\, d \le [a_i x - b + \lfloor b\rfloor(1-f)]_+\big\},$$

and

$$\alpha_i x = \left[\frac{a_i - b + \lfloor b\rfloor(1-f)}{M}\right]_+ x \le [a_i - b + \lfloor b\rfloor(1-f)]_+ \le [a_i x - b + \lfloor b\rfloor(1-f)]_+.$$

On the other hand, for $x = 0$ it is clear that $0 \in \Upsilon(0)$. □

3.4 Polyhedral Approximations of p-Order Cones

Observe that lifted cuts (10) and (13) for, respectively, 0-1 and mixed integer p-order cone programming problems have the form of p-order cones themselves. Thus, one may expect that while addition of such cuts can reduce the number of nodes explored in the branch-and-bound tree, the computational cost of solving the relaxed problem with extra p-cone constraints at the nodes may increase. In view of this, we propose to replace the nonlinear p-order cone cuts (10) and (13) with their polyhedral approximations during the branch-and-cut procedure. A detailed discussion of polyhedral approximations of p-order cones can be found in [3].

Since in our case the lifted cuts have the form of 3-dimensional p-cones, we use a simple gradient polyhedral approximation. Particularly, a gradient polyhedral approximation for the conic set $\mathcal{K}_p^{(3)} = \{\xi \in \mathbb{R}_+^3 \,:\, \xi_3 \ge \|(\xi_1,\xi_2)\|_p\}$, $p \in]1,\infty[$, can be constructed as

$$\mathcal{H}_{p,\ell}^{(3)} = \big\{\xi \in \mathbb{R}_+^3 \,:\, \xi_3 \ge \alpha_i^{(p)}\xi_1 + \beta_i^{(p)}\xi_2,\ i = 0,\ldots,\ell\big\}, \tag{14}$$

where

$$\begin{bmatrix}\alpha_i^{(p)}\\ \beta_i^{(p)}\end{bmatrix} = (\cos^p\theta_i + \sin^p\theta_i)^{\frac{1-p}{p}}\begin{bmatrix}\cos^{p-1}\theta_i\\ \sin^{p-1}\theta_i\end{bmatrix}, \quad \theta_i = \frac{\pi i}{2\ell}, \quad i = 0,\ldots,\ell.$$

Here $\mathcal{H}_{p,\ell}^{(3)}$ is an approximation of $\mathcal{K}_p^{(3)}$ in the sense that $\xi \in \mathcal{K}_p^{(3)}$ implies $\xi \in \mathcal{H}_{p,\ell}^{(3)}$, and $\xi \in \mathcal{H}_{p,\ell}^{(3)}$ implies $(1+\varepsilon)\xi_3 \ge \|(\xi_1,\xi_2)\|_p$, where $\varepsilon = \varepsilon(\ell)$ is the accuracy of approximation. In the case of polyhedral approximation (14), the latter can be estimated as [16]

$$\varepsilon(\ell) \approx \begin{cases} \frac{1}{p}\Big(1-\frac{1}{p}\Big)^p\Big(\frac{\pi}{2\ell}\Big)^p, & p \in]1,2[,\\ \frac{1}{8}(p-1)\Big(\frac{\pi}{2\ell}\Big)^2, & p \in [2,\infty[. \end{cases}$$

For example, for $p = 4.0$ it suffices to have $\ell = 25$ facets in the approximation to ensure an accuracy of 10^{-3}.

4 Computational Results

In this section we report the results of numerical experiments on applying the derived MIR and lifted conic cuts to MIpOCP problem instances. In our case study, three types of problem instances were considered: the first type represents the "generic" MIpOCP instances with randomly generated data, and the second and third types of instances represent portfolio optimization problems with cardinality constraints and lot-buying constraints, respectively. Historical financial data were used for both types of portfolio optimization problems. A detailed description of each problem type is given below.

Computations were ran on a 3GHz PC with 4GB RAM, and CPLEX 12.2 solver was used. Since CPLEX cannot natively handle p-cone constraints with $p \neq 2$, a second-order cone reformulation [17–19] was applied to p-order cone constraints with rational $p > 2$. The derived cuts were added at the root node of the branch-and-bound tree using CPLEX callback routines. In addition, each instance was solved using the default mixed integer CPLEX solver with built-in cuts. In both cases, default solver configuration was used, with the exceptions that the number of threads was limited to one and QCP relaxations of the model were used at each node.

4.1 Problem Formulations

Randomly generated MIpOCP problems The first set of problem instances consisted of randomly generated mixed integer p-order cone programming problems of the general form. Specifically, the following formulation was used:

$$\begin{aligned} \min \quad & (\mathbf{c}^\top \mathbf{x} + y^+ + y^-) \\ \text{s. t.} \quad & \big\| [\mathbf{A}\mathbf{x} + y^+\mathbf{1} - y^-\mathbf{1} - \mathbf{b}]_+ \big\|_p \leq \mathbf{e}^\top \mathbf{x} + f y^+ - g y^- - h \\ & \mathbf{x} \in \mathbb{Z}^n_+,\ y^+, y^- \in \mathbb{R}_+, \end{aligned} \tag{15}$$

where $\mathbf{A} \in \mathbb{R}^{n\times m}$, $\mathbf{c}, \mathbf{b}, \mathbf{e} \in \mathbb{R}^n$, $f, g, h \in \mathbb{R}$, and $\mathbf{1} = (1, \ldots, 1)^\top$. Each of the parameters $\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{e}, f, g, h$ in (15) was selected from the uniform $U(1, 1000)$ distribution.

Portfolio optimization with cardinality constraints. The second set of problem instances consisted of portfolio optimization problems with cardinality constraints. Specifically, portfolio risk as given by HMCR measure was minimized while requiring that the portfolio's expected return was not below some prescribed level r_0. No short sales were allowed, and the cardinality constraint ensured that the portfolio was comprised of no more than K assets:

$$\min_{\mathbf{y}\in\mathbb{R}^n_+,\, \mathbf{x}\in\{0,1\}^n} \Big\{ \mathsf{HMCR}_{\alpha,p}(-\mathbf{r}^\top \mathbf{y}) \ :\ \mathsf{E}(\mathbf{r}^\top \mathbf{y}) \geq r_0,\ \mathbf{1}^\top \mathbf{y} \leq 1,\ \mathbf{y} \leq \mathbf{x},\ \mathbf{1}^\top \mathbf{x} \leq K \Big\}, \tag{16}$$

where vectors $\mathbf{y}$ and $\mathbf{r} = \mathbf{r}(\omega)$ represented the weights of assets in the portfolio and the assets' uncertain returns, respectively. Using definition (2) of HMCR measures and assuming that the stochastic vector $\mathbf{r}(\omega)$ is discretely distributed with m scenarios $\mathbf{r}(\omega_i)$, $i = 1, \ldots, m$, the portfolio optimization problem (16) can be formulated as a 0-1 MIpOCP problem with $(m+1)$-dimensional p-cone constraint. In our computations we set $K = 5$ and $\alpha = 0.9$ in (16).

Portfolio optimization with lot-buying constraints. The last type of problems considered in this case study represents portfolio optimization problems with lot-buying constraints. The lot-buying constraints reflect the real-life trading policies of many financial markets (see, e.g., [20–22] and references therein), where the investors are allowed to buy or sell shares of financial instruments only in *lots* of standard size L, e.g., in multiples of $L = 1{,}000$ shares. Following the same setup as above, a risk-minimizing portfolio allocation problem with lot-buying constraints is formulated as

$$\min_{\mathbf{y}\in\mathbb{R}^n_+,\,\mathbf{x}\in\mathbb{Z}^n_+} \left\{ \mathsf{HMCR}_{\alpha,p}(-\mathbf{r}^\top\mathbf{y}) \;:\; \mathsf{E}(\mathbf{r}^\top\mathbf{y}) \geq r_0,\;\; \mathbf{1}^\top\mathbf{y} \leq 1,\;\; \mathbf{y} = \frac{L}{C}\,\mathrm{Diag}(\mathbf{p})\,\mathbf{x} \right\}. \tag{17}$$

Here $L \in \mathbb{N}$ is the given lot size, $C > 0$ is the available capital (in dollars), vector $\mathbf{p} \in \mathbb{R}^n_+$ represents the current (observable) asset prices per share, and $\mathrm{Diag}(\mathbf{a})$ denotes a matrix whose diagonal elements are equal to the corresponding elements of vector $\mathbf{a}$, and off-diagonal elements are zero. Similarly to the above, portfolio problem (17) reduces to a MIpOCP problem with $(m+1)$-dimensional p-cone constraint, where m is the number of scenarios in stochastic representation of the vector of assets' returns $\mathbf{r}$. The values of parameters L and C in our experiments were set at $L = 1{,}000$ and $C = \$100{,}000$.

For portfolio optimization problems, we used historical data for n stocks chosen at random from the S&P500 index, and returns over m consequent 10-day periods starting at a (common) randomized date were used to construct the set of m scenarios for the stochastic vector $\mathbf{r}$ in (16), (17).

4.2 Discussion of Results: Conic MIR Cuts

Randomly generated MIpOCP problems For each pair of parameters (n,m) that determine the number of integer variables and the dimensionality of p-cone, 50 randomly generated instances of problem (15) were solved. The results are summarized in Table 1, where the average computational time (in seconds), the average number of nodes explored in the search tree, and the average number of cuts added during the solution procedure are reported. In addition, we report the percentage of cases in which addition of conic MIR cuts improves the computational time and the number of nodes explored, respectively, as compared to the default CPLEX routines. It has also been noted that randomly generated problems are relatively easy to solve; in fact, many instances were solved at the root node. Therefore, in addition to the results averaged over all instances of a given problem size (n,m), Table 1 presents the results averaged over "difficult" instances, i.e., instances that could not be solved at the root node by CPLEX solver with default parameter settings. As one can see, in most cases utilization of conic MIR cuts reduces the average solution time and the number of nodes explored in the solution tree, with the improvement being more noticeable for "difficult" instances and larger sizes of the problem. It is also worth noting that while solution times vary for different values of the parameter p, the observed improvement due to implementation of conic MIR cuts stays approximately the same.

Portfolio optimization with cardinality constraints. For each problem size we generated 30 problem instances. The obtained results are summarized in Table 2. We can again conclude that for the majority of the instances, introduction of conic MIR cuts leads to an improved performance in comparison to the default CPLEX solution procedures, although the improvement is considerably smaller comparing to that observed on randomly generated

Table 1 Performance of conic MIR cuts for randomly generated MIpOCP problems. The "% better" column represents the percentage of problem instances for which conic MIR cuts approach outperformed CPLEX with default parameters in terms of solution time and number of nodes, respectively. "Difficult" instances represent problem instances that cannot be solved at the root node.

			$p = 2.0$					
			all instances			"difficult" instances		
n	m		default CPLEX	conic MIR	% better	default CPLEX	conic MIR	% better
500	200	time	26.88	22.88	29.41%	58.22	43.77	61.11%
		nodes	2.0	0.75	100.00%	5.67	2.11	100.0%
		cuts	16.74	48.65	–	16.06	50.94	–
	600	time	218.0	224.72	52.83%	356.27	369.85	67.86%
		nodes	3.34	3.17	92.45%	6.32	6.0	85.71%
		cuts	73.45	53.90	–	19.08	55.82	–
	1000	time	1117.45	856.59	45.61%	2045.46	1418.66	65.22%
		nodes	1.68	0.60	96.49%	4.17	1.48	91.30%
		cuts	102.54	63.40	–	76.00	50.87	–

			$p = 3.0$					
			all instances			"difficult" instances		
n	m		default CPLEX	conic MIR	% better	default CPLEX	conic MIR	% better
500	200	time	12.60	11.10	37.25%	24.11	20.68	76.92%
		nodes	0.88	0.31	100.00%	1.23	3.46	100.0%
		cuts	11.71	49.65	–	11.38	50.94	–
	600	time	189.76	71.90	51.92%	421.64	133.0	87.50%
		nodes	6.92	2.13	100.00%	22.94	7.06	100.00%
		cuts	18.92	54.58	–	15.37	48.26	–
	1000	time	910.04	560.12	66.67%	1741.93	974.53	61.90%
		nodes	1.53	0.35	98.25%	4.14	0.95	95.24%
		cuts	32.81	63.40	–	22.0	50.87	–

			$p = 4.0$					
			all instances			"difficult" instances		
n	m		default CPLEX	conic MIR	% better	default CPLEX	conic MIR	% better
500	200	time	31.92	26.54	35.29%	62.04	48.06	52.17%
		nodes	2.29	0.98	98.04%	5.09	2.17	95.65%
		cuts	26.16	48.65	–	29.17	63.83	–
	600	time	582.88	324.86	43.40%	875.88	471.92	55.88%
		nodes	9.25	8.0	88.84%	14.41	12.47	82.36%
		cuts	76.75	53.91	–	37.87	60.01	–

problems. Note also that a significantly smaller number of cuts were generated in problem instances of this type; moreover, in many cases the default CPLEX optimizer did not add any cuts to the problem.

Portfolio optimization with lot-buying constraints. The results averaged over 30 instances for each problem size are summarized in Table 3. Note that in many instances of problems of this type, no user cuts of the proposed structure have been found. It can also be noted that regardless of the number of cuts found, solution times are rather comparable to those of the default CPLEX optimizer, which may indicate that conic MIR cuts do not make a significant difference in problems of this type.

4.3 Discussion of Results: Lifted Conic Cuts

Portfolio Optimization. For evaluation of the performance of lifted cuts derived in Section 3, we used both types of portfolio optimization problems, with parameters set up as described above. As it has been already noted, each lifted nonlinear cut was replaced by its outer gradient polyhedral approximation. Specifically, the approximation accuracy was set

Table 2 Performance of conic MIR and lifted cuts in cardinality constrained portfolio optimization problems. Entries in bold correspond to the minimum solution time for each row. Results are averaged over 30 instances for each problem size.

					$p = 2.0$					
		default CPLEX			conic MIR cuts			lifted conic cuts		
n	m	time	nodes	cuts	time	nodes	cuts	time	nodes	cuts
100	600	360.97	31.31	0.10	315.98	31.90	3.00	**281.34**	30.59	2.00
	1000	787.16	31.15	0.00	772.44	77.90	3.00	**595.66**	30.77	2.00
	1400	916.18	37.58	0.00	766.14	55.50	3.00	**664.73**	25.8	2.00
150	600	446.11	41.80	0.00	400.02	41.20	3.00	**377.87**	40.20	2.00
	1000	1566.79	53.44	0.00	1436.57	53.20	3.00	**1326.74**	52.33	2.00
	1400	2601.84	40.69	0.00	2343.03	38.83	3.00	**2196.61**	39.92	2.00
					$p = 3.0$					
		default CPLEX			conic MIR cuts			lifted conic cuts		
n	m	time	nodes	cuts	time	nodes	cuts	time	nodes	cuts
100	600	813.62	47.93	0.00	**537.14**	45.63	3.00	610.98	45.35	2.00
	1000	1449.75	49.78	0.00	1216.24	49.90	3.00	**1213.02**	49.67	2.00
	1400	1671.64	36.38	0.00	1518.44	59.87	3.00	**1428.81**	40.2	2.00
150	600	488.07	41.40	0.20	415.92	40.67	3.00	**354.40**	39.80	2.00
	1000	2877.30	80.81	0.05	2661.90	83.87	3.00	**2514.82**	86.71	2.00
	1400	4307.80	70.72	0.11	4006.54	70.43	3.00	**3739.91**	69.89	2.00
					$p = 4.0$					
		default CPLEX			conic MIR cuts			lifted conic cuts		
n	m	time	nodes	cuts	time	nodes	cuts	time	nodes	cuts
100	600	1234.58	47.08	0.10	1186.99	45.83	3.00	**1062.46**	45.58	2.00
	1000	2368.82	45.05	0.00	2204.83	48.20	3.00	**2062.06**	47.87	2.00
	1400	3243.04	33.49	0.00	2630.18	34.40	3.00	**2552.70**	31.48	2.00
150	600	435.52	34.50	0.17	371.95	58.65	3.00	**340.62**	33.33	2.00
	1000	5913.61	94.71	0.00	5451.90	47.95	3.00	**5168.28**	97.57	2.00
	1400	6442.82	62.50	0.05	6087.91	31.30	3.00	**5286.47**	62.85	2.00

at 10^{-3}. Since in this case each cut results in multiple additional linear constraints, we restricted the number of lifted cuts to be added at the root node to two. The results obtained for portfolio optimization problems with cardinality constraints (16) and lot-buying constraints (17), each averaged over 30 problem instances, are summarized in Tables 2 and 3, respectively. We observed similar improvements in computational time for both types of problems. Also, it has been observed that utilization of lifted cuts in portfolio optimization with lot-buying constraints does not generally lead to a reduction in the number of nodes explored in the solution tree. Thus, based on this observation and results of the experiments of the previous section, we can suggest that the observed improvement is probably partially due to considerably less time spent while looking for cuts. In contrast, in portfolio problems with cardinality constraints we observe reductions in both the number of nodes and solution times due to utilization of lifted cuts.

5 Concluding Remarks

The recent progress in solving mixed integer programming problems can partially be attributed to the advances in utilization of valid inequalities for integer and mixed integer sets. Mixed integer cuts allow for tightening of the bounds given by the continuous relaxation of the problem during the branch-and-cut procedure and, as a result, can lead to reductions in the number of nodes explored in the branch-and-bound tree and in the overall computational time. Typically, valid inequalities exploit specific structure of the feasible set of the problem.

This paper presents two families of valid inequalities for mixed integer p-order programming problems that arise in risk-averse stochastic optimization with downside risk measures.

Table 3 Performance of conic MIR and lifted cuts in portfolio optimization problems with lot-buying constraints. Entries in bold correspond to the minimum solution time for each row. Results are averaged over 30 instances for each problem size.

