AFRL-AFOSR-VA-TR-2015-0322

Selective Optimization

Shabbir Ahmed
GEORGIA TECH RESEARCH CORPORATION

07/06/2015
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)* 16-06-2015	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* 4/2012-4/2015

4. TITLE AND SUBTITLE
Selective Optimization

5a. CONTRACT NUMBER

5b. GRANT NUMBER
AFOSR FA9550-12-1-0154

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Shabbir Ahmed and Santanu S. Dey

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Georgia Institute of Technology

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

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

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution/A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project focuses on developing algorithms for optimization problems that have intrinsic limitations preventing the utilization of all available decision alternatives (problem variables) and/or the satisfaction of all constraints. Part of the optimization decision in these problems is the selection of which variables to use and/or which subset of constraints to satisfy. We refer to these problems as selective optimization (SO) problems. The combinatorial aspects of selection make these problems extremely difficult. In this project we develop a set of generic tools applicable to a wide class of selective optimization problems. Our approach is based on standard mixed-integer programming (MIP) formulations of selective optimization problems. While such formulations can be attacked by commercial optimization solvers, they typically exhibit extremely poor performance. We develop a variety of effective model and algorithm enhancement techniques for the standard MIP formulations. These techniques are easily integrable into commercial MIP solvers, thereby making them readily usable in applications of selective optimization.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					19b. TELEPHONE NUMBER *(Include area code)*

Reset

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.

Final Report

AFOSR FA9550-12-1-0154
Selective Optimization

Project term: 4/2012-4/2015

Shabbir Ahmed and Santanu S. Dey
Georgia Institute of Technology

Project Overview

This project focuses on developing algorithms for optimization problems that have intrinsic limitations preventing the utilization of all available decision alternatives (problem variables) and/or the satisfaction of all constraints. Part of the optimization decision in these problems is the selection of which variables to use and/or which subset of constraints to satisfy. We refer to these problems as selective optimization (SO) problems. A typical example is the partial coverage problem that seeks levels of resources (facilities, sensors, transmission locations) to cover or serve a set of specified tasks (customers, targets, receivers) while minimizing resource usage costs. In many settings, full coverage is economically or physically impossible, and a typical goal is to achieve partial (say 95%) coverage, i.e. only a subset of the covering constraints need to be satisfied. Applications of selective optimization arise in numerous diverse areas ranging from defense to medicine.

The combinatorial aspects of selection make these problems extremely difficult. In this project we develop a set of generic tools applicable to a wide class of selective optimization problems. Our approach is based on standard mixed-integer programming (MIP) formulations of selective optimization problems. While such formulations can be attacked by commercial optimization solvers, they typically exhibit extremely poor performance. We develop a variety of effective model and algorithm enhancement techniques for the standard MIP formulations. These techniques are easily integrable into commercial MIP solvers, thereby making them readily usable in applications of selective optimization.

In the following we describe our research activities, key results, and publications. Details of the results appear in the papers appended at the end of this report.

Performance period 4/2012-4/2013

The first year of the project was devoted to the following three activities:

1. Partial covering problems where a subset of the constraints is required to be satisfied: We analyzed the complexity of this class of problems, and developed strengthened formulations and algorithmic techniques which perform significantly better than standard MIP approaches. A paper on this work has been published.
2. Transportation problems where a subset of the demands is to be satisfied: When resources are constrained, one may be able to satisfy only a subset of demands, rather than all the demands. Therefore the transportation problem now includes two sets of decisions: the decision of which demands to satisfy and a decision of how to satisfy

these demands. We study the complexity of this problem and present cases where the problem is tractable. For all other cases, we developed cutting-plane based techniques to improve the efficacy of a general purpose IP solver solving this problem.

3. Scenario decomposition of stochastic combinatorial problems: We developed a new decomposition algorithm for stochastic 0-1 problems based on selecting solutions from the set of single scenario solutions, and refining this set iteratively. The algorithm is very easily parallelizable and has the best computational performance to-date on various benchmark problems. A paper on this work has been published.

The project provided partial support for PhD students Feng Qiu and Gustavo Angulo. Feng Qiu graduated in 2013 and is currently a postdoc in Argonne national labs. One month for each of the two PIs was also supported. The projects funds were also used for travel to various conferences (INFORMS, MIP2013, ICSP2013) for dissemination of research results.

The following publications resulted from the 2012-2013 activities:

1. F. Qiu, S. Ahmed, S.S. Dey, L. Wolsey. ‘‘Covering linear programming with violations,’’ INFORMS Journal on Computing, vol. 26, pp. 531-546, 2014.
2. P. Damci-Kurt, S.S. Dey, S. Kucukyavuz. "On the transportation problem with market choice” Discrete Applied Mathematics, vol.181, pp.54-77, 2015.
3. S. Ahmed. ‘‘A scenario decomposition algorithm for 0-1 stochastic programs,’’ Operations Research Letters, vol. 41, pp. 565-569, 2013.

Performance period 4/2013-4/2014:

The second year of the project was devoted to the following three activities:

1. Strengthening the bounds for estimating the probability of k-out-of-n events: Given a set of n random events, represented by n Bernoulli variables, we consider the computation of bounds on the probability that k out of n events occur when partial distribution information is available. Upper or lower bound can be computed for this probability using a linear program. We designed inequalities that can be added to this linear program to significantly strengthen these bounds. The bounding approach is very useful in deriving relaxations and restrictions of probabilistic set covering problems.
2. Forbidding vertices of a polytope: Given a polytope P and a subset X of its vertices, we study the complexity of linear optimization over the subset of vertices of P that are not contained in X. This problem is closely related to finding the k best basic solutions to a linear problem. We show that the complexity of the problem changes significantly depending on the encoding of both P and X. Using these results we show that optimizing on the binary all different polytope can be accomplished in polynomial-time.
3. Improved integer L-shaped method: The Integer L-shaped method is the state-of-the-art algorithm for solving two-stage stochastic programs with integer recourse. We develop two enhancements to this algorithm to significantly improve its performance. The first approach relies on carefully alternating between solving integer and linear subproblems, and the second approach uses our results from forbidding vertices of a polytope (item 2 above) to strengthen integer L-shaped cuts.

The project provided partial support for PhD students Gustavo Angulo and Kevin Ryan. Gustavo Angulo graduated in Summer 2014. He is currently a postdoctoral researcher at CORE, Belgium for a year and will bee joining as a faculty at Pontificia Universidad Católica de Chile. One month for each of the two PIs was also supported.

The following publications resulted from the 2013-2014 activities:

1. F. Qiu, S. Ahmed, S.S. Dey. “Strengthened bounds for probability of k-out-of-n events,” accepted for publication in Discrete Applied Math.
2. G. Angulo, S. Ahmed, S.S. Dey, V. Kaibel. ‘‘Forbidden vertices,’’ *Mathematics of Operations Research*, vol.40, pp.350-360, 2015.
3. G. Angulo, S. Ahmed, S.S. Dey. “Improving the integer L-shaped method,” under review in the INFORMS J. on Computing.

Performance period 4/2014-4/2015:

1. Transportation problems with a cardinality constraint on the number of demands to be satisfied: It is well-known that the intersection of the matching polytope with a cardinality constraint is integral. In this project we prove a similar result for the polytope corresponding to the transportation problem with market choice (TPMC) (studied in performance period 4/2012-4/2014) when the demands are in a specific set. This result generalizes the result regarding the matching polytope and also implies that some special classes of minimum weight perfect matching problem with a cardinality constraint on a subset of edges can be solved in polynomial time.
2. Intersection on mixing sets with cardinality constraint: Intersection of two mixing sets with a cardinality constraint arises as a relaxation of deterministic equivalent of chance-constrained programming problems with finite discrete distributions. We study an extended formulation of this set and describe the convex hull in some special cases.

The project provided partial support for PhD students Kevin Ryan and Qianyi Wang. One month for each of the two PIs was also supported.

The following publications resulted from the 2013-2014 activities:

1. M. Walter, P. Damci-Kurt, S.S. Dey, S. Kucukyavuz. “On a Cardinality-Constrained Transportation Problem With Market Choice” submitted for publication, 2015.
2. K. Ryan, S. Ahmed, S.S. Dey. “Intersection of mixing sets with cardinality constraint,” working paper, 2015.

Covering Linear Programming with Violations

Feng Qiu, Shabbir Ahmed, Santanu S. Dey and Laurence A. Wolsey

May 8, 2012

Abstract

We consider a class of linear programs involving a set of covering constraints of which at most k are allowed to be violated. We show that this covering linear program with violation is strongly $\mathcal{NP}$-hard. In order to improve the performance of mixed-integer programming (MIP) based schemes for these problems, we introduce and analyze a coefficient strengthening scheme, adapt and analyze an existing cutting plane technique, and present a branching technique. Through computational experiments, we empirically verify that these techniques are significantly effective in improving solution times over the CPLEX MIP solver. In particular, we observe that the proposed schemes can cut down solution times from as much as six days to under four hours in some instances.

1 Introduction

A point belongs to the feasible region of a linear program (LP) if it satisfies all the linear constraints defining the LP. However, when certain problems are being modeled, the feasibility requirement is soft. That is, a point is considered feasible even if it violates no more than a specified number of the constraints defining the problem. Such a linear program is called a *k-violation linear program* (KVLP) [19]:

$$\begin{array}{ll} \min & c^\top x \\ \text{s.t.} & a_i^\top x \geq b_i \quad i = 1, ..., m, \\ & \text{at most } k \text{ of the } m \text{ constraints can be violated,} \\ & x \in \mathbb{R}^n_+. \end{array} \tag{1}$$

The feasible region of a KVLP is the union of $\binom{m}{k}$ polyhedral sets, each of which are defined by the intersection of some subset of $(m-k)$ inequalities from the m inequalities in (1). In general, such a feasible region is nonconvex and KVLP is a strongly $\mathcal{NP}$-hard optimization problem [1]. Much of the existing work on this class of problems focuses on polynomial time algorithms for low dimensional problems (i.e. n is fixed and small) (cf. [5] for a survey).

This paper addresses KVLPs where the linear system (1) consists of covering type linear inequalities, i.e., a_i and b_i are non-negative for all i. We call such a problem a *covering-type k-violation linear program* (CKVLP). CKVLPs, which are an important subclass of KVLPs, have many applications.

As a concrete example, consider a probabilistically-constrained portfolio optimization problem [16] to determine a minimum cost distribution of a unit investment among n assets with uncertain returns, requiring the overall return to be at least r with a probability of $1-\epsilon$, where $\epsilon \in (0,1)$ is a prespecified risk level. A formulation of this problem is

$$\begin{array}{ll} \min & c^\top x \\ \text{s.t.} & e^\top x = 1 \\ & \mathbb{P}\{\tilde{a}^\top x \geq r\} \geq 1 - \epsilon \\ & x \in \mathbb{R}^n_+, \end{array} \tag{2}$$

where $\tilde{a}$ is the random return vector for n assets following some known distribution, $\mathbb{P}\{A\}$ denotes the probability of the random event A, and c is the cost vector. A common approach to dealing with the probabilistic constraint in (2) is the sample average approximation method [12] where the distribution of $\tilde{a}$ is approximated by an empirical distribution corresponding to an i.i.d sample of return vectors $\{a_i\}_{i=1}^m$. The approximated problem then reads as follows:

$$\begin{array}{rll} \min & c^\top x & \\ \text{s.t.} & e^\top x = 1 & \\ & a_i^\top x \geq r \quad i = 1, ..., m, & \qquad (3) \\ & \text{at most } k \text{ of the } m \text{ constraints can be violated,} & \\ & x \in \mathbb{R}^n_+, & \end{array}$$

where $k = \lfloor m\epsilon \rfloor$. Since the return is non-negative and only nonnegative investments are allowed, (3) is an example of CKVLP with an additional equality constraint. In Section 6, we discuss a similar application of CKVLP in an optimal vaccine allocation under probabilistic constraints [18]. Additional applications of CKVLP arise in intensity modulated radiation therapy (IMRT) planning [20] and signal broadcasting coverage design [17].

A CKVLP can be modeled as a mixed integer program (MIP) in a straight-forward manner. First, note that if $b_i = 0$ for any $i \in \{1, \ldots, m\}$, then the corresponding inequality is redundant since then the inequality is implied by the non-negativity constraints on the x variables. Thus, we assume henceforth that $b_i > 0$ for all $i \in \{1, \ldots, m\}$ and so they can be scaled to 1. Then, an MIP formulation of CKVLP is

$$\begin{array}{rll} \min & c^\top x & \\ \text{s.t.} & a_i^\top x + z_i \geq 1 \quad i = 1, ..., m, & \qquad (4) \\ & \sum_{i=1}^m z_i \leq k & \\ & x \in \mathbb{R}^n_+, z_i \in \{0, 1\} \quad i = 1, ..., m, & \end{array}$$

where we have introduced the binary variables z_i taking the value 1 if the i-th constraint is violated. For large scale CKVLPs, the above MIP formulation performs very poorly. The goal of this paper is to study a number of enhancement schemes to improve the computational performance of MIP-based approaches for solving CKVLPs.

We begin by studying the theoretical complexity of CKVLPs and illustrating the difficulty of solving realistic instances directly by the CPLEX MIP solver (Section 2) as well. Next, in order to improve the performance of standard solvers on the MIP model (4) of CKVLPs, we introduce and analyze a coefficient strengthening scheme (Section 3), adapt and analyze an existing cutting plane technique (Section 4), and present a branching technique (Section 5). Through computational experiments on the probabilistic portfolio optimization problem (3) and an optimal vaccination allocation problem, we empirically verify that these techniques are extremely effective in improving solution times (Section 6). In particular, we observe that the proposed schemes can cut down solution times from as much as six days to under four hours in some instances.

We close this section by pointing out that all three enhancement schemes studied here are applicable when there are additional side constraints in the MIP (4). This follows since these schemes attempt to tighten the LP relaxation of (4), which is a valid relaxation even when additional side constraints are present.

2 Difficulty of Solving CKVLP

2.1 Computational Complexity

General KVLP has been shown to be $\mathcal{NP}$-complete [1]. However, to the best of our knowledge, the complexity of CKVLP, a sub-class of KVLP, has not been addressed. In a recent paper [20], Tunçel et al. showed that a packing version KVLP is weakly $\mathcal{NP}$-hard (the linear inequalities in KVLP are packing inequalities) by reduction from the partition problem. This result can be modified to show the $\mathcal{NP}$-hardness of CKVLP. In this paper we provide a direct proof that CKVLP is strongly $\mathcal{NP}$-hard.

By complementing the binary variables z in (4), we have the following equivalent formulation of CKVLP:

$$\begin{array}{ll} \min & c^\top x \\ \text{s.t.} & Ax \geq z \\ & e^\top z \geq p \\ & x \in \mathbb{R}^n_+ \\ & z \in \{0,1\}^m, \end{array} \tag{5}$$

where $A = [a_1^\top, ..., a_m^\top] \in \mathbb{Q}_+^{m \times n}$, $c \in \mathbb{Q}^n_+$, $p = m - k$, e is the column vector with each entry equal to 1, and $\mathbb{Q}$ is the set of rationals.

To prove that CKVLP (5) is $\mathcal{NP}$-hard, we first verify that the following intermediate decision problem is $\mathcal{NP}$-complete.

> *Intermediate CKVLP Feasibility Problem*: Given $\eta \in \mathbb{Q}$, $A \in \mathbb{Q}_+^{m \times n}$ and $c \in \mathbb{Q}^n$, does there exist a solution $(x, z) \in \mathbb{R}^n_+ \times \{0,1\}^m$ to the following system?
>
> $$\begin{array}{lcl} c^\top x - e^\top z & \leq & \eta \\ Ax & \geq & z. \end{array} \tag{6}$$

Lemma 1. *The Intermediate CKVLP Feasibility Problem (6) is strongly $\mathcal{NP}$-complete.*

Proof. Since (6) is a decision version of a mixed integer linear program, it is in $\mathcal{NP}$. In order to show that determining the feasibility of (6) is strongly $\mathcal{NP}$-complete, we polynomially reduce an arbitrary instance of the strongly $\mathcal{NP}$-complete vertex cover problem [8] to an instance of (6). An instance of the vertex cover problem is defined as follows:

> *Vertex Cover*: Given a graph $G = (V, E)$ and $q \in \mathbb{N}$, does there exist $S \subseteq V$ such that (i) $|S| \leq q$ and (ii) S is a vertex cover, that is for all $(i, j) \in E$, either $i \in S$ or $j \in S$?

Given an instance of the vertex cover problem, we construct an instance of (6) by setting $m := |V| + E|$, $n := |V|$, $\eta := q - |E|$, $c := 2e$, $A := \begin{bmatrix} H \\ I \end{bmatrix}$, where H is the node-arc incidence matrix of G and I is a $|V| \times |V|$ identity matrix. The resulting instance of (6) is then:

$$2\sum_{j\in V} x_j - \sum_{j \in V} z_j - \sum_{(i,j)\in E} y_{ij} \leq q - |E| \tag{7}$$

$$x_i + x_j \geq y_{ij} \qquad \forall\ (i,j) \in E \tag{8}$$

$$x_i \geq z_i \qquad \forall\ i \in V \tag{9}$$

$$x \in \mathbb{R}_+^{|V|} \tag{10}$$

$$z \in \{0,1\}^{|V|} \tag{11}$$

$$y \in \{0,1\}^{|E|}. \tag{12}$$

Note that the size of (7)-(12) is polynomial in the encoding length of G and q. We complete the proof by showing that a vertex cover instance has an answer yes if and only if the associated system (7)-(12) has a solution.

($\Rightarrow$) Let S be a vertex cover for G such that $|S| \leq q$. Then, consider a solution $(\tilde{x}, \tilde{y}, \tilde{z}) \in \mathbb{R}_+^{|V|} \times \{0,1\}^{|E|} \times \{0,1\}^{|V|}$ defined as:

$$\begin{array}{rcl} \tilde{x}_j = \tilde{z}_j & = & \begin{cases} 1 & \forall\ j \in S \\ 0 & \forall\ j \in V \setminus S, \end{cases} \\ \tilde{y}_{i,j} & = & 1\ \forall\ (i,j) \in E. \end{array}$$

The solution $(\tilde{x}, \tilde{y}, \tilde{z})$ satisfies (9)-(12) by construction, and since S is a vertex cover it also satisfies (8). Finally, $2\sum_{j\in V}\tilde{x}_j - \sum_{j\in V}\tilde{z}_j - \sum_{(i,j)\in E}\tilde{y}_{ij} = |S| - |E| \le q - |E|$. Thus the system (7)-(12) has a solution.

($\Leftarrow$) Now assume that the system (7)-(12) has a solution. Note that an arbitrary feasible solution to (7)-(12) may have fractional x components that cannot be directly converted to a vertex cover for G. We show that there exists a feasible solution of (7)-(12) with integral values of x and $y = e$ whenever (7)-(12) is feasible. Towards this end, we first present some properties of feasible solutions to (7)-(12). Given $(x, y, z) \in \mathbb{R}_+^{|V|} \times \{0,1\}^{|E|} \times \{0,1\}^{|V|}$, which satisfies (8)-(12), let

$$f(x, y, z) := 2\sum_{j\in V} x_j - \sum_{j\in V} z_j - \sum_{(i,j)\in E} y_{ij},$$

i.e., if (x, y, z) is feasible for (7)-(12), then $f(x, y, z) \le q - |E|$.

Claim a. Given (x^1, y^1, z^1) satisfying (8)-(12), there exists (x^2, y^2, z^2) satisfying (8)-(12) such that $y^2 = e$ i.e. a vector of ones, and $f(x^2, y^2, z^2) \le f(x^1, y^1, z^1)$.

Proof of Claim a. Suppose there exists $(\tilde{i}, \tilde{j}) \in E$ such that $y^1_{\tilde{i}\tilde{j}} = 0$. Construct (x^3, y^3, z^3) as follows:

$$x^3_j = \begin{cases} 1 & j = \tilde{i} \\ x^1_j & j \in V \setminus \{\tilde{i}\} \end{cases},$$

$$z^3_j = \begin{cases} 1 & j = \tilde{i} \\ z^1_j & j \in V \setminus \{\tilde{i}\} \end{cases},$$

$$y^3_{ij} = \begin{cases} 1 & (i,j) = (\tilde{i}, \tilde{j}) \\ y^1_{ij} & (i,j) \in E \setminus \{(\tilde{i}, \tilde{j})\}. \end{cases}$$

It is easy to see that (x^3, y^3, z^3) satisfies (8)-(12). We observe that $f(x^1, y^1, z^1) - f(x^3, y^3, z^3) = (2x^1_{\tilde{i}} - z^1_{\tilde{i}} - y^1_{\tilde{i},\tilde{j}}) - (2 \times 1 - 1 - 1) = x^1_{\tilde{i}} + (x^1_{\tilde{i}} - z^1_{\tilde{i}}) \ge 0$, where the last inequality holds due to the fact that (x^1, y^1, z^1) satisfies (9). By repeating the above construction at most $|E|$ times we arrive at a solution (x^2, y^2, z^2) satisfying the claim. $\diamondsuit$

We now restrict our attention to feasible solutions of (7)-(12) with the vector y fixed to e. Next, we show that a feasible solution with integral x components exists.