					$p = 2.0$					
		default CPLEX			conic MIR cuts			lifted conic cuts		
n	m	time	nodes	cuts	time	nodes	cuts	time	nodes	cuts
10	200	9.09	4.13	1.50	9.59	5.10	0.00	**8.03**	5.31	2.00
	600	45.53	4.67	2.61	40.08	5.57	0.13	**32.98**	6.17	2.00
	1000	117.78	11.47	2.37	111.44	13.97	0.33	**102.81**	14.74	2.00
20	200	42.49	20.79	3.64	37.17	23.13	0.40	**32.00**	25.36	2.00
	600	103.28	12.80	5.00	101.67	16.93	0.13	**94.96**	20.16	2.00
	1000	188.04	13.63	3.19	177.53	13.83	1.10	**168.88**	13.63	2.00
50	200	54.50	42.94	4.38	51.21	45.40	0.50	**46.55**	47.44	2.00
	600	307.66	33.19	6.19	286.28	41.27	1.50	**268.13**	46.75	2.00
	1000	640.82	49.71	3.71	**635.54**	62.03	0.00	664.29	69.35	2.00

					$p = 3.0$					
		default CPLEX			conic MIR cuts			lifted conic cuts		
n	m	time	nodes	cuts	time	nodes	cuts	time	nodes	cuts
10	200	18.56	4.79	3.57	17.33	7.73	0.03	**15.50**	9.50	2.00
	600	49.60	8.33	2.22	42.32	9.73	0.03	**34.46**	10.39	2.00
	1000	96.15	10.19	2.38	94.93	12.97	0.03	**90.25**	15.38	2.00
20	200	34.05	9.06	3.11	27.11	10.97	1.10	**21.23**	12.00	2.00
	600	96.98	9.51	4.22	79.78	12.00	1.10	**66.74**	13.84	2.00
	1000	**130.59**	4.53	4.35	134.93	4.67	1.23	141.49	4.53	2.00
50	200	78.29	30.55	5.10	70.07	35.93	0.03	**57.25**	39.95	2.00
	600	316.89	37.39	5.33	275.04	38.17	0.03	**210.81**	37.67	2.00
	1000	540.25	22.58	5.37	500.46	36.87	1.00	**459.55**	47.74	2.00

					$p = 4.0$					
		default CPLEX			conic MIR cuts			lifted conic cuts		
n	m	time	nodes	cuts	time	nodes	cuts	time	nodes	cuts
10	200	23.29	6.29	2.29	17.93	6.13	2.00	**13.58**	5.71	2.00
	600	44.50	3.57	2.21	41.56	3.93	7.03	**37.73**	4.21	2.00
	1000	**122.08**	8.00	2.29	123.10	10.13	25.03	125.04	12.71	2.00
20	200	49.11	7.93	4.07	43.88	16.07	0.13	**40.19**	20.40	2.00
	600	110.42	16.47	3.31	101.32	18.00	12.50	**89.95**	18.24	2.00
	1000	315.87	10.89	4.94	279.44	11.10	34.23	**256.45**	10.89	2.00
50	200	127.20	43.78	5.17	118.54	46.67	0.46	**112.06**	48.06	2.00
	600	416.48	36.76	4.68	344.87	33.93	21.40	**294.47**	29.32	2.00
	1000	993.53	44.50	5.71	825.43	46.20	33.17	**682.21**	56.59	2.00

Particularly, we developed mixed integer rounding cuts and nonlinear lifted cuts for mixed integer p-order conic sets, extending the corresponding results for mixed integer second order programming problems [7,8]. Computational studies on randomly generated problems as well as discrete portfolio optimization problems with historical data demonstrate that both conic MIR cuts and lifted conic cuts lead to improved solution times.

In general, nonlinear cuts are not yet as prevalent as linear ones, partly due to the fact that additional nonlinear inequalities in the bounding (relaxed) problem tend to have deteriorating effect on the computational time of branch-and-bound procedure. In order to improve the computational tractability of the derived nonlinear lifted cuts within the branch-and-cut framework, we proposed replacing them with their polyhedral approximations; since the nonlinear lifted cuts constitute low-dimensional p-cones, the corresponding polyhedral approximations are relatively inexpensive. In this respect, our computational results are among the first successful applications of nonlinear cuts in nonlinear mixed integer programming problems.

A A Direct Derivation of Conic Mixed Integer Rounding Cuts for Mixed Integer p-Order Cone Programming Problems

Following [7], let us first consider a simple case of the following set

$$T = \{(y,w,t,x) \in \mathbb{R}^3_+ \times \mathbb{Z} : [x+y-w-b]_+ \le t\}.$$

Let us denote by $\mathrm{relax}(T)$ the continuous relaxation of T and by $\mathrm{conv}(T)$ its convex hull. It can be seen that the extreme rays of $\mathrm{relax}(T)$ are as follows: $(1,0,0,1)$, $(-1,0,0,0)$, $(1,0,1,0)$, $(-1,1,0,0)$, and its only extreme point is $(b,0,0,0)$. Let us also denote $f = b - \lfloor b \rfloor$. Clearly, the case of $f = 0$ is not interesting, hence it can be assumed that $f > 0$, whereby $\mathrm{conv}(T)$ has four extreme points: $(\lfloor b \rfloor,0,0,0)$, $(\lfloor b \rfloor, f,0,0)$, $(\lceil b \rceil,0,1-f,0)$, $(\lceil b \rceil,0,0,1-f)$. With these observations in mind we can formulate the following proposition.

Proposition A.1 *Inequality*

$$(1-f)(x-\lfloor b \rfloor) \le t+w \tag{18}$$

is valid for T and cuts off all points in $\mathrm{relax}(T) \setminus \mathrm{conv}(T)$.

Proof First, let us show the validity of (18). The base inequality for T is

$$[x+y-w-b]_+ \le t. \tag{19}$$

Now, let $x = \lfloor b \rfloor - \alpha$ and $\alpha \ge 0$. In this case, (19) turns into $t \ge [y-w-f-\alpha]_+$ and (18) becomes $t \ge -(1-f)\alpha - w$. Observing that $[y-w-f-\alpha]_+ - (-(1-f)\alpha - w) = \max\{y-f-\alpha f, (1-f)\alpha + w\} \ge 0$, one obtains that (19) implies (18) for $x \le \lfloor b \rfloor$.

On the other hand, if $x = \lceil b \rceil + \alpha$ with $\alpha \ge 0$, then (19) becomes $t \ge [y-w+(1-f)+\alpha]_+$ and (18) turns into $t \ge (1-f)(1+\alpha) - w$. Similarly to above,

$$\begin{aligned}
&[y-w+(1-f)+\alpha]_+ - ((1-f)(1+\alpha)-w) \\
&= \max\{y-w+(1-f)+\alpha-(1-f)-\alpha(1-f)+w, w-(1-f)(1+\alpha)\} \\
&= \max\{y+\alpha f, w-(1-f)(1+\alpha)\} \ge 0,
\end{aligned}$$

which means that (19) implies (18) for $x \ge \lceil b \rceil$. Hence, (18) is valid for T.

To prove the remaining part of the proposition, consider the polyhedron $\acute{T}$ defined by the inequalities

$$x+y-w-b \le t, \tag{20}$$
$$0 \le t, \tag{21}$$
$$0 \le y, \tag{22}$$
$$0 \le w, \tag{23}$$
$$(1-f)(x-\lfloor b \rfloor) \le t+w. \tag{24}$$

Since $\acute{T}$ has four variables, the basic solutions of $\acute{T}$ are defined by four of these inegualities at equality. They are:

- Inequalities (20), (21), (22), (23): $(x,y,w,t) = (b,0,0,0)$ is infeasible if $f \ne 0$.
- Inequalities (20), (21), (22), (24): $(x,y,w,t) = (\lceil b \rceil,0,1-f,0)$.
- Inequalities (20), (21), (23), (24): $(x,y,w,t) = (\lfloor b \rfloor,f,0,0)$.
- Inequalities (20), (22), (22), (24): $(x,y,w,t) = (\lceil b \rceil,0,0,1-f)$.
- Inequalities (21), (23), (22), (24): $(x,y,w,t) = (\lfloor b \rfloor,0,0,0)$.

Hence, $\mathrm{conv}(T)$ has exactly the same extreme points as $\acute{T}$, which completes the proof. □

In the general case, let

$$\widehat{T} = \{(y^+,y^-,t,\mathbf{x}) \in \mathbb{R}^3_+ \times \mathbb{Z}^n_+ : [\mathbf{a}^\top \mathbf{x} + y^+ - y^- - b]_+ \le t\}, \tag{25}$$

and consider the following function

$$\phi_f(a) = \begin{cases} (1-f)n, & n \le a < n+f \\ (1-f)n + (a-n) - f, & n+f \le a < n+1. \end{cases}$$

Proposition A.2 *For $\alpha \neq 0$ the following inequality*

$$\sum_{j=1}^{n} \phi_{f_{|\alpha|}}\Big(\frac{a_j}{|\alpha|}\Big)x_j - \phi_{f_{|\alpha|}}\Big(\frac{b}{|\alpha|}\Big) \leq \frac{t+y^-}{|\alpha|}, \tag{26}$$

where $f_{|\alpha|} = \frac{b}{|\alpha|} - \lfloor\frac{b}{|\alpha|}\rfloor$, is valid for $\widehat{T}$.

Proof First consider the case $\alpha = 1$. We can rewrite the base inequality for (25) as

$$\Big[\Big(\sum_{f_j \leq f} \lfloor a_j \rfloor x_j + \sum_{f_j > f} \lceil a_j \rceil x_j\Big) + \Big(\sum_{f_j \leq f} f_j x_j + y^+\Big) - \Big(\sum_{f_j > f} (1-f_j)x_j + y^-\Big) - b\Big]_+ \leq t,$$

where $f_j = a_j - \lfloor a_j \rfloor$. Observe that

$$\acute{x} = \sum_{f_j \leq f} \lfloor a_j \rfloor x_j + \sum_{f_j > f} \lceil a_j \rceil x_j \in \mathbb{Z}, \quad \acute{y} = \sum_{f_j \leq f} f_j x_j + y^+ \geq 0, \quad \acute{w} = \sum_{f_j > f} (1-f_j)x_j + y^- \geq 0.$$

Hence, we can apply simple conic MIR inequality (18) with variables $(\acute{x}, \acute{y}, \acute{w}, t)$:

$$(1-f)\Big(\sum_{f_j \leq f} \lfloor a_j \rfloor x_j + \sum_{f_j > f} \lceil a_j \rceil x_j - \lfloor b \rfloor\Big) \leq t + \sum_{f_j > f} (1-f_j)x_j + y^-.$$

Rewriting it with the help of function $\phi_f(a)$, we obtain that

$$\sum_{j=1}^{n} \phi_f(a_j)x_j - \phi_f(b) \leq t + y^-.$$

So, by Proposition A.1 inequality (26) is valid for $\alpha = 1$. In order to see that the result holds for all $\alpha \neq 0$ we only need to scale the base inequality:

$$\Big[\frac{1}{|\alpha|}(\mathbf{a}^\top \mathbf{x} + y^+ - y^- - b)\Big]_+ \leq \frac{t}{|\alpha|}.$$

□

Proposition A.3 *Inequalities* (26) *with $\alpha = a_j$, $j = 1, \ldots, n$ are sufficient to cut off all fractional extreme points of* relax$(\widehat{T})$.

Proof The set relax$(\widehat{T})$ is defined by $n+3$ variables and $n+4$ constraints. Therefore, if $x_j > 0$ in an extreme point, then the remaining $n+3$ constraints must be active. Thus, the continuous relaxation has at most n fractional extreme points $(x^j, 0, 0, 0)$ of the form $x^j_j = \frac{b}{a_j} > 0$, and $x^j_i = 0$, for $i \neq j$. Such points are infeasible if $\frac{b}{a_j} \notin \mathbb{Z}$. Now, let $a_j > 0$. For such a fractional extreme point inequality (26) reduces to

$$\phi_{f_{a_j}}(1)x_j - \phi_{f_{a_j}}\Big(\frac{b}{a_j}\Big) \leq \frac{t+y^-}{a_j}, \quad \text{or} \quad (1-f_{a_j})x_j - (1-f_{a_j})\Big\lfloor\frac{b}{a_j}\Big\rfloor \leq \frac{t+y^-}{a_j},$$

which by Proposition A.1 cuts off fractional extreme point with $x^j_j = \frac{b}{a_j}$.

Now, let us consider $a_j < 0$. In this case we observe that the inequality (26) reduces to

$$\phi_{f_{|a_j|}}(-1)x_j - \phi_{f_{|a_j|}}\Big(\frac{b}{|a_j|}\Big) \leq \frac{t+y^-}{|a_j|}, \quad \text{or} \quad -(1-f_{|a_j|})x_j - (1-f_{|a_j|})\Big\lfloor\frac{b}{|a_j|}\Big\rfloor \leq \frac{t+y^-}{|a_j|},$$

which again, cuts off fractional extreme point with $x^j_j = \frac{b}{a_j}$. □

References

1. Vielma, J.P., Ahmed, S., Nemhauser, G.L.: A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. INFORMS J. Comput. **20**, 438–450 (2008)
2. Ben-Tal, A., Nemirovski, A.: On polyhedral approximations of the second-order cone. Math. Oper. Res. **26**, 193–205 (2001)
3. Vinel, A., Krokhmal, P.: On polyhedral approximations for solving *p*-order conic programming problems. Working paper (2012)
4. Drewes, S.: Mixed integer second order cone programming. Ph.D. thesis, Technische Universität Darmstadt, Germany (2009)
5. Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wächter, A.: An algorithmic framework for convex mixed integer nonlinear programs. Discrete Optim. **5**, 186–204 (2008)
6. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Math. Program. **103**, 225–249 (2005)
7. Atamtürk, A., Narayanan, V.: Conic mixed-integer rounding cuts. Math. Program. **122**, 1–20 (2010)
8. Atamtürk, A., Narayanan, V.: Lifting for conic mixed-integer programming. Math. Program. **126**, 351–363 (2011)
9. Belotti, P., Goez, J., Polik, I.and Ralphs, T., Terlaky, T.: A conic representation of the convex hull of disjunctive sets and conic cuts for integer second order cone optimization. Working paper (2012)
10. Dadush, D., Dey, S.S., Vielma, J.P.: The split closure of a strictly convex body. Oper. Res. Lett. **39**, 121–126 (2011)
11. Stubbs, R.A., Mehrotra, S.: A branch-and-cut method for 0-1 mixed convex programming. Math. Program. **86**, 515–532 (1999)
12. Çezik, M.T., Iyengar, G.: Cuts for mixed 0-1 conic programming. Math. Program. **104**, 179–202 (2005)
13. Krokhmal, P.A.: Higher moment coherent risk measures. Quant. Finance **7**, 373–387 (2007)
14. Krokhmal, P., Zabarankin, M., Uryasev, S.: Modeling and optimization of risk. Surveys in Operations Researh and Management Science **16**, 49–66 (2011)
15. Nemhauser, G.L., Wolsey, L.A.: Integer and combinatorial optimization. Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons Inc., New York, a Wiley-Interscience Publication (1988)
16. Krokhmal, P.A., Soberanis, P.: Risk optimization with *p*-order conic constraints: A linear programming approach. European J. Oper. Res. **201**, 653–671 (2010)
17. Nesterov, Y.E., Nemirovski, A.: Interior Point Polynomial Algorithms in Convex Programming, volume 13 of Studies in Applied Mathematics. SIAM, Philadelphia, PA (1994)
18. Alizadeh, F., Goldfarb, D.: Second-order cone programming. Math. Program. **95**, 3–51 (2003)
19. Morenko, Y., Vinel, A., Yu, Z., Krokhmal, P.: On *p*-cone linear discrimination. Submitted for publication (2012)
20. Perold, A.F.: Large-scale portfolio optimization. Management Sci. **30**, 1143–1160 (1984)
21. Bonami, P., Lejeune, M.A.: An exact solution approach for portfolio optimization problems under stochastic and integer constraints. Oper. Res. **57**, 650–670 (2009)
22. Scherer, B., Martin, R.D.: Introduction to modern portfolio optimization with NUOPT and S-PLUS. Springer, New York (2005)

On risk-averse weighted *k*-club problems

Maciej RYSZ [a], Foad PAJOUH [b], Pavlo KROKHMAL [a,1], and Eduardo PASILIAO [c]

[a] *Department of Mechanical and Industrial Engineering,*
The University of Iowa, 3131 Seamans Center, Iowa City, IA, 52242
[b] *Department of Industrial & Systems Engineering,*
Research and Engineering Education Facility (UF-REEF),
University of Florida, 1350 N. Poquito Road, Shalimar, FL 32579
[c] *Air Force Research Lab,*
101 West Eglin Blvd, Eglin AFB, FL 32542

Abstract. In this work, we consider a risk-averse maximum weighted *k*-club problems. It is assumed that vertices of the graph have stochastic weights whose joint distribution is known. The goal is to find the *k*-club of minimum risk contained in the graph. A stochastic programming framework that is based on the formalism of coherent risk measures is used to find the corresponding subgraphs. The selected representation of risk of a subgraph ensures that the optimal solutions are maximal *k*-clubs. A combinatorial branch-and-bound solution algorithm is proposed and solution performances are compared with an equivalent mathematical programming counterpart problem for instances with $k = 2$.

Keywords. *k*-club, clique relaxation, risk-averse subgraph problem, stochastic weights, coherent risk measures.

1. Introduction

A principal class of graph theoretical problems involves the identification of embodied subgraphs corresponding to some structural property. One particular setting of fundamental importance entails finding the largest "perfectly" cohesive group within a network such that the confined members are all interconnected, i.e., the largest *clique* (*complete subgraph*). Several prominent studies founded the basis for exact combinatorial solution algorithms for the maximum clique problem [1, 2, 3]. In particular, Carraghan and Pardalos [2] introduced a recursive branch-and-bound method for efficient finding maximum cliques by exploiting the heredity property [4] of complete subgraphs. Subsequent extensions of their work enhanced the process of eliminating solution space via vertex coloring schemes for *branching* and upper-bounds estimation on the maximal achievable subgraph sizes during the algorithmic processing (e.g. [5, 6, 7]). In many practical applications, the requirement that the desired subgraph must be complete may, however, impose excessive restrictions, and warrant some structural relaxation in terms of member connectivity. As a consequence, several clique relaxation models have been proposed in graph theory literature. A comprehensive review on clique relaxation models is provided

[1] Corresponding Author, E-mail: krokhmal@engineering.uiowa.edu.

in [4]. In this work we focus on a specific model, the *k-club* [8], where subgraph members may also be indirectly connected via at most k intermediary members.

A popular extension of the described above class of problems involves the imposition topologically exogenous information in the form of deterministic vertex weights, and correspondingly finding a subset of maximum weight that conforms to a defined structural property. Similar exact weight-based branch-and-bound solution techniques have been developed for determining the maximum-weight subgraphs [9, 10, 11].

Particular circumstances may further justify the imposition of uncertain exogenous information over the graph's edges that influences network flow distribution, robustness, and costs [12, 13, 14, 15, 16, 17]. However, far fewer endeavors concern decision making regarding optimal resource allocation over defined subgraph topologies when uncertainties are induced by stochastic factors associated with network vertices. In this study, we adopt this setting and extend the techniques introduced in [18] to address problems seeking subgraphs of minimum risk that represent a k-club. A statistical framework utilizing the distributional information of stochastic vertex weights by means of *coherent risk measures* [19, 20] is employed to define a *risk-averse maximum weighted k-club* (R-MWK) problem as finding the lowest risk k-club in a network. As an illustrative example, we focus on instances when $k = 2$ and utilize a mathematical programming formulation for the maximum 2-club problem introduced in [21]. A branch-and-bound method for finding maximum k-clubs [22] is modified to accommodate the conditions of R-MWK problems by bounding solutions in a coherent risk measure context. We compare the solution performance of the proposed algorithm relative to an equivalent mathematical programming counterpart problem for R-MWK problems when $k = 2$.

The remainder of the paper is organized as follows. In Section 2 we examine the general representation of R-MWK problems and consider their properties. Section 3 presents a mathematical programming formulation and a combinatorial branch-and-bound method for R-MWK problems with $k = 2$. Finally, Section 4 furnishes numerical studies demonstrating the computational performance of the developed branch-and-bound method on problems where risk is quantified using higher-moment coherent risk measures [23].