Claim b. Given (x^1, e, z^1) satisfying (8)-(12), there exists a solution (x^2, e, z^2) satisfying (8)-(12) such that $x^2 \in \{0,1\}^{|V|}$ and $f(x^2, e, z^2) \le f(x^1, e, z^1)$.

Proof of Claim b. If $x^1 \in \{0,1\}^{|V|}$, then there is nothing to verify. If there exists j such that $x^1_j > 1$, then we can set $x^1_j = 1$. The resulting solution still satisfies (8)-(12), and the value of the function f reduces. Therefore, the non-trivial case is when there exists $\tilde{j}$ such that $x^1_{\tilde{j}} \in (0,1)$. In this case, we construct a solution (x^3, e, z^3) as follows. Examine the set of neighboring vertices $N(\tilde{j})$ of the vertex $\tilde{j}$. If $x^1_{\tilde{i}} + x^1_{\tilde{j}} > y^1_{\tilde{i}\tilde{j}} = 1$ for all $\tilde{i} \in N(\tilde{j})$ then we may reduce the value of $x^1_{\tilde{j}}$ by a sufficiently small positive value so that (x^1, e, z^1) still satisfies (8)-(12) and the value of $f(x^1, e, z^1)$ reduces. Therefore, we may assume that there exists a vertex $\tilde{i} \in N(\tilde{j})$ such that $x^1_{\tilde{i}} + x^1_{\tilde{j}} = 1$. Without loss of generality, we may assume that $1 > x^1_{\tilde{j}} \ge \frac{1}{2}$ (otherwise we can swap $\tilde{i}$ and $\tilde{j}$), which also implies that $z^1_{\tilde{j}} = 0$. Next, we construct (x^3, e, z^3) as follows:

$$x^3_j = \begin{cases} 1 & j = \tilde{j} \\ x^1_j & j \in V \setminus \{\tilde{j}\} \end{cases},$$
$$z^3_j = \begin{cases} 1 & j = \tilde{j} \\ z^1_j & j \in V \setminus \{\tilde{j}\} \end{cases}.$$

It is easy to see that (x^3, e, z^3) with $x^3_{\tilde{j}} \in \{0, 1\}$ as constructed above satisfies (8)-(12). Furthermore $f(x^1, e, z^1) - f(x^3, e, z^3) = 2x^1_{\tilde{j}} - (2-1) = 2x^1_{\tilde{j}} - 1 \geq 0$. By repeating the above construction at most $|V|$ times, we obtain the required (x^2, e, z^2) satisfying the claim. ♢

From the claims a and b, it is clear that there exists a feasible solution of the form (x, y, z) with (i) $y = e$ and (ii) $x \in \{0, 1\}^{|V|}$. If $x_j = 1$ and $z_j = 0$ for some j, then we can set $z_j = 1$, and the resulting solution is still feasible for (7)-(12). Therefore, we may assume that the feasible solution also satisfies $x_j = z_j$ for all $j \in V$. We select any such feasible solution and let $S = \{j : \ x_j = 1\}$. Clearly, S is a vertex cover for G since $y = e$. Notice that $f(x, y, z) = 2|S| - |S| - |E| \leq q - |E|$ or equivalently $|S| \leq q$. □

Theorem 1. *CKVLP is strongly $\mathcal{NP}$-hard.*

Proof. To verify that (5) is $\mathcal{NP}$-hard, we show that if there exists a polynomial time algorithm for solving (5), then there is a polynomial time algorithm for deciding the feasibility of (6). This completes the proof, since by Lemma 1, we have that deciding the feasibility of (6) is $\mathcal{NP}$-complete.

Let $v(p)$ denote the optimal value of (5) as a function of $p \in \{0, \dots, m\}$. Consider the following algorithm for deciding the feasibility of (6):

1. Given $A \in \mathbb{Z}^{m \times n}_+$, $c \in \mathbb{Z}^n$, and $\eta \in \mathbb{Z}$, compute $v(p)$ for all $p \in \{0, \dots, m\}$, using the polynomial-time algorithm for solving (5).

2. Compute $\eta^* := \min_{0 \leq p \leq m} \{v(p) - p\}$. If $\eta^* \leq \eta$, return "yes," (i.e. (6) is feasible); otherwise return "no."

Notice that the above algorithm is a polynomial time algorithm in the size of the encoding of (6). It remains to verify the validity of the above algorithm.

Suppose $\eta^* \leq \eta$ and $p^* \in \operatorname{argmin}\{v(p) - p\}$. Consider an optimal solution (x^*, z^*) to (5) corresponding to $p = p^*$. Since $\eta \geq \eta^* = v(p^*) - p^* \geq v(p^*) - e^\top z^* = c^\top x^* - e^\top z^*$, the instance of (6) is feasible.

Suppose $\eta^* > \eta$. Assume by contradiction that the instance of (6) is feasible and let (x^*, z^*) be a feasible point. Let $p^* = \sum_{j=1}^m z^*_j$. Then, observe that (x^*, z^*) is feasible to (5) corresponding to $p = p^*$. Thus, $\eta^* \leq v(p^*) - p^* \leq c^\top x^* - p^* \leq \eta$, a contradiction. □

2.2 Performance of a standard MIP solver on CKVLP instances

Given the significant advancements made in MIP solution technology, many instances of $\mathcal{NP}$-hard problems are not necessarily difficult to solve in practice. To assess the practical computational difficulty of CKVLP, we next report on the performance of CPLEX, a state-of-the-art MIP solver, on randomly generated instances of the MIP (4).

We consider instances with $n = 20$, $m = 200$ and $k \in \{15, 20\}$. The data is generated as follows:

1. "Dense Data": Each left-hand-side coefficient a_{ij} is generated uniformly between 0.8 and 1.5, and then the coefficients are divided by 1.1. The cost vector is a vector of ones.

2. "Sparse Data": This uses the same input data as in "Dense Data", except that half of the left-hand-side coefficients are randomly set to zero.

3. "Random Objective": These instances have the same constraint coefficients as in "Dense Data", but with random integer cost coefficients between 1 and 100.

For each of the six combinations of two values of k and three data classes, we considered 10 instances with a total for 60 instances. The computations are run on Intel Xeon 2.27 GHz dual core Linux server installed with 4 Gb RAM. The model is implemented with the callable libraries and solved by the MIP solver in CPLEX 12.1 with default settings.

The average results over ten instances in each size-data combination are presented in Table 1. The 'Gap' column in the table reports the root node LP relaxation gap closed by CPLEX cuts. That is, the value $\left(\frac{z^{LP+Cuts}-z^{LP}}{z^{*}-z^{LP}}\right) \times 100$, where $z^{LP+Cuts}$, z^{LP}, and z^{*} are the objective function values of the LP relaxation with CPLEX cuts at the root node, of just the LP relaxation, and of the MIP, respectively. The 'Nodes' and the 'Time' columns report the number of branch-and-bound tree nodes generated and the time in seconds needed to solve the instances to optimality, respectively.

	Dense Data			Sparse Data			Random Objective		
k	Gap	Nodes	Time	Gap	Nodes	Time	Gap	Nodes	Time
15	2%	3,537,864	2,454	7%	158,039	83	17%	1,777	1
20	2%	43,296,679	25,948	6%	1,769,574	917	21%	6,227	2

Table 1: Performance of CPLEX on CKVLP

Following are some observations based on the above computations.

1. The effect of k: Setting k to a larger value results in a substantial increase in time and memory consumption (measured in the number of nodes in the branch-and-bound tree), as seen by a ten-fold increase for the first two types of instances. This phenomenon can perhaps be explained by the combinatorial nature of CKVLP, which is to choose the linear program with the best objective value among $\binom{m}{k}$ linear programs. When k increases to $\lfloor \frac{m}{2} \rfloor$, the number of possible linear programs increases rapidly.

2. The effect of sparsity: The coefficient matrix density, measured by the number of non-zeros, can make instances significantly harder to solve, as seen by a 20-time increase in nodes and 30-time increase in time when the density increases from 50% to 100%. The dense coefficients not only make the LP relaxation hard to solve, but also make it hard for CPLEX to find effective cuts, e.g., CPLEX default cuts close only 2% of the LP relaxation gap in the "Dense Data" instances, whereas 6-7% of the gap is closed in the "Sparse Data" instances.

3. The effect of objective function: The objective function coefficients play a crucial role in determining the computational difficulty, as demonstrated by the contrast between "Dense Data" and "Random Objective". The instances with random objective coefficients can be solved in seconds; however, the instances with the same constraints but uniform objective coefficients in "Dense Data" take hours to solve. When the cost coefficients and the constraint coefficients are set up in a way so that the objective values of linear programs formed by different choices of linear constraints are close, the branch-and-bound procedure generates a great number of nodes, of which the LPs are similar in terms of bounds, and the MIP solver spends an enormous amount of time on proving optimality.

In the rest of the paper, we focus on variants of the most difficult class of the above instances, that is, instances that are very similar in type to "Dense Data," and attempt to tighten the root node lower bound and reduce the size of the search tree.

3 Iterative Coefficient Strengthening

In this section, we propose and analyze a scheme for strengthening the coefficients of the binary variables in the MIP formulation (4) of CKVLP. Let X denote the set of feasible x solutions of (4), i.e.

$$X := \{x \in \mathbb{R}^n_+ : \ \exists z \in \{0,1\}^m \text{ s.t. } a_i^\top x + z_i \geq 1 \ \forall \ i = 1, ..., m \text{ and } \sum_{i=1}^{m} z_i \leq k\}. \tag{13}$$

Definition 1. *A vector $\ell \in \mathbb{R}^m$ is called a* valid bound vector *if $\ell_i \leq \min\{a_i^\top x : \ x \in X\}$ for all $i = 1, \ldots, m$.*

Given a valid bound vector ℓ, let

$$X(\ell) := \{x \in \mathbb{R}^n_+ : \ \exists z \in [0,1]^m \text{ s.t. } a_i^\top x + (1 - \ell_i)z_i \geq 1 \ \forall \ i = 1, ..., m \text{ and } \sum_{i=1}^{m} z_i \leq k\}.$$

Proposition 2. *(i) If ℓ is a valid bound vector then $X(\ell) \supseteq X$. (ii) The bound vector $\ell = 0$ is valid. (iii) For valid bounds ℓ^1 and ℓ^2, if $\ell^2 \geq \ell^1$ then $X(\ell^1) \supseteq X(\ell^2)$.*

Proof. (i) If $x \in X$ then there exists $z \in \{0,1\}^m$ such that $a_i^\top x \geq \max\{1 - z_i, \ell_i\}$ for all $i = 1, \ldots, m$ and $\sum_{i=1}^m z_i \leq k$. Since $\max\{1 - z_i, \ell_i\} = 1 - (1 - \ell_i)z_i$ when $z_i \in \{0,1\}$, it follows that $a_i^\top x + (1 - \ell_i)z_i \geq 1$ and $x \in X(\ell)$. (ii) Since $a^\top x \geq 0$ for all $x \in \mathbb{R}^n_+$, we obtain that $\ell = 0$ is a valid bound vector. (iii) If $x \in X(\ell^2)$ then there exists $z \in [0,1]^m$ such that $a_i^\top x \geq 1 - (1 - \ell_i^2)z_i$ for all $i = 1, \ldots, m$ and $\sum_{i=1}^m z_i \leq k$. Since $z_i \geq 0$ this implies that $a_i^\top x \geq 1 - (1 - \ell_i^1)z_i$, hence $x \in X(\ell^1)$. □

Note that $X(0)$ is the projection, on to the x variables, of the LP relaxation of the MIP formulation (4). Proposition 2 suggests that we can strengthen this LP relaxation by iteratively tightening the bound vector ℓ and hence the coefficients of the binary variables in (4), starting from $\ell = 0$. Algorithm 1 describes such a coefficient strengthening procedure. Note that procedure requires solving m feasible linear programs with bounded objectives in each iteration.

Algorithm 1 Iterative Coefficient Strengthening

Input : A threshold parameter $\varepsilon > 0$ and the data (m, n, k, a_{ij}) describing X
Output : A valid bound vector $\hat{\ell} \in \mathbb{R}^m_+$

```
$\Delta \leftarrow 2\varepsilon$, $t \leftarrow 1$, $\ell^t \leftarrow 0$
while $\Delta > \varepsilon$ do
  for $i = 1, ..., m$ do
    $\ell_i^{t+1} = \min\{a_i^\top x : \ x \in X(\ell^t)\}$
  end for
  $\Delta \leftarrow ||\ell^{t+1} - \ell^t||_\infty$
  $t \leftarrow t + 1$
end while
$\hat{\ell} \leftarrow \ell^t$
```

Proposition 3. *Let $\{\ell^t\}$ be the sequence of bound vectors produced in Algorithm 1. We have (i) $\ell^{t+1} \geq \ell^t$ and (ii) ℓ^t is a valid bound vector for all t. Accordingly, Algorithm 1 terminates finitely returning a valid bound vector $\hat{\ell}$.*

Proof. We proceed by induction on t. For the base case $t = 1$ we have $\ell^1 = 0$, then (ii) holds from part (ii) of Proposition 2. Moreover $\ell_i^2 = \min\{a_i^\top x : \ x \in X(0)\} \geq 0$ for all i, hence (i) holds. Suppose now that (i) and (ii) hold for some $t > 1$. By definition $\ell_i^{t+1} = \min\{a_i^\top x : \ x \in X(\ell^t)\}$ for all $i = 1, \ldots, m$. Thus, for each $i = 1, \ldots, m$, $\ell_i^{t+1} \leq a_i^\top x$ for all $x \in X(\ell^t)$ and hence for all $x \in X$ since $X \subseteq X(\ell^t)$ from the validity of ℓ^t. Thus ℓ^{t+1} is a valid bound vector and (ii) holds for all t. By our induction hypothesis $\ell^{t+1} \geq \ell^t$ thus $X(\ell^{t+1}) \subseteq X(\ell^t)$ by part (iii) of Proposition 2. Thus $\ell_i^{t+2} = \min\{a_i^\top x : \ x \in X(\ell^{t+1})\} \geq \min\{a_i^\top x : \ x \in X(\ell^t)\} = \ell_i^{t+1}$ for all $i = 1, \ldots, m$, and so (i) holds for all t. Finally note that, for any t, $X(\ell^t) \supseteq X$ from part (i) of Proposition 2, thus $\ell_i^t = \min\{a_i^\top x : \ x \in X(\ell^t)\} \leq \min\{a_i^\top x : \ x \in X\} =: \bar{\ell}_i^*$, where $\bar{\ell}_i^*$ is a well defined finite value, for all $i = 1, \ldots, m$. Thus, for each $i = 1, \ldots, m$, $\{\ell_i^t\}$ is a bounded nondecreasing sequence, and hence convergent. It follows that for any $\varepsilon > 0$ there exists a sufficiently large value of t such that $||\ell^{t+1} - \ell^t||_\infty \leq \varepsilon$ ensuring finite termination of the algorithm. □

Next we analyze the strength of the LP relaxation of (4) using tightened coefficients derived using Algorithm 1. Given a cost vector c, let

$$v^* = \min\{c^\top x : \ x \in X\} \ \text{ and } \ z^L(\ell) = \min\{c^\top x : \ x \in X(\ell)\}, \tag{14}$$

be the optimal value of the MIP (4) and the optimal value of the LP relaxation corresponding to bound vector ℓ, respectively. Note that these values are finite as long as $c \geq 0$. Recall that $v^L(0)$ is the natural LP relaxation bound for (4), and the coefficient tightening scheme in Algorithm 1 is aimed to improve this bound. In the following we analyze this improvement as a function of the instance data. For simplicity of the analysis we assume that $c_j > 0$ and $a_{ij} > 0$ for all i and j. Let

$$\rho = \min_{i=1,\ldots,m} \min_{j=1,\ldots,n} \left\{ \frac{a_{ij}}{(1/m)\sum_{i'=1}^m a_{i'j}} \right\}. \tag{15}$$

Note that ρ is a measure of the variability of the constraint coefficient data and $\rho \in (0, 1]$. Let $\{\ell^t\}$ be the sequence of bound vectors produced by the scheme in Algorithm 1 with a threshold of $\varepsilon = 0$. From Proposition 3 we know that this sequence is convergent. Let

$$\ell^* = \lim_{t \to \infty} \ell^t. \tag{16}$$

Recall that m is the number of constraints in (4) and k is maximum number of constraints allowed to be violated.

Lemma 2. *Assuming $a_{ij} > 0$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$,*

$$\ell_i^* \geq \frac{m-k}{m-\rho k}\rho \ \ \forall \ i = 1, \ldots, m,$$

where ρ and ℓ^* are as defined in (15) and (16), respectively.

Proof. Let $\{u^t\}$ be a sequence of m dimensional vectors defined by the following recursion:

$$u_i^1 = 0 \ \text{ and } \ u_i^{t+1} = \rho(1 - (1 - u_i^t)k/m) \ \forall \ i = 1, \ldots, m, \ \forall \ t \geq 1. \tag{17}$$

First, we claim that

$$\ell^t \geq u^t \geq 0 \ \forall \ t \geq 1. \tag{18}$$

We prove this claim by induction on t. Note that (18) holds for $t = 1$ since $\ell_i^1 = u_i^1 = 0$ for all $i = 1, \ldots, m$. Suppose now that (18) holds for some $t > 1$. Since $u_i^t \geq 0$ and $0 < k/m \leq 1$ we have that $(1 - (1 - u_i^t)k/m) = (1 - k/m) + u_i^t k/m \geq 0$, and hence $u_i^{t+1} \geq 0$. Let $\mu_j = \sum_{i=1}^m a_{ij}/m$ for $j = 1, \ldots, n$ and μ be the corresponding

n-dimensional vector. For any $i = 1, \dots, m$,

$$\begin{aligned}
\ell_i^{t+1} &= \min\{a_i^\top x :\ x \in X(\ell^t)\} && (19)\\
&\geq \min\{a_i^\top x :\ x \in X(u^t)\} && (20)\\
&= \min\{a_i^\top x :\ a_{i'}^\top x + (1-u_{i'}^t) z_{i'} \geq 1\ \forall\ i' = 1, \dots, m, \sum_{i'=1}^{m} z_{i'} \leq k, x \in \mathbb{R}_+^n, z \in [0,1]^m\} && (21)\\
&\geq \min\{a_i^\top x :\ \mu^\top x \geq 1 - (1-u_i^t)k/m, x \in \mathbb{R}_+^n\} && (22)\\
&= (1-(1-u_i^t)k/m)(\min_{j=1,\dots,n}\{a_{ij}/\mu_j\}) && (23)\\
&\geq \rho(1-(1-u_i^t)k/m) && (24)\\
&= u_i^{t+1}, && (25)
\end{aligned}$$

where (20) follows from the induction hypothesis $\ell^t \geq u^t$ since $X(\ell^t) \subseteq X(u^t)$ by Proposition 2(iii); (21) follows from the definition of $X(u^t)$; (22) follows by aggregating the m rows of the linear program defined in (21), noting that $u_{i'}^t = u_i^t$ for all i and i', and eliminating the z variables; since $(1-(1-u_i^t)k/m) \geq 0$, (23) follows from the optimal solution of the single constrained linear program defined in (22); (24) follows from the definition of ρ; and (25) follows from the definition of u_i^{t+1}. Thus (18) holds.

Next we claim that, for all $i = 1, \dots, m$, $\{u_i^t\}$ is convergent and

$$\lim_{t\to\infty} u_i^t = \frac{m-k}{m-\rho k}\rho. \tag{26}$$

Consider any $i \in \{1, \dots, m\}$. We first verify that $u_i^t \leq \frac{m-k}{m-\rho k}\rho$ for all t. We proceed by induction on t. By definition $u_t^1 = 0 \leq \frac{m-k}{m-\rho k}\rho$. By induction hypothesis, we have that $u_i^t \leq \frac{m-k}{m-\rho k}\rho$. Now $u_i^{t+1} = \rho - \rho\frac{k}{m} + \rho\frac{k}{m}u_i^t \leq \rho - \rho\frac{k}{m} + \rho\frac{k}{m}\left(\frac{m-k}{m-\rho k}\rho\right) = \frac{m-k}{m-\rho k}\rho$. Now we verify that the sequence $\{u_i^t\}$ is non-decreasing. Observe that $u_i^t - u^{t+1} = u_i^t - \left(\rho - \rho\frac{k}{m} + \rho\frac{k}{m}u_i^t\right) = u_i^t\left(1 - \rho\frac{k}{m}\right) - \rho + \rho\frac{k}{m} \leq \left(\frac{m-k}{m-\rho k}\rho\right)\left(1-\rho\frac{k}{m}\right) - \rho + \rho\frac{k}{m} = 0$. Finally suppose by contradiction that the sequence $\{u_i^t\}$ converges to a value $\frac{m-k}{m-\rho k}\rho - \delta$, where $\delta > 0$. Therefore, there exists a t such that $\frac{m-k}{m-\rho k}\rho - \delta > u_i^t > \frac{m-k}{m-\rho k}\rho - \delta - \epsilon$, in which $\epsilon = \delta\left(1 - \rho\frac{k}{m}\right)$. Since $\rho\frac{k}{m} < 1$, we have $\left(1-\rho\frac{k}{m}\right) < 1$. Hence, we obtain $u_i^t - u_i^{t+1} < \left(\frac{m-k}{m-\rho k}\rho - \delta\right)\left(1-\rho\frac{k}{m}\right) - \rho + \rho\frac{k}{m} = -(\delta)\left(1-\rho\frac{k}{m}\right) = -\epsilon$. Thus, $u_i^{t+1} > u_i^t + \epsilon > \frac{m-k}{m-\rho k}\rho - \delta$ which is a contradiction. Thus (26) holds.

It then follows from (18) and (26) that

$$\ell_i^* \geq \frac{m-k}{m-\rho k}\rho \ \ \forall\ i = 1, \dots, m.$$

□

Theorem 4. *Assuming $c_j > 0$ and $a_{ij} > 0$ for all $i = 1, \dots, m$ and $j = 1, \dots, m$,*

$$\frac{v^* - v^L(\ell^*)}{v^*} \leq \frac{m(1-\rho)}{m-\rho k}. \tag{27}$$

Proof. Note that

$$\begin{aligned}
v^L(\ell^*) &= \min\left\{c^\top x:\ a_i^\top x + (1-\ell_i^*)z_i \geq 1\ \forall\ i=1,\ldots,m,\ \sum_{i=1}^m z_i \leq k,\ x\in\mathbb{R}_+^n,\ z\in[0,1]^m\right\} && (28)\\
&\geq \min\left\{c^\top x:\ a_i^\top x + (1-\frac{m-k}{m-\rho k}\rho)z_i \geq 1\ \forall\ i=1,\ldots,m,\ \sum_{i=1}^m z_i \leq k,\ x\in\mathbb{R}_+^n,\ z\in[0,1]^m\right\} && (29)\\
&\geq \min\left\{c^\top x:\ \mu^\top x + (1-\frac{m-k}{m-\rho k}\rho)\frac{k}{m} \geq 1\ \forall\ i=1,\ldots,m,\ x\in\mathbb{R}_+^n\right\} && (30)\\
&= \frac{c_{\hat{j}}}{\mu_{\hat{j}}}\frac{m-k}{m-\rho k} && (31)
\end{aligned}$$

where

$$\hat{j}\in \operatorname{argmin}\left\{\frac{c_j}{\mu_j}\ :\ j=1,\ldots,n\right\}. \quad (32)$$

In the above, (29) follows from Lemma 2; (30) follows from aggregating the rows of the LP defined in (29) and eliminating the z variables; and (31) follows from solving the single constrained LP defined in (30).

Note that

$$v^* = \min\left\{c^\top x:\ a_i^\top x + z_i \geq 1\ \forall i\in\{1,\ldots,m\},\ \sum_{i=1}^m z_i \leq k,\ x\in\mathbb{R}_+^n,\ z\in\{0,1\}^m\right\}.$$

Next we obtain an upper bound on v^*. For $\hat{j}$ defined in (32):

1. Sort $a_{i\hat{j}}$'s from smallest to largest.
2. Let $a_{\hat{i}\hat{j}}$ be the $(k+1)^{\text{st}}$ smallest number.
3. Let $v^H = \frac{c_{\hat{j}}}{a_{\hat{i}\hat{j}}}$. This corresponds to the objective function value of the feasible solution $x_j = 0$ for $j \neq \hat{j}$ and $x_{\hat{j}} = \frac{1}{a_{\hat{i}\hat{j}}}$. Thus $v^* \leq v^H$.

Now observe that

$$\frac{c_{\hat{j}}}{\mu_{\hat{j}}}\frac{m-k}{m-\rho k} \leq v^L \leq v^* \leq \frac{c_{\hat{j}}}{a_{\hat{i}\hat{j}}} = z^H. \quad (33)$$

Therefore using the definition of ρ we obtain that,

$$\frac{v^*-v^L}{v^*} \leq \frac{z^H-v^L}{z^H} \leq \frac{m(1-\rho)}{m-\rho k}. \quad (34)$$

□

4 Mixing Set Inequalities

In this section, we study valid inequalities derived from a mixing set relaxation of CKVLP. A *mixing set* is defined as follows:

$$P = \{(y,z)\in\mathbb{R}_+\times\{0,1\}^n\ :\ y + h_i z_i \geq h_i, i=1,...,n\}, \quad (35)$$

where $h_1 \geq h_2 \geq \cdots \geq h_n$. The mixing set was introduced by Günlük and Pochet [10], and its variants in different contexts have also been studied in [6, 15, 7, 21, 9, 11]. The following inequalities, known as *mixing (set) inequalities*, are valid for P:

$$y + \sum_{j=1}^{l} (h_{t_j} - h_{t_{j+1}}) z_{t_j} \geq h_{t_1} \quad \forall\, T = \{t_1, ..., t_l\} \subseteq N, \tag{36}$$

where $h_{t_1} > h_{t_2} > \cdots > h_{t_l}$ and $h_{t_{l+1}} := 0$. Furthermore, these inequalities can be separated in polynomial time, are facet-defining for P when $t_1 = 1$, and are sufficient to describe the convex hull of P [2, 10].

Recently, the mixing set inequalities have been applied to solve the MIP formulation of chance-constrained problems, which has a k-violation-type substructure, i.e., a feasible solution must satisfy the constraints corresponding to at least k out of m scenarios [13, 14]. CKVLPs can be viewed as a special case of this substructure in which each scenario consists of only one covering linear constraint. We next describe and analyze the mixing set inequalities for CKVLPs.

Let the set of (x, z)-solutions to the MIP (4) be denoted by X_{MIP}, and recall from (13) that the set of x-solutions to (4) is denoted by X. Note that X is the projection of X_{MIP} into x-space, i.e., $X = \text{Proj}_x(X_{\text{MIP}})$. Following [14], we can obtain a mixing set relaxation of X_{MIP} as follows. Given a vector $\alpha \in \mathbb{R}^n_+$, calculate $\beta_i^\alpha, i \in \{1, \ldots, m\}$ as below:

$$\begin{aligned} \beta_i^\alpha := \quad & \min \quad \alpha^\top x \\ & \text{s.t.} \quad a_i^\top x \geq 1,\ x \in \mathbb{R}^n_+, \end{aligned}$$

where a_i is the coefficient vector for the i-th constraint in the MIP (4). Assume without loss of generality that $\beta_1^\alpha \geq \beta_2^\alpha \geq ... \geq \beta_m^\alpha$, and consider the following set

$$Y(\alpha) := \{(x, z) \in \mathbb{R}^n_+ \times \{0, 1\}^m \,:\, \alpha^\top x + (\beta_i^\alpha - \beta_{k+1}^\alpha) z_i \geq \beta_i^\alpha, \quad i = 1, ..., k\}. \tag{37}$$

Proposition 5. *For any $\alpha \in \mathbb{R}^n_+$, $X_{\text{MIP}} \subseteq Y(\alpha)$ and $X \subseteq \text{Proj}_x(Y(\alpha))$*

Proof. Let $(\bar{x}, \bar{z}) \in X_{\text{MIP}}$. Then the non-negativity constraints and integrality constraints in $Y(\alpha)$ are satisfied by $(\bar{x}, \bar{z})$. Without loss of generality, we may assume that the indexes $1, ..., k$ in $Y(\alpha)$ are the first k indexes in X_{MIP}. It remains to verify that $(\bar{x}, \bar{z})$ satisfies the constraints $\alpha^\top \bar{x} + (\beta_i^\alpha - \beta_{k+1}^\alpha)\bar{z}_i \geq \beta_i^\alpha$ for all $i = 1, ..., k$.

(i) For i such that $\bar{z}_i$=1: We require to verify that $\alpha^\top \bar{x} \geq \beta_{k+1}^\alpha$. Since $(\bar{x}, \bar{z}) \in X_{\text{MIP}}$, there exists some $u \in \{1, \ldots, k+1\}$ such that $a_u^\top \bar{x} \geq 1$. Moreover as $\beta_u^\alpha = \min\{\alpha^\top x : x \in \mathbb{R}^n_+,\ a_u^\top x \geq 1\}$, we obtain that $\alpha^\top \bar{x} \geq \beta_u^\alpha \geq \beta_{k+1}^\alpha$, where the last inequality is due to the fact that $u \leq k+1$.

(ii) For i such that $\bar{z}_i$=0: We require to verify that $\alpha^\top \bar{x} \geq \beta_i^\alpha$. Since $(\bar{x}, \bar{z}) \in X_{\text{MIP}}$, we obtain that $a_i^\top \bar{x} \geq 1$. Moreover as $\beta_i^\alpha = \min\{\alpha^\top x : x \in \mathbb{R}^n_+,\ a_i^\top x \geq 1\}$, we have that $\alpha^\top \bar{x} \geq \beta_i^\alpha$.

Therefore, $(\bar{x}, \bar{z}) \in Y(\alpha)$ and $X_{\text{MIP}} \subseteq Y(\alpha)$. The result $X \subseteq \text{Proj}_x(Y(\alpha))$ follows from the fact that $X = \text{Proj}_x(X_{\text{MIP}})$. □

The set $Y(\alpha)$ is a valid relaxation of X_{MIP} and it is in the form of a mixing set. This can be noted by considering $y := (\alpha^\top x - \beta_{k+1}^\alpha)$ as a nonnegative continuous variable to obtain the mixing system

$$y + (\beta_i^\alpha - \beta_{k+1}^\alpha) z_i \geq \beta_i^\alpha - \beta_{k+1}^\alpha \ \ \forall\, i = 1, \ldots, k.$$

Thus, we have the complete description of conv(Y(α)) using the inequalities (36), which are also valid for X_{MIP}, i.e., conv(X_{MIP}) $\subseteq$ conv(Y(α)). Let us call $\bigcap_{\alpha \in \mathbb{R}^n_+}$ conv(Y(α)) the *mixing closure*. Clearly, the mixing closure is a valid relaxation of conv(X_{MIP}). Let v^{MIX} be the optimal objective value of optimizing over the mixing closure, and v^* be the optimal objective value of the MIP (4). Then, the best root node gap that can be potentially achieved by the mixing inequality procedure is bounded by $(v^* - v^{\text{MIX}})/v^*$. To study this gap quantitatively, e.g., deriving a bound for $(v^* - v^{\text{MIX}})/v^*$, we analyze the projection of the mixing closure on the x-space, i.e., $\text{Proj}_x(\bigcap_{\alpha \in \mathbb{R}^n_+} \text{conv(Y}(\alpha)))$ in the following subsections.

4.1 The mixing closure

Note that

$$\operatorname{conv}(X) = \operatorname{Proj}_x(\operatorname{conv}(X_{\text{MIP}})) \subseteq \operatorname{Proj}_x(\bigcap_{\alpha\in\mathbb{R}^n_+} \operatorname{conv}(Y(\alpha))) \tag{38}$$

$$\subseteq \bigcap_{\alpha\in\mathbb{R}^n_+} \operatorname{Proj}_x(\operatorname{conv}(Y(\alpha))) = \bigcap_{\alpha\in\mathbb{R}^n_+} \operatorname{conv}(\operatorname{Proj}_x(Y(\alpha))).$$

Thus, minimizing over $\bigcap_{\alpha\in\mathbb{R}^n_+} \operatorname{conv}(\operatorname{Proj}_x(Y(\alpha)))$ yields a lower bound for v^{MIX}.

Proposition 6. $\operatorname{Proj}_x(Y(\alpha)) = \{x \in \mathbb{R}^n_+ : \alpha^\top x \geq \beta^\alpha_{k+1}\}$.

Proof. $\subseteq$: Let $\bar{x} \in \operatorname{Proj}_x(Y(\alpha))$, then there exists $\bar{z} \in \{0,1\}^k$ such that $\alpha^\top\bar{x} + (\beta^\alpha_i - \beta^\alpha_{k+1})\bar{z}_i \geq \beta^\alpha_i, i = 1, ..., k$. Thus $\alpha^\top\bar{x} \geq \beta^\alpha_i(1-\bar{z}_i) + \beta^\alpha_{k+1}\bar{z}_i \geq \beta^\alpha_{k+1}$ since $\beta^\alpha_i \geq \beta^\alpha_{k+1}$ and $\bar{z}_i \in [0,1]$.

$\supseteq$: Let $\bar{x} \in \{x \in \mathbb{R}^n_+ : \alpha^\top x \geq \beta^\alpha_{k+1}\}$, set $\bar{z}_i = 1, i = 1, ..., k$, then $(\bar{x}, \bar{z}) \in Y(\alpha)$ and $\bar{x} \in \operatorname{Proj}_x(Y(\alpha))$. □

Since $\operatorname{Proj}_x(Y(\alpha))$ is a half space in the non-negative orthant and hence convex, the convex hull operator in $\bigcap_{\alpha\in\mathbb{R}^n_+} \operatorname{conv}(\operatorname{Proj}_x(Y(\alpha)))$ is unnecessary.

Proposition 7.

$$\bigcap_{\alpha\in\mathbb{R}^n_+} \operatorname{conv}(\operatorname{Proj}_x(Y(\alpha))) = \bigcap_{\alpha\in\mathbb{R}^n_+} \operatorname{Proj}_x(Y(\alpha)) = \bigcap_{\alpha\in\mathbb{R}^n_+} \{x \in \mathbb{R}^n_+ : \alpha^\top x \geq \beta^\alpha_{k+1}\}.$$

Proposition 7 and (38) indicate that the projection of the mixing closure onto the x-space is contained in the closure constituted by infinitely many half spaces. To study this closure, we give a formal definition as below:

Definition 2 (Basic Mixing Closure)**.** The *Basic Mixing Closure* is defined as

$$\mathcal{M} := \bigcap_{\alpha\in\mathbb{R}^n} \{x \in \mathbb{R}^n_+ \,:\, \alpha^\top x \geq \beta^\alpha\}, \tag{39}$$

where $\beta_\alpha := \beta^\alpha_{k+1}$.

We call $\alpha^\top x \geq \beta^\alpha$ a *basic mixing inequality* corresponding to α. In order to understand the basic mixing closure, we describe another class of inequalities.

Definition 3 (Simple Disjunctive Cuts and Closure)**.**

1. Select a subset S of $k+1$ constraints. Since at least one of these constraints must be satisfied, we obtain the simple disjunction:

$$(a_{i_1}^\top x \geq 1, x \in \mathbb{R}^n_+) \vee (a_{i_2}^\top x \geq 1, x \in \mathbb{R}^n_+) \vee \cdots \vee (a_{i_{k+1}}^\top x \geq 1, x \in \mathbb{R}^n_+), \tag{40}$$

   where $S = \{i_1, \dots i_{k+1}\}$.

2. Define $a_S \in \mathbb{R}^n$ as

$$(a_S)_j = \max_{i\in S}\{a_{ij}\} \quad \forall j = 1, ..., n.$$

   The convex hull of (40) is

$$(a_S)^\top x \geq 1, x \in \mathbb{R}^n_+,$$

   and we call $(a_S)^\top x \geq 1$ a *simple disjunctive cut.*

We define the *simple disjunctive closure* as

$$\mathcal{D} := \bigcap_{S \subseteq \{1,\dots,m\}, |S|=k+1} \{x \in \mathbb{R}^n_+ \,:\, (a_S)^\top x \geq 1\}. \tag{41}$$

Proposition 8. $\mathcal{D} = \mathcal{M}$

Proof. $\mathcal{D} \subseteq \mathcal{M}$: For any given α, without loss of generality, let $\beta_1 \geq \dots \beta_k \geq \beta_{k+1} \geq \cdots \geq \beta_m$. Then $\beta_\alpha = \beta_{k+1}$. Since $\alpha^\top x \geq \beta_i$ is a valid inequality for the set $\{a_i^\top x \geq 1, x \in \mathbb{R}^n_+\} \quad \forall i = 1, ..., k+1$, $\alpha^\top x \geq \beta_\alpha$ is a valid inequality for the convex hull of the set

$$(a_1^\top x \geq 1, x \in \mathbb{R}^n_+) \vee (a_2^\top x \geq 1, x \in \mathbb{R}^n_+) \vee \cdots \vee (a_{k+1}^\top x \geq 1, x \in \mathbb{R}^n_+),$$

or equivalently $\alpha^\top x \geq \beta_\alpha$ is dominated by the inequality $(a_S)^\top x \geq 1$.

$\mathcal{M} \subseteq \mathcal{D}$: Let $S \subseteq \{1, \dots, m\}$ such that $|S| = k+1$. We set $\alpha = \alpha_S$. Then for any $i \in \{1, \dots, m\}$,

$$\begin{aligned} \beta_i = \quad & \min \quad (a_S)^\top x \\ & \text{s.t.} \quad (a_i)^\top x \geq 1, x \in \mathbb{R}^n_+. \end{aligned}$$

Since $a_{ij} \leq (a_S)_j$, we obtain that $\beta_i = \min_{1 \leq j \leq n} \frac{(a_S)_j}{a_{ij}} \geq 1$. Therefore, $\beta_{a_S} \geq 1$. Hence, the basic mixing inequality is

$$(a_S)^\top x \geq \beta_{(a_S)}$$

which dominates the inequality $(a_S)^\top x \geq 1$. □

Because m and k are finite numbers, the number of simple disjunctive cuts is also finite, the following result is immediate:

Corollary 9. *$\mathcal{M}$ is polyhedral.*

4.2 Bound Quality

Using the equivalence of $\mathcal{D}$ and $\mathcal{M}$, and the fact that $\mathcal{D}$ has an explicit form and simple structure, we derive a lower bound for v^{MIX} by studying $\mathcal{D}$. We then provide a bound on the best possible gap achievable by the addition of all possible mixing inequalities, i.e., $(v^* - v^{\text{MIX}})/v^*$.

Proposition 10. *Suppose $c > 0$ and $a_{ij} > 0$ for all i, j. Let $\underline{a} = \min_{ij}\{a_{ij}\}$ and $\overline{a} = \max_{ij}\{a_{ij}\}$. Let v^* be the optimal objective value over X and v^M be the optimal value over the basic mixing closure, then*

$$0 \leq \frac{v^* - v^{\text{MIX}}}{v^*} \leq \frac{v^* - v^M}{v^*} \leq \frac{\overline{a} - \underline{a}}{\overline{a}}.$$

Proof. Let $\underline{c} = \min_j\{c_j\}$. Note that $v^* \leq \min\{c^\top x : \ a_i^\top x \geq 1 \forall \ i = 1, \dots, m, \ x \in \mathbb{R}^n_+\} \leq \min\{c^\top x : \ (e^\top x) \geq 1/\underline{a}, \ x \in \mathbb{R}^n_+\} = \underline{c}/\underline{a}$. By the equivalence of $\mathcal{D}$ and $\mathcal{M}$, we obtain that $v^M = \min\{c^\top x : (a_S)^\top x \geq 1 \ \forall S \subseteq \{1, ..., m\}, |S| = k+1, x \in \mathbb{R}^n_+\} \geq \min\{c^\top x : \ \overline{a}(e^\top x) \geq 1, x \in \mathbb{R}^n_+\} = \underline{c}/\overline{a}$. Thus, $(v^* - v^M)/v^* = 1 - v^M/v^* \leq 1 - (\underline{c}/\overline{a})/(\underline{c}/\underline{a}) = (\overline{a} - \underline{a})/\overline{a}$. □

The above result implies that the relaxations $\mathcal{D}$ and equivalently $\mathcal{M}$ can be tight when the variation of the constraint coefficients is small. However, the separation of the most violated simple disjunctive cut from $\mathcal{D}$ is $\mathcal{NP}$-complete. Consider an arbitrary $x^* \in \mathbb{R}^n_+$ that we want to separate. Let $M := \{i \in \{1, \dots, m\} :$

$a_i^\top x^* < 1\}$. Clearly, $|M| > k$, because otherwise, x^* belongs to the feasible region of the k-violation problem X and therefore belongs to $\mathcal{D}$. When $|M| \geq k+1$, we solve the following separation problem:

$$\begin{aligned}
\eta = \quad & \min \quad \sum_{j=1}^{n} \pi_j x_j^* - 1 \\
& \text{s.t.} \quad \pi_j + a_{ij} w_i \geq a_{ij} \; j = 1, ..., n; \; \forall i \in M \\
& \qquad \sum_{i \in \mathcal{M}} w_i = |M| - (k+1) \\
& \qquad \pi_j \geq 0 \; \forall j \in \{1, \ldots, n\} \\
& \qquad w_i \in \{0, 1\} \; \forall i \in M,
\end{aligned}$$

where π_j is the cut coefficient for variable x_j and w_i is a binary variable taking value 0 whenever the i-th row is considered in the disjunction (40). The inequality $\sum_{j=1}^{n} \pi_j x_j \geq 1$ separates x^* from $\mathcal{D}$ if and only if $\eta < 0$. This separation problem is NP-hard [13]. Notice that although the mixing closure is contained in $\mathcal{D}$ and separation over $\mathcal{D}$ is $\mathcal{NP}$-complete, we do not know the complexity of the separation over $\text{Proj}_x(\bigcap_\alpha \text{conv}(Y(\alpha)))$.