2. Risk-averse stochastic maximum k-club problem

Given an undirected graph $G = (V, E)$ and any subset of its vertices $S \subseteq V$, let $G[S]$ represent the subgraph of G induced by S such that any pair of vertices (i, j) share an edge in S only if (i, j) is an edge in G. To ease notation, define $\mathscr{Q}$ as a desired property which the induced graph $G[S]$ must satisfy. The present work considers the case when $\mathscr{Q}$ represents a certain relaxation of the *completeness* property, such that a subgraph with property $\mathscr{Q}$ represents a *clique relaxation*.

Depending on the characteristic of a complete graph that is relaxed, the clique relaxations can be categorized into *density-based*, *degree-based*, and *diameter-based* relaxations. The density of a graph $G = (V, E)$ is defined as a ratio $D(G) = |E|/\binom{|V|}{2}$, where the denominator represents the number of edges in a complete graph with $|V|$ vertices. Evidently, a complete graph (clique) has a density of 1. Then, for a fixed $\gamma \in (0, 1)$, graph G is called a *γ-quasi-clique* [24], if its density is at least γ:

$$D(G) \geq \gamma, \quad \text{or, equivalently,} \quad |E| \geq \gamma \binom{|V|}{2}.$$

The γ-quasi-clique is, therefore, a density-based relaxation of the clique concept, and as such is different from the *k-clique*, which is one of the diameter-based clique relaxations. Namely, let $d_G(i,j)$ be the distance between nodes $i,j \in V$, measured as the number of edges in the shortest path between i and j in G. Then, the subgraph $G[S]$ induced by a subset of nodes $S \subset V$ of the graph G is called a k-clique if

$$\max_{i,j \in S} d_G(i,j) = k.$$

Note that the definition of the k-clique does not require that the shortest path between $i,j \in S$ belong to $G[S]$. If one requires that the shortest path between any two vertices i,j in S belong to the induced subgraph $G[S]$, then the subset S such that

$$\max_{i,j \in S} d_{G[S]}(i,j) = k, \tag{1}$$

is called a *k-club*. Note that a k-club is also a k-clique, while the inverse is not true in general. The shortest path connecting two vertices in a clique is 1, thus 1-clique and 1-club are cliques. For a vertex $i \in V$, its *degree* $\deg_G(i)$ is defined as the number of adjacent vertices: $\deg_G(i) = |\{j \in V : (i,j) \in E\}|$. A degree-based clique relaxation, known as *k-plex,* is defined as a subset S of V such that the degree of each vertex in the induced subgraph $G[S]$ is at least $|S| - k$ [25]:

$$\deg_{G[S]}(i) \geq |S| - k \quad \text{for all} \quad i \in S,$$

(observe that the degree of each vertex in a clique of size n is equal to $n-1$).

The present work considers the case when $\mathscr{Q}$ represents a distance-based relaxation of the clique model in the sense of k-club definition (1) when $k \geq 2$. Throughout the remainder of this study we let property $\mathscr{Q}_{G[S]}$ define a k-club as

$$\mathscr{Q}_{G[S]} = \{S \subseteq V \mid \forall i,j \in S : d_{G[S]}(i,j) \leq k\}. \tag{2}$$

A popular instance of graph-theoretic problems arises when seeking a subgraph S with the maximum additive vertex weights, $w_i > 0$, that satisfies property $\mathscr{Q}_{G[S]}$. When $\mathscr{Q}_{G[S]}$ is defined by (2) a *maximum weight k-club problem* can take the form

$$\max_{S \subseteq V} \left\{ \sum_{i \in S} w_i : G[S] \text{ satisfies } \mathscr{Q}_{G[S]} \right\}. \tag{3}$$

Clearly, the optimal subgraph $G[S]$ in problem (3) will be *maximal*, but not necessarily the *maximum* (of the largest order) subgraph with property $\mathscr{Q}_{G[S]}$.

In this work, we consider an extension of problem (3) that assumes stochastic vertex weights. In this case, a direct translation into a stochastic framework is not trivial due to the fact that the maximization of random weights would be ill-posed in context of stochastic programming resulting from the absence of a deterministic optimal solution. Likewise, maximization of the expected weight of the sought subgraph is not interesting in the sense that it reduces to the deterministic version of the problem presented above. A more suitable approach, thus, involves computing the subgraph's weight via a statistical functional that utilizes the distributional information about the weights' uncertainties, rather than as a simple sum of its (random) weights. To this end, we pursue a *risk-averse* approach so as to find the subgraph of G that has the *lowest risk* and satisfies the property $\mathscr{Q}$. Let X_i denote random variables that represent costs of losses associated with vertices $i \in V$, such that the joint distribution of vector $\mathbf{X}_G = (X_1, \ldots, X_{|V|})$ is known. The problem of finding the *minimum-risk* subgraph in G with property $\mathscr{Q}$, or the risk-averse maximum weighted $\mathscr{Q}$ problem take the form:

$$\min_{S \subseteq V} \big\{ \mathscr{R}(S; \mathbf{X}_G) : G[S] \text{ satisfies } \mathscr{Q} \big\}, \tag{4}$$

where $\mathscr{R}(S; \mathbf{X}_G)$ is the risk of the induced subgraph $G[S]$ given the distributional information $\mathbf{X}_G$.

A formal representation of risk $\mathscr{R}(S; \mathbf{X}_G)$ is invoked via the well-known concept of *risk measure* in stochastic optimization literature [26]. Namely, given a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, where Ω is the set of random events, $\mathscr{F}$ is the σ-algebra, and $\mathbb{P}$ is a probability measure, a risk measure is defined as a mapping $\rho : \mathscr{X} \mapsto \mathbb{R}$, where $\mathscr{X}$ is a linear space of $\mathscr{F}$-measurable functions $X : \Omega \mapsto \mathbb{R}$. Further, assuming that risk measure ρ is lower semi-continuous (l.s.c.), the risk $\mathscr{R}(S; \mathbf{X}_G)$ of subgraph of $G[S]$ with uncertain vertex weights X_i can be defined as an optimal value of the following stochastic programming problem:

$$\mathscr{R}(S; \mathbf{X}_G) = \min \left\{ \rho\left(\sum_{i \in S} u_i X_i \right) : \sum_{i \in S} u_i = 1,\ u_i \geq 0,\ i \in S \right\}. \tag{5}$$

Notice that this definition of the subgraph risk function $\mathscr{R}(\cdot)$ admits risk reduction through diversification as illustrated by the following proposition:

Proposition 1 ([18]) *Given a graph $G = (V, E)$ with stochastic weights X_i, $i \in V$, and a l.s.c. risk measure ρ, the subgraph risk function $\mathscr{R}$ defined by* (5) *satisfies*

$$\mathscr{R}(S_2; \mathbf{X}_G) \leq \mathscr{R}(S_1; \mathbf{X}_G) \quad \textit{for all} \quad S_1 \subseteq S_2. \tag{6}$$

The following observation regarding the optimal solution of the risk-averse maximum weighted $\mathscr{Q}$ problem (4) stems directly from property (6):

Corollary 1 *There exists an optimal solution of the risk-averse maximum weighted $\mathscr{Q}$ problem* (4) *with $\mathscr{R}(S; \mathbf{X}_G)$ defined by* (5) *that is a maximal $\mathscr{Q}$-subgraph in G.*

Additional properties of $\mathscr{R}(S; \mathbf{X}_G)$ ensue from the assumption that risk measure ρ belongs to the family of coherent measures of risk. Namely, the definition of ρ is augmented with the properties of monotonicity, subadditivity, transitional invariance, and positive homogeneity (see [19]). Assuming that risk measure ρ in (5) is coherent, or satisfies the first three properties and is l.s.c, then the corresponding subgraph risk function $\mathscr{R}(S; \mathbf{X}_G)$ satisfies analogous properties with respect to the stochastic weights vector $\mathbf{X}_G$,

(G1) *monotonicity*: $\mathscr{R}(S;\mathbf{X}_G) \leq \mathscr{R}(S;\mathbf{Y}_G)$ for all $\mathbf{X}_G \leq \mathbf{Y}_G$;
(G2) *positive homogeneity*: $\mathscr{R}(S;\lambda\mathbf{X}_G) = \lambda\mathscr{R}(S;\mathbf{X}_G)$ for all $\mathbf{X}_G$ and $\lambda > 0$;
(G3) *transitional invariance*: $\mathscr{R}(S;\mathbf{X}_G + a\mathbf{1}) = \mathscr{R}(S;\mathbf{X}_G) + a$ for all $a \in \mathbb{R}$;

where $\mathbf{1}$ is the vector of ones, and the vector inequality $\mathbf{X}_G \leq \mathbf{Y}_G$ is interpreted component-wise.

Observe that $\mathscr{R}(S;\mathbf{X}_G)$ violates the sub-additivity requirements with respect to the stochastic weights. However, risk reduction via diversification is guaranteed by (6), which ensures that the inclusion of additional vertices to the existing feasible solution is always beneficial. Further, under an assumption of non-negative stochastic vertex weights, $\mathbf{X}_G \geq \mathbf{0}$, the subgraph risk $\mathscr{R}(S;\mathbf{X}_G)$ can be shown to be subadditive in relative to induced subgraphs in G,

$$\mathscr{R}(S_1 \cup S_2;\mathbf{X}_G) \leq \mathscr{R}(S_1;\mathbf{X}_G) + \mathscr{R}(S_2;\mathbf{X}_G), \quad S_1, S_2 \subseteq V. \tag{7}$$

Clearly, it is required that S_1, S_2, and $S_1 \cup S_2$ satisfy property $\mathscr{Q}$ in conformance to the context of risk-averse maximum weighted $\mathscr{Q}$ problems.

3. Solution approaches for risk-averse maximum weighted 2-club problems

In this section we consider a mathematical programming formulation for the R-MWK problem when $k = 2$, and where the risk $\mathscr{R}(S)$ of induced subgraph $G[S]$ is defined by (5). Also, we propose a combinatorial branch-and-bound algorithm utilizing the solution space processing principals for finding maximum k-clubs introduced by Pajouh and Balasundaram [22].

3.1. A mathematical programming formulation

Let binary decision variables x_i indicate whether node $i \in V$ belongs to a subset S:

$$x_i = \begin{cases} 1, & i \in S \text{ such that } G[S] \text{ satisfies } \mathscr{Q} \\ 0, & \text{otherwise.} \end{cases}$$

When the property $\mathscr{Q}$ denotes a 2-club, one can choose the *edge formulation* of the maximum 2-club problem proposed by Balasundaram et al. [21], whereby the mathematical programming formulation of the R-MWK problem with $k = 2$ takes the form

$$\begin{aligned} \min \quad & \rho\Big(\sum_{i \in V} u_i X_i\Big) \\ \text{s. t.} \quad & \sum_{i \in V} u_i = 1, \\ & u_i \leq x_i, \quad i \in V, \\ & x_i + x_j - \sum_{l \in \mathscr{N}^{\cap}(i,j)} x_l \leq 1, \quad (i,j) \in \overline{E}, \\ & x_i \in \{0,1\}, \;\; u_i \geq 0, \quad i \in V, \end{aligned} \tag{8}$$

where $\overline{E}$ represents the complement edges of graph G, and $\mathscr{N}^{\cap}(i,j)$ denotes the vertices that are both adjacent to vertex i and vertex j. Appropriate (nonlinear) mixed integer programming solvers can be used to solve formulation (8) with risk measures ρ whose representations admits some form of mathematical programming problems. A combinatorial branch-and-bound algorithm for solving R-MWK problems is described next.

3.2. A combinatorial branch-and-bound algorithm

The following branch-and-bound (BnB) algorithm for solving R-MWK problems entails efficient processing of solution space by traversing "levels" of the BnB tree until a subgraph $G[S]$ that represents a maximal 2-club of minimum risk in G as measured by (5) is found. The algorithm begins at level $\ell = 0$ with a partial solution $Q := \emptyset$, incumbent solution $Q^* := \emptyset$, and an upper bound on risk $L^* := +\infty$ (risk induced by Q^*), where Q consists of the vertices of the induced subgraph with property $\mathscr{Q}$, and Q^* contains vertices corresponding to a maximal $\mathscr{Q}$-subgraph whose risk equals L^* in G. A set of "candidate" vertices C_ℓ is maintained at each level ℓ, from which a certain *branching* vertex q is selected and added to the partial solution Q, or simply deleted from set C_ℓ without being added to Q. In order to ensure that the proper vertices are removed from Q when the algorithm backtracks between levels of the BnB tree, we introduce set $F := \emptyset$ to account for the levels at which nodes were created to delete a vertex q from C_ℓ.

Due to the distance-based properties of k-clubs, considerations are warranted upon transferring or deleting a vertex q from candidate set C_ℓ, as the structural integrity of corresponding to the graph induced by Q and the candidate set at the subsequent level $\cup C_{\ell+1}$ may be affected. Thus, the removal of q from C_ℓ to add to Q, and the deletion of q from C_ℓ without adding it to Q are considered independently via the construction of two BnB tree nodes for any given current node at level ℓ. The first node is created to include q in Q, while the other to delete q from C_ℓ. The necessary structural properties of Q and $C_{\ell+1}$ at each node are described next.

Consider a k-clique in graph G as a subset S that satisfies

$$\{S \subseteq V \mid \forall i,j \in S : d_G(i,j) \leq k\},$$

and observe that any k-club in G also satisfies the properties of a k-clique, while a k-clique is not necessarily a k-club for $k \geq 2$. Further, both reduces to a complete graph in the case of $k = 1$. By this notion, an incumbent solution Q^* defines a k-club if the following conditions are maintained for all graphs $G[Q \cup C_{\ell+1}]$:

(C1) Q is a k-clique in $G[Q \cup C_{\ell+1}]$
(C2) $d_{G[Q \cup C_{\ell+1}]}(i,j) \leq k, \ \forall i \in Q, \ \forall j \in C_{\ell+1}$

The algorithm is then initialized with $C_0 := V$. Whenever a vertex q is selected from C_ℓ and added to Q, the candidate set at level $\ell+1$ must be accordingly constructed by removing all vertices from C_ℓ whose distances to vertex in q are larger than k,

$$C_{\ell+1} := \{j \in C_\ell : d_{G[Q \cup C_\ell]}(q,j) \leq k\}.$$

In situations when the deleted vertices serve as intermediaries, their removal from C_ℓ may, however, impose pairwise distance violations among the vertices in $Q \cup q$ with respect to condition (C2). In other words, after removing vertex q from C_ℓ, the distance between a pair of vertices $(i,j) \in Q$ follows $d_{G[Q \cup C_{\ell+1}]}(i,j) > k$. In such cases, the corresponding node of the BnB tree is fathomed and the algorithm backtracks to level ℓ. If a BnB tree node is created to delete vertex q, the candidate set $C_{\ell+1}$ is likewise constructed by eliminating vertices that violate (C2). If the removal of vertices from the candidate sets in either of the above cases results in a violation of (C1), then the corresponding BnB node is fathomed.

The subsequent step entails evaluating the quality of the solution that can be obtained from the subgraph induced by vertices in $Q \cup C_{\ell+1}$. An exact approach of directly finding the 2-club with the lowest possible risk that is contained in $G[Q \cup C_{\ell+1}]$ would involve solving problems (8) where $x_i = 0, \quad i \in V \setminus (Q \cup C_{\ell+1})$. However, solving a mixed 0–1 problem at every node of the BnB tree is impractical, and a lower bound problem is obtained by eliminating variables x_i, $i \in V$, and the graph structural constraints,

$$\begin{aligned} \mathscr{R}(Q \cup C_{\ell+1}; \mathbf{X}_G) \geq \mathscr{L}(Q \cup C_{\ell+1}) := \min \quad & \rho\Big(\sum_{i \in V} u_i X_i\Big) \\ \text{s. t.} \quad & \sum_{i \in V} u_i = 1 \\ & u_i = 0, \quad i \in V \setminus (Q \cup C_{\ell+1}) \\ & u_i \geq 0, \quad i \in Q \cup C_{\ell+1}. \end{aligned} \tag{9}$$

This notion admits the assumption that $G[Q \cup C_{\ell+1}]$ is a 2-club, under which all the mentioned graph structural constraints would be satisfies and thus vanish. Therefore, by virtue of Proposition 1, the solution to (9) provides a lower bound on the risk achievable by any 2-club contained in the graph induced via the union of vertices in Q and any subset of vertices in $C_{\ell+1}$. As a result, the risk at any subsequent level ℓ' along the current branch of the BnB tree cannot deteriorate as the set $Q \cup C_{\ell'+1}$ is refined.

The computed values of $\mathscr{L}(Q \cup C_{\ell+n})$ determine whether the algorithm branches further or prunes/backtracks. If $\mathscr{L}(Q \cup C_{\ell+n}) \geq L^*$, then the corresponding branch of the BnB tree is fathomed due to the fact that sequential refinement can not achieve a further reduction in risk. If $C_\ell \neq \emptyset$, another branching vertex is selected and either removed from C_ℓ and added to Q, or deleted from C_ℓ. Alternatively, if $C_\ell = \emptyset$, the algorithm backtracks to level $\ell - 1$.

In the case when $\mathscr{L}(Q \cup C_{\ell+1}) < L^*$ and $C_{\ell+1} \neq \emptyset$, the a branching vertex q is selected at the next level $\ell + 1$. In the case of $\mathscr{L}(Q \cup C_{\ell+1}) < L^*$ and $C_{\ell+1} = \emptyset$, the $G[Q]$ represents a maximal 2-club in G and is assigned as the new incumbent solution, $Q^* := Q$, and the global upper bound on risk is updated $L^* := \mathscr{L}(Q \cup C_{\ell+1})$. The algorithm then backtracks to level $\ell - 1$.

Empirical experimental observations suggest that branching on a vertex q with the smallest value of $\rho(X_q)$ or $\mathbb{E}X_q$ can significantly enhance computational performance. To this end, the vertices in any candidate set C_ℓ are ordered in descending order with respect to their risks $\rho(X_i)$ or expected values $\mathbb{E}X_i$, and the last vertex in C_ℓ is always selected for branching.

The described branch-and-bound algorithm procedure for R-MWK problems is formalized in Algorithm 1. Notice that it is applicable to any positive integer value k.