5 Branching Scheme

As demonstrated in Table 1, the branch and bound search tree could be enormously large even for a small-sized instance of the MIP (4). Part of the reason for the excessive number of nodes is the overlap in the search tree. Without loss of generality, we assume that z_j is the binary variable to branch on at the root node. The left branch with z_j fixed at zero consists of the following set

$$\mathcal{B}^L := \{(x, z) : \sum_{i \neq j} z_i \leq k, a_j^\top x \geq 1, (x, z) \in X_{\text{MIP}}^j\},$$

where X_{MIP}^j represents the set X_{MIP} with the constraint $a_j^\top x + z_j \geq 1$ dropped and the variable z_j removed from the formulation. The right branch with z_j fixed at one consists of the following set

$$\mathcal{B}^R := \{(x, z) : \sum_{i \neq j} z_i \leq k-1, a_j^\top x \geq 0, (x, z) \in X_{\text{MIP}}^j\},$$

which is the union of the following two sets:

$$\mathcal{B}_{\geq}^R := \{(x, z) : \sum_{i \neq j} z_i \leq k-1, a_j^\top x \geq 1, (x, z) \in X_{\text{MIP}}^j\}$$

and

$$\mathcal{B}_{\leq}^R := \{(x, z) : \sum_{i \neq j} z_i \leq k-1, a_j^\top x \leq 1, (x, z) \in X_{\text{MIP}}^j\}.$$

Note that $\mathcal{B}_{\geq}^R$ is in fact a restriction of $\mathcal{B}^L$ and hence a overlap between the left and right branches. Re-exploring $\mathcal{B}_{\geq}^R$ in the right branch is a redundancy which could also hinder the infeasibility-based pruning: When $\mathcal{B}_{\leq}^R$ is infeasible but $\mathcal{B}_{\geq}^R$ is feasible, the overall right branch will be treated as a feasible node that, otherwise, would have been pruned. We can safely take $\mathcal{B}_{\geq}^R$ out of the right branch and the remaining search tree will still cover the whole solution space. This logic applies to any node with a z_i fixed at one.

One way to remove the overlap from the search tree is to introduce extra constraints and use a big-M

formulation to model the dichotomy of z_is:

$$\begin{aligned} \min \quad & c^\top x \\ & a_i^\top x + z_i \geq 1 \quad i = 1, ..., m \\ & a_i^\top x + M z_i \leq 1 + M \quad i = 1, ..., m \\ & \sum_{i=1}^{m} z_i \leq k \\ & x \in \mathbb{R}_+^n, z_i \in \{0, 1\} \quad \forall i = 1, ..., m. \end{aligned}$$

With this big-M formulation, however, the number of constraints doubles, and an appropriate large number M is not obvious. Instead, we remove the overlap during the branch-and-bound process as follows: whenever a branching variable z_i is fixed at one, we reverse the sign of $a_i^\top x \geq 1$ and add it as a local cut to this node. The addition of these local cuts keeps the search tree compact and could improve infeasibility-based node pruning. Another example exploring similar infeasiblity-based pruning can be found in [4].

6 Computational Experiments

In this section, we examine the potential impact of the proposed MIP approaches in solving two classes of problems with the CKVLP structure, i.e. MIPs of the form of (4). We implement the algorithms using CPLEX callable libraries (version 12.1), run the programs on Intel Xeon 2.27 GHz dual core Linux servers installed with 4 Gb RAM, and compare the performance against the CPLEX MIP solver with default settings.

6.1 Implementation Details

The implementation of the coefficient strengthening technique (described in Section 3) straightforwardly follows Algorithm 1. Notice that, we could obtain a tighter ℓ^t by enforcing integrality constraints on some binary variables in $X(\ell^t)$, but the series of minimization problems in Algorithm 1 would become more time-consuming. We keep $X(\ell^t)$ in Algorithm 1 as the set in Definition 1. The threshold parameter Δ is chosen to be 0.001.

In the implementation of the mixing set inequality procedure (described in Section 4), we add cuts only at the root nodes of search trees. We first solve the root node LP relaxation and obtain an optimal solution $(\bar{x}, \bar{z})$. Next we select the vector α from the following two sets:

- those constraint vectors a_i's for which $a_i^\top \bar{x} < 1$; and
- the cost vector c, if all a_is have been used as α.

Then we build a mixing set $Y(\alpha)$ as described in Section 4. Other than the most violated mixing inequality from (36), we also add violated inequalities (36) with $|T| = 2$ and $t_1 = 1$ to the root-node LP relaxation and solve it. The choice of these inequalities is based on recommendations in [14]. We iterate this process until one of the following stopping criteria is reached: (1) no cut with a violation of more than 0.00001 is identified, (2) the solution time exceeds 10,000 seconds, or (3) the cut generation procedure has run for 1000 iterations. To obtain the most violated mixing inequality, we implemented the separation algorithm in [2]. At the end of the cut generation phase, we keep only the tight cuts in the final model that is passed on to the branch-and-bound phase.

In the implementation of the branching rule, we add $a_i^\top x \leq 1$ as a local cut to the nodes in which z_i is fixed at one.

6.2 Probabilistic Portfolio Optimization

The first class of instances we test are from the probabilistically-constrained portfolio optimization model (2) introduced in Section 1. This problem can be approximated by the sample approximation approach as in (3) and reformulated as the following MIP [16]:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & e^\top x = 1 \\ & a_i^\top x + r z_i \geq r, \quad \forall i = 1, ..., m \\ & \sum_{i=1}^{m} z_i \leq k, \\ & x \in \mathbb{R}^n_+, z_i \in \{0,1\} \quad \forall i = 1, ..., n. \end{aligned}$$

where a_i is the i-th sample drawn from the distribution of $\tilde{a}_i$ and $k = \lfloor m \times \epsilon \rfloor$. The k-violation substructure in this formulation enforces that the number of sampled scenarios in which the overall return is not achieved must not exceed $\lfloor m \times \epsilon \rfloor$. Hence, $\frac{k}{m}$, the frequency, approximates the risk level ϵ. The constraint $e^\top x = 1$ is the budget constraint obtained by scaling the investment levels to a unit budget. We also considered instances where there is no budget constraint.

Each component of a_i is drawn from an independent uniform distribution between 0.8 and 1.5, which, in this context, represents the range between a 20% loss on one's investment and a 50% profit. The required return r is chosen to be 1.1, and ϵ is set at 0.05, indicating a ten percent average return with a probability of 95%. We set $n = 20$, $m = 200$, and $k = 15$, allowing, at most, 15 of 200 linear inequalities to be violated. The cost coefficients in the model with a budget constraint take on integer values uniformly distributed between 1 and 100. For the model without the budget constraint, we use the vector with all components equal to one as the cost vector, since the instances with this particular cost vector are especially difficult to solve. We select ten randomly generated instances for each model that can be solved by CPLEX within ten hours, and compare the proposed methods against CPLEX with default settings.

Tables 2 and 3 present the computational results for the model with a budget constraint and the model without a budget constraint, respectively. The first column gives the instance number. The second and third columns give the branch-and-bound (B&B) time (in seconds) and nodes of the CPLEX MIP solver (CPX). Columns 4-6 give the root node gap closed by the cuts generated by CPX, the coefficient strengthening (CS), and the mixing set inequalities (MIX), respectively. Finally, columns 7-9 and 10-12 compare the percentage improvements of the three schemes: the branching rule (BR), CS, and MIX, over the CPLEX MIP solver on branch-and-bound time and nodes, respectively. The percentage improvement in time for BR is computed as 100 × (Time(CPX) - Time(BR))/Time(CPX), where Time(CPX) is the branch-and-bound time for default CPLEX and Time(BR) is the branch-and-bound time using the proposed branching rule. The percentage improvements for the other two schemes, and the nodes saved are computed analogously.

The reported solution times are only for the branch-and-bound phase of the overall procedure. The mixing set cutting plane algorithm spends 20 to 30 seconds on root node until no more cuts can be separated. The time spent on coefficient strengthening, which amounts to solving a series of linear programming problems, is under 20 seconds. The local cuts added in the branching scheme can be obtained instantly by simply reversing the sign of the corresponding constraint. Since the preprocessing times in these instances are negligible in comparison with the branch-and-bound times, we do not include them in the solution time.

Table 2: Percentage Improvements Over CPLEX (Portfolio Optimization with Budget Constraint)

Instance	CPX Default		Root Gap Closed			B&B Time Saved			B&B Nodes Saved		
Number	B&B time	B&B nodes	CPX	CS	MIX	BR	CS	MIX	BR	CS	MIX
1	11,094	13,640,260	19%	24%	25%	91%	89%	60%	89%	85%	62%
2	968	1,657,606	19%	23%	24%	75%	82%	51%	70%	79%	54%
3	223	505,037	36%	36%	41%	77%	86%	25%	77%	86%	36%
4	19,830	25,651,409	7%	10%	13%	90%	91%	79%	88%	89%	80%
5	400	786,701	29%	31%	33%	70%	69%	-4%	66%	62%	-3%
6	5,044	9,786,835	14%	19%	22%	68%	82%	27%	66%	80%	37%
7	10,923	14,365,495	29%	33%	35%	82%	88%	26%	80%	85%	35%
8	1,822	3,177,889	12%	18%	21%	87%	90%	20%	86%	88%	35%
9	2,115	3,454,682	17%	24%	28%	71%	69%	-17%	66%	65%	-2%
10	13,017	10,526,548	15%	18%	21%	75%	88%	11%	79%	76%	-15%
Average	6,544	8,355,246	20%	24%	26%	79%	83%	28%	77%	80%	32%

Table 3: Percentage Improvements Over CPLEX (Portfolio Optimization Without Budget Constraint)

Instance	CPX Default		Root Gap Closed			B&B Time Saved			B&B Node Saved		
Number	B&B Time	B&B Nodes	CPX	CS	MIX	BR	CS	MIX	BR	CS	MIX
1	11,903	23,786,322	2%	64%	57%	-15%	85%	-176%	22%	91%	85%
2	14,584	24,366,521	4%	66%	61%	29%	95%	65%	38%	95%	89%
3	8,730	17,586,672	2%	64%	58%	14%	92%	-181%	19%	93%	88%
4	5,516	10,898,121	5%	64%	59%	7%	90%	51%	19%	91%	83%
5	12,462	18,021,273	4%	66%	62%	19%	95%	65%	19%	94%	89%
6	21,475	30,948,921	2%	64%	58%	58%	92%	65%	46%	93%	84%
7	6,928	14,634,688	2%	64%	60%	-27%	86%	44%	17%	88%	80%
8	15,547	20,957,656	2%	65%	61%	42%	93%	68%	33%	94%	89%
9	34,512	68,752,624	2%	64%	55%	41%	89%	63%	50%	94%	84%
10	5,314	9,376,843	2%	65%	60%	-2%	94%	69%	14%	94%	88%
Average	13,697	23,932,964	3%	65%	59%	17%	91%	13%	28%	93%	86%

From Tables 2 and 3 we observe that the mixing set inequalities and coefficient strengthening have comparable performance in terms of closing root node gaps. They both close more gap than the CPLEX default cuts, especially in the model without a budget constraint. However, in the branch-and-bound process afterwards, the mixing set inequalities cannot take full advantage of the tighter lower bounds to reduce overall time and nodes. In fact, in four of the 20 instances, the mixing set inequalities even worsen the performance. The reason lies in the difficulty of selecting effective cuts to keep in the model throughout the branch-and-bound process. In our experiment, we also try to employ the CPLEX cut pool to dynamically manage all the cuts generated at root nodes, but we have not been successful in identifying the most useful cuts.

The coefficient strengthening technique closes gap amounts similar to those closed by the mixing inequalities, but the improvement in the overall branch-and-bound process is significantly larger than for the mixing inequalities. The coefficient strengthening is able to cut down the time and nodes by an average of over 80%. This achievement can be attributed to the fact that the coefficient strengthening tightens the lower bound without introducing any extra variables or constraints at the root node.

The branching rule performs remarkably better in the model with the budget constraint, over 70% savings on nodes and time versus less than 30% savings in the model without the budget constraint. This sizable difference can be explained by the presence of the budget constraint. The budget constraint, as one type of side constraint, greatly reduces the feasible region of the node problems. Consequently, the feasibility of the node problems that have budget constraints is more sensitive to the addition of local cuts obtained by reversing the signs of the corresponding covering inequalities. Therefore, adding the local cuts to the models with budget constraints is more likely to lead to infeasible node problems, triggering the infeasibility-based node pruning more frequently.

6.3 Optimal Vaccination Allocation

The second class of test instances is the optimal vaccination allocation problem under uncertainty addressed in [18]. In this application, a scarce vaccine is allocated to households in a community to prevent an epidemic from breaking out by restricting the post-vaccination reproductive number to be strictly less than one. The sample average approximation approach to this problem yields a MIP formulation which has a CKVLP structure, plus some side constraints of form $\sum_{i \in S} x_i = 1$, where S is some subset of the index set of the decision variables. A full description of the model is provided in the Appendix. We use the same test instances of this problem as in [18]. These instances have 302 continuous variables and m binary variables (see Column 1 in Table 4). The risk level ϵ is set to 0.05, and the value of k can be determined accordingly by $k = \lfloor m \times \epsilon \rfloor$.

Table 4 compares the performance of three schemes against the performance of the CPLEX MIP solver. The first two columns describe the sizes of the instances. The next three columns provide the root node gaps closed by the cuts generated by the CPLEX MIP solver, the coefficient strengthening procedure, and the mixing inequalities, respectively. Columns 6-7 present the time (in seconds) spent on coefficient strengthening and generating mixing inequalities at the root node, respectively. Columns 8-11 and columns 12-15 compare the time (in seconds) and the number of nodes in the branch-and-bound phase by the CPLEX MIP solver and the three proposed schemes, respectively. Table 5 summarizes the percentage improvements of the three schemes over the CPLEX MIP solver with default settings. The percentage improvements in total time (root node time + branch-and-bound time) for CS is computed as 100 × (Time(CPX) - Time(CS))/Time(CPX), where Time(CPX) is the total time for default CPLEX and Time(CS) is the total time using coefficient strengthening. The percentage improvements in the branch-and-bound time (excluding the coefficient strengthening time) and the nodes saved are computed analogously. The percentage improvements for MIX and BR are computed similarly.

Table 5: Percentage Improvements Over CPLEX (Optimal Vaccination Allocation Problem)

Size		B&B Node Saved			B&B Time Saved			Total Time Saved		
m	k	CS	MIX	BR	CS	MIX	BR	CS	MIX	BR
250	12	43%	17%	40%	100%	100%	50%	-5033%	-7669%	50%
		88%	81%	80%	100%	100%	33%	-3220%	-5237%	33%
		92%	95%	92%	100%	100%	60%	-2186%	-3257%	60%
		99%	96%	95%	100%	100%	78%	-1037%	-1836%	78%
		78%	84%	78%	100%	100%	50%	-5191%	-7379%	50%
500	25	96%	88%	86%	90%	86%	69%	-1083%	-4142%	69%
		95%	87%	92%	91%	78%	78%	-701%	-2637%	78%
		96%	93%	84%	93%	82%	61%	-1808%	-6459%	61%
		91%	91%	82%	91%	82%	50%	-2251%	-7785%	50%
		100%	100%	98%	98%	96%	82%	-913%	-3833%	82%
750	37	100%	99%	97%	99%	99%	96%	-24%	-590%	96%
		78%	72%	-82%	75%	74%	-159%	-2506%	-14788%	-159%
		95%	91%	84%	89%	83%	57%	-1418%	-8252%	57%
		97%	96%	83%	97%	93%	63%	-380%	-3399%	63%
		95%	94%	89%	94%	86%	73%	-764%	-6194%	73%
1000	50	100%	99%	99%	100%	99%	97%	54%	-58%	97%
		97%	96%	96%	98%	96%	87%	12%	-180%	87%
		99%	96%	96%	99%	96%	92%	30%	-127%	92%
		91%	46%	67%	93%	61%	47%	-827%	-2800%	47%
		88%	59%	54%	88%	55%	7%	-1498%	-5689%	7%
2000	100	99%	78%	97%	99%	78%	92%	93%	72%	92%
		99%	59%	99%	99%	52%	96%	97%	49%	96%
		100%	87%	100%	99%	85%	98%	96%	81%	98%
		98%	85%	99%	99%	82%	96%	91%	74%	96%
		100%	30%	100%	100%	14%	98%	98%	12%	98%

Table 4: Computational Results (Optimal Vaccination Allocation Problem)

Size		Root Gap Closed			Root Node Time		B&B Time				B&B Nodes			
m	k	CPX	CS	MIX	CS	MIX	CPX	CS	MIX	BR	CPX	CS	MIX	BR
250	12	62%	95%	95%	103	155	2	0	0	1	81	46	67	49
		61%	93%	92%	100	160	3	0	0	2	673	81	129	134
		64%	93%	93%	114	168	5	0	0	2	1,096	83	52	91
		58%	92%	90%	102	174	9	0	0	2	3,054	39	112	149
		55%	94%	94%	106	150	2	0	0	1	153	33	24	33
500	25	52%	91%	89%	493	1,776	42	4	6	13	7,082	286	820	995
		50%	93%	90%	507	1,738	64	6	14	14	20,166	1,004	2,600	1,551
		55%	91%	90%	532	1,832	28	2	5	11	5,329	218	390	839
		54%	90%	90%	515	1,731	22	2	4	11	3,907	347	359	694
		48%	94%	92%	506	1,964	50	1	2	9	18,822	38	26	458
750	37	30%	91%	90%	1,656	9,230	1,340	7	16	59	206,179	463	1,648	5,276
		44%	91%	91%	1,574	9,066	61	15	16	158	7,124	1,554	1,989	12,944
		49%	93%	92%	1,522	8,418	101	11	17	43	16,857	804	1,436	2,729
		45%	91%	90%	1,106	8,101	232	8	16	87	40,393	1,057	1,497	6,692
		42%	91%	90%	1,116	8,164	130	8	18	35	23,886	1,198	1,332	2,658
1000	50	26%	90%	89%	2,928	10,162	6,449	30	35	182	829,738	3,142	4,773	6,351
		32%	91%	89%	3,107	10,056	3,636	86	143	482	433,591	11,998	16,333	15,915
		29%	92%	90%	3,157	10,144	4,546	28	166	376	348,976	2,552	13,239	15,253
		38%	91%	88%	3,218	10,012	350	25	137	185	28,423	2,424	15,406	9,278
		33%	91%	88%	2,791	10,109	176	21	79	163	15,241	1,821	6,310	7,065
2000	100	16%	88%	84%	10,684	10,804	166,074	843	36,096	13,992	9,978,113	57,601	2,237,052	294,053
		15%	89%	85%	10,259	11,084	386,246	2,740	185,168	15,960	24,523,780	191,138	10,109,370	316,629
		14%	89%	86%	10,570	11,276	324,023	1,779	49,923	6,305	24,463,163	116,273	3,297,139	114,177
		15%	89%	85%	11,082	11,091	141,172	2,023	24,990	5,129	9,032,616	138,575	1,350,094	89,084
		15%	89%	84%	11,257	11,448	574,819	1,889	493,400	10,795	39,500,399	128,181	27,842,365	184,543

The results in Table 4 and 5 show the effectiveness of the coefficient strengthening technique in both closing root node gaps and reducing nodes and time of the branch-and-bound phase. We observe that the performance of the coefficient strengthening algorithm is significantly more consistent than the other two methods and exhibits a certain stability. For example when $m = 1000$, the branch-and-bound time saved by the branching scheme ranges from 7.4% to 97.2%; the branch-and-bound time saved by the mixing set inequalities ranges from 55.1% to 99.5%; in contrast, the coefficient strengthening algorithm varies only from 88.1% to 99.5%. This consistent behavior is also observed for the probabilistic portfolio optimization instances in Tables 2 and 3. The branching scheme has a comparable impact on reducing the search tree size to the coefficient strengthening in the vaccination instances, especially for the difficult ones with $m = 2000$. Since this model consists of equalities as side constraints, the local cuts added by the branching rule cause infeasibility in the node problems frequently, therefore, effectively reducing the search tree size.