Algorithm 1 Graph-based branch-and-bound method for problem (8)

```
1.  Initialize: $\ell := 0$; $C_0 := V$; $Q := \emptyset$; $Q^* := \emptyset$; $L^* := \infty$; $F := \emptyset$;
2.  While (not STOP) do
3.    if $C_\ell \neq \emptyset$ then
4.      select a vertex $q \in C_\ell$;
5.      $C_\ell := C_\ell \setminus q$;
6.      $Q := Q \cup q$;
7.      $C_{\ell+1} := \{i \in C_\ell : d_{G[Q \cup C_\ell]}(q,i) \leq k \, \forall i \in C_\ell\}$;
8.      if $Q$ is a $k$-clique in $G[Q \cup C_{\ell+1}]$ then
9.        solve $\mathscr{L}(Q \cup C_{\ell+1})$;
10.       if $\mathscr{L}(Q \cup C_{\ell+1}) < L^*$ then
11.         if $C_{\ell+1} \neq \emptyset$ then
12.           $\ell := \ell + 1$;
13.         else
14.           $Q^* := Q$;
15.           $L^* := \mathscr{L}(Q \cup C_{\ell+1})$;
16.           $Q := Q \setminus q$;
17.           if $\ell \notin F$ then
18.             $Q := Q \setminus q$
19.             $C_{\ell+1} := \{j \in C_\ell : d_{G[Q \cup C_\ell]}(i,j) \leq k, \, \forall i \in Q, \}$;
20.             if $C_{\ell+1} \neq \emptyset$ then
21.               if $Q$ is a $k$-clique in $G[Q \cup C_{\ell+1}]$ then
22.                 $F := F \cup \ell$;
23.                 go to step 9;
24.               else
25.                 go to step 3;
26.           else
27.             $F := F \setminus \ell$;
28.       else
29.         if $\ell \notin F$ then
30.           $Q := Q \setminus q$;
31.         else
32.           $F := F \setminus \ell$;
33.     else
34.       $Q := Q \setminus q$;
35.       $C_{\ell+1} := \{j \in C_\ell : d_{G[Q \cup C_\ell]}(i,j) \leq k, \, \forall i \in Q, \}$;
36.       if $Q$ is a $k$-clique in $G[Q \cup C_{\ell+1}]$ then
37.         $F := F \cup l$;
38.         go to step 9;
39.       else
40.         go to step 3;
41.   else
42.     $\ell := \ell - 1$;
43.     if $\ell = -1$ then
44.       STOP
45.     if $\ell \notin F$ then
46.       $Q := Q \setminus q$;
47.     else
48.       $F := F \setminus l$;
49. return $Q^*$
```

4. Case study: Risk-averse maximum weighted 2-club problem with higher moment coherent risk measures

In this section we present a computational framework for problem (8) and conduct numerical experiments demonstrating the computational performance enhancements associated with the proposed BnB algorithm. We adopt higher-moment coherent risk (HMCR) measure class that was introduced in [23] as optimal values to the following stochastic programming problem:

$$\mathrm{HMCR}_{\alpha,p}(X) = \min_{\eta \in \mathbb{R}} \; \eta + (1-\alpha)^{-1} \left\| (X-\eta)^+ \right\|_p, \quad \alpha \in (0,1), \quad p \geq 1, \tag{10}$$

where $X^+ = \max\{0, X\}$ and $\|X\|_p = \left(\mathbb{E}|X|^p\right)^{1/p}$. Mathematical programming problems that contain HMCR measures can be formulated using p-order cone constraints. Typically, in stochastic programming models, the set of random events Ω is assumed to be discrete, $\Omega = \{\omega_1, \ldots, \omega_N\}$, with the probabilities $\mathbb{P}(\omega_k) = \pi_k > 0$, and $\pi_1 + \cdots + \pi_N = 1$. The corresponding mathematical programming model (8) with $\rho(X) = \mathrm{HMCR}_{p,\alpha}(X)$ takes the following mixed integer p-order cone programming form:

$$\begin{aligned}
\min\quad & \eta + (1-\alpha)^{-1} t \\
\text{s. t.}\quad & t \geq \|(y_1, \ldots, y_N)\|_p, \\
& \pi_k^{-1/p} y_k \geq \sum_{i \in V} u_i X_{ik} - \eta, \quad k = 1, \ldots, N, \\
& \sum_{i \in V} u_i = 1, \\
& u_i \leq x_i, \;\; i \in V, \\
& x_i + x_j - \sum_{l \in \mathscr{N}^\cap(i,j)} x_l \leq 1, \quad (i,j) \in \overline{E}, \\
& x_i \in \{0,1\},\; u_i \geq 0,\; i \in V;\;\; y_k \geq 0,\; k = 1, \ldots, N,
\end{aligned} \tag{11}$$

where X_{ik} represents the realization of the stochastic weight of vertex $i \in V$ under scenario $k \in \mathscr{N}$. Analogously, the lower bound problem (9) takes the form

$$\begin{aligned}
\mathscr{L}(Q \cup C_{\ell+1}) = \min\quad & \eta + (1-\alpha)^{-1} t \\
\text{s. t.}\quad & t \geq \|y_1, \ldots, y_N\|_p, \\
& \pi_k^{-1/p} y_k \geq \sum_{i \in V} u_i X_{ik} - \eta, \quad k = 1, \ldots, N, \\
& \sum_{i \in V} u_i = 1, \\
& u_i \geq 0, \;\; i \in Q \cup C_{\ell+1}, \\
& u_i = 0, \;\; i \in V \setminus (Q \cup C_{\ell+1}), \\
& y_k \geq 0,\; k = 1, \ldots, N.
\end{aligned} \tag{12}$$

For instances when $p = 1$ or 2, problems (11) and (12) reduce to linear programming (LP) and second order cone programming (SOCP) models, respectively. However, in cases when when $p \in (1,2) \cup (2,\infty)$ the p-cone is not self-dual and there exist no efficient long-step self-dual interior point solution methods. Consequently, we employ the methods for representing p-order cones into a higher dimensional space [27] that are based on polyhedral approximations of p-order cones and representation of rational-order p-cones via second order cones.

4.1. Setup of the numerical experiments and results

Numerical experiments of the risk-averse maximum weighted 2-club problem were conducted on randomly generated Erdös-Rényi graphs of orders $|V| = 25, 50, 100$ with average densities $d = 0.0125, 0.025, 0.05, 0.1, 0.15$. The specified edge probabilities were chosen due to empirical observations indicating that a graph of order $|V| \geq 50$ commonly reduces to a 2-club when the density is in the range [0.15,0.25]. The stochastic weights of graphs' vertices were generated as i.i.d. samples from the uniform $U(0,1)$ distribution. Scenario sets with $N = 100$ were generated for each combination of graph order and density. The HMCR risk measure (10) with $p = 1,2,3$, and $\alpha = 0.9$ was used.

The BnB algorithm has been coded in C++, and we used the CPLEX Simplex and Barrier solvers for the polyhedral approximations and SOCP reformulations of the p-order cone programming lower bound problem (12), respectively (see [27]). For instances when $p = 1$, the CPLEX Simplex solver was utilized to solve problem (12) directly. The computations were conducted on an Intel Xeon 3.30GHz PC with 128GB RAM, and the CPLEX 12.5 solver in Windows 7 64-bit environment was used.

The computational performance of the mathematical programming model (11) was compared with that of developed BnB algorithm. In the case of $p = 1$, problem (11) was solved with CPLEX MIP solver. The CPLEX MIP Barrier solver was used for the SOCP version in the case of $p = 2$, and using the SOCP reformulation in the case of $p = 3$.

Table 1 presents the computational times, averaged over five instances. Observe that the BnB algorithm outperforms the CPLEX MIP solver over all the listed graph configurations, and one to two orders of magnitude in performance improvements were witnesses for the majority of instances. Further, the relative differences in performance also become more pronounced with an increase in p. Also noteworthy is improvement in relative performance of the BnB method for problems with $p = 3$ in comparison to $p = 2$. This results from properties of the cutting-plane algorithm for solving polyhedral approximations of p-order cone programming problems, which becomes more effective as p increases [27].

		$d = 0.0125$		$d = 0.025$		$d = 0.05$		$d = 0.1$		$d = 0.15$	
p	$\lvert V\rvert$	CPLEX	BnB	CPLEX	BnB	CPLEX	BnB	CPLEX	BnB	CPLEX	BnB
1	25	0.47	0.06	0.54	0.04	0.46	0.04	0.31	0.04	0.32	0.08
	50	1.32	0.13	0.74	0.14	0.79	0.18	1.29	0.33	2.47	1.91
	100	1.99	0.07	3.25	0.38	6.00	2.19	57.62	40.90	—	—
2	25	11.00	0.56	9.63	0.72	6.24	0.33	6.38	0.37	10.57	0.43
	50	16.20	0.69	14.89	0.52	19.01	0.46	46.19	1.10	167.51	4.91
	100	38.25	0.61	119.15	1.15	253.27	2.91	973.18	70.45	—	—
3	25	40.48	0.90	25.65	0.81	15.53	0.42	15.26	0.66	27.25	0.86
	50	35.89	1.11	31.80	1.21	42.39	1.09	90.74	1.55	232.49	5.36
	100	70.47	1.08	188.71	1.54	316.38	3.13	1455.73	62.73	—	—

Table 1. Average computation times (in seconds) obtained by solving problem (8) using the proposed BnB algorithm and CPLEX with risk measure (10) and scenarios $N = 100$. All running times are averaged over 5 instances and symbol "—" indicates that the time limit of 7200 seconds was exceeded.

5. Conclusions

We have considered a R-MWK problems which entail finding a k-club of minimum risk in a graph. HMCR risk measures were utilized for quantifying the distributional information of the stochastic factors associated with vertex weights. It was shown that the optimal solutions to R-MWK problems are maximal k-clubs. A combinatorial BnB solution algorithm was developed and tested on a special case of the R-MWK problem when $k = 2$. Numerical experiments on randomly generated graphs of various configurations suggest that the proposed BnB algorithm significantly reduces solution times in comparison with the mathematical programming model solved using the CPLEX MIP solver.

6. Acknowledgements

This work was supported in part by the AFOSR grant FA9550-12-1-0142 and the U.S. Department of Air Force grant FA8651-12-2-0010. In addition, support by the AFRL Mathematical Modeling and Optimization Institute is gratefully acknowledged.

References

[1] E. Balas and C. S. Yu, “Finding a maximum clique in an arbitrary graph,” *SIAM Journal on Computing*, vol. 15, pp. 1054–1068, Nov. 1986.

[2] R. Carraghan and P. M. Pardalos, “An exact algorithm for the maximum clique problem,” *Operations Research Letters*, vol. 9, no. 6, pp. 375 – 382, 1990.

[3] P. R. J. Östergård, “A fast algorithm for the maximum clique problem,” *Discrete Applied Mathematics*, vol. 120, no. 1–3, pp. 197–207, 2002. Special Issue devoted to the 6th Twente Workshop on Graphs and Combinatorial Optimization.

[4] S. Trukhanov, C. Balasubramaniam, B. Balasundaram, and S. Butenko, “Algorithms for detecting optimal hereditary structures in graphs, with application to clique relaxations,” *Computational Optimization and Applications*, vol. 56, no. 1, pp. 113–130, 2013.

[5] E. Tomita, Y. Sutani, T. Higashi, S. Takahashi, and M. Wakatsuki, “A simple and faster branch-and-bound algorithm for finding a maximum clique,” in *WALCOM: Algorithms and Computation* (M. Rahman and S. Fujita, eds.), vol. 5942 of *Lecture Notes in Computer Science*, pp. 191–203, Springer Berlin Heidelberg, 2010.

[6] J. Konc and D. Janezic, “An improved branch and bound algorithm for the maximum clique problem,” *proteins*, vol. 4, p. 5, 2007.

[7] R. Carmo and A. Zge, “Branch and bound algorithms for the maximum clique problem under a unified framework,” *Journal of the Brazilian Computer Society*, vol. 18, no. 2, pp. 137–151, 2012.

[8] R. D. Alba, “A graph-theoretic definition of a sociometric clique,” *Journal of Mathematical Sociology*, vol. 3, pp. 3–113, 1973.

[9] L. Babel, “A fast algorithm for the maximum weight clique problem,” *Computing*, vol. 52, no. 1, pp. 31–38, 1994.

[10] P. R. J. Östergård, “A new algorithm for the maximum-weight clique problem,” *Nordic Journal of Computing*, vol. 8, pp. 424–436, Dec. 2001.

[11] D. Kumlander, “A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search,” in *Proceedings of The Fourth International Conference on Engineering Computational Technology*, pp. 137–138, Civil-Comp Press, 2004.

[12] G. D. Glockner and G. L. Nemhauser, “A dynamic network flow problem with uncertain arc capacities: Formulation and problem structure,” *Operations Research*, vol. 48, no. 2, pp. 233–242, 2000.

[13] A. Atamtürk and M. Zhang, “Two-stage robust network flow and design under demand uncertainty,” *Operations Research*, vol. 55, no. 4, pp. 662–673, 2007.

[14] A. M. Campbell and B. W. Thomas, “Probabilistic traveling salesman problem with deadlines,” *Transportation Science*, vol. 42, no. 1, pp. 1–21, 2008.

[15] A. Gupta, V. Nagarajan, and R. Ravi, “Technical noteapproximation algorithms for vrp with stochastic demands,” *Operations Research*, vol. 60, no. 1, pp. 123–127, 2012.

[16] Y. P. Aneja, R. Chandrasekaran, and K. P. K. Nair, “Maximizing residual flow under an arc destruction,” *Networks*, vol. 38, no. 4, pp. 194–198, 2001.

[17] B. Verweij, S. Ahmed, A. Kleywegt, G. Nemhauser, and A. Shapiro, “The sample average approximation method applied to stochastic routing problems: A computational study,” *Computational Optimization and Applications*, vol. 24, no. 2-3, pp. 289–333, 2003.

[18] M. Rysz, M. Mirghorbani, P. Krokhmal, and E. Pasiliao, “On risk-averse maximum weighted subgraph problems,” *Submitted for publication*.

[19] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, “Coherent measures of risk,” *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.

[20] F. Delbaen, “Coherent risk measures on general probability spaces,” in *Advances in Finance and Stochastics* (K. Sandmann and P. Schnbucher, eds.), pp. 1–37, Springer Berlin Heidelberg, 2002.

[21] B. Balasundaram, S. Butenko, and S. Trukhanov, “Novel approaches for analyzing biological networks,” *Journal of Combinatorial Optimization*, vol. 10, pp. 23–39, 2005.

[22] F. M. Pajouh and B. Balasundaram, “On inclusionwise maximal and maximum cardinality *k*-clubs in graphs,” *Discrete Optimization*, vol. 9, no. 2, pp. 84 – 97, 2012.

[23] P. A. Krokhmal, “Higher moment coherent risk measures,” *Quantitative Finance*, vol. 7, pp. 373–387, 2007.

[24] A. Veremyev, V. Boginski, P. Krokhmal, and D. Jeffcoat, “Dense percolation in large-scale mean-field random networks is provably “explosive”,” *PLOS ONE*, 2012.

[25] B. Balasundaram, S. Butenko, and I. Hicks, “Clique relaxations in social network analysis: The maximum k-plex problem,” *Operations Research*, vol. 59, pp. 133–142, 2011.

[26] P. Krokhmal, M. Zabarankin, and S. Uryasev, “Modeling and optimization of risk,” *Surveys in Operations Researh and Management Science*, vol. 16, no. 2, pp. 49–66, 2011.

[27] P. A. Krokhmal and P. Soberanis, “Risk optimization with *p*-order conic constraints: A linear programming approach,” *European Journal of Operational Research*, vol. 201, no. 3, pp. 653–671, 2010.

On p-norm linear discrimination

Yana Morenko[*] Alexander Vinel[*] Zhaohan Yu[*] Pavlo Krokhmal[*,†]

Abstract

We consider a p-norm linear discrimination model that generalizes the model of Bennett and Mangasarian (1992) and reduces to a linear programming problem with p-order cone constraints. The proposed approach for handling linear programming problems with p-order cone constraints is based on reformulation of p-order cone optimization problems as second order cone programming (SOCP) problems when p is rational. Since such reformulations typically lead to SOCP problems with large numbers of second order cones, an "economical" representation that minimizes the number of second order cones is proposed. A case study illustrating the developed model on several popular data sets is conducted.

1 Introduction

Consider two discrete sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$ containing k and m points, respectively: $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_k\}$, $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$. One of the principal tasks arising in machine learning and data mining is that of *discrimination* of such sets, namely, constructing a surface $f(\mathbf{x}) = 0$ such that $f(\mathbf{x}) < 0$ for any $\mathbf{x} \in \mathcal{A}$ and $f(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{B}$. Of particular interest is the linear separating surface (hyperplane): $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} - \gamma = 0$. From the simple fact that any two points $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^n$ satisfying the inequalities $\mathbf{w}^\top \mathbf{y}_1 - \gamma > 0$, $\mathbf{w}^\top \mathbf{y}_2 - \gamma < 0$ for some $\mathbf{w}$ and γ are located on the opposite sides of the hyperplane $\mathbf{w}^\top \mathbf{x} - \gamma = 0$, it follows that the discrete sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$ are considered *linearly separable* if and only if there exist $\mathbf{w} \in \mathbb{R}^n$ such that $\mathbf{w}^\top \mathbf{a}_i > \gamma > \mathbf{w}^\top \mathbf{b}_j$ for all $i = 1, \ldots, k$, $j = 1, \ldots, m$, with an appropriately chosen γ, or, equivalently,

$$\min_{\mathbf{a}_i \in \mathcal{A}} \mathbf{a}_i^\top \mathbf{w} > \max_{\mathbf{b}_j \in \mathcal{B}} \mathbf{b}_j^\top \mathbf{w}. \tag{1}$$

Clearly, existence of such a separating hyperplane is not guaranteed (namely, a separating hyperplane exists if the convex hulls of sets $\mathcal{A}$ and $\mathcal{B}$ are disjoint); thus, in general, a separating hyperplane that minimizes some sort of *misclassification error* is desired.

In the next section we introduce a new linear separation model that is based on p-order cone programming, and discuss its key properties. The proposed solution approach, based on a reformulation of p-cone programming problems as second order cone programming (SOCP) problems when p is rational, is presented in Section 3. Section 4 contains a case study on several popular data sets that illustrates the developed discrimination model.

2 p-Norm linear separation: A stochastic optimization analogy

Since definition (1) involves strict inequalities, it is not well suited for mathematical programming models of selecting the "best" linear separator. However, the fact that the separating hyperplane can be scaled by any non-negative factor allows one to formulate the following observation:

Proposition 1 ([4]) *Discrete sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$ represented by matrices $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_k)^\top \in \mathbb{R}^{k\times n}$ and $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_m)^\top \in \mathbb{R}^{m\times n}$, respectively, are linearly separable if and only if*

$$\mathbf{A}\mathbf{w} \geq \mathbf{e}\gamma + \mathbf{e}, \quad \mathbf{B}\mathbf{w} \leq \mathbf{e}\gamma - \mathbf{e} \quad \textit{for some} \quad \mathbf{w} \in \mathbb{R}^n,\ \gamma \in \mathbb{R}, \tag{2}$$

[*] Department of Mechanical and Industrial Engineering, University of Iowa, Iowa City, IA 52242, USA

[†] Corresponding author. E-mail: krokhmal@engineering.uiowa.edu

where $\mathbf{e}$ *is the vector of ones of an appropriate dimension,* $\mathbf{e} = (1, \ldots, 1)^\top$.