The performance improvement in the branch-and-bound phase comes at the expense of computational effort in coefficient strengthening and separation of mixing inequalities at the root node. Unlike the portfolio optimization instances, this effort is quite significant for the vaccination instances (see columns 6-7 in Table 4). Each iteration of the coefficient strengthening requires solving m linear programs – for the instances with $m = 1000$ and $m = 2000$, several thousand linear programs need to be solved. Similarly, in generating the mixing set inequalities, m linear programs need to be solved in order to form one mixing set for a given α, and there are m possible choices for α. Accordingly, the cut generation time increases in the order of m^2. Comparing column 8 in Table 4 and column 9 in Table 5, we observe that significant effort on coefficient strengthening is not justifiable for instances that CPLEX can solve in under 1500 seconds. For example, for the instances with $m = 1000$, the coefficient strengthening technique takes around 3000 seconds. Recall that we impose a time limit of 10000 seconds, so for these instances coefficient strengthening is run till no coefficients can be further tightened. Considering the fact that CPLEX takes only one to two hours to solve these instances, running the strengthening procedure to termination is not economical. Similarly, we observe (by comparing column 8 in Table 4 and column 10 in Table 5) that the effort on mixing inequalities is not justified for instances with $m < 2000$ that CPLEX can solve within 6500 seconds. On the overall solution time, the branching rule has a more consistent performance since it requires no additional effort at the root node. For the larger size instances with $m = 2000$, it is worth spending about three hours on strengthening to reduce the branch-and-bound time from days to minutes. The CPLEX MIP solver takes one to six days to solve these instances to optimality, whereas the coefficient strengthening reduces the overall effort to under four hours.

7 Concluding Remarks

In this paper, we study covering-type k-violation linear programs. We show that such problems are strongly NP-hard, and study empirically the computational difficulty of MIP-based approaches for these problems. We introduce and analyze a coefficient strengthening scheme, adapt and analyze an existing cutting plane technique, and present a branching technique to improve the performance of MIP approaches. Computational experiments on two classes of problems show that the proposed methods are effective in significantly reducing running times. The coefficient strengthening is most effective for large instances and reduces the solution time and the number of search tree nodes by 80% to 98% in these instances. The branching scheme reduces the size of search trees by removing overlaps between branches and incurring infeasiblity-based node pruning. It takes no effort to implement and works most effectively on the CKVLP models with side constraints. The mixing set cuts are capable of closing a large percentage of root node gaps. However, the impact of these cuts on the branch-and-bound process are mixed. Perhaps better performance might be achieved by a more effective separation procedure for mixing inequalities. We have also investigated the performance of various combinations of the three schemes, but the gains are not significant.

References

[1] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147(1-2):181–210, 1995.

[2] A. Atamtürk, G. L. Nemhauser, and M. W. P. Savelsbergh. The mixed vertex packing problem. *Mathematical Programming*, 89(1, Ser. A):35–53, 2000.

[3] N.G. Becker and D.N. Starczak. Optimal vaccination strategies for a community of households. *Mathematical Biosciences*, 139(2):117–132, 1997.

[4] D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74(2):121–140, 1996.

[5] T.M. Chan. Low-dimensional linear programming with violations. *SIAM Journal on Computing*, 34(4):879–893, 2005.

[6] M. Conforti and L.A. Wolsey. Compact formulations as a union of polyhedra. *Mathematical Programming*, 114(2):277–289, 2008.

[7] M. Constantino, A.J. Miller, and M. Van Vyve. Mixing MIR inequalities with two divisible coefficients. *Mathematical Programming*, 123(2):451–483, 2010.

[8] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* WH Freeman & Co. New York, NY, USA, 1979.

[9] Y. Guan, S. Ahmed, and G.L. Nemhauser. Sequential pairing of mixed integer inequalities. *Discrete Optimization*, 4(1):21–39, 2007.

[10] O. Günlük and Y. Pochet. Mixing mixed integer inequalities. *Mathematical Programming*, 90(3):429–457, 2001.

[11] S. Küçükyavuz. On mixing sets arising in chance-constrained programming. *Mathematical Programming*, 132(1-2):31–56, 2012.

[12] J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.

[13] J. Luedtke, S. Ahmed, and G.L. Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272, 2010.

[14] J. Luedtke. An integer programming and decomposition approach to general chance-constrained mathematical programs. In F. Eisenbrand and F. Shepherd, editors, *Integer Programming and Combinatorial Optimization*, volume 6080 of *Lecture Notes in Computer Science*, pages 271–284. Springer Berlin / Heidelberg, 2010.

[15] A.J. Miller and L.A. Wolsey. Tight formulations for some simple mixed integer programs and convex objective integer programs. *Mathematical Programming*, 98(1):73–88, 2003.

[16] B. Pagnoncelli, S. Ahmed, and A. Shapiro. Computational study of a chance constrained portfolio selection problem. *Journal of Optimization Theory and Applications*, 142:399–416, 2009.

[17] F. Rossi, S. Smriglio, and A. Sassano. Models and algorithms for terrestrial digital broadcasting. *Annals of Operations Research*, 107(1):267–283, 2001.

[18] M.W. Tanner and L. Ntaimo. IIS branch-and-cut for joint chance-constrained stochastic programs and application to optimal vaccine allocation. *European Journal of Operational Research*, 207(1):290–296, 2010.

[19] R. Thomas and P. Widmayer. k-violation linear programming. *Information Processing Letters*, 52:109–114, 1994.

[20] A. Tunçel, F. Preciado, R. Rardin, M. Langer, and J.-P.P. Richard. Strong valid inequalities for fluence map optimization problem under dose-volume restrictions. *Annals of Operations Research*, pages 1–22, 2010.

[21] M. Zhao and I.R. de Farias Jr. The mixing-MIR set with divisible capacities. *Mathematical Programming*, 115(1):73–103, 2008.

Appendix

Optimal Vaccination Allocation Model of [18]

The vaccination allocation problem allocate a scarce vaccine to households in a community to prevent an epidemic from breaking out. The epidemic will die out if the post-vaccination reproductive number is strictly less than one. Assume a community has a set F of types of households and each type of household $f \in F$ consists of a combination of person types $t \in T$, e.g., child, adult, or elderly. A vaccination policy $v \in V$ is a delivery of vaccine to certain types of persons in a household $f \in F$. For example, a vaccination policy could be a delivery of vaccine only to the two children in a household type that consists of two adults and two children. The decision problem is to determine an implementation of vaccination policies for each type of household in this community with a minimal cost which guarantees that the post-vaccination reproductive number is strictly below one with a high probability $1 - \epsilon$. We state below the probabilistically-constrained model in [18]:

$$
\begin{aligned}
\min : \quad & \sum_{f\in F}\sum_{v\in V}\sum_{t\in T} v_t h_f x_{fv} \\
\text{s.t.} \quad & \sum_{v\in V} x_{fv} = 1 \quad \forall f \in F \\
& \mathbb{P}\{\sum_{f\in F}\sum_{v\in V} a_{fv}(\omega) x_{fv} \le 1\} \ge 1-\epsilon \\
& 0 \le x_{fv} \le 1 \quad \forall f \in F,\, v \in V,
\end{aligned}
$$

where x_{fv} is the decision variable representing the percentage of policy v to be implemented for household type f, v_t is the number of people of type t vaccinated in policy v, h_f is the proportion of households in the community that are of type f, and $a_{fv}(\omega)$ is the computed random parameter for impact of the vaccination policy v for household type f, which is a function of different random numbers following some known distributions. For more details, see [3, 18].

After m i.i.d. samples are taken from $a_{fv}(\omega)$s, the above probabilistically-constrained problem can be approximated by the following MIP, which has a CKVLP structure:

$$
\begin{aligned}
\max : \quad & \sum_{f\in F}\sum_{v\in V}\sum_{t\in T} v_t h_f x'_{fv} - \sum_{f\in F}\sum_{v\in V}\sum_{t\in T} v_t h_f \\
\text{s.t.} \quad & \sum_{v\in V} x'_{fv} = 1 \quad \forall f \in F \\
& \sum_{f\in F}\sum_{v\in V} a^i_{fv} x'_{fv} + b_i z_i \ge b_i \quad i = 1, ..., m \\
& \sum_{i=1}^{m} z_i \le k \\
& 0 \le x'_{fv} \le 1 \quad \forall f \in F,\, v \in V,\; z_i \in \{0,1\}\ i = 1, .., m,
\end{aligned}
$$

where a^i_{fv} is the i-th sample of $a_{fv}(\omega)$, $x'_{fv} = 1 - x_{fv}$, $b_i = \sum_{f\in F}\sum_{v\in V} a^i_{fv} - 1$, and $k = \lfloor \epsilon \times m \rfloor$.

On the Transportation Problem With Market Choice

Pelin Damcı-Kurt[1], Santanu S. Dey[2], and Simge Küçükyavuz[1]

[1]Department of Integrated Systems Engineering, The Ohio State University
[2]School of Industrial and Systems Engineering, Georgia Institute of Technology

April 3, 2013

Abstract

We study a variant of the classical transportation problem in which suppliers with limited capacities have a choice of which demands (markets) to satisfy. We refer to this problem as the transportation problem with market choice (TPMC). While the classical transportation problem is known to be strongly polynomial-time solvable, we show that its market choice counterpart is strongly NP-complete. For the special case when all potential demands are no greater than two, we show that the problem reduces in polynomial time to minimum weight perfect matching in a general graph, and thus can be solved in polynomial time. Next, we consider the convex hull of solutions to the problem when a cardinality constraint is introduced on the number of rejected markets. We show that the cardinality constraint does not introduce new fractional extreme points for the case when TPMC is polynomially solvable. We give valid inequalities and coefficient update schemes for general mixed-integer sets that are substructures of TPMC. Finally, we give conditions under which these inequalities define facets, and report our preliminary computational experiments with using them in a branch-and-cut algorithm.

Keywords: Transportation problem, market choice, complexity, cardinality constraint, facet

1 Introduction

We consider a variant of the classical transportation problem in which suppliers with limited capacities have a choice of which demands (markets) to satisfy. In this problem, if a market is selected its demand must be satisfied fully through shipments from the suppliers. If a market is rejected, then the corresponding potential revenue is lost. The objective is to minimize the total cost of shipping and lost revenues. We refer to this problem as the transportation problem with market choice (TPMC).

More formally, we are given a set of supply and demand nodes that form a bipartite graph $G(V_1 \cup V_2, E)$. The nodes in set V_1 represent the supply nodes, where for $i \in V_1$, $s_i \in \mathbb{N}$ represents the capacity of supplier i. The nodes in set V_2 represent the potential markets, where for $j \in V_2$, $d_j \in \mathbb{N}$ represents the demand of market j. The edges between supply and demand nodes have weights that represent shipping costs w_{ij}, where $(i,j) \in E$. For each $j \in V_2$, r_j is the revenue lost if the market j is rejected. For a given vector of parameters γ_j for $j \in S$ and $S' \subseteq S$, we let $\gamma(S') := \sum_{j \in S'} \gamma_j$, throughout the paper.

Let x_{ij} be the amount of demand of market j satisfied by supplier i for $(i,j) \in E$, and let z_j be an indicator variable taking a value 1 if market j is rejected and 0 otherwise. A mixed-integer programming (MIP) formulation of the problem is given where the objective is to minimize the transportation costs and the lost revenues due to unchosen markets:

$$\min \sum_{(i,j)\in E} w_{ij}x_{ij} + \sum_{j\in V_2} r_j z_j \tag{1a}$$

$$\text{s.t.} \sum_{i:(i,j)\in E} x_{ij} = d_j(1-z_j) \qquad \forall j \in V_2 \tag{1b}$$

$$\sum_{j:(i,j)\in E} x_{ij} \le s_i \qquad \forall i \in V_1 \tag{1c}$$

$$z \in \{0,1\}^{|V_2|} \tag{1d}$$

$$x \in \mathbb{R}_+^{|E|}. \tag{1e}$$

We refer to problem description (1a)-(1e) as TPMC. The first set of constraints (1b) is the demand constraint. In TPMC either a demand for a market is fully satisfied or rejected altogether, which necessitates the introduction of the additional binary variables. The second set of constraints (1c) model the supply restrictions.

TPMC is closely related to the capacitated facility location (CFL) problem. In CFL, given a set of potential facilities $j \in V_2$ with capacities $\bar{d}_j, j \in V_2$ and customers $i \in V_1$ with demands $\bar{s}_i, i \in V_1$, we would like to determine which facilities to open so that the demand of all customers can be satisfied from shipments from the open facilities. A MIP formulation of CFL is

$$\sum_{i:(i,j)\in E} \bar{x}_{ij} \le \bar{d}_j \bar{z}_j \qquad \forall j \in V_2 \tag{2a}$$

$$\sum_{j:(i,j)\in E} \bar{x}_{ij} = \bar{s}_i \qquad \forall i \in V_1 \tag{2b}$$

$$\bar{z} \in \{0,1\}^{|V_2|} \tag{2c}$$

$$\bar{x} \in \mathbb{R}_+^{|E|}. \tag{2d}$$

Therefore one may view the CFL problem as a 'complement' of the TPMC problem where the constraints (1b) and (1c) of TPMC change signs in the constraints (2a) and (2b) in CFL respectively. Note that there is no straightforward way of 'complementing' the variables of TPMC in order to construct an instance of CFL or vice versa. While the CFL problem has been extensively studied with respect to its complexity, polyhedral structure, and approximability ([1, 8] and references therein), TPMC is less understood.

Recently, approximation algorithms and heuristics have been proposed for various supply chain planning and logistics problems with market choice [11, 18]. It is assumed that these problems are uncapacitated or that they have *soft* capacities. A two-stage approach is utilized in solving these classes of problems that admit a facility location formulation. In the first stage, the problem is to determine a subset of markets and reject the others. In the second stage, the goal is to minimize the production cost and lost revenues due to unselected markets. In particular, for the *uncapacitated* lot-sizing problem, the facility location formulation is used to model the market choice counterpart. It is shown that the LP relaxation solution can be rounded in a way that guarantees a constant factor approximation algorithm. However, this algorithm relies on scaling continuous variables up, so it does not immediately generalize to our problem with hard capacity constraints (1c). Van den Heuvel et al. [25] consider a maximization version of the same problem and show that no constant factor approximation algorithm exists for this version, unless P=NP. The authors also give several polynomially solvable special cases, and test heuristics for the general case.

The rest of the paper is organized as follows. In Section 2 we explore the complexity of TPMC. We show that while the classical transportation problem admits a strongly polynomial algorithm [16], its market choice counterpart is strongly NP-complete. We also identify a polynomially solvable case when the demands of all potential markets are no more than two. In Section 3 we consider a version of the problem with a service level constraint on the maximum number of markets that can be rejected. We show that for the case in which the original problem is polynomial, its cardinality-constrained version is also polynomial. Furthermore, in this case, we show that adding the cardinality constraint to the convex hull of solutions to the original problem does not create any new fractional extreme points. In Section 4 we present methods for constructing valid inequalities for mixed integer cover sets and mixed-integer knapsack sets with variable upper bound constraints, which appear as substructures of TPMC. We show that these methods are useful for generating valid inequalities for TPMC. We also study the strength of the proposed valid inequalities. Our preliminary computations, summarized in Section 5, show that there is a reduction in the root gap when our valid inequalities are incorporated to the branch-and-cut algorithm. However, we do not give an extensive computational study and the heuristic separation we use needs significant improvement.

2 Complexity

We first show that TPMC is strongly NP-hard in general.

Proposition 1. *The decision version of TPMC is NP-complete even when:*

1. $s_i = 1$ *for all* $i \in V_1$, $d_j = d \geq 3$ *for all* $j \in V_2$, $w_{ij} = 0$ *for all* $(i,j) \in E$ *and* $r_j = 1$ *for all* $j \in V_2$.
2. $|V_1| = 1$ *and* $w_{ij} = 0$ *for all* $(i,j) \in E$.

The proof for Proposition 1 Part 1 is similar to the proof of a related result presented in [22]. For completeness, we provide its proof and the proof of Part 2 in the Appendix. Because the reduction of Part 1 is from the Exact 3-Cover problem, which is strongly NP-complete [10], we conclude that TPMC is strongly NP-hard even for the case where all demands are equal to three. In contrast, Proposition 2 shows that TPMC is polynomially solvable when demands of all markets do not exceed two.

Proposition 2. *Suppose that* $d_j \leq 2$ *for all* $j \in V_2$. *Then there exists a polynomial-time algorithm to solve TPMC.*

This result is proven by a polynomial time reduction to a minimum weight perfect matching problem on a general graph (provided in the Appendix). The key ideas of the reduction are based on those presented in [3]. This result can also be proven by a polynomial time reduction to the b-matching problem [9], see also Theorem 36.1 in [23].

A matrix A is said to have the Edmonds-Johnson property if the sum of the absolute values of the entries in any column of A is less than or equal to 2. Edmonds and Johnson [9] show that the convex hull of integer solutions to a system $Ax \leq b$, where A has this property is given by the so-called blossom inequalities. Note that the constraint matrix defined by inequalities (1b), (1c), (1e), and $z \in \mathbb{R}_+^{|V_2|}$ have the Edmonds-Johnson property when $d_j \leq 2$ for all $j \in V_2$. Hence adding the blossom inequalities to the original formulation is enough to give the convex hull of solutions to TPMC in this case. The blossom inequality for TPMC is

$$\sum_{i \in U_1, j \in U_2 : (i,j) \in E} x_{ij} + \sum_{j \in U_2} \lfloor d_j/2 \rfloor z_j \leq \left\lfloor \frac{s(U_1) + d(U_2)}{2} \right\rfloor, \tag{3}$$

where $U_1 \subseteq V_1$, $U_2 \subseteq V_2$ such that the sum of total supply in U_1 and total demand in U_2, $s(U_1) + d(U_2)$, is odd. The separation of blossom inequalities (3) is polynomial [12, 17, 20]. We propose other classes of valid inequalities for the general case in Section 4.

3 TPMC with a cardinality constraint

An important and natural constraint that one may add to the TPMC problem is that of a service level, i.e., the number of rejected markets is restricted to be at most k. This restriction can be modelled using a *cardinality constraint*, $\sum_{j \in V_2} z_j \leq k$, appended to (1a)-(1e). We call the resulting problem cardinality constrained TPMC (CCTPMC). If we are able to solve CCTPMC in polynomial-time, then we can solve TPMC in polynomial time by solving CCTPMC for all $k \in \{0, \dots, |V_2|\}$. Therefore by Proposition 1, we obtain that CCTPMC is NP-hard in general. In this section, we examine the specific case where we know that TPMC admits a polynomial-time algorithm.

In light of the proof of Proposition 2, via the reduction to a minimum weight perfect matching problem on a general (non-bipartite) graph $G' = (V', E')$, it is possible to reduce CCTPMC with $d_j \leq 2$ for all $j \in V_2$ to a *minimum weight perfect matching problem with a cardinality constraint on a subset of edges* (specifically the cardinality constraint is applied only on the edges $(j, j') \in E'$ for each $j \in V_2$; see proof of Proposition 2 in Appendix). To the best of our knowledge, the complexity status of minimum weight perfect matching problem on a general graph with a cardinality constraint on a subset of edges is open. This can be seen by observing that if one can solve minimum weight perfect matching problem with a cardinality constraint on a subset of edges, then one can solve the exact perfect matching problem; see discussion in the last section in [6]. On the other hand, we will prove in this section that CCTPMC with $d_j \leq 2$ for all $j \in V_2$, which is a special case of a minimum weight perfect matching problem with cardinality constraint on a specific subset of edges, in fact admits a polynomial-time algorithm. Our approach will be following: We will prove that the TPMC polytope (when $d_j \leq 2$ for all $j \in V_2$) along with the constraint $\sum_{j \in V_2} z_j \leq k$ is integral. Therefore by invoking the ellipsoid algorithm it is possible to solve CCTPMC in polynomial time. This result also allows for solving CCTPMC (when $d_j \leq 2$ for all $j \in V_2$) by a Lagrangian relaxation approach, where we relax the cardinality constraint.

Before we proceed, we briefly note that the intersection of the perfect matching polytope with a cardinality constraint on a strict subset of edges is not always integral.

Example 1. *Consider the bipartite graph $G(V_1 \cup V_2, E)$ with $V_1 = \{1, 2, 3\}, V_2 = \{4, 5, 6\}$, $E = \{(1, 4), (1, 5), (2, 4), (2, 5), (2, 6), (3, 5), (3, 6)\}$, and the cardinality constraint $x_{14} + x_{25} \leq 1$. It is straightforward to show that $x_{14} = x_{15} = x_{24} = x_{25} = 0.5, x_{26} = x_{35} = 0, x_{36} = 1$ is a fractional extreme point of the intersection of the perfect matching polytope with the cardinality constraint.*

Let $X \in \mathbb{R}_+^{|E|} \times \{0, 1\}^{|V_2|}$ be the set of feasible solutions of TPMC. Our main result of this section is presented next.

Theorem 1. *Let $k \in \mathbb{Z}_+$ and $k \leq |V_2|$. Let $X^k := \operatorname{conv}(X \cap \{(x, z) \in \mathbb{R}_+^{|E|} \times \{0, 1\}^{|V_2|} \mid \sum_{j \in V_2} z_j \leq k\})$. If $d_j \leq 2$ for all $j \in V_2$, then $X^k = \operatorname{conv}(X) \cap \{(x, z) \in \mathbb{R}_+^{|E|} \times [0, 1]^{|V_2|} \mid \sum_{j \in V_2} z_j \leq k\}$.*

Corollary 3. *CCTPMC is polynomially solvable when $d_j \leq 2$ for all $j \in V_2$.*

Observation 1. *Theorem 1 is a generalization of the well-known result, Matching Cardinality Theorem: Let $G(V, E)$ be a graph with n vertices and m edges. Let $M \subset \mathbb{R}^m$ be the matching polytope and let $M^k \subset \mathbb{R}^m$ be the convex hull of incidence vectors of matchings with at least k edges. Then $M^k = M \cap \{x \in \mathbb{R}^m \mid \sum_{i=1}^m x_i \geq k\}$. (See [23] for a proof.)*

We construct a bipartite graph $\hat{G}(\hat{V}^1 \cup \hat{V}^2, \hat{E})$ as follows: $\hat{V}^1$ is a set of n vertices corresponding to the n vertices in G. $\hat{V}^2$ corresponds to the set of edges of G, i.e., $\hat{V}^2$ contains m vertices. We use (i, j) to refer to the vertex in $\hat{V}^2$ corresponding to the edge (i, j) in E. The set of edges in $\hat{E}$ are of the form $(i, (i, j))$ and $(j, (i, j))$ for every $i, j \in V$ such that $(i, j) \in E$. Now we can construct (the feasible region of) an instance of TPMC with respect to $\hat{G}(\hat{V}^1 \cup \hat{V}^2, \hat{E})$ as follows:

$$Q = \{(x, z) \in \mathbb{R}^{2m} \times \mathbb{R}^m \mid x_{i,(i,j)} + x_{j,(i,j)} + 2z_{(i,j)} = 2 \ \forall (i, j) \in \hat{V}^2 \tag{4}$$

$$\sum_{j:(i,j) \in E} x_{i,(i,j)} \leq 1 \ \forall i \in \hat{V}^1 \tag{5}$$

$$z_{(i,j)} \in \{0, 1\} \ \forall (i, j) \in \hat{V}^2\}. \tag{6}$$

We can construct an instance of CCTPMC by adding the constraint $\sum_{(i,j) \in E} z_{(i,j)} \leq k$ (call this set Q^k). It is straightforward to verify that the Matching Cardinality Theorem is equivalent to stating $\operatorname{conv}(Q^k) =$

$\text{conv}(Q) \cap \{(x,z) \,|\, \sum_{(i,j)\in E} z_{(i,j)} \leq k\}$. *Thus, the Matching Cardinality Theorem follows from Theorem 1 applied to the bipartite graph* $\hat{G}$.

Now note that the graph $\hat{G}$ *has a very special structure. In particular, the degree of every node in the second set of vertices* ($\hat{V}^2$) *is* 2. *On the other hand, Theorem 1 holds for a general instance of TPMC with* $d_j \leq 2$ *for all* $j \in V_2$, *i.e. in particular for instances corresponding to general bipartite graphs where the degree of the vertices can be more than* 2 *and the value of* d_j *can be either* 1 *or* 2. □

To prove Theorem 1, one approach could be to appeal to the reduction to minimum weight perfect matching problem and then use the well-known adjacency properties of the vertices of the perfect matching polytope. However, as illustrated in Example 1, the integrality result does not hold for the perfect matching polytope on a general graph with a cardinality constraint on any subset of edges. Therefore a generic approach considering the perfect matching polytope appears to be less fruitful. We use an alternative approach to prove this result. In particular, we apply a technique similar to that used in [2]. Consider the following desirable property:

Definition 1 (Edge Property). *Let* $T \subseteq \mathbb{R}^p_+ \times \{0,1\}^n$ *be some mixed integer set. We say that* T *satisfies the* edge property *if for all* $(w,r) \in \mathbb{R}^{p+n}$ *such that* $\min\{w^\top x + r^\top z \,|\, (x,z) \in T\}$ *is bounded and has at least two optimal solutions,* (x^1,z^1) *and* (x^2,z^2) *where* $\sum_{j=1}^n z^1_j = k^1$, $\sum_{j=1}^n z^2_j = k^2$ *and* $k^1 \leq k^2 - 2$, *then there is an optimal solution* (x^3,z^3) *such that* $\sum_{j=1}^n z^3_j = k^3$ *and* $k^1 < k^3 < k^2$.

Proposition 4. *Let* $T \subseteq \mathbb{R}^p_+ \times \{0,1\}^n$ *be a mixed integer set such that* $\text{conv}(T)$ *is a pointed polyhedron and let* $T^k := \text{conv}(T \cap \{(x,z) \in \mathbb{R}^p_+ \times \{0,1\}^n \mid \sum_{j=1}^n z_j \leq k\})$. *If* T *satisfies the edge property, then* $T^k = \text{conv}(T) \cap \{(x,z) \in \mathbb{R}^p_+ \times [0,1]^n | \sum_{j=1}^n z_j \leq k\}$.

Proof. Assume by contradiction that

$$T^k \neq \text{conv}(T) \cap \{(x,z) \in \mathbb{R}^p_+ \times [0,1]^n | \sum_{j=1}^n z_j \leq k\},$$

for some $k = k' \in \{0,1,\ldots,n\}$. By definition $T^k = \text{conv}(T \cap \{(x,z) \in \mathbb{R}^p_+ \times \{0,1\}^n | \sum_{j=1}^n z_j \leq k\})$ so $T^k \subseteq \text{conv}(T) \cap \{(x,z) \in \mathbb{R}^p_+ \times [0,1]^n | \sum_{j=1}^n z_j \leq k\}$ holds for all $k \in \{0,1,\ldots,n\}$. By assumption we obtain $T^{k'} \subset \text{conv}(T) \cap \{(x,z) \in \mathbb{R}^p_+ \times [0,1]^n | \sum_{j=1}^n z_j \leq k'\}$. Since $\text{conv}(T)$ is pointed this implies that there exists a vertex (x',z') of $\text{conv}(T) \cap \{(x,z) \in \mathbb{R}^p_+ \times [0,1]^n | \sum_{j=1}^n z_j \leq k'\}$ such that $(x',z') \notin T^{k'}$. Therefore z' is fractional and $\sum_{j=1}^n z'_j = k'$ (if $\sum_{j=1}^n z'_j < k'$, then this point is also a vertex of $\text{conv}(T)$, therefore integral and belonging to $T^{k'}$ - a contradiction).

Since (x',z') is not a vertex of $\text{conv}(T)$, there exists (w,r) such that the vertex (x',z') is the intersection of the face defined by $\{(x,z) \in \mathbb{R}^p_+ \times [0,1]^n | \sum_{j=1}^n z_j = k'\}$ and an edge of $\text{conv}(T)$ defined as:

$$\{(x,z) \in \text{conv}(T) \,|\, w^\top x + r^\top z = \delta\}, \tag{7}$$

where $\delta = \min\{w^\top x + r^\top z \,|\, (x,z) \in \text{conv}(T)\} = w^\top x' + r^\top z'$. Let (x^1,z^1) and (x^2,z^2) be two feasible points of T that belong to the edge (7) such that (x',z') is a convex combination of (x^1,z^1) and (x^2,z^2). Note that $\delta = w^\top x' + r^\top z' = w^\top x^1 + r^\top z^1 = w^\top x^2 + r^\top z^2$. Hence, (x^1,z^1) and (x^2,z^2) are two optimal solutions corresponding to the objective function (w,r). Furthermore, due to our selection of δ, $\sum_{j\in V_2} z^1_j < k' < \sum_{j\in V_2} z^2_j$. The edge property ensures that there exists an integral optimal solution (x^3,z^3) with $k^3 = \sum_{j\in V_2} z^3_j = k'$ such that $\sum_{j\in V_2} z^1_j < k^3 < \sum_{j\in V_2} z^2_j$. However, this implies that (x^3,z^3) belongs to the edge defined by (7). Thus, (x^3,z^3) must be a convex combination of (x^1,z^1) and (x^2,z^2) or equivalently, we must have $(x^3,z^3) = (x',z')$ with z' integral, a contradiction. □

Now, we show how edge property and Proposition 4 can be applied to TPMC with an additional constraint that at most k markets can be rejected. To prove Theorem 1 we use Proposition 4. Similar to the argument in the proof of Proposition 2, we assume that all data are integral, and that $s_i = 1$ for all $i \in V_1$ without loss of generality. It is straightforward to verify that the polyhedron X corresponding to the original instance with $s_i > 1$ for some $i \in V_1$ satisfies the edge property if and only if X corresponding to the corresponding instance with $s_i = 1$ for all $i \in V_1$ satisfies the edge property. We are now ready to present the proof of Theorem 1.

Proof of Theorem 1. By hypothesis $d_j \leq 2$ for all $j \in V_2$. From Proposition 4 it is sufficient to prove that the edge property holds.

Suppose that (x^1, z^1) and (x^2, z^2) are optimal solutions to $\min\{w^\top x + r^\top z \,|\, (x, z) \in X\}$ and that x^1 is fractional. Then we can solve a simple transportation problem with the set of demand nodes j such that $z^1_j = 0$. Since all data is integral, there exists an optimal solution with integral flows. Therefore, we may assume that x^1 (and similarly x^2) are integral.

Claim 1. *Suppose we have two feasible solutions of X, namely (x^3, z^3) and (x^4, z^4), such that*

1. $\sum_{j \in V_2} z^3_j = k^1 + 1$ *and* $\sum_{j \in V_2} z^4_j = k^2 - 1$ *and*
2. *The objective function value of (x^3, z^3) is $\rho - \delta$ and that of (x^4, z^4) is $\rho + \delta$, where ρ is the objective function value of the solution (x^1, z^1) and $\delta \in \mathbb{R}$,*

then the proof of Theorem 1 is complete.

Proof. Since ρ is the optimal objective function value, we obtain that $\delta = 0$ since otherwise the objective function value of either (x^3, z^3) or (x^4, z^4) is better than that of (x^1, z^1). Therefore (x^3, z^3) is an optimal solution with $k^1 < \sum_{j \in V_2} z^3_j < k^2$. Because edge property is satisfied by Proposition 4, the proof of Theorem 1 is complete. □

Given an integral point $(\tilde{x}, \tilde{z})$ of X, let $S(\tilde{z}) := \{j \in V_2 \,|\, \tilde{z}_j = 0\}$ be the set of nodes in V_2 whose demands are met. For $j \in S(\tilde{z})$, let $I_j(\tilde{x}, \tilde{z}) = \{i \in V_1 \,|\, \tilde{x}_{ij} > 0\} = \{i \in V_1 \,|\, \tilde{x}_{ij} = 1\}$ be the set of suppliers that sends one unit to j.

Given the optimal solutions (x^1, z^1) and (x^2, z^2), let $F := \big(S(z^1) \setminus S(z^2)\big) \cup \big(S(z^2) \setminus S(z^1)\big)$, $P := S(z^1) \cap S(z^2)$ and $R := V_2 \setminus (F \cup P)$. For $j \in F$, observe that only the set $I_j(x^1, z^1)$ or the set $I_j(x^2, z^2)$ is defined. So for $j \in F$, we define I_j as:

$$I_j := \begin{cases} I_j(x^1, z^1) & \text{if } j \in S(z^1) \setminus S(z^2) \\ I_j(x^2, z^2) & \text{if } j \in S(z^2) \setminus S(z^1). \end{cases} \tag{8}$$

As a first step towards constructing (x^3, z^3) and (x^4, z^4) required in Claim 1, we construct a bipartite (conflict) graph $G^*(U_1 \cup U_2, \mathcal{E})$. The set of nodes is constructed as follows:

1. If $j \in S(z^1) \setminus S(z^2)$, then $j \in U_1$ and j is called a *full node.* Let $W_1 = S(z^1) \setminus S(z^2)$ be the set of full nodes of U_1.
2. Similarly, if $j \in S(z^2) \setminus S(z^1)$, then $j \in U_2$ and j is called a *full node.* Let $W_2 = S(z^2) \setminus S(z^1)$ be the set of full nodes of U_2.
3. If $j \in S(z^1) \cap S(z^2)$ and $d_j = 2$ then we place two copies of node j in U_1 (call these j_1 and j_2) and two copies of j in U_2 (call these j_3 and j_4). These nodes are called *partial nodes* of j. Each partial node of j is distinct: If $I_j(x^1, z^1) = \{t_1, t_2\}$, then associate (WLOG) t_1 with j_1 and t_2 with j_2, that is define $I_{j_1} := \{t_1\}$ and $I_{j_2} := \{t_2\}$. Similarly if $I_j(x^2, z^2) = \{t_3, t_4\}$, then associate (WLOG) t_3 with j_3 and t_4 with j_4, that is define $I_{j_3} := \{t_3\}$ and $I_{j_4} := \{t_4\}$. If $j \in S(z^1) \cap S(z^2)$ and $d_j = 1$, then we place one copy of node j in U_1 (call this j_1) and one copy of j in U_2 (call this j_3). Similar to the $d_j = 2$ case these nodes are called *partial nodes* of j. If $I_j(x^1, z^1) = \{t_1\}$ and $I_j(x^2, z^2) = \{t_3\}$, then set $I_{j_1} = \{t_1\}$ and $I_{j_3} = \{t_3\}$. Let $P = P^1 \cup P^2$, where $P^1 = \{j \in P : d_j = 1\}$ and $P^2 = \{j \in P : d_j = 2\}$.

Thus $U_1 = W_1 \cup \left(\bigcup_{j\in P^2}\{j_1, j_2\}\right) \cup \left(\bigcup_{j\in P^1}\{j_1\}\right)$ and for each element $a \in U_1$ the set I_a is well-defined and non-empty. Similarly, $U_2 = W_2 \cup \left(\bigcup_{j\in P^2}\{j_3, j_4\}\right) \cup \left(\bigcup_{j\in P^1}\{j_3\}\right)$ and for each element $b \in U_2$ the set I_b is well-defined and non-empty. Now we construct the edges $\mathcal{E}$ as follows: For all $a \in U_1$ and $b \in U_2$, there is an edge $(a, b) \in \mathcal{E}$ if and only if a and b have at least one common supplier, i.e.,

$$I_a \cap I_b \neq \emptyset \text{ iff } (a, b) \in \mathcal{E}. \tag{9}$$

Let $G'(V', E')$ be a subgraph of $G^*(U_1 \cup U_2, \mathcal{E})$. Since the elements in $V' \cap (W_1 \cup W_2)$ correspond to unique elements in V_2, whenever required we will (with slight abuse of notation) treat $V' \cap (W_1 \cup W_2) \subseteq V_2$.

Claim 2. *Let $G'(V', E')$ be a subgraph of $G^*(U_1 \cup U_2, \mathcal{E})$ satisfying the following properties:*

1. *There are no edges in G^* between the nodes in V' and the nodes in $(U_1 \cup U_2) \setminus V'$.*
2. *For each $j \in P^1$, $|V' \cap \{j_1\}| = |V' \cap \{j_3\}|$ and for each $j \in P^2$, $|V' \cap \{j_1, j_2\}| = |V' \cap \{j_3, j_4\}|$.*
3. *$|W_1 \cap V'| = |W_2 \cap V'| + 1$.*

Now construct

$$z_j^3 = \begin{cases} z_j^1 & \text{if } j \in V_2 \setminus (V' \cap F) \\ 1 & \text{if } j \in V' \cap W_1 \\ 0 & \text{if } j \in V' \cap W_2. \end{cases} \tag{10}$$

$$x_{ij}^3 = \begin{cases} 1 & \text{if } j \in F, z_j^3 = 0, i \in I_j \\ 1 & \text{if } j \in P, j_1 \in (U_1 \cup U_2) \setminus V', i \in I_{j_1} \\ 1 & \text{if } j \in P, j_2 \in (U_1 \cup U_2) \setminus V', i \in I_{j_2} \\ 1 & \text{if } j \in P, j_3 \in V', i \in I_{j_3} \\ 1 & \text{if } j \in P, j_4 \in V', i \in I_{j_4} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

and

$$z_j^4 = \begin{cases} z_j^2 & \text{if } j \in V_2 \setminus (V' \cap F) \\ 0 & \text{if } j \in V' \cap W_1 \\ 1 & \text{if } j \in V' \cap W_2. \end{cases} \tag{12}$$

$$x_{ij}^4 = \begin{cases} 1 & \text{if } j \in F, z_j^4 = 0, i \in I_j \\ 1 & \text{if } j \in P, j_3 \in (U_1 \cup U_2) \setminus V', i \in I_{j_3} \\ 1 & \text{if } j \in P, j_4 \in (U_1 \cup U_2) \setminus V', i \in I_{j_4} \\ 1 & \text{if } j \in P, j_1 \in V', i \in I_{j_1} \\ 1 & \text{if } j \in P, j_2 \in V', i \in I_{j_2} \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Then (x^3, z^3) and (x^4, z^4) are feasible solutions of X that satisfy the requirements of Claim 1.

Proof. 1. We verify that (x^3, z^3) is a valid solution to X. A similar proof can be given for the validity of (x^4, z^4). Clearly x^3 and z^3 satisfy the variable restrictions. We verify that the constraint $\sum_{i:(i,j)\in E} x_{ij}^3 + d_j z_j = d_j$ is satisfied for all $j \in V_2$. If $j \in R$, then $z_j^3 = z_j^1 = 1$ and $x_{ij}^3 = 0$ for all $(i, j) \in E$; therefore the constraint is satisfied. If $j \in F$, then using the first and last entry in (11), we have $\sum_{i:(i,j)\in E} x_{ij}^3 + d_j z_j^3 = d_j$. If $j \in P$, then $j \in V_2 \setminus (V' \cap F)$. Therefore $z_j^3 = z_j^1 = 0$. Now it is straightforward to verify that $\sum_{i:(i,j)\in E} x_{ij}^3 = 2 = d_j$ for each $j \in P^2$ since $|V' \cap \{j_1, j_2\}| = |V' \cap \{j_3, j_4\}|$ and by the use of the last five entries in (11). For $j \in P^1$ we have $\sum_{i:(i,j)\in E} x_{ij}^3 = 1 = d_j$ since $|V' \cap \{j_1\}| = |V' \cap \{j_3\}|$ and by the use of the second, fourth and sixth entries in (11).

Now we verify that the constraint $\sum_{j:(i,j)\in E} x_{ij} \leq 1$ is satisfied for all $i \in V_1$. Given $i \in V_1$, assume for contradiction that $x^3_{ig} = x^3_{ih} = 1$ for some $g, h \in V_2$ and $g \neq h$. By construction of (x^3, z^3), $x^3_{ij} = 0$ for all $j \in R$. Thus, $g, h \notin R$. Moreover since $\sum_{i:(i,j)\in E} x^3_{ij} + d_j z_j = d_j$ is satisfied for all $j \in V_2$, we have $z^3_g = z^3_h = 0$. Now, there are three cases to consider:

(a) $g, h \in F$. By construction of x^3 we have $i \in I_g \cap I_h$. Now if $g \notin V'$ and $h \notin V'$, then by construction of z^3 (first entry in (10)) we have $z^1_g = z^3_g = 0 = z^3_h = z^1_h$ and thus $g, h \in S(z^1)$. Therefore by the validity of (x^1, z^1) we have $I_g \cap I_h = \emptyset$. This contradicts $i \in I_g \cap I_h$. Now consider the case where $g \in V'$ and $h \in V'$. Since $i \in I_g \cap I_h$ by (9) there is an edge between g and h in $G^*(U_1 \cup U_2, \mathcal{E})$. Thus we may assume without loss of generality that $g \in V' \cap W_1$ and $h \in V' \cap W_2$. However, this implies that $z^3_g = 1$, a contradiction. Now, without loss of generality, assume that $g \in V'$ and $h \notin V'$. Since $i \in I_g \cap I_h$ by (9) there is an edge between g and h in $G^*(U_1 \cup U_2, \mathcal{E})$. On the other hand, by assumption there is no edge between nodes in V' and those not in V', which is the required contradiction.