Given the linear separability condition (2), the (non-negative) vectors $\mathbf{x}_{\mathcal{A}} = (-\mathbf{A}\mathbf{w} + \mathbf{e}\gamma + \mathbf{e})_+$, $\mathbf{x}_{\mathcal{B}} = (\mathbf{B}\mathbf{w} - \mathbf{e}\gamma + \mathbf{e})_+$, where $t_+ = \max\{0, t\}$, represent misclassification errors: $\mathbf{x}_{\mathcal{A}}$ and/or $\mathbf{x}_{\mathcal{B}} > \mathbf{0}$ if sets $\mathcal{A}$ and $\mathcal{B}$ are not linearly separable. If one considers that points of sets $\mathcal{A}$ and $\mathcal{B}$ represent realizations of (discretely distributed) random vectors $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^n$, respectively, the corresponding elements of vectors $\mathbf{x}_{\mathcal{A}}$, $\mathbf{x}_{\mathcal{B}}$ may be regarded as realizations of random variables $X_{\mathcal{A}}(\mathbf{a}; \mathbf{w}, \gamma) = (-\mathbf{a}^\top\mathbf{w} + \gamma + 1)_+$, $X_{\mathcal{B}}(\mathbf{b}; \mathbf{w}, \gamma) = (\mathbf{b}^\top\mathbf{w} - \gamma + 1)_+$, respectively, that depend parametrically on the decision variables $\mathbf{w}$ and γ. Then, a plausible strategy for selecting $\mathbf{w}$ and γ is one that minimizes, for example, the expected misclassification errors, and which can be formulated as the following stochastic programming problem:

$$\min_{(\mathbf{w},\gamma)\,\in\,\mathbb{R}^{n+1}} \Big\{\delta_1 \mathsf{E}\big[(-\mathbf{a}^\top\mathbf{w} + \gamma + 1)_+\big] + \delta_2 \mathsf{E}\big[(\mathbf{b}^\top\mathbf{w} - \gamma + 1)_+\big]\Big\},$$

where $\delta_{1,2}$ serve as "importance" weights of the misclassification errors for points of sets $\mathcal{A}$ and $\mathcal{B}$, respectively. Further, instead of minimizing the expected misclassification error, one may select the parameters $\mathbf{w}$ and γ so as to minimize the risk of misclassification. As it is well known in stochastic optimization and risk analysis, the "risk" associated with random outcome of a decision under uncertainty is often attributed to the "heavy" tails of the corresponding probability distribution. The risk-inducing "heavy" tails of probability distributions, are, in turn, characterized by the distribution's higher moments. Thus, if the misclassifications introduced by a separating hyperplane can be viewed as "random", the misclassification risk may be controlled better if one minimizes not the average, or expected misclassification errors, but their moments of order $p > 1$. This gives rise to the following formulation for linear discrimination of sets $\mathcal{A}$ and $\mathcal{B}$:

$$\min_{(\mathbf{w},\gamma)\,\in\,\mathbb{R}^{n+1}} \delta_1 \big\|(-\mathbf{a}^\top\mathbf{w} + \gamma + 1)_+\big\|_p + \delta_2 \big\|(\mathbf{b}^\top\mathbf{w} - \gamma + 1)_+\big\|_p, \quad p \in [1, +\infty], \tag{3}$$

where $\|\cdot\|_p$ is the usual $\mathcal{L}_p$ norm: $\|Y\|_p = \big(\mathsf{E}|Y|^p\big)^{1/p}$ if $p \in [1, \infty)$, and $\|Y\|_\infty = \operatorname{ess\,sup}|Y|$. If $\mathbf{a}$ and $\mathbf{b}$ are uniformly distributed with support sets $\mathcal{A}$ and $\mathcal{B}$, respectively:

$$\mathsf{P}(\mathbf{a} = \mathbf{a}_i) = 1/k, \quad \mathsf{P}(\mathbf{b} = \mathbf{b}_j) = 1/m \quad \text{for all} \quad \mathbf{a}_i \in \mathcal{A}, \;\; \mathbf{b}_j \in \mathcal{B}, \tag{4}$$

the p-norm linear discrimination problem takes the form

$$\min_{(\mathbf{w},\gamma)\in\mathbb{R}^{n+1}} \frac{\delta_1}{k^{1/p}} \big\|(-\mathbf{A}\mathbf{w} + \mathbf{e}\gamma + \mathbf{e})_+\big\|_p + \frac{\delta_2}{m^{1/p}} \big\|(\mathbf{B}\mathbf{w} - \mathbf{e}\gamma + \mathbf{e})_+\big\|_p, \tag{5}$$

where $\|\cdot\|_p$ is a norm in Euclidean space of an appropriate dimension: $\|\mathbf{u}\|_p = (|u_1|^p + \ldots + |u_l|^p)^{1/p}$, $p \in [1, \infty)$ and $\|\mathbf{u}\|_\infty = \max_{i=1,\ldots,l}\{u_i\}$ (in the sequel, it shall be clear from the context whether the $\mathcal{L}_p$ or Euclidean p-norm is used). Further, (5) can be formulated as a p-order cone programming problem (pOCP)

$$\begin{aligned}
\min \quad & \delta_1 k^{-1/p}\,\xi + \delta_2 m^{-1/p}\,\eta && \text{(6a)}\\
\text{s. t.} \quad & \xi \ge \|\mathbf{y}\|_p, && \text{(6b)}\\
& \eta \ge \|\mathbf{z}\|_p, && \text{(6c)}\\
& \mathbf{y} \ge -\mathbf{A}\mathbf{w} + \mathbf{e}\gamma + \mathbf{e}, && \text{(6d)}\\
& \mathbf{z} \ge \mathbf{B}\mathbf{w} - \mathbf{e}\gamma + \mathbf{e}, && \text{(6e)}\\
& \mathbf{z}, \mathbf{y} \ge \mathbf{0}. && \text{(6f)}
\end{aligned}$$

Note that the special case of $p = 1$ and $\delta_1 = \delta_2$ corresponds to the linear discrimination model of Bennett and Mangasarian [4]. The p-cone programming linear separation model (3)–(6) shares many key properties with the LP separation model [4], including the guarantee that an optimal solution of (6) is non-zero in $\mathbf{w}$ for linearly separable sets.

Proposition 2 *When sets* $\mathcal{A}$ *and* $\mathcal{B}$, *represented by matrices* $\mathbf{A}$ *and* $\mathbf{B}$, *are linearly separable, the separating hyperplane* $\mathbf{w}^{*\top}\mathbf{x} = \gamma^*$ *given by an optimal solution of* (5)–(6) *satisfies* $\mathbf{w}^* \neq \mathbf{0}$.

Proof: Zero optimal value of (6a) entails that $-\mathbf{A}\mathbf{w}^* + \mathbf{e}\gamma^* + \mathbf{e} \le \mathbf{0}$, $\mathbf{B}\mathbf{w}^* - \mathbf{e}\gamma^* + \mathbf{e} \le \mathbf{0}$ at optimality, which requires that $\gamma^* \le -1$ and $\gamma^* \ge 1$ simultaneously for $\mathbf{w}^* = \mathbf{0}$ to hold. ■

Secondly, the p-norm separation model (6) can produce a solution with $\mathbf{w} = \mathbf{0}$ only in a rather special case that is identified by Theorem 1 below.

Theorem 1 *Consider the p-order cone programming problem* (6)–(5), *where it is assumed without loss of generality that* $0 < \delta_1 \le \delta_2$. *Then, for any $p \in (1, \infty)$ the p-order cone programming problem* (6) *has an optimal solution with $\mathbf{w}^* = \mathbf{0}$ if and only if*

$$\frac{\mathbf{e}^\top}{k}\mathbf{A} = \mathbf{v}^\top\mathbf{B}, \quad \textit{where} \quad \mathbf{e}^\top\mathbf{v} = 1, \ \ \mathbf{v} \ge \mathbf{0}, \ \ \|\mathbf{v}\|_q \le \frac{\delta_2}{\delta_1 m^{1/p}}, \tag{7a}$$

where q satisfies $p^{-1} + q^{-1} = 1$. In other words, the arithmetic mean of the points in $\mathcal{A}$ must be equal to some convex combination of points in $\mathcal{B}$. In the case of $\delta_1 = \delta_2$ condition (7a) *reduces to*

$$\frac{\mathbf{e}^\top}{k}\mathbf{A} = \frac{\mathbf{e}^\top}{m}\mathbf{B}, \tag{7b}$$

i.e., the arithmetic means of the points of sets $\mathcal{A}$ and $\mathcal{B}$ must coincide.

Proof: First, let us consider the case when the p-cone discrimination model (6) has an optimal solution with $\mathbf{w}^* = \mathbf{0}$ and demonstrate that (7) must then hold. From the formulation (5) of problem (6) it follows that in the case when $\mathbf{w} = \mathbf{0}$ at optimality, the corresponding optimal value of the objective (6a) is determined as

$$\min_{\gamma\in\mathbb{R}} \left\{ \frac{\delta_1}{k^{1/p}} \left(\sum_{i=1}^{k} (1+\gamma)_+^p \right)^{1/p} + \frac{\delta_2}{m^{1/p}} \left(\sum_{j=1}^{m} (1-\gamma)_+^p \right)^{1/p} \right\} = 2\delta_1,$$

due to the assumption $0 < \delta_1 \le \delta_2$. Next, consider the dual of the p-cone programming problem (6):

$$\begin{aligned} \max \quad & \mathbf{e}^\top\mathbf{u} + \mathbf{e}^\top\mathbf{v} \\ \text{s. t.} \quad & -\mathbf{A}^\top\mathbf{u} + \mathbf{B}^\top\mathbf{v} = \mathbf{0}, \\ & \mathbf{e}^\top\mathbf{u} - \mathbf{e}^\top\mathbf{v} = \mathbf{0}, \\ & \mathbf{0} \le \mathbf{u} \le -\mathbf{s}, \\ & \mathbf{0} \le \mathbf{v} \le -\mathbf{t}, \\ & \|\mathbf{s}\|_q \le \delta_1 k^{-1/p}, \\ & \|\mathbf{t}\|_q \le \delta_2 m^{-1/p}, \end{aligned} \tag{8}$$

where q is such that $1/p + 1/q = 1$. Note that (6) is strictly feasible and bounded from below, since for any $\mathbf{w}_0$, γ_0 and $\varepsilon > 0$ one can select $\mathbf{y}_0 = \varepsilon\mathbf{e} + (-\mathbf{A}\mathbf{w}_0 + \mathbf{e}\gamma_0 + \mathbf{e})_+ > \mathbf{0}$, $\mathbf{z}_0 = \varepsilon\mathbf{e} + (\mathbf{B}\mathbf{w}_0 - \mathbf{e}\gamma_0 + \mathbf{e})_+ > \mathbf{0}$, $\xi_0 = (1+\varepsilon)\|\mathbf{y}_0\|_p > \|\mathbf{y}_0\|_p > 0$, and $\eta_0 = (1+\varepsilon)\|\mathbf{z}_0\|_p > \|\mathbf{z}_0\|_p > 0$ that are feasible to (6). Thus, the duality gap for the primal-dual pair of p-order cone programming problems (6) and (8) is zero [12]. Then, from the first two constraints of (8) we have $\mathbf{A}^\top\mathbf{u}^* = \mathbf{B}^\top\mathbf{v}^*$ as well as $\mathbf{e}^\top\mathbf{u}^* = \mathbf{e}^\top\mathbf{v}^*$, which, given that the optimal objective value of (8) is $2\delta_1$, implies that an optimal $\mathbf{u}^*$ must satisfy

$$\mathbf{e}^\top\mathbf{u}^* = \delta_1. \tag{9a}$$

Also, from (8) it follows that

$$\|\mathbf{u}^*\|_q \le \delta_1 k^{-1/p}. \tag{9b}$$

Then, it is easy to see that the unique solution of system (9) is

$$\mathbf{u}^* = \frac{\delta_1}{k}\mathbf{e} = \left(\frac{\delta_1}{k}, \ldots, \frac{\delta_1}{k} \right)^\top,$$

which corresponds to the point where the surface $(u_1^q + \ldots + u_k^q)^{1/q} = \delta_1 k^{-1/p}$ is tangent to the hyperplane $u_1 + \ldots + u_k = \delta_1$ in the positive of $\mathbb{R}^k$.

Likewise, an optimal $\mathbf{v}^*$ must satisfy $\mathbf{e}^\top \mathbf{v}^* = \delta_1$ and $\|\mathbf{v}^*\|_q \leq \delta_2 m^{-1/p}$, but such $\mathbf{v}^*$ is not unique in the case $\delta_2/\delta_1 > 1$. By substituting the obtained characterizations for $\mathbf{u}^*$ and $\mathbf{v}^*$ in the constraint $\mathbf{A}^\top \mathbf{u}^* = \mathbf{B}^\top \mathbf{v}^*$ of the dual, we obtain (7a). When $\delta_1 = \delta_2$, the optimal $\mathbf{v}^*$ is unique: $\mathbf{v}^* = \dfrac{\delta_1}{m}\mathbf{e}$, and yields (7b).

To prove the statement of the Theorem in the opposite direction, assume that, for instance, (7a) holds for certain $\mathbf{u}$ and $\mathbf{v}$. Selecting $\mathbf{u}^* = (\delta_1/k)\,\mathbf{e}$, $\mathbf{v}^* = \delta_1 \mathbf{v}$, and $\mathbf{s}^* = -\mathbf{u}^*$, $\mathbf{t}^* = -\mathbf{v}^*$, it is easy to see that $(\mathbf{u}^*, \mathbf{v}^*, \mathbf{s}^*, \mathbf{t}^*)$ represents a feasible solution of the dual problem (8) with the dual cost of $2\delta_1$. Similarly, the tuple $(\mathbf{w}^*, \gamma^*, \mathbf{y}^*, \mathbf{z}^*, \xi^*, \eta^*)$, where $\mathbf{w}^* = \mathbf{0}$, $\gamma^* = 1$, $\mathbf{y}^* = (\mathbf{e}\gamma^* + \mathbf{e})_+ = 2\mathbf{e}$, $\mathbf{z}^* = (-\mathbf{e}\gamma^* + \mathbf{e})_+ = \mathbf{0}$, $\xi^* = \|\mathbf{y}^*\|_p = 2k^{1/p}$, $\eta^* = \|\mathbf{z}^*\|_p = 0$, represents a feasible solution of the primal problem (6) with the corresponding objective value of $2\delta_1$. Noting the zero duality gap for the constructed pair of feasible solutions of (6) and (8), and recalling that the primal problem is bounded and strictly feasible, we immediately obtain that this pair of primal-dual solutions is optimal [12]. Hence, from (7a) it follows that an optimal solution of (6) exists with $\mathbf{w}^* = \mathbf{0}$. ■

Observe that Theorem 1 implies that in the case of $\delta_1 = \delta_2$, the p-norm discrimination model (6) produces a null separating hyperplane only when the "geometric centers" of the sets $\mathcal{A}$ and $\mathcal{B}$ coincide. In practice, this means that such sets cannot be efficiently separated, at least by a hyperplane, thus an occurrence of a $\mathbf{w}^* = \mathbf{0}$ solution in (6) may be regarded not as a shortfall of formulation (6), but rather as the general unsuitability of such sets $\mathcal{A}$ and $\mathcal{B}$ to linear discrimination. In the case of $\delta_1 < \delta_2$, occurrence of a $\mathbf{w}^* = \mathbf{0}$ solution in (6) does not necessarily signify that sets $\mathcal{A}$ and $\mathcal{B}$ are hardly amenable to linear separation. In this case Theorem 1 only claims that the "geometric center" of $\mathcal{A}$ must lie within the convex hull of set $\mathcal{B}$, so that linear discrimination can still be a feasible approach, albeit at a cost of significant misclassification errors.

In order for a $\mathbf{w}^* = \mathbf{0}$ solution to occur only under the stricter condition (7b) when misclassification preferences for sets $\mathcal{A}$ and $\mathcal{B}$ are different, the p-norm linear discrimination model can be extended by applying norms of different orders to misclassifications of points in $\mathcal{A}$ and $\mathcal{B}$:

$$\min_{(\mathbf{w},\gamma)\in\mathbb{R}^{n+1}} \; k^{-1/p_1} \left\| (-\mathbf{A}\mathbf{w} + \mathbf{e}\gamma + \mathbf{e})_+ \right\|_{p_1} + m^{-1/p_2} \left\| (\mathbf{B}\mathbf{w} - \mathbf{e}\gamma + \mathbf{e})_+ \right\|_{p_2}, \quad p_{1,2} \in (1, \infty). \tag{10}$$

Intuitively, a norm of higher order places more "weight" on the outliers. For example, use of $p = 1$ norm entails minimization of the average of misclassifications; in contrast, application of the $p = \infty$ norm implies minimization of the largest misclassification for a set. Thus, by selecting appropriately the orders p_1 and p_2 in (10) one may introduce tolerance preferences on misclassifications of points of sets $\mathcal{A}$ and $\mathcal{B}$. At the same time, it can be shown that the occurrence of $\mathbf{w}^* = \mathbf{0}$ solution in (10) would signal the presence of the aforementioned singularity about the sets $\mathcal{A}$ and $\mathcal{B}$. Namely, we have

Theorem 2 *The p-order cone programming problem* (10)*, where $p_1, p_2 \in (1, \infty)$, has an optimal solution with $\mathbf{w}^* = \mathbf{0}$ if and only if* (7b) *holds.*

We conclude this section by pointing out a connection between the p-norm separation model and the classical Support Vector Machine (SVM) model. SVM models are widely used in classification problems (see some recent works in, e.g., [5, 9, 14]). The linear SVM for non-separable sets can be written as a quadratic programming problem of the form

$$\begin{aligned}
\min \quad & \tfrac{1}{2}\|\mathbf{w}\|_2 + C_1 \mathbf{e}^\top \boldsymbol{\varepsilon}_1 + C_2 \mathbf{e}^\top \boldsymbol{\varepsilon}_2 && \text{(11a)}\\
\text{s. t.} \quad & \mathbf{A}\mathbf{w} - \mathbf{e}\gamma \geq \mathbf{e} - \boldsymbol{\varepsilon}_1 && \text{(11b)}\\
& -\mathbf{B}\mathbf{w} + \mathbf{e}\gamma \geq \mathbf{e} - \boldsymbol{\varepsilon}_2 && \text{(11c)}\\
& \boldsymbol{\varepsilon}_{1,2} \geq \mathbf{0} && \text{(11d)}
\end{aligned}$$

where $\boldsymbol{\varepsilon}_1$ and $\boldsymbol{\varepsilon}_2$ are misclassification vectors for sets $\mathcal{A}$ and $\mathcal{B}$, respectively, and $C_1, C_2 > 0$.

Proposition 3 *If the misclassification weight coefficients in the p-norm separation model* (6) *and the SVM model* (11) *coincide, $C_1 = \delta_1/k_1$ and $C_2 = \delta_2/k_2$, the optimal value V^*_{SVM} of SVM problem* (11) *can be* *bounded as*

$$V_p^* \le V_{\text{SVM}}^* \le V_p^* + \tfrac{1}{2}\|\mathbf{w}^*\|_2,$$

where V_p^* is the optimal value of p-norm problem (6) *and* $\mathbf{w}^*$ *is an optimal solution of* (6).