(b) $g \in F$ and $h \in P$. Without loss of generality we may assume that $g \in W_1$. If $g \in V'$, then $z^3_g = 1$, a contradiction. Therefore, we have $g \notin V'$. Thus $z^1_g = z^3_g = 0$. Therefore by validity of (x^1, z^1) we have $i \notin I_h(x^1, z^1)$ or equivalently $i \in I_h(x^2, z^2)$. Without loss of generality we may assume that $i \in I_{h_3}$. Note that h_3 belongs to V' (by the construction of x^3 and the fact that $x^3_{ih} = 1$ and $i \in I_{h_3}$). Since $i \in I_g$, there exists an edge between g and h_3. However, since $g \notin V'$ and $h_3 \in V'$, we get a contradiction to the fact that there are no edges between the nodes in V' and the nodes in $(U_1 \cup U_2) \setminus V'$.

(c) $g, h \in P$. In this case we may assume without loss of generality that $i \in I_g(x^1, z^1)$ and $i \in I_h(x^2, z^2)$. Therefore without loss of generality, we may assume that $i \in I_{g_1}$ and $i \in I_{h_3}$. Since $x^3_{ig} = x^3_{ih} = 1$, we have $g_1 \notin V'$ and $h_3 \in V'$. By assumption on G', this implies that there is no edge between g_1 and h_3. On the other hand, since $i \in I_{g_1} \cap I_{h_3}$ by (9) we have an edge $(g_1, h_3) \in \mathcal{E}$, a contradiction.

2. Next we verify that the objective function value of (x^3, z^3) is $\rho - \delta$ and that of (x^4, z^4) is $\rho + \delta$ where ρ is the objective function value of the solution (x^1, z^1) and $\delta \in \mathbb{R}$. This result is verified by showing that (x^3, z^3) and (x^4, z^4) are obtained by 'symmetrically' updating demands from (x^1, z^1) and (x^2, z^2) respectively. In particular, we examine each demand node and examine the cost of either satisfying it or not satisfying it in each solution. We consider the different cases next:

(a) $j \in R$. Then $z^4_j = z^3_j = z^1_j = z^2_j = 1$.

(b) $j \in V' \cap W_1$. Then $z^1_j = 0$ and $z^3_j = 1$. On the other hand $z^2_j = 1$ and $z^4_j = 0$. Notice that in each solution where d_j is satisfied, this is done by using the same set of input nodes (and thus using the same arcs). Therefore the difference in objective function value between (x^1, z^1) and (x^3, z^3) due to demand node j is $-\sum_{i\in I_j} w_{ij} + r_j$ and the difference in objective function value between the solutions (x^2, z^2) and (x^4, z^4) due to demand node j is $\sum_{i\in I_j} w_{ij} - r_j$.

(c) $j \in V' \cap W_2$. Similar to the above case the difference in objective function value between (x^1, z^1) and (x^3, z^3) due to demand node j is $\sum_{i\in I_j} w_{ij} - r_j$ and the difference in objective function value between (x^2, z^2) and (x^4, z^4) due to demand node j is $-\sum_{i\in I_j} w_{ij} + r_j$.

(d) $j \in F \setminus V'$, then $z^1_j = z^3_j$ and $z^2_j = z^4_j$.

(e) $j \in P^2$ such that $j_1, j_2 \in (U_1 \cup U_2) \setminus V'$ and $j_3, j_4 \in (U_1 \cup U_2) \setminus V'$. Then the demand d_j is satisfied by the nodes in $I_j(x^1, z^1)$ in (x^1, z^1) and (x^3, z^3). Therefore there is no difference in objective function value between (x^1, z^1) and (x^3, z^3) with respect to demand node j. Similarly, the demand d_j is satisfied by the nodes in $I_j(x^2, z^2)$ in (x^2, z^2) and (x^4, z^4) and there is no difference in objective function value between (x^2, z^2) and (x^4, z^4) with respect to demand node j. We can make a similar argument for $j \in P^1$ such that $j_1 \in (U_1 \cup U_2) \setminus V'$ and $j_3 \in (U_1 \cup U_2) \setminus V'$.

(f) $j \in P^2$ such that $j_1 \in V'$, $j_2 \in (U_1 \cup U_2) \setminus V'$, $j_3 \in (U_1 \cup U_2) \setminus V'$, $j_4 \in V'$ without loss of generality. Then the demand d_j is satisfied by the nodes in $(I_{j_1} \cup I_{j_2})$ in (x^1, z^1) and by nodes $(I_{j_2} \cup I_{j_4})$ in (x^3, z^3). Therefore the difference in objective function value between (x^1, z^1) and (x^3, z^3) with respect to demand node d_j is $\sum_{i \in I_{j_1}} w_{ij} - \sum_{i \in I_{j_4}} w_{ij}$. The demand d_j is satisfied by the nodes in $(I_{j_3} \cup I_{j_4})$ in (x^2, z^2) and by the nodes in $(I_{j_1} \cup I_{j_3})$ in (x^4, z^4). Therefore the difference in objective function value between (x^2, z^2) and (x^4, z^4) with respect to demand node j is $\sum_{i \in I_{j_4}} w_{ij} - \sum_{i \in I_{j_1}} w_{ij}$. We can make a similar argument for the cases: $j_1 \in (U_1 \cup U_2) \setminus V'$, $j_2 \in V'$, $j_3 \in V'$, $j_4 \in (U_1 \cup U_2) \setminus V'$; $j_1 \in V'$, $j_2 \in (U_1 \cup U_2) \setminus V'$, $j_3 \in V'$, $j_4 \in (U_1 \cup U_2) \setminus V'$ and $j_1 \in (U_1 \cup U_2) \setminus V'$, $j_2 \in V'$, $j_3 \in (U_1 \cup U_2) \setminus V'$, $j_4 \in V'$.

(g) $j \in P^2$ such that $j_1 \in V'$, $j_2 \in V'$, $j_3 \in V'$, $j_4 \in V'$. Then the demand d_j is satisfied by the nodes in $(I_{j_1} \cup I_{j_2})$ in (x^1, z^1) and by the nodes in $(I_{j_3} \cup I_{j_4})$ in (x^3, z^3). Therefore, the difference in the objective function value between (x^1, z^1) and (x^3, z^3) with respect to satisfying demand d_j is $\sum_{i \in (I_{j_1} \cup I_{j_2})}(w_{ij} + w_{ij}) - \sum_{i \in (I_{j_3} \cup I_{j_4})}(w_{ij} + w_{ij})$. The demand d_j is satisfied by the nodes in $(I_{j_3} \cup I_{j_4})$ in (x^2, z^2) and by the nodes in $(I_{j_1} \cup I_{j_2})$ in (x^4, z^4). Therefore, the difference in the objective function value between (x^2, z^2) and (x^4, z^4) with regards to satisfying demand d_j is $-\sum_{i \in (I_{j_1} \cup I_{j_2})}(w_{ij} + w_{ij}) + \sum_{i \in (I_{j_3} \cup I_{j_4})}(w_{ij} + w_{ij})$. For $j \in P^1$, we can similarly consider j_1 and j_3 with $j_1 \in V'$, $j_3 \in V'$.

Therefore, the objective function value of (x^3, z^3) is $\rho - \delta$ and that of (x^4, z^4) is $\rho + \delta$ where ρ is the objective function value of the solution (x^1, z^1) and (x^2, z^2) and $\delta \in \mathbb{R}$.

3. Finally we verify that $\sum_{j \in V_2} z_j^3 = k^1 + 1$ and $\sum_{j \in V_2} z_j^4 = k^2 - 1$. We prove this for (x^3, z^3). The proof is similar for the case of (x^4, z^4). Observe that if $j \in R$, then $z_j^1 = z_j^3 = 1$. If $j \in P$, then $z_j^1 = z_j^3 = 0$. If $j \in F \setminus V'$, then $z_j^1 = z_j^3$. If $j \in W_1 \cap V'$, then $z_j^1 = 0$ and $z_j^3 = 1$ and if $j \in W_2 \cap V'$, then $z_j^1 = 1$ and $z_j^3 = 0$. Thus $\sum_{j \in V_2} z_j^1 - \sum_{j \in V_2} z_j^3 = |V' \cap W_2| - |V' \cap W_1| = -1$, where the last equality is by assumption (3) of G'. Thus, $\sum_{j \in V_2} z_j^3 = k^1 + 1$.

□

Now the proof of Theorem 1 is complete by showing that a subgraph $G'(V', E')$ of $G^*(U_1 \cup U_2, \mathcal{E})$ always exists that satisfies the conditions of Claim 2. In order to prove this, we verify a few results.

Claim 3. *Connected components of G^* are paths or cycles of even length and all the cycles involve only full nodes.*

Proof. This is evident from the fact that G^* is bipartite and degree of $a \in (U_1 \cup U_2)$ is bounded from above by $|I_a|$. □

We associate a value v_j to each node $j \in U_1 \cup U_2$. In particular:

1. If $j \in W_1$, then $v_j = 1$.
2. If $j \in U_1$ and j is a partial node, then $v_j = \frac{1}{2}$.
3. If $j \in U_2$ and j is a partial node, then $v_j = -\frac{1}{2}$.
4. If $j \in W_2$, then $v_j = -1$.

For a subgraph $\tilde{G}(\tilde{V}, \tilde{E})$ of G^* we call $v(\tilde{V}) = \sum_{j \in \tilde{V}} v_j$ the *value of the path.*

Claim 4. $v(U_1 \cup U_2) = k^2 - k^1 \geq 2.$

Proof. $\sum_{j \in U_1 \cup U_2} v_j = \sum_{j \in W_1} v_j + \sum_{j \in P^2}(v_{j_1} + v_{j_2}) + \sum_{j \in P^1} v_{j_1} + \sum_{j \in W_2} v_j + \sum_{j \in P^2}(v_{j_3} + v_{j_4}) + \sum_{j \in P^1} v_{j_3} = |S(z^1) \setminus S(z^2)| - |S(z^2) \setminus S(z^1)| = |S(z^1)| - |S(z^2)| = k^2 - k^1$. □

Claim 5. *If $\tilde{G}(\tilde{V}, \tilde{E})$ is a cyclic subgraph of $G^*(U_1 \cup U_2, \mathcal{E})$, then $v(\tilde{V}) = 0$.*

Proof. By Claim 3, a cycle has only full nodes. Moreover, since a cycle is of even length, it contains equal number of nodes from W_1 and W_2. □

Note that a partial node must be a leaf node in a path. Using this observation and by some simple case analysis the following three claims can be verified.

Claim 6. *If $\tilde{G}(\tilde{V}, \tilde{E})$ is a path containing exactly one partial node, then $v(\tilde{V}) \in \{-\frac{1}{2}, \frac{1}{2}\}$.*

Claim 7. *If $\tilde{G}(\tilde{V}, \tilde{E})$ is a path containing two partial nodes, then $v(\tilde{V}) = 0$.*

Claim 8. *If $\tilde{G}(\tilde{V}, \tilde{E})$ is a path containing only full nodes, then $v(\tilde{V}) \in \{-1, 0, 1\}$.*

For the subgraph $\tilde{G}(\tilde{V}, \tilde{E})$, consider a $k \in \tilde{V} \setminus F$ such that $k = j_t$ where $t \in \{1, 2, 3, 4\}$ and $j \in P^2$. Suppose $k = j_1$ or j_2, then we say that a path $\tilde{G}(\tilde{V}, \tilde{E})$ is a *mirror path* for j, if $\tilde{V}$ contains either j_3 or j_4. Moreover we call one of j_3 or j_4 (whichever belongs to $\tilde{V}$ or arbitrarily select one of these if both belong to $\tilde{V}$) as the *mirror node.* Similarly if $k = j_3$ or j_4, then we say that a path $\tilde{G}(\tilde{V}, \tilde{E})$ is a *mirror path* for j, if $\tilde{V}$ contains either j_1 or j_2. *Mirror node* is similarly defined in this case. For $j \in P^1$ we consider $k = j_1$ and $k = j_3$. Suppose $k = j_1$, then we say that a path $\tilde{G}(\tilde{V}, \tilde{E})$ is a *mirror path* for j, if $\tilde{V}$ contains j_3 and we call j_3 the *mirror node.* Similarly if $k = j_3$, then we say that a path $\tilde{G}(\tilde{V}, \tilde{E})$ is a *mirror path* for j, if $\tilde{V}$ contains j_1 and we call j_1 the *mirror node.*

Algorithm 1 constructs $G'(V', E')$ that satisfies all the properties of Claim 2. We next verify that Algorithm 1 is well-defined, that is all the steps can be carried out. Moreover we show that the algorithm generates a subgraph $G'(V', E')$ that satisfies the conditions of Claim 2.

Algorithm 1 Construction of $G'(V', E')$

```
Input: $G^*(U_1 \cup U_2, \mathcal{E})$.
Output: $G'(V', E')$ that satisfies all conditions of Claim 2.

1. If there exists a path $\tilde{G}(\tilde{V}, \tilde{E})$ in $G^*(U_1 \cup U_2, \mathcal{E})$ containing only full nodes with $v(\tilde{V}) = 1$, then set
   $G' := \tilde{G}$. STOP.

2. Tag all paths in $G^*(U_1 \cup U_2, \mathcal{E})$ as 'unmarked.'

3. Select a path $\tilde{G}(\tilde{V}, \tilde{E})$ from the set of 'unmarked' paths containing a partial node such that $v(\tilde{V}) = \frac{1}{2}$.
   Tag this path as 'marked.' Note that by Claim 6 and Claim 7, $\tilde{V}$ contains a unique partial node $j^*$.

4. Select a path from the list of 'unmarked' paths, such that it is a mirror path for $j^*$. Tag this path as
   'marked.'

5. There are three cases:

    (a) The mirror path tagged as 'marked' in (4) contains a unique partial node and its value is $\frac{1}{2}$.
        GO TO Step 6
    (b) The mirror path tagged as 'marked' in (4) contains a unique partial node and its value is $-\frac{1}{2}$.
        GO TO Step 3.
    (c) The mirror path tagged as 'marked' in (4) contains two partial nodes (then its value is 0):
        One of the partial nodes corresponds to the mirror node. Set $j^*$ to be the other partial node. GO
        TO Step 4.

6. Set $G'(V', E')$ to be disjoint union of the paths tagged as 'marked.' STOP.
```

Claim 9. *Algorithm 1 is well-defined.*

1. *At the beginning of Step (3), the total value of all marked paths is* 0.
2. *Let* $\hat{V} := \bigcup_{\tilde{G}(\tilde{V},\tilde{E}) \text{ is marked before Step (3)}} \tilde{V}$*. Then* $|\hat{V} \cap \{j_1, j_2\}| = |\hat{V} \cap \{j_3, j_4\}|$ *for all* $j \in P^2$ *and* $|\hat{V} \cap \{j_1\}| = |\hat{V} \cap \{j_3\}|$ *for all* $j \in P^1$*.*
3. *Step (3) is well-defined, that is as long as the algorithm does not terminate, Step (3) can be carried out.*
4. *At the end of Step (3), the total value of all marked paths is* $\frac{1}{2}$*.*
5. *Step (4) is well-defined, that is as long as the algorithm does not terminate, Step (4) can be carried out.*

Proof. We prove Claim 9 by induction on the iteration number (n) of the algorithm visiting Step (5). When $n = 0$:

1. At the beginning of Step (3) there are no 'marked' paths and therefore the total value of all marked paths is 0.
2. $\hat{V} = \emptyset$.
3. By Step (1), we know that there exists no path containing only full nodes with $v(\tilde{V}) = 1$. Moreover by Claim 4 we have $v(U_1 \cup U_2) \geq 2$. Since by Claim 5 all cycles have a value of 0, there must exist at least one path with partial nodes with positive value. Since this is only possible (Claim 6 and Claim 7) if there exists exactly one partial node in the path, we see that Step (3) is well-defined.
4. At Step (3) one path is marked which has a value of half.
5. Since one path is tagged as marked in Step (3), it contains exactly one partial node, $j^* \in P$. Suppose that $j^* \in P^2$ and $j^* = j_i^*$ for some $i \in \{1, \ldots, 4\}$. Then there exists paths (at least two) which contain the other three partial nodes corresponding to j^*. If $j^* \in P^1$ then there exists one path which contains the other partial node. Therefore this step is well-defined.

Now for any $n \in \mathbb{Z}_+$, assuming by the induction hypothesis that the result is true for $n' = 0, \ldots, n-1$:

1. Step (3) is arrived at via Step (5b). Let $n' < n$ be the last iteration when Step (3) is invoked. By the induction hypothesis the total value of all the marked paths at the end of Step (3) in iteration n' is $\frac{1}{2}$. From iterations $n'+1, \ldots, n-1$, the algorithm alternates between Step (4) and Step (5c). The total value of all the marked paths here is 0. Finally, the value of the last path tagged as marked in Step (4) is $-\frac{1}{2}$ (since the algorithm invokes Step (5b)). Hence, the total value of all the marked paths is 0 at the beginning of Step (3) in iteration n.
2. Let $n' < n$ be the last iteration when Step (3) is invoked. By the induction hypothesis $|\hat{V} \cap \{j_1, j_2\}| = |\hat{V} \cap \{j_3, j_4\}|$ for all $j \in P^2$ and $|\hat{V} \cap \{j_1\}| = |\hat{V} \cap \{j_3\}|$ for all $j \in P^1$ where $\hat{V} := \bigcup_{\tilde{G}(\tilde{V},\tilde{E}) \text{ is marked before Step (3) iteration n'}} \tilde{V}$. From iterations $n'+1, \ldots, n-1$, the algorithm alternates between Step (4) and Step (5c). Since in iteration $n-1$ at Step (4), we add one path that contains only the mirror node to j^* (the unique partial node from the previous iteration), we arrive at this result.
3. Proof same as that in the case where $n = 0$.
4. The total value of paths at the end of Step (3) = value of marked path + total value of previously marked path $= \frac{1}{2} + 0$.

5. Step (4) is invoked after either Step (3) or Step (5c). In case we arrive via Step (3), by the induction hypothesis $|\hat{V} \cap \{j_1, j_2\}| = |\hat{V} \cap \{j_3, j_4\}|$ for all $j \in P^2$ and $|\hat{V} \cap \{j_1\}| = |\hat{V} \cap \{j_3\}|$ for all $j \in P^1$ where $\hat{V} := \bigcup_{\tilde{G}(\tilde{V},\tilde{E}) \text{ is marked before Step (3) iteration n'}} \tilde{V}$. Moreover the path marked in step (3) contains exactly a unique partial node j^* then, there must exist an unmarked path containing a mirror node to j^*. In case of we arrive via Step (5c), again the proof is essentially the same by observing that at the start of Step (4), there is a unique partial node j^* that is not paired with a mirror partial node.

□

Claim 10. *Algorithm 1 terminates in finite time.*

Proof. This is true since there are a finite number of edges and at each iteration of the algorithm at least one unmarked path is tagged as marked. □

Claim 11. *Algorithm 1 generates a subgraph $G'(V', E')$ that satisfies the properties of Claim 2.*

Proof. First observe that since the output $G'(V', E')$ of the algorithm is a disjoint union of paths, there exists no edge between V' and $(U_1 \cup U_2) \setminus V'$ in $\mathcal{E}$, so property 1 is satisfied.

By Claim 9, 2. we have $|\hat{V} \cap \{j_1, j_2\}| = |\hat{V} \cap \{j_3, j_4\}|$ for all $j \in P^2$ and $|\hat{V} \cap \{j_1\}| = |\hat{V} \cap \{j_3\}|$ for all $j \in P^1$ where

$$\hat{V} := \bigcup_{\tilde{G}(\tilde{V},\tilde{E}) \text{ is marked before Step (3)}} \tilde{V}.$$

Therefore, it is easily verified that in the last iteration before termination, a path with a unique partial node, which is a mirror node to j^*, is marked in Step (4). This is because before termination we arrive at Step (5a) implying that the value of the path marked in Step (4) is $\frac{1}{2}$. Hence Claim 6 and Claim 7 imply that there is a unique partial node in this path. Thus, $|V' \cap \{j_1, j_2\}| = |V' \cap \{j_3, j_4\}|$ for all $j \in P^2$ and $|V' \cap \{j_1\}| = |V' \cap \{j_3\}|$ for all $j \in P^1$, so property 2 is satisfied.

Finally, since $v(V') = 1$ and $|V' \cap \{j_1, j_2\}| = |V' \cap \{j_3, j_4\}|$ for all $j \in P^2$ and $|V' \cap \{j_1\}| = |V' \cap \{j_3\}|$ for all $j \in P^1$ we have

$$\sum_{j \in V' \cap W_1} v_j + \sum_{j \in V' \cap W_2} v_j = 1.$$

As a result, $|V' \cap W_1| = |V' \cap W_2| + 1$, so property 3 is satisfied. □

We showed that the set of solutions to TPMC satisfies the edge property. Theorem 1 then follows from Proposition 4.