Proof: By renaming variables $\boldsymbol{\varepsilon}_1 = \mathbf{y}$, $\boldsymbol{\varepsilon}_2 = \mathbf{z}$, problem (11) can be rewritten as

$$\min \left\{ \tfrac{1}{2}\|\mathbf{w}\|_2 + C_1\xi + C_2\eta \;\middle|\; \xi \ge \|\mathbf{y}\|_1, \;\; \eta \ge \|\mathbf{z}\|_1, \; (6d), \; (6e), \; (6f) \right\}. \tag{12}$$

Setting $C_1 = \delta_1/k$, $C_2 = \delta_2/m$ and taking into account that $\|\mathbf{x}\|_p \ge \|\mathbf{x}\|_q$ for $1 \le p < q$, it is easy to see that

$$\begin{aligned} &\tfrac{1}{2}\|\mathbf{w}^*\|_2 + C_1\xi^* + C_2\eta^* \ge \tfrac{1}{2}\|\mathbf{w}^{**}\|_2 + C_1\xi^{**} + C_2\eta^{**} \\ &\ge C_1\xi^{**} + C_2\eta^{**} \ge C_1\,\xi^*_{(1)} + C_2\,\eta^*_{(1)} \ge C_1\xi^* + C_2\eta^*, \end{aligned}$$

where $\mathbf{w}^{**}, \xi^{**}, \eta^{**}$ are the optimal values of the variables in the SVM problem (12), $\mathbf{w}^*, \xi^*, \eta^*$ are optimal solutions of the p-norm separation model (6), and $\xi^*_{(1)}, \eta^*_{(1)}$ are optimal solutions of (6) with $p = 1$. ■

In the next section we discuss the details of practical implementation of the p-norm linear discrimination model (6).

3 A second order cone programming approach to p-order cone programming problems

The p-order cone constraints (6b)–(6c) are central to practical implementation of the p-norm separation method (6). In the special cases of $p = 1$ or $p = \infty$, p-order cone constraints reduce to linear inequalities; specifically, the $p = 1$ version of model (6) has been studied in [4]. In general, the amenability of 1-norm to implementation via linear constraints has been exploited in a variety approaches and applications, too numerous to cite here. Another prominent special case of is that of $p = 2$, when (6b)–(6c) represent second order, or quadratic cones. The second order cone programming (SOCP) constitutes a well-developed subject of convex optimization, and a number of efficient self-dual "long-step" interior point (IP) SOCP algorithms have been developed in the literature and implemented in software [1, 2, 13]. The "general" case of $p \in (1,2) \cup (2,\infty)$, when the p-cone is not self-dual, has received relatively limited attention in the literature. IP approaches to p-order cone programming have been considered in, e.g., [6, 11, 15]; a polyhedral approximation approach was proposed in [10].

In this work, we pursue an approach to solving p-cone programming problems that is based on the possibility to represent a p-order cone via a sequence of second order cones when p is rational [1, 12]. Reformulation of a rational-order p-cone programming problem as a SOCP problem allows for employing the efficient self-dual SOCP methods, albeit at a cost of a large number of second order cones required for such a reformulation. Moreover, since such a reformulation is not unique, in Section 3.2 we introduce a constructive "economical" representation of rational-order p-cones via second order cones.

3.1 Representation of rational-order p-cones with second order cones

Without loss of generality, consider a p-cone in the positive orthant of $\mathbb{R}^{n+1}$

$$t \ge (w_1^p + \ldots + w_n^p)^{1/p}, \quad (t, w_1, \ldots, w_n)^\top \ge \mathbf{0}. \tag{13}$$

In the case when the parameter p is a positive rational number, $p = r/s$, where $r, s \in \mathbb{N}$, then, for instance, the following "lifted" representation of the p-cone set (13) can be constructed in $\mathbb{R}_+^{2n+1}$ [1, 10]:

$$t \ge u_1 + \ldots + u_n, \qquad u_j \ge 0, \quad j = 1, \ldots, n, \tag{14a}$$

$$w_j^R \le u_j^s t^{r-s} w^{R-r}, \qquad j = 1, \ldots, n, \tag{14b}$$

where $R = 2^\rho$, $\rho = \lceil \log_2 r \rceil$. Then, each nonlinear inequality (14b) can equivalently be replaced by a sequence of three-dimensional (3D) rotated quadratic cones $z^2 \le xy$; such a representation, however, is not unique. Observe that each side of inequalities (14b) contains 2^ρ factors; this allows one to construct a lifted representation for (14b) via $2^\rho - 1$ 3D rotated quadratic cones using the "tower of variables" technique [3]:

$$\begin{aligned}
& w^2 \le v_{\rho-1,1} v_{\rho-1,2} && && \text{(15a)}\\
& v_{l,i}^2 \le v_{l-1,2i-1}\, v_{l-1,2i}, && i = 1, \ldots, 2^{\rho-l}, \quad l = 2, \ldots, \rho-1, && \text{(15b)}\\
& v_{1,i}^2 \le u^2, && i = 1, \ldots, \lfloor s/2 \rfloor, && \text{(15c)}\\
& v_{1,i}^2 \le ut, && i = \lfloor s/2 \rfloor + 1, \ldots, \lceil s/2 \rceil, && \text{(15d)}\\
& v_{1,i}^2 \le t^2, && i = \lceil s/2 \rceil + 1, \ldots, \lfloor r/2 \rfloor, && \text{(15e)}\\
& v_{1,i}^2 \le tw, && i = \lfloor r/2 \rfloor + 1, \ldots, \lceil r/2 \rceil, && \text{(15f)}\\
& v_{1,i}^2 \le w^2, && i = \lceil r/2 \rceil + 1, \ldots, \lfloor R/2 \rfloor, && \text{(15g)}\\
& w, v_{l,i}, u, t \ge 0,
\end{aligned}$$

where subscripts j are suppressed for brevity. The set of inequalities (15) can be visualized as a binary tree whose nodes represent the variables in (15). Each inequality in (15) can then be viewed as a subgraph with two arcs that connect the "parent" node (the variable at the left-hand side of the inequality) to the two "child" nodes (the variables at the right-hand side of the same inequality). Given this binary structure, the set of second order cones in (15) can be regarded as partitioned into ρ levels indexed by l, where the variable w in (15a) constitutes the root node of the tree, and belongs to ρ-level, while variables u, t, w in (15d)–(15g) represent the leaf nodes, or 0-level nodes of the tree.

In [10] it has been shown that among the $2^\rho - 1$ inequalities (15) there are only $O(\rho) = O(\log_2 r)$ non-degenerate second order cones, while the rest reduce to linear inequalities that can be omitted. The following bounds on the number of non-degenerate quadratic cones in (15) follow directly from the arguments in [10]:

Proposition 4 ([10]) *When p is a positive rational number, $p = r/s$, such that $r > s$ and the greatest common divisor of r and s is 1, a p-order cone in the positive orthant of $\mathbb{R}^{n+1}$ can equivalently be represented by C_p three-dimensional quadratic cones, where C_p satisfies*

$$n\rho \le C_p \le n(2\rho - 1), \quad \rho = \lceil \log_2 r \rceil. \tag{16}$$

It it easy to see that the order in which the variables u, t, and w are assigned to the leaf nodes in the binary tree (15) can significantly affect the number of non-degenerate quadratic cones needed to represent a rational-order p-cone in $\mathbb{R}^{n+1}$. As an illustration, consider the case $p = 3$; direct application of (15) yields $\rho = 2$, $R = 4$, and a representation of $p = 3$ cone (13) that involves $3n$ 3D rotated quadratic cones:

$$t \ge u_1 + \ldots + u_n; \quad w_j^2 \le v_{1j} v_{2j}, \quad v_{1j}^2 \le u_j t, \quad v_{2j}^2 \le t w_j, \quad j = 1, \ldots, n. \tag{17}$$

On the other hand, it is easy to verify that reordering the leaf nodes inequalities (15c)–(15g) allows for reducing the number of 3D quadratic cones necessary to represent a $p = 3$ cone in $\mathbb{R}_+^{n+1}$ to $2n$:

$$t \ge u_1 + \ldots + u_n; \quad w_j^2 \le t v_j, \quad v_j^2 \le u_j w_j, \quad j = 1, \ldots, n. \tag{18}$$

Observe that the number of second order cones in representations (17) and (18) correspond to the upper and lower bounds in (16), respectively.

Since a reduction in the number of second order cone inequalities in (15) leads to a reduction in the number of quadratic cones representing a rational-order p-cone (13) by the order of dimensionality n of the p-cone, it is of interest to devise an "economical" second order cone representation of rational-order cones.

3.2 An "economical" representation of rational-order p-cone via second order cones

Below we demonstrate that the lower bound on C_p in (16) is achievable for any rational $p \ge 1$. To this end, consider the following convex pointed cone in $\mathbb{R}_+^4$:

$$\mathcal{P} = \left\{ \mathbf{y} \in \mathbb{R}_+^4 \;\middle|\; y_0^{k_0} - y_1^{k_1} y_2^{k_2} y_3^{k_3} \le 0 \right\}, \tag{19}$$

that satisfies the next four properties:
(P1) $k_0, k_1, k_2, k_3 \in \mathbb{Z}_+$;
(P2) $k_0 = k_1 + k_2 + k_3$;
(P3) $k_1 + k_2 + k_3 = 2^q$ for some integer $q \geq 1$;
(P4) exactly two numbers among k_1, k_2, and k_3 are odd.

Proposition 5 *Cone* $\mathcal{P}$ (19) *that satisfies (P1)–(P4) can be represented as an intersection of at most* q *three-dimensional cones of the form* $\{\mathbf{x} \in \mathbb{R}^3_+ \mid x_3^2 \leq x_1 x_2\}$.

Proof: The process of building such a representation of $\mathcal{P}$ is based on successive lifting of $\mathcal{P}$ into spaces of dimensions greater than previous by 1, in such a way that the degree of the polynomial in (19) is reduced in half each time. First, assume that $k_1, k_3, k_3 > 0$ are all different, and $q \geq 2$. Without loss of generality, let k_1, k_2 be odd and such that $k_2 > k_1$, and consider the following set in $\mathbb{R}^5_+$:

$$\begin{gathered}\mathcal{P}^* = \left\{\mathbf{y} \in \mathbb{R}^5_+ \;\middle|\; y_0^{\nu_0} - y_4^{\nu_4} y_2^{\nu_2} y_3^{\nu_3} \leq 0,\;\; y_4^2 \leq y_1 y_2\right\}, \\ \text{where} \quad \nu_0 = k_0/2, \quad \nu_2 = (k_2 - k_1)/2, \quad \nu_4 = k_1, \quad \nu_3 = k_3/2.\end{gathered} \tag{20}$$

It is easy to see that any $(y_0, \ldots, y_3) \in \mathcal{P}$ can be extended to $(y_0, \ldots, y_4) \in \mathcal{P}^*$, and any $(y_0, \ldots, y_4) \in \mathcal{P}^*$ is such that $(y_0, \ldots, y_3) \in \mathcal{P}$. As k_1 and k_2 are odd and positive integers by assumption, due to (P4) k_3 is even, whence ν_3 is a positive integer. The above assumption also implies that $k_2 - k_1$ is even, meaning that ν_2 is a positive integer. Similarly, ν_0 is integer and $\nu_0 = 2^{q-1}$. Also, observe that $\nu_1 + \nu_2 + \nu_3 = (k_1 + k_2 + k_3)/2 = k_0/2 = \nu_0$. So, the first cone in (20) satisfies properties (P1)–(P3). Next, observe that $\nu_4 = k_1$ is odd, thus out of two integers ν_2, ν_3 exactly one should be odd for $\nu_2 + \nu_3 + \nu_4 = 2^{q-1}$ to hold. Thus, condition (P4) holds as well.

Note that if in our assumption $k_1 = k_2$, then $\nu_2 = 0$ in (20), but all conditions still hold. Consider the case when $q \geq 2$ and one of k_1, k_2, k_3 is zero, assume it is k_3. Then k_1, k_2 should be odd by (P4). Performing the same transformation, we obtain

$$\mathcal{P}^{**} = \left\{\mathbf{y} \in \mathbb{R}^5_+ \;\middle|\; y_0^{\nu_0} - y_4^{\nu_4} y_2^{\nu_2} \leq 0,\; y_4^2 \leq y_1 y_2\right\},\; \nu_0 = k_0/2,\; \nu_2 = (k_2 - k_1)/2,\; \nu_4 = k_1. \tag{21}$$

The first cone of $\mathcal{P}^{**}$ still has properties (P1)–(P4), and $(y_0, \ldots, y_3) \in \mathcal{P}$ can be extended to $(y_0, \ldots, y_4) \in \mathcal{P}^{**}$, and any $(y_0, \ldots, y_4) \in \mathcal{P}^{**}$ is such that $(y_0, \ldots, y_3) \in \mathcal{P}$.

If $q = 1$, then one of k_1, k_2, k_3 is zero, and two others are necessarily equal to 1. In this case $\mathcal{P}$ is already a quadratic cone. Thus, the above lifting transformation can be carried out no more than $q - 1$ times, and the conic set $\mathcal{P}$ (19) can be represented by at most q quadratic cones using at most $q - 1$ new variables. ■

With the help of Proposition 5 we can now establish the following result on second order cone representation of rational-order p-cones:

Theorem 3 *Let* $p > 1$ *be a positive rational number,* $p = r/s$*, where the greatest common divisor of* r *and* s *is 1. Then a* p*-order cone in the positive orthant of* $\mathbb{R}^{n+1}$ *can equivalently be represented by* $n\lceil \log_2 r \rceil$ *three-dimensional rotated quadratic cones.*

Proof: In accordance to (13)–(14b), the problem of representing a (r/s)-cone in $\mathbb{R}^{n+1}_+$ via second order cones can be reduced to finding a second order cone representation of n sets of the form

$$\mathcal{Q} = \left\{\mathbf{y} \in \mathbb{R}^3_+ \;\middle|\; y_3^R - y_1^s y_2^{r-s} y_3^{R-r} \leq 0\right\}, \tag{22}$$

where $R = 2^\rho$, $\rho = \lceil \log_2 r \rceil$. Observe that cone $\mathcal{Q}$ is equivalent to intersection of cone $\mathcal{P}$ (19), where $k_1 = s$, $k_2 = r - s$, $k_3 = R - r$, with a hyperplane $y_0 = y_3$. Indeed, properties (P1)–(P3) are obvious, and (P4) holds since if r and s do not have common divisor greater than 1, neither do $r - s$ and s, whereby $r - s$ and s cannot be both even.

Note that an iteration of the lifting procedure described in Proposition 5 corresponds to a specific order in which the variables at some level of the binary tree are arranged. For example, the first iteration of lifting corresponds to arranging the 0-level variables $\{w, t, u\} = \{y_1, y_2, y_3\}$ in pairs corresponding to second order cone constraints, such that y_1 and y_2 make k_1 pairs, or $y_4^2 \leq y_1 y_2$ non-degenerate cones; the remaining $k_2 - k_1$ variables y_2 form $(k_2 - k_1)/2$ pairs, or degenerate cones ${y_4'}^2 \leq y_2^2$, and k_3 variables y_3 form $k_3/2$ pairs, or degenerate cones ${y_4''}^2 \leq y_3^3$, assuming that $k_1 < k_2$ are odd. Obviously, the degenerate cones can simply be disregarded.

Hence, by Proposition 5, $\mathcal{Q}$ admits representation by at most $\rho = \lceil \log_2 r \rceil$ second order cones; combining this with Proposition 4, one obtains that each of n sets of the form $\mathcal{Q}$ admits representation using exactly $\rho = \lceil \log_2 r \rceil$ second order cones. ■

It is well known that second order cone sets admit an equivalent semidefinite representation in the form of linear matrix inequalities (LMIs). In general, p-order cones are not LMI-representable in the space of original variables (see an example for $p = 4$ cone in [7, 8]), but admit lifted LMI representations.

Corollary 1 *Conic set $\mathcal{Q}$ (22) admits a lifted representation in the form of LMI*

$$\mathcal{Q}^* = \left\{ \mathbf{y} \in \mathbb{R}_+^{\rho+2} \;\middle|\; \sum_{i=1}^{\rho+2} \mathbf{A}_i y_i \succeq \mathbf{0} \right\},$$

where $\mathbf{A}_i \in \mathbb{R}^{2\rho \times 2\rho}$ are symmetric matrices, in the sense that the projection of $\mathcal{Q}^*$ onto the space of variables (y_1, y_2, y_3) coincides with $\mathcal{Q}$.

4 Computational study

In this section we report computational results on using the p-norm discrimination model (5)–(6) for linear separation of sets. In particular, we employ the presented above "economical" SOCP reformulation approach to solving pOCP problem (6) in the case when p is rational, and compare it with the polyhedral approximation technique of [10].

In our computational experiments we used three data sets from UCI Machine Learning Repository. The first data set is Wisconsin Breast Cancer data set with a total of 683 instances and 9 attributes. It contains 444 instances with benign diagnosis and 239 instances with malignant diagnosis. The second data set, Cleveland Heart Disease data set, contains 281 instances with 13 attributes, of them 125 instances correspond to positive diagnosis and 156 instances correspond to negative diagnosis. Finally, the Pima Indians Diabetes data set reports 768 instances with 8 attributes, including 266 instances of positive diagnosis and 502 instances of negative diagnosis. Both the Wisconsin Breast Cancer and Cleveland Heart Disease data sets (in their then-up-to-date versions) were used in [4].

For each data set, training and testing was performed by randomly selecting 100 training sets with equal number of points of both types, and testing the obtained separator on the data not included in the training set. For computational purposes, the data in training data sets was normalized and scaled by a factor of 10^4; the same transformation was then applied to testing data. After the training and testing procedures were performed, the average misclassification error on testing set was computed. It is important to comment on selection of parameter p in (6): as a general rule that follows from our numerical experiments and is consistent with the motivation presented in Section 2, smaller values of p (around $p = 2$) are beneficial for well-separable data sets with smaller misclassification errors, whereas larger values of $p \geq 3$ allow for reducing large misclassification errors in linear separation. With this in mind, a particular value of p can be selected during the training procedure.

Table 1 reports the average out-of-sample misclassification error for each data set, together with the respective "best" value of p at which this error was obtained. It also includes results for the cases of $p = 1$, which corresponds to minimization of the average of misclassifications due to [4], $p = \infty$, corresponding to minimization of the largest misclassification errors, and SVM model (11). Figures 1, 2, and 3 illustrate the behavior of the misclassification error in the described data sets with respect to the value of parameter p in (5)–(6), which was varied in the range of 1.0 to 4.0 with a 0.1 step. As it follows from Table 1 and Figures 1–3, the p-norm separation model (5)–(6) with $p > 1$ allows for an improved classification accuracy as compared to the cases of $p = 1$ proposed in [4], the SVM model (11), and the worst-error approach of $p = \infty$.

In addition to classification capabilities of the p-norm linear separation model (5)–(6), its computational properties were investigated. In particular, for all the data sets described above we compared the running times of the cutting plane procedure for polyhedral approximations of problem (6) due to [10], denoted as LP/CP, and the "economical" SOCP reformulation of (6), along with the corresponding results for SVM model (11) and $p = \infty$ case. All models were coded in C++ and CPLEX 12.2 solver was used to solve the resulting LP, SOCP, and QP problems. A dual-core 3GHz CPU computer with 2GB of RAM was used to run the computations. Figure 4 illustrates corresponding running times on the example of the Wisconsin Breast Cancer data set, along with the values of the parameter $\rho = \lceil \log_2 r \rceil$, where $p = r/s$, which is proportional to the number of second order cones in the SOCP reformulation of rational-order p-cone programming problem (6). From Figure 4 it follows that the solution times for SOCP reformulation of a rational-order p-cone programming model (6) are highly correlated to the number of second order cones in the reformulated problem. On the other hand, solution times of a polyhedral approximation of (6) solved with a cutting plane method (LP/CP) exhibit relatively little dependence on the value of the parameter p, and are competitive with the running times of the SVM model. Computational performance of the considered models on other data sets is very similar to that presented in Figure 4.

Table 1: Classification results for different data sets: the lowest average misclassification error, the corresponding value of p, and misclassification error for the cases of $p = 1$, $p = \infty$, and SVM model (11).

Dataset	Error	Best p	$p = 1$	SVM	$p = \infty$
Wisconsin Breast Cancer Dataset	3.95%	1.8	4.11%	4.03%	4.21%
Cleveland Heart Disease Dataset	18.7%	3.8	19.5%	18.98%	19.11%
Pima Indians Diabetes Dataset	31.82%	3.4	35.29%	34.02%	33.51%

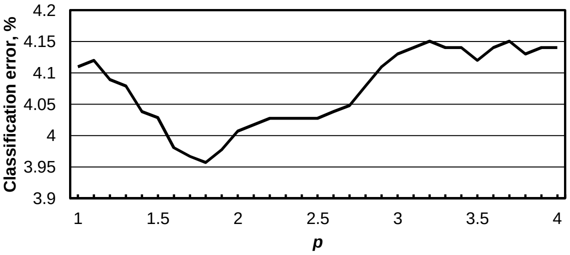


Figure 1: Misclassification error as a function of p for Wisconsin Breast Cancer data set.

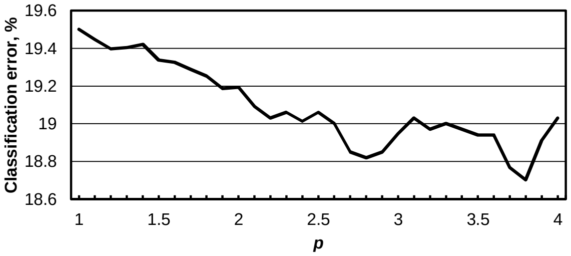


Figure 2: Misclassification error as a function of p for Cleveland Heart Disease data set.

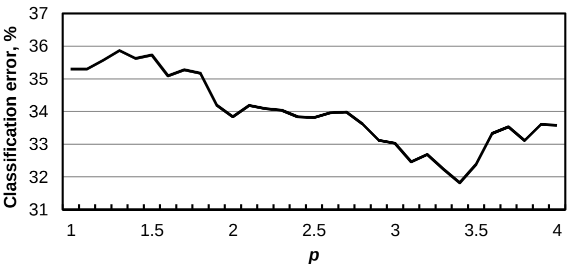


Figure 3: Misclassification error as a function of p for Pima Indians Diabetes data set.

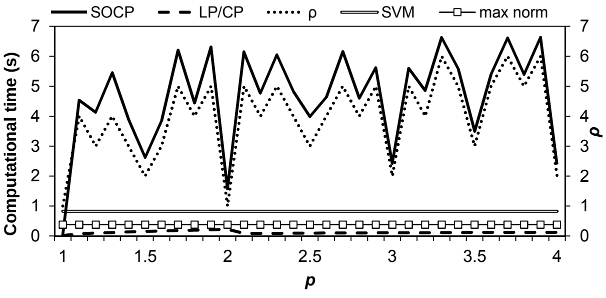


Figure 4: Average running time for instances of Wisconsin Breast Cancer data set.

Acknowledgments

The authors would like to acknowledge partial support of the U. S. Air Force Office of Scientific Research and National Science Foundation.

References

[1] Alizadeh, F. and Goldfarb, D. (2003) "Second-order cone programming," *Mathematical Programming*, **95** (1), 3–51.

[2] Andersen, E. D., Roos, C., and Terlaky, T. (2003) "On implementing a primal-dual interior-point method for conic quadratic optimization," *Mathematical Programming*, **95** (2), 249–277.

[3] Ben-Tal, A. and Nemirovski, A. (2001) "On polyhedral approximations of the second-order cone," *Mathematics of Operations Research*, **26** (2), 193–205.

[4] Bennett, K. P. and Mangasarian, O. L. (1992) "Robust linear programming separation of two linearly inseparable sets," *Optimization Methods and Software*, **1** (1), 23–34.

[5] Carrizosa, E. and Morales, D. R. (2013) "Supervised classification and mathematical optimization," *Computers & Operations Research*, **40**, 150–165.

[6] Glineur, F. and Terlaky, T. (2004) "Conic Formulation for l_p-Norm Optimization," *Journal of Optimization Theory and Applications*, **122** (2), 285–307.

[7] Helton, W. and Nie, J. (2010) "Semidefinite representation of convex sets," *Mathematical Programming*, **122** (1), 21–64.

[8] Helton, W. and Vinnikov, V. (2007) "Linear matrix inequality representation of sets," *Communications in Pure and Applied Mathematics*, **60** (5), 654–674.

[9] Khemchandani, R., Jayadeva, and Chandra, S. (2009) "Knowledge based proximal support vector machines," *European Journal of Operational Research*, **195**, 914–923.

[10] Krokhmal, P. and Soberanis, P. (2010) "Risk optimization with p-order conic constraints: A linear programming approach," *European Journal of Operational Research*, **301** (3), 653–671.

[11] Nesterov, Y. E. (2012) "Towards non-symmetric conic optimization," *Optimization Methods & Software*, **27** (4–5), 893–917.

[12] Nesterov, Y. E. and Nemirovski, A. (1994) *Interior Point Polynomial Algorithms in Convex Programming*, volume 13 of *Studies in Applied Mathematics*, SIAM, Philadelphia, PA.

[13] Sturm, J. F. (1998) "Using SeDuMi 1.0x, a MATLAB toolbox for optimization over symmetric cones," *Manuscript*.

[14] Trafalis, T. and Gilbert, R. (2006) "Robust classification and regression using support vector machines," *European Journal of Operational Research*, **173**, 893–909.

[15] Xue, G. and Ye, Y. (2000) "An efficient algorithm for minimizing a sum of p-norms," *SIAM Journal on Optimization*, **10** (2), 551–579.

On finding k-cliques in k-partite graphs

M. Mirghorbani and P. Krokhmal*

Department of Mechanical and Industrial Engineering
University of Iowa, Iowa City, IA 52242
e-mail: {smirghor,krokhmal}@engineering.uiowa.edu

Abstract

In this paper, a branch-and-bound algorithm for finding all cliques of size k in a k-partite graph is proposed that improves upon the method of Grunert et al (2002). The new algorithm uses bit-vectors, or bitsets, as the main data structure in bit-parallel operations. Bitsets enable a new form of data representation that improves branching and backtracking of the branch-and-bound procedure. Numerical studies on randomly generated instances of k-partite graphs demonstrate competitiveness of the developed method.

Keywords: maximum clique enumeration problem, k-partite graph, k-clique, bit parallelism

1 Introduction

Given an (undirected) graph $G = (V, E)$, where V is set of nodes and E is the set of arcs, a *clique* in G is defined as a complete subset of G, i.e., a set of nodes in V that are pairwise adjacent. A clique of size k is called *k-clique*;[1] the largest clique in a graph is called the *maximum clique* and its size is denoted by $\omega(G)$. Note that G may contain several cliques of size $\omega(G)$. Closely related to the concept of a clique is that of an *independent set* of G, defined as an induced subgraph of V whose nodes are pairwise disjoint.

The Maximum Clique Problem (MCP) consists in finding the largest clique in a graph, and is of fundamental importance in discrete mathematics, computer science, operations

*Corresponding author.

[1] It is worth noting that the term *k-clique* is used in several different contexts in the literature; for instance, one of its alternative interpretations is that of a subgraph where any two nodes are connected by a path of length at least k [10]. In this work, we use the definition of k-clique as given above.

research, and related fields [1]. In many applications it is of interest to identify all maximum cliques in a graph. This problem is known as the Maximum Clique Enumeration Problem (MCEP). In the present work, we consider a special case of the MCEP, concerned with finding all k-cliques in a k-partite graph. A graph $G = (V, E)$ is called k-partite if the set of nodes V can be partitioned into k independent sets, or *partites* V_r, $r = 1, \ldots, k$:

$$V = \bigcup_{r=1}^{k} V_r, \quad V_r \cap V_s = \emptyset, \quad r \neq s, \text{ such that for all } i, j \in V_r : (i, j) \notin E. \tag{1}$$

Clearly, one has that $\omega(G) \leq k$ in a k-partite graph G, since the maximum clique cannot contain more than one node from each independent set V_r. Note also that the problem of finding all k-cliques in a k-partite graph is not equivalent to MCEP since it does not account for maximum cliques with $\omega(G) < k$.

The problem of finding k-cliques in k-partite graphs has applications in many areas of science and engineering, including textile industry [3], where the braiding problem can be reduced to the problem of finding k-cliques in the path compatibility graph that represents a k-partite graph; data mining, particularly for clustering of categorical attributes over k-domains [12]; identification of protein structures [9], where protein interaction network is represented by a k-partite graph that is mined for k-cliques. Recently, it has been shown that the problem of finding k-cliques in k-partite graphs can be used to find high-quality solutions of large-scale randomized instances of multidimensional assignment problem (MAP) [6, 7, 11].

Grunert et al [3] proposed branch-and-bound algorithm FINDCLIQUE for the problem of finding all k-cliques in k-partite graphs, which takes as an input a graph $G = (V, E)$, where V satisfies (1), and produces the set Q of k-cliques contained in G as an output. FINDCLIQUE is a recursive method, such that level t of recursion corresponds to the level t of branch-and-bound tree, which in turn, is associated with the t-th partite that is branched on in V. Starting at the root ($t = 0$) of the branch-and-bound tree with a partial solution $S = \emptyset$, at each step of branch-and-bound procedure a node is added to or removed from S until S amounts to a k-clique in G, i.e., $|S| = k$, or it is verified that G contains no k-cliques, $\omega(G) < k$.

Let $B = \{1, \ldots, k\}$ be the index set of partites in G, $V = \bigcup_{b \in B} V_b$, and B_S denote the set of partites that have a node in S:

$$B_S = \{b \in B \mid V_b \cap S \neq \emptyset\}.$$

Given a partial solution S, a node is called *compatible* if it is adjacent to all the nodes in S; the set of compatible nodes w.r.t. S is denoted by C_S:

$$C_S = \{i \in V \mid (i, j) \in E \ \ \forall j \in S\}.$$

The set C_S is further partitioned into subsets containing nodes from the same partite:

$$C_S = \bigcup_{b \in \overline{B}_S} C_{S,b},$$

where $\overline{B}_S = B \setminus B_S$, and $C_{S,b} \subseteq V_b$ is given by

$$C_{S,b} = \bigcup_{s \in S} (V_b \cap N(s)),$$

with $N(s)$ being the set of nodes adjacent to node s.

At the root node of the branch-and-bound tree ($t = 0$), one has $S = \emptyset$, $B = \overline{B}_S = \{1, \ldots, k\}$, $B_S = \emptyset$, and $C_{S,b} = V_b$ for all $b \in B$. At a level t of the branch-and-bound tree, $b_t \in \overline{B}_S$ is selected as the partition to branch on. In order to achieve the greatest reduction in the size of the branch-and-bound tree when pruning, b_t is selected as the partition with the smallest number of nodes:

$$b_t \in \arg\min_b \{|C_{S,b}| \mid b \in \overline{B}_S\}. \tag{2}$$

As long as there is a node $n_t \in C_{S,b_t}$ that is not traversed, the search process is restarted from this point with $S := S \cup \{n_t\}$ as the new partial solution. To this end, the set C_S of compatible nodes is updated with respect to $S \cup \{n_t\}$:

$$C_{S,b} := C_{S,b} \cap N(n_t) \text{ for all } b \in \overline{B}_S. \tag{3}$$

Maintaining the sets $C_{S,b}$ of nodes compatible with the current partial solution S is a key aspect of the algorithm, thus for backtracking purposes the nodes that are removed from $C_{S,b}$ during (3) are added to the set $\overline{C} = \bigcup_{t=1}^{k} \overline{C}_t$, which is similarly partitioned into k levels $\overline{C}_t$, each level corresponding to level t of the branch-and-bound tree. In other words, $\overline{C}_t$ contains the nodes in $C_{S,b}$ that are not adjacent to node n_t:

$$\overline{C}_t = \{i \in C_{S,b} \mid (i, n_t) \notin E,\ b \in \overline{B}_S\}.$$

Obviously, after this step, $C_{S,b_t} = \emptyset$. A subproblem with a partial solution S is *promising* if all of the partitions in C_S that do not share a node in the partial solution are nonempty:

$$|C_{S,b}| > 0 \text{ for all } b \in \overline{B}_S,\ \ b \neq b_t. \tag{4}$$

Let P be the number of partitions $C_{S,b} \subseteq C_S$ that contain at least one node; then, an upper bound on the size of the largest clique containing S is given by $|S| + P$. If $|S| + P = k$, the current subproblem is feasible, meaning S may be part of a k-clique. For a feasible subproblem, the algorithm traverses deeper into the branch-and-bound tree, $t := t + 1$, and a new subproblem is created.

Accordingly, a subproblem with partial solution S is pruned if

$$|S| + P < k, \tag{5}$$

i.e., there exists no clique of size k that contains S. For a nonpromising subproblem, set C_{S,b_t} is restored by moving the nodes in $\overline{C}_t$ back to C_S, $C_S := C_S \cup \overline{C}_t$. The last operation implicitly requires that the nodes from $\overline{C}_t$ are put back into the partitions of C_S that they were removed from:

$$C_{S,\pi(v)} := C_{S,\pi(v)} \cup v \text{ for all } v \in \overline{C}_t, \tag{6}$$

where $\pi(i)$ is the index of the partite that node i belongs to: $i \in V_{\pi(i)}$; moreover, the relative orders of nodes in the partites V_b should be preserved in $C_{S,b}$, given that the nodes in G are assumed to be ordered/numbered.

The search process is then restarted, provided that there exists a node in partition C_{S,b_t} that is not traversed. If there is no such node, FINDCLIQUE returns to the previous level $t - 1$ of the branch-and-bound tree.

2 A bitwise algorithm for finding k-cliques in a k-partite graph

In this section, we present an algorithm, referred to as BitCLQ, for the k-clique enumeration problem in a k-partite graph, which improves upon the FINDCLIQUE algorithm of Grunert et al [3] by introducing bitset data structures and utilizing bit parallelism for updating the set of compatible nodes and improving backtracking.

2.1 Bitsets

Bitsets are essentially binary vectors, or sequences of bits, and as such can be utilized efficiently in computer codes. Particularly, bitsets are useful for storing adjacency matrices of graphs, or specific subsets of ordered sets. For example, in a graph on six nodes $\{v_1, \ldots, v_6\} = V$, a clique with nodes v_1, v_2, v_3, v_5 can be represented by a bitset $\{111010\}$, where each bit corresponds uniquely to a node in the graph, with the *significant* bits (i.e., bits equal to 1) indicating the nodes in the clique. *Bit parallelism* is a form of parallel computing that achieves computational improvements by representing the problem data in bitsets of size R, where R is the machine word size (e.g., 32 or 64), such that they can be processed together within a single processor instruction. Bit parallelism has been successfully used in many computational algorithms, particularly for string matching [2, 4, 5]. Recently, bit parallelism has been employed for solving hard combinatorial problems, such as SAT [14] and the Maximum Clique Problem [13].

In the present work, bit parallelism is used to improve the computational procedure for updating the set of compatible nodes in (3), and, moreover, to achieve faster backtracking by eliminating the need for set $\overline{C}$. In addition, use of bitsets allows for improvements in memory storage efficiency for problem data structures, such as the set of compatible nodes and the adjacency matrix of the graph.

Of particular significance in the context of the present work is the operation of indexing the first significant bit in a bitset, also known as the forward bit scanning. One of the techniques for this purpose relies on use of the De Bruijn sequence with a perfect hash table [8]. The value to be looked up in the hash table is given by H_R below:

$$H_R := (x \wedge -x)\, D \gg (R - \log_2 R), \tag{7}$$

where x is the bitset for which the first significant bit has to be indexed, D is an instance of De Bruijn sequence, R is the machine word size, and $\gg$ stands for the binary *shift right* operator. H_R is effective for bitsets of maximum size equal to R. For larger bitsets, special containers need to be devised. The hash table required to look up the value of H_R is created based on the particular De Bruijn sequence used in (7).

Note that in (7) multiplication is performed modulo R and only the last $\log_2 R$ bits of the result will be retained. More details on forward bit scanning and the specification of the De Bruijn sequence used in (7) can be found in [8].

2.2 BitCLQ

Below we present a modification of FINDCLIQUE, which we refer to as BitCLQ, that uses bitset data structures and bit parallelism for keeping track of the nodes in G that are compatible to the current partial solution S, while simultaneously reducing the computational cost of backtracking.

To this end, we introduce a set Z consisting of k levels, $Z_1, \dots, Z_k$. Each of these k levels will be used to represent the compatible nodes to the partial solution S at the t-th level of the branch-and-bound tree, where $1 \leq t \leq k$. Every level in Z is further partitioned into k sets, each corresponding to a partite V_b in G:

$$Z_t = \bigcup_{b \in B} Z_{t,b}, \quad t = 1, \dots, k.$$

The sets $Z_{t,b}$ are represented by bitsets of size $|V_b|$. Let $Z_{t,b,i}$ be the i-th bit in $Z_{t,b}$ corresponding to the i-th node in V_b, such that $Z_{t,b,i} = 1$ if the i-th node in V_b is compatible with all the nodes in the partial solution S at the t-th level of the branch-and-bound tree in BitCLQ:

$$Z_{t,b,i} = \begin{cases} 1, & \text{if } (i,j) \in E \text{ for all } j \in S_t; \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, each level Z_t of Z is an ordered set of combination of bitsets with the total size $|V|$. Further, the adjacency matrix M of graph G is stored in the bitset form, with the convention that the i-th row (column) corresponds to the i-th bit in Z_t, $t = 1, \dots, k$.

BitCLQ is initialized by setting $t := 0$, $S := \emptyset$, $B = \overline{B}_S := \{1, \dots, k\}$, and $Q := \emptyset$, where Q is the set of all k-cliques in G. Note that since at the beginning all the nodes in G can be added to S to extend its size, all the bits in Z_1 are significant:

$$Z_{1,b,i} = 1 \text{ for all } b \in \overline{B}(S_t),\ i \in V_b.$$

At level t of the branch-and-bound tree, the partition b_t to branch on is selected as

$$b_t \in \arg\min_b \{|Z_{t,b}| \mid b \in \overline{B}_S\}, \tag{8}$$

where $|Z_{t,b}|$ is defined as the number of significant bits in the bitset $Z_{t,b}$. The forward bit scanning method discussed in Section 2.1 is used to identify node $n_t \in V_{b_t}$ that has not been traversed and thus can be added to the partial solution. As long as such a node exists in V_{b_t}, the search process is restarted with $S := S \cup \{n_t\}$ as the partial solution, and the corresponding bit in Z_{t,b_t} is set to 0.

Utilizing bitsets also facilitates the process of updating the compatible nodes: when n_t is added to partial solution, Z_{t+1} is created by performing a logical AND operation with Z_t and the row $M(n_t)$ of the adjacency matrix corresponding to the node n_t as operands:

$$Z_{t+1} = Z_t \wedge M(n_t). \tag{9}$$

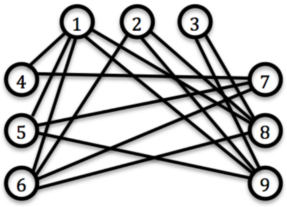
Similarly to FINDCLIQUE, let P denote the number of partitions $Z_{t,b}$ with $|Z_{t,b}| > 0$ at level the t of the branch-and-bound tree. If $|S| + P = k$, the current partial solution is promising, so that a new subproblem is created, and BitCLQ proceeds one level deeper into the branch-and-bound tree, $t := t + 1$. If the partial solution is not promising, the method presented in Section 2.1 is used to select nodes in V_{b_t} that have not been traversed. If such a node is found, the search process is restarted, otherwise backtracking is performed by simply updating $t := t - 1$. Note that due to the special structure of Z, BitCLQ does not need to restore the set of compatible nodes during backtracking, in contrast to the update procedure (6) for the set C_S that is performed in FINDCLIQUE.

2.3 Example

As an illustration, consider the 3-partite graph that is shown along with its adjacency matrix M in Figure 1, where the partite 1 consists of nodes $\{1, 2, 3\}$, partite 2 contains nodes $\{4, 5, 6\}$, and partite 3 contains nodes $\{7, 8, 9\}$. BitCLQ is initialized by setting $S := \emptyset$, $\overline{B}_S := \{1, 2, 3\}$ and $Z_1 := \{111|111|111\}$. Since all the partites are of the same size, i.e. $|Z_{1,b}| = 3$ for all $b \in \overline{B}_S$, the one to branch on is chosen arbitrarily; assume that the first partite $Z_{1,1}$ is chosen for branching. The search process from this point restarts 3 times, each time adding one of the three nodes in $Z_{1,1}$. The first node to add to S is node 1, $Z_{1,1,1}$ is then set to 0, and Z_2 is subsequently created by performing logical AND operation with Z_1 and the corresponding row of the adjacency matrix M as operands:

Algorithm 1 BitCLQ(t)

```
$b_t \in \arg\min_b \{|Z_{t,b}| \mid b \in \overline{B}_S\}$
$i :=$ the first significant bit in $Z_{t,b_t}$
repeat
    $n_t :=$ the $i$-th node in $b_t$
    $Z_{t,b,i} := 0$
    $S := S \cup \{n_t\}$
    if $|S| = k$ then
        $Q := Q \cup S$
        $S := S \setminus \{n_t\}$
    else
        $Z_{t+1,b} := Z_{t,b} \wedge M(n_t)$ for all $b \in \overline{B}_S$
        $B_S := B_S \cup \{b_t\}$;  $\overline{B}_S := \overline{B}_S \setminus \{b_t\}$
        $P :=$ number of partitions $Z_{t,b}$ with $|Z_{t,b}| > 0$, $b \in \overline{B}_S$
        if $|S| + P = k$ then
            BitCLQ($t+1$)
            $S := S \setminus \{n_t\}$
            $B_S := B_S \setminus \{b_t\}$;  $\overline{B}_S := \overline{B}_S \cup \{b_t\}$
        else
            $S := S \setminus \{n_t\}$
            $B_S := B_S \setminus \{b_t\}$;  $\overline{B}_S := \overline{B}_S \cup \{b_t\}$
        end if
    end if
    $i :=$ the first significant bit in $Z_{t,b_t}$
until $i \leq |V_{b^t}|$
```



$$M = \begin{array}{cc} & \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} & \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Figure 1: A 3-partite graph and its adjacency matrix.

$t := 1$,
$S := \{1\}$,
$Z_2 := Z_1 \wedge M(1) = \{011|111|111\} \wedge \{000|111|011\} = \{000|111|011\}$,
$\overline{B}_S := \{2, 3\}$.

As a result, the set Z_2 of nodes compatible with the partial solution $S = \{1\}$ contains nodes $\{4, 5, 6, 8, 9\}$. Since none of the partites in $\overline{B}_S$ is empty, the partial solution S is promising and a new subproblem is created. The objective in the new subproblem is to find a $|\overline{B}_S|$-clique in Z_2. A node from $Z_{2,3}$ will be added to S (since $|Z_{2,3}| < |Z_{2,2}|$). The first node in $Z_{2,3}$ to add to the partial solution is node 8. The bit corresponding to node 8 is set $Z_{2,3,2} := 0$, and we have

$t := 2$,
$S := \{1, 8\}$,
$Z_3 := Z_2 \wedge M(8) = \{000|111|001\} \wedge \{111|001|000\} = \{000|001|000\}$,
$\overline{B}_S := \{2\}$.

Again, the partites in $\overline{B}_S$ contain at least 1 node (node 6) in Z_3. So the partial solution is promising, and a new subproblem is created. In the next step, node 5 is added to S:

$t := 3$,
$S := \{1, 8, 6\}$.

At this point, since $|S| = k = 3$, i.e., a k-clique is found. To continue the search for other k-cliques, the last node in S is removed. BitCLQ searches $Z_{3,2}$ for another node that can be added to S. Since such a node does not exist, the algorithm backtracks: $t := 2$, node 8 is removed from S, and BitCLQ restarts with $S = \{1, 9\}$ as the partial solution.

Table 1: Average computational time (in seconds) to find all the k-cliques (#CLQ) contained in randomly generated k-partite graphs.

k	m	$\|V\|$	p	#CLQ	FINDCLIQUE	BitCLQ
3	100	300	0.1	1004	0.005	0.002
4	100	400	0.15	1124	0.008	0.002
5	100	500	0.2	1047	0.015	0.003
6	100	600	0.25	939	0.031	0.006
7	50	350	0.35	192	0.009	0.004
8	50	400	0.4	299	0.021	0.007
9	50	450	0.45	683	0.055	0.021
10	50	500	0.5	2672	0.176	0.071

3 Numerical Results

In order to illustrate the performance of the proposed method, the k-clique enumeration problem for k-partite graphs has been solved by BitCLQ and FINDCLIQUE for randomly generated graph instances of several types. Both algorithms were implemented in C++ and ran on a 64-bit Windows machine with 3GHz dual-core processor and 4GB of RAM. It is worth noting that the original implementation of FINDCLIQUE algorithm by Grunert et al [3] relies on the use of `vectors` and `links` data types from the C++ standard template library (STL). In our experiments, we observed that by replacing the original data structure of vectors of lists with arrays, up to 300% improvement in FINDCLIQUE running time is achieved on the data sets used in our case study. The numerical results reported for the FINDCLIQUE algorithm are obtained using this "improved" implementation.

Our numerical experiments involve randomly generated instances of k-partite graphs of two types. The first set of instances consists of two groups: small-size instances and large-size instances. In the small-size instances, k-partite graphs are randomly generated with the number of partites in the range $k \in [3, 10]$. For each value of k, the reported running times and the number of k-cliques in the graph are averaged over 10 instances. Table 1 shows the summary of the experimental results for this first group. The columns of the table show the number k of partites in the k-partite graph, the number m of nodes in each partite of the graph, the total number $|V|$ of nodes in the graph, the graph's density p, and the total number of k-cliques in the graph (#CLQ). The density parameter p is used for generation of the graphs, and is equal to the probability of an edge connecting two nodes from different partites: $\Pr\{(v_i, v_j) \in E\} = p$.

The second group include instances of larger size with the values of $k \in \{25, 50, 75, 100\}$. For each value of k in this group, 10 random instances of the k-partite graph have been generated and solved by FINDCLIQUE and BitCLQ. Table 2 summarizes the results of the experiments for this group. Since the graphs used in this set of experiments are rather large and the list of all k-cliques contained in them may not be found in a reasonable time,

Table 2: Average number of k-cliques found in randomly generated instances of k-partite graphs after 200 seconds.

k	m	$\|V\|$	p	time	FINDCLIQUE	BitCLQ
25	40	1000	0.8	200	13,556,733	23,516,581
50	30	1500	0.9	200	800,369	1,032,111
75	30	2250	0.95	200	557,042,389	735,722,241
100	30	3000	0.95	200	348,416	365,799

Table 3: Average computational time (in seconds) needed to find the first n-clique in an n-partite graph corresponding to a randomized instance of the Multidimensional Assignment Problem with d dimensions and n elements per dimension.

n	d	m	$\|V\|$	p	BitCLQ	FINDCLIQUE
10	3	10	100	0.74	0.00	0.00
20	3	12	240	0.86	0.00	0.00
30	3	13	390	0.91	0.02	0.00
40	3	13	520	0.93	0.76	1.38
50	3	14	700	0.94	0.42	0.42
60	3	14	840	0.95	55.28	86.87
70	3	14	980	0.96	251.78	395.34
10	4	22	220	0.65	0.00	0.00
20	4	28	480	0.82	0.08	0.20
30	4	31	930	0.87	8.18	22.41
10	5	48	480	0.59	0.00	0.01
20	5	68	1360	0.77	13.29	28.23

the solution process has been terminated after 200 seconds and the number of k-cliques found by each method was recorded. BitCLQ outperformed FINDCLIQUE in all cases.

The third set of experiments was conducted to compare the performance of BitCLQ with FINDCLIQUE on randomly generated instances of Multidimensional Assignment Problem (MAP). As was mentioned before, high-quality solutions for randomized MAPs can be obtained as n-cliques in an n-partite subgraph of the underlying graph representing the MAP instance. graphs that are constructed in a special way from the problem's data (in this case, m denotes the number of elements per dimension in a d-dimensional MAP). For MAPs with random iid costs, the resulting n-partite graph can be viewed as randomly generated with a certain density. The corresponding results are reported in Table 3, where n denotes the number of partitions in the graphs, and d is the number of dimensions in the MAP. For each value of n, 10 instances are solved, and the computational time to find the first n-clique is recorded. Algorithms are terminated after finding the first n-clique. The average computational time over 10 runs is reported for each n for each algorithm. In all cases but one, BitCLQ performs better or equally well compared to FINDCLIQUE.

4 Conclusions

In this paper, bitset-based data structures are proposed for the algorithm presented by Grunert et al [3] for the problem of enumerating all k-cliques in a k-partite graph. Utilization of bitsets and the associated bit parallelism enables one to reduce the computational cost of branching and backtracking in the branch-and-bound procedure. Numerical experiments on small- and large-scale randomly generated k-partite graphs show that the proposed approach allows for achieving substantial computational improvements over the original method of [3].

Acknowledgements

The authors would like to acknowledge partial support of AFOSR grant FA9550-12-1-0142 and NSF grant EPS1101284.

References

[1] Immanuel M. Bomze, Marco Budinich, Panos M. Pardalos, and Marcello Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization*, pages 1–74. Kluwer Academic Publishers, 1999.

[2] S. Grabowski and K. Fredriksson. Bit-parallel string matching under hamming distance in $O(n\lceil m/w\rceil)O(n\lceil m/w\rceil)$ worst case time. *Information Processing Letters*, 105(5):182–187, 2008.

[3] T. Grunert, S. Irnich, H.J. Zimmermann, M. Schneider, and B. Wulfhorst. Finding all k-cliques in k-partite graphs, an application in textile engineering. *Computers & Operations Research*, 29(1):13–31, 2002.

[4] H. Hyyrö. Bit-parallel approximate string matching algorithms with transposition. *Journal of Discrete Algorithms*, 3(2–4):215–229, 2005.

[5] H. Hyyrö and G. Navarro. Bit-parallel witnesses and their applications to approximate string matching. *Algorithmica*, 41(3):203–231, 2004.

[6] P. Krokhmal, D. Grundel, and P. Pardalos. Asymptotic behavior of the expected optimal value of the multidimensional assignment problem. *Mathematical Programming*, 109(2–3):525–551, 2007.

[7] P. A. Krokhmal and P. M. Pardalos. Limiting optimal values and convergence rates in some combinatorial optimization problems on hypergraph matchings. *Submitted for publication*, 2011.

[8] Charles E. Leiserson, Harald Prokop, and Keith H. Randall. Using de Bruijn sequences to index a 1 in a computer word. *Working paper*, 1998, http://supertech.csail.mit.edu/papers/debruijn.ps.

[9] Q Liu and YPP Chen. High functional coherence in k-partite protein cliques of protein interaction networks. *Bioinformatics and Biomedicine*, 2009.

[10] R. D. Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15(2):169–190, 1950.

[11] M. Mirghorbani, P. Krokhmal, and E. L. Pasiliao. Computational studies of randomized multidimensional assignment problems. In Alexey Sorokin, My T. Thai, and Panos M. Pardalos, editors, *Dynamics of Information Systems*, page in press. Springer.

[12] M Peters. CLICK: Clustering categorical data using k-partite maximal cliques. *IEEE International Conference on Data Engineering*, 2005.

[13] Pablo San Segundo, Diego Rodríguez-Losada, and Agustín Jiminez. An exact bit-parallel algorithm for the maximum clique problem. *Computers & Operations Research*, 38(2):571–581, February 2011.

[14] Pablo San Segundo, Cristbal Tapia, Julio Puente, and Diego Rodrguez-Losada. A new exact bit-parallel algorithm for sat. In *ICTAI (2)'08*, pages 59–65, 2008.

1.

1. Report Type

Final Report

Primary Contact E-mail

Contact email if there is a problem with the report.

lynn-hudachek@uiowa.edu

Primary Contact Phone Number

Contact phone number if there is a problem with the report

319-335-2123

Organization / Institution name

University of Iowa

Grant/Contract Title

The full title of the funded effort.

Combinatorial Optimal Stopping Problems

Grant/Contract Number

AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".

FA9550-12-1-0142

Principal Investigator Name

The full name of the principal investigator on the grant or contract.

Dr. Pavlo Krokhmal

Program Manager

The AFOSR Program Manager currently assigned to the award

Dr. Jean-Luc Cambier

Reporting Period Start Date

04/01/2012

Reporting Period End Date

12/31/2015

Abstract

Optimal resource utilization is one of the most general "meta"-settings in operations research: many hard optimization problems can be casted as problems of optimal resource utilization. Additional challenges are introduced by uncertainties; the difficulties are further multiplied in a dynamic context. This project has considered a class of discrete and combinatorial optimal resource utilization problems under uncertainties that arise in the context of the optimal stopping problems. In addition, as a generalization of traditional stochastic formulations that optimize the expected payoff or cost, we considered risk averse discrete and combinatorial optimization problems, where the risk of the stopping decision was estimated using a coherent or convex risk measure. In particular, we developed a special class of certainty equivalent (CE) measures of risk that can be represented via solution of a specially formulated (stochastic) optimization problem. A number of solution techniques for discrete and combinatorial problems involving CE measures have been developed, including exact methods based on polyhedral approximations, branch-and-bound and branch-and-cut algorithms, scenario decomposition techniques, and combinatorial branch-and-bound methods for risk-averse combinatorial optimization problems.

Distribution Statement

This is block 12 on the SF298 form.

Distribution A - Approved for Public Release

Explanation for Distribution Statement

If this is not approved for public release, please provide a short explanation. E.g., contains proprietary information.

SF298 Form

Please attach your SF298 form. A blank SF298 can be found here. Please do not password protect or secure the PDF The maximum file size for an SF298 is 50MB.

SF298.pdf

Upload the Report Document. File must be a PDF. Please do not password protect or secure the PDF . The maximum file size for the Report Document is 50MB.

Report.pdf

Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.

Archival Publications (published) during reporting period:

Rysz, M., Vinel, A., Krokhmal, P., and E. L. Pasiliao (2015) A scenario decomposition algorithm for stochastic programming problems with a class of downside risk measures, INFORMS Journal on Computing, 27(2), 416-430.

Vinel, A. and P. Krokhmal (2015) Certainty equivalent measures of risk, Annals of Operations Research, DOI:10.1007/s10479-015-1801-0.

Chernikov, D., Krokhmal, P., Zhupanska, O. I., and C. L. Pasiliao (2015) A two-stage stochastic PDE-constrained optimization approach to vibration control of an electrically conductive composite plate subjected to mechanical and electromagnetic loads, Structural and Multidisciplinary Optimization, 52(2), 227-352.

Vinel, A. and P. Krokhmal (2014) Polyhedral approximations in p-order cone programming, Optimization Methods and Software, 29(6), 1210-1237.

Rysz, M., Mirghorbani, M., Krokhmal, P. and E. L. Pasiliao (2014) On risk-averse maximum weighted subgraph problems, Journal of Combinatorial Optimization, 28(1), 167-185.

Vinel, A. and P. Krokhmal (2014) On valid inequalities for mixed integer p-order cone programming, Journal of Optimization Theory and Applications, 160(2), 439-456.

Rysz, M. , Krokhmal, P., and E.L. Pasiliao (2013) Minimum risk maximum clique problem, in: A. Sorokin and P. M. Pardalos (Eds), Dynamics of Information Systems: Algorithmic Approaches, Springer Proceedings in Mathematics & Statistics, vol. 51, 251–267

Morenko, Y., Vinel, A., Yu, Z., and P. Krokhmal (2013) On p-norm linear discrimination, European Journal of Operational Research, 231(3), 784-789.

Rysz, M., Pajouh, F., Krokhmal, P. and E. L. Pasiliao (2014) On risk-averse weighted k-club problems, Examining Robustness and Vulnerability of Critical Infrastructure Networks, NATO Science for Peace and Security Series - D: Information and Communication Security, vol. 37, 231-242.

Mirghorbani, M. and P. Krokhmal (2013) On finding k-cliques in k-partite graphs, Optimization Letters, 7(6), 1155-1165.

Papers under review:

Vinel, A. and P. Krokhmal (2015) Mixed-Integer Programming with a Class of Nonlinear Convex Constraints, under review in Discrete Optimization.

Rysz, M., Krokhmal, P., and E. L. Pasiliao (2015) Identifying resilient structures in stochastic networks: A two-stage stochastic optimization approach, under review in Networks.

Rysz, M., Pajouh, F., Krokhmal, P., and E. L. Pasiliao (2015) Identifying risk-averse low-diameter clusters in graphs with stochastic vertex weights, under review in Annals of Operations Research.

Changes in research objectives (if any):

During the last project period, more emphasis has been placed on the development of solution methods for risk-averse combinatorial problems.

Change in AFOSR Program Manager, if any:

Dr. Donald Hearn was replaced by Dr. Fariba Fahroo, who was replaced by Dr. Jean-Luc Cambier.

Extensions granted or milestones slipped, if any:

A no-cost extension for the period from 04/01/2015 to 12/31/2015 was requested and granted. The no-cost extension was requested due to the fact that the PI, Dr. Pavlo Krokhmal, was on sabbatical leave in 2015, during which he was awarded the National Research Council Senior Research Associateship Award that, as a condition, required that the recipient did not conduct conducting research on other grants during the period of the award.

AFOSR LRIR Number

LRIR Title

Reporting Period

Laboratory Task Manager

Program Officer

Research Objectives

Technical Summary

Funding Summary by Cost Category (by FY, $K)

	Starting FY	FY+1	FY+2
Salary			
Equipment/Facilities			
Supplies			
Total			

Report Document

Report Document - Text Analysis

Report Document - Text Analysis

Appendix Documents

2. Thank You

E-mail user

Mar 29, 2016 14:28:44 Success: Email Sent to: lynn-hudachek@uiowa.edu