Finally we ask the natural question: Does the edge property hold for TPMC when there exist demands that are greater than 2? The next example illustrates that the edge property can fail to hold even if $d_j > 2$ for only one $j \in V_2$.

Example 2. *Consider an instance of TPMC where $G(V_1 \cup V_2, E)$ is a bipartite graph with $V_1 = \{1, 2, \ldots, 6\}$, $V_2 = \{1, 2, 3, 4\}$, $E = \{(1,1), (2,2), (3,3), (4,1), (4,4), (5,2), (5,4), (6,3), (6,4)\}$, $s_i = 1$, $i \in V_1$, $d_j = 2$, $j = \{1, 2, 3\}$, $d_4 = 3$. For $k = 2$ we obtain a non-integer extreme point of $\operatorname{conv}(T) \cap \{(x, z) \in \mathbb{R}^p_+ \times [0, 1]^n \,|\, \sum_{j=1}^n z_j \leq k\}$, given by $x_{11} = x_{22} = x_{33} = x_{41} = x_{44} = x_{52} = x_{54} = x_{63} = x_{64} = z_1 = z_2 = z_3 = z_4 = \frac{1}{2}$. Therefore, $T^k \neq \operatorname{conv}(T) \cap \{(x, z) \in \mathbb{R}^p_+ \times [0, 1]^n \,|\, \sum_{j=1}^n z_j \leq k\}$ in this example. Next we show how the conflict graph construction fails for this example. In fact, it can be shown that the edge property is not satisfied in this example by using an alternative characterization defined in [2]. Let $w_{11} = w_{22} = w_{33} = w_{41} = w_{44} = w_{52} = w_{54} = w_{63} = w_{64} = 1$ and $r_j = 3$, $j = \{1, 2, 3\}$ and $r_4 = 6$. For $k = 0$ the problem is infeasible. For $k = 1$, an optimal solution is $x_{11} = x_{22} = x_{33} = x_{41} = x_{52} = x_{63} = z_4 = 1$ and all other variables are zero, with an objective function value 12. For $k = 3$, an optimal solution is $x_{44} = x_{54} = x_{64} = z_1 = z_2 = z_3 = 1$ and all other variables are zero, with an objective function value 12. We show that Algorithm 1 fails to find a subgraph $G'(V', E')$ of $G^*(U_1 \cup U_2, \mathcal{E})$ that satisfies the properties given in Claim 2 for this example. We use two feasible solutions, namely solution for $k = 1$ and $k = 3$ to build the bipartite graph given in Figure 1.*

Note that $I_1 = \{1,4\}$, $I_2 = \{2,5\}$, $I_3 = \{3,6\}$ and $I_4 = \{4,5,6\}$. In Step (1) of Algorithm 1 we find a path with $v(\tilde{V}) = 1$ which is $1-4-2$ then the algorithm stops. We have $V' = \{1,4,2\}$ and $(U_1 \cup U_2) \setminus V' = \{3\}$. However, property 1 does not hold since there exists an edge between 3 and 4 but $3 \in (U_1 \cup U_2) \setminus V'$ and $4 \in V'$. Hence, Algorithm 1 fails.

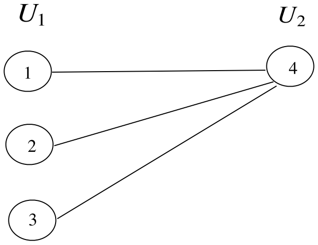


Figure 1: Bipartite Graph $G^*(U_1 \cup U_2, \mathcal{E})$ for Example 2

4 Valid Inequalities

In this section we give valid inequalities for TPMC and study their strength. First, observe that the variable upper bound inequalities (VUB) for $(i,j) \in E$

$$x_{ij} \leq \min\{s_i, d_j\}(1 - z_j) \tag{14}$$

are valid for X.

Proposition 5. *Let $I \subseteq V_1$, $J \subseteq V_2$ such that $d(J) \geq s(V_1 \setminus I)$. The inequality*

$$\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} + \sum_{j \in J} \left(\min\left\{d(J) - s(V_1 \setminus I), d_j\right\}\right) z_j \geq d(J) - s(V_1 \setminus I) \tag{15}$$

is valid for X.

Proof. Given a feasible solution (x,z) we consider two cases.

1. If $z_{j'} = 1$ for some $j' \in J$ such that $\min\{d(J) - s(V_1 \setminus I), d_{j'}\} = d(J) - s(V_1 \setminus I)$, then the feasible solution satisfies inequality (15) because we have

$$\begin{aligned} & \sum_{i \in I, j \in J : (i,j) \in E} x_{ij} + \sum_{j \in J} \left(\min\left\{d(J) - s(V_1 \setminus I), d_j\right\}\right) z_j \\ = & \sum_{i \in I, j \in J \setminus \{j'\} : (i,j) \in E} x_{ij} + \sum_{j \in J \setminus \{j'\}} \left(\min\left\{d(J) - s(V_1 \setminus I), d_j\right\}\right) z_j + d(J) - s(V_1 \setminus I) \\ \geq & \; d(J) - s(V_1 \setminus I) \end{aligned}$$

   where the last inequality holds because $\min\{d(J) - s(V_1 \setminus I), d_j\} \geq 0$ for all $j \in J$, and all x and z variables are non-negative.

2. If $z_j = 0$ for all $j \in J$ satisfying $\min\{d(J) - s(V_1 \setminus I), d_j\} = d(J) - s(V_1 \setminus I)$, then $\sum_{j \in J} \left(\min\left\{d(J) - s(V_1 \setminus I), d_j\right\}\right) z_j = \sum_{j \in J} d_j z_j$. Moreover, observe that $\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} + s(V_1 \setminus$

I) is at least as large as the total flow sent to the demand nodes in J in the solution (x,z), i.e., $\sum_{i\in I, j\in J:(i,j)\in E} x_{ij} + s(V_1 \setminus I) \geq \sum_{j\in J} d_j(1-z_j)$. Therefore we have

$$\sum_{i\in I, j\in J:(i,j)\in E} x_{ij} + \sum_{j\in J} \left(\min\left\{d(J) - s(V_1\setminus I), d_j\right\}\right) z_j + s(V_1\setminus I) \geq \sum_{j\in J} d_j z_j + \sum_{j\in J} d_j(1-z_j) = d(J),$$

so inequality (15) is valid.

□

Next, we give valid inequalities for general mixed-integer sets that are substructures of TPMC.

4.1 A Coefficient Update Scheme for Mixed-Integer Covers

Consider the mixed integer cover set $\mathcal{S}_1$ defined by

$$t + \sum_{j\in J} \beta_j z_j \geq \beta_0 \tag{16}$$

$$t \geq 0 \tag{17}$$

$$z_j \in \{0,1\} \quad \forall j \in J, \tag{18}$$

for given $\beta_j \geq 0$ for all $j \in J$ and $\beta_0 \geq 0$. We assume that $\beta_j \leq \beta_0$ for all $j \in J$ without loss of generality. Let $\mathcal{T}_1 = conv(\mathcal{S}_1)$. We refer to inequalities in the form of (16) as type-I base inequalities. Note that inequalities (15) for TPMC are in the form of (16) since we can replace $\sum_{i\in I, j\in J:(i,j)\in E} x_{ij}$ by t and $t \geq 0$. Therefore, (16)-(18) is a relaxation of TPMC.

Proposition 6. *Given a type-I base inequality* (16) *valid for a mixed-integer program (MIP) with* (17)-(18)*, let* $\tilde{J} := \{j_1, j_2, \ldots, j_p\} \subseteq J$ *be a minimal cover, i.e.,* $\sum_{j\in\tilde{J}} \beta_j > \beta_0$ *and* $\sum_{j\in\tilde{J}\setminus\{j_k\}} \beta_j \leq \beta_0$ *for all* $k \in \{1,\ldots,p\}$*. Let* $\beta_{j_p} \geq \beta_{j_k}$ *for all* $k \in \{1,\ldots,p\}$*. Let* $J^* := \tilde{J} \cup \{j \in J : \beta_j \geq \beta_{j_p}\}$, $\beta = \sum_{j\in\tilde{J}} \beta_j - \beta_0$ *and* $\beta_0^{'} := \beta_0 - (p-1)\beta$*. Then,*

$$t + \sum_{j\in J^*} \min\left\{(\beta_j - \beta), \beta_0^{'}\right\} z_j + \sum_{j\in J\setminus J^*} \min\left\{\beta_0^{'}, \beta_j\right\} z_j \geq \beta_0^{'} \tag{19}$$

is a valid inequality for $\mathcal{S}_1$.

Proof. We first claim that $\beta_j \geq \beta$ for all $j \in J^*$. Suppose, without loss of generality, that $\beta_{j_1} \leq \beta_{j_2} \leq \cdots \leq \beta_{j_p}$, and recall that $\beta_j \geq \beta_{j_p}$ for all $j \in J^* \setminus \tilde{J}$. Assume by contradiction that $\beta_{j_1} < \beta$ or equivalently $\beta_{j_1} - (\sum_{k=1}^{p} \beta_{j_k} - \beta_0) < 0$. This is a contradiction to the minimality of the cover $\tilde{J}$.

Next we claim that $\beta_0^{'} \geq 0$: By the previous claim we have $\beta \leq \beta_{j_k}$ for $k = 1,\ldots,p$. Therefore, we obtain

$$\beta_0^{'} = \beta_0 - (p-1)\beta \geq \beta_0 - \sum_{k=1}^{p-1} \beta_{j_k} \geq 0,$$

where the last inequality follows from the fact that $\tilde{J}$ is a minimal cover.

Given a feasible solution (x,z), let $J_1 = \{j \in J : z_j = 1\}$ and $J_1^* = \{j \in J^* : z_j = 1\}$. Consider the following cases:

1. Suppose that there exists $j' \in J_1^*$ such that $\min\left\{\beta_0^{'}, \beta_{j'} - \beta\right\} = \beta_0^{'}$. Then,

$$\begin{aligned} & t + \sum_{j\in J^*} \min\left\{(\beta_j - \beta), \beta_0^{'}\right\} z_j + \sum_{j\in J\setminus J^*} \min\left\{\beta_0^{'}, \beta_j\right\} z_j \\ \geq \quad & t + \sum_{j\in J^*\setminus\{j'\}} \min\left\{(\beta_j - \beta), \beta_0^{'}\right\} z_j + \sum_{j\in J\setminus J^*} \min\left\{\beta_0^{'}, \beta_j\right\} z_j + \beta_0^{'} \geq \beta_0^{'}, \end{aligned}$$

where the last inequality follows from the fact that all variables are non-negative, $\beta_j \geq \beta$ for all $j \in J^*$ and $\beta_0^{'} \geq 0$. The proof for the case where there exists $j' \in J_1 \setminus J_1^*$ such that $\min\left\{\beta_0^{'}, \beta_{j'}\right\} = \beta_0^{'}$ follows similarly.

2. Suppose that for all $j \in J_1^*$, we have $\min\left\{\beta_0^{'}, \beta_j - \beta\right\} = \beta_j - \beta$ and for all $j \in (J_1 \setminus J_1^*)$ we have $\min\left\{\beta_0^{'}, \beta_j\right\} = \beta_j$. There are two cases to consider:

   (a) Suppose that $|J_1^*| \leq p-1$. In this case,

$$\begin{aligned} t + \sum_{j \in J^*}(\beta_j - \beta)z_j + \sum_{j \in J \setminus J^*} \beta_j z_j &= t + \sum_{j \in J_1^*}(\beta_j - \beta) + \sum_{j \in J_1 \setminus J_1^*} \beta_j \\ &= t + \sum_{j \in J_1^*} \beta_j + \sum_{j \in J_1 \setminus J_1^*} \beta_j - |J_1^*|\beta \\ &\geq \beta_0 - |J_1^*|\beta \geq \beta_0 - (p-1)\beta, \end{aligned}$$

   where the first inequality follows because inequality (16) is valid and the second inequality follows because of our assumption $|J_1^*| \leq p-1$.

   (b) Suppose that $|J_1^*| \geq p$. In this case,

$$\begin{aligned} t + \sum_{j \in J^*}(\beta_j - \beta)z_j + \sum_{j \in J \setminus J^*} \beta_j z_j &= t + \sum_{j \in J_1^*}(\beta_j - \beta) + \sum_{j \in J_1 \setminus J_1^*} \beta_j \\ &\geq \sum_{j \in J_1^*}(\beta_j - \beta) \geq \sum_{k=1}^{p}(\beta_{j_k} - \beta) \\ &= \sum_{k=1}^{p} \beta_{j_k} - p\beta = \beta_0 - (p-1)\beta. \end{aligned}$$

   The second inequality holds since $|J_1^*| \geq p$ and since $\beta \leq \beta_{j_1} \leq \beta_{j_2} \leq \cdots \leq \beta_{j_p} \leq \beta_j$ for $j \in J^* \setminus \tilde{J}$.

□

Given type-I base inequalities (16) valid for any MIP with $t \geq 0$, and $z_j \in \{0, 1\}$, $j \in J$, we can derive a new class of valid inequalities (19). Similarly, inequality (19) is in the form of (16), so this process can be repeated by letting the valid inequality (19) be the type-I base inequality to derive other classes of valid inequalities.

Inequality (19) is related to the weight inequalities of Weismantel [26] for the 0/1 knapsack polytope. Note that inequality (19) is valid when J^* is replaced with $\tilde{J}$. After complementing the z variables, we can show that inequality (19) where J^* is replaced with $\tilde{J}$ and the condition $\beta_j \leq \beta_0'$ for all $j \in J \setminus \tilde{J}$ is satisfied is equivalent to the weight inequalities for the 0/1 knapsack polytope (ignoring the continuous term t). However, if $J^* \supsetneq \tilde{J}$ then inequality (19) with J^* dominates inequality (19) with $\tilde{J}$. Additionally if $J^* = \tilde{J}$ and there exists $j \in J \setminus \tilde{J}$ such that $\beta_j > \beta_0'$ then inequality (19) dominates the corresponding weight inequality. Weismantel also proposes weight-reduction and extended weight inequalities for the 0/1 knapsack polytope. In Example 3 we show that weight-reduction inequalities and inequalities (19) are not equivalent. We also show that the extended weight inequality is dominated by the inequalities found using Proposition 6 for this example.

Example 3. *Consider the type-I base inequality*

$$3z_1 + 4z_2 + 5z_3 + 6z_4 \geq 6, \tag{20}$$

for $t = 0$. *Next, we give examples of inequality* (19) *for different choices of* $\tilde{J}$.

1. *Let* $\tilde{J} = \{1, 4\}$. *Then* $J^* = \tilde{J}$ *and* $\beta = (3 + 6) - 6 = 3$. *Then corresponding inequality* (19) *defined by this choice of* $\tilde{J}$ *is* $\min\{4, 3\}z_2 + \min\{5, 3\}z_3 + 3z_4 \geq 3$, *or*
$$z_2 + z_3 + z_4 \geq 1. \tag{21}$$
2. *Let* $\tilde{J} = \{2, 4\}$. *Then* $J^* = \tilde{J}$ *and* $\beta = (4 + 6) - 6 = 4$ *Then corresponding inequality* (19) *defined by this choice of* $\tilde{J}$ *is* $\min\{3, 2\}z_1 + \min\{5, 2\}z_3 + 2z_4 \geq 2$, *or*
$$z_1 + z_3 + z_4 \geq 1. \tag{22}$$
3. *Let* $\tilde{J} = \{3, 4\}$. *Then* $J^* = \tilde{J}$ *and* $\beta = (5 + 6) - 6 = 5$. *Then corresponding inequality* (19) *defined by this choice of* $\tilde{J}$ *is* $\min\{3, 1\}z_1 + \min\{4, 1\}z_2 + z_4 \geq 1$, *or*
$$z_1 + z_2 + z_4 \geq 1. \tag{23}$$

Inequalities (21)-(23) *dominate the corresponding weight inequalities since for all the inequalities there exists* $j \in J \setminus \tilde{J}$ *such that* $\beta_j > \beta'_0$. *Inequality* (23) *cannot be obtained by weight-reduction inequalities in [26]. On the other hand, the weight-reduction inequality*

$$3z_1 + z_3 + 2z_4 \geq 2,$$

cannot be obtained using Proposition 6. For this example, the only valid extended weight inequality is

$$z_1 + z_2 + 2z_3 + 2z_4 \geq 2,$$

which is dominated by the inequalities (21) *and* (22).

4.2 A Coefficient Update Scheme for Mixed-Integer Knapsacks with Variable Upper Bounds

Next, we consider another substructure of TPMC consisting of a mixed integer knapsack and variable upper bound constraints. We define set $\mathcal{S}_2$ as follows:

$$\sum_{j\in J} t_j + \sum_{j\in J} \alpha_j z_j \leq \alpha_0 \tag{24}$$
$$t_j \leq d_j(1 - z_j)\ \forall j \in J \tag{25}$$
$$z \in \{0, 1\}^{|J|}, t_j \in \mathbb{R}_+^{|J|}, \tag{26}$$

for given $\alpha_j \geq 0$ for all $j \in J$ and $\alpha_0 \geq 0$.

Let $\mathcal{T}_2 = conv(\mathcal{S}_2)$. We refer to inequalities in the form of (24) as type-II base inequalities. If we replace $t_j := \sum_{i\in I:(i,j)\in E} x_{ij}$, $I \subseteq V_1$ then the sum of relaxation of the supply constraints (1c) over I is in the form of (24) (with $\alpha_j = 0$ for all $j \in J$) for TPMC, and (25) is a relaxation of the demand constraints (1b). In this case, we observe that TPMC contains the fixed-charge network flow substructure. Therefore, the lifted flow cover and pack inequalities [4, 5, 13, 21, 24], and submodular inequalities [1, 27] are all valid for TPMC. Furthermore, these inequalities and the blossom inequalities (3) are in the form of (24). Next we describe valid inequalities for the set $\mathcal{S}_2$.

Proposition 7. *Given the mixed-integer set* $\mathcal{S}_2$, *let* $\tilde{J} = \{j_1, j_2, \ldots, j_u\} \subseteq J$ *such that* $d_{j_1} - \alpha_{j_1} \geq d_{j_2} - \alpha_{j_2} \geq \cdots \geq d_{j_u} - \alpha_{j_u}$ *and there exists* $m = \max\{l \in \{0, \ldots, u-1\} : \sum_{k=1}^{l} d_{j_k} + \sum_{k=l+1}^{u} \alpha_{j_k} < \alpha_0 - \sum_{j\in J\setminus\tilde{J}} \max\{d_j, \alpha_j\}\}$. *Let* $M = \{j_1, j_2, \ldots, j_m\}$ *(*$M = \emptyset$ *if* $m = 0$*) and* $\alpha = \alpha_0 - \sum_{j\in J\setminus\tilde{J}} \max\{d_j, \alpha_j\} - d(M) - \alpha(\tilde{J} \setminus M)$. *Then the inequality given by*

$$\sum_{j\in J} t_j + \sum_{j\in\tilde{J}} (\alpha_j + \alpha)z_j + \sum_{j\in J\setminus\tilde{J}} \alpha_j z_j \leq \alpha_0 + (u - m - 1)\alpha \tag{27}$$

is valid for $\mathcal{S}_2$.

Proof. Given a feasible solution (t, z) to $\mathcal{S}_2$, let $\tilde{J}_1 = \{j \in \tilde{J} : z_j = 1\}$ and $\tilde{J}_0 = \{j \in \tilde{J} : z_j = 0\}$. Consider the following cases:

1. Suppose that $u - m - 1 \geq |\tilde{J}_1|$. In this case,

$$\begin{aligned}\sum_{j \in J} t_j + \sum_{j \in \tilde{J}} (\alpha_j + \alpha) z_j + \sum_{j \in J \setminus \tilde{J}} \alpha_j z_j &= \sum_{j \in J \setminus \tilde{J}_1} t_j + \sum_{j \in \tilde{J}_1} \alpha_j + \sum_{j \in J \setminus \tilde{J}} \alpha_j z_j + |\tilde{J}_1| \alpha \\ &\leq \alpha_0 + |\tilde{J}_1| \alpha \\ &\leq \alpha_0 + (u - m - 1)\alpha.\end{aligned}$$

2. Suppose that $u - m \leq |\tilde{J}_1|$, or equivalently $m \geq u - |\tilde{J}_1| = |\tilde{J}_0|$. Then,

$$\begin{aligned}&\sum_{j \in J} t_j + \sum_{j \in \tilde{J}} (\alpha_j + \alpha) z_j + \sum_{j \in J \setminus \tilde{J}} \alpha_j z_j \\ =\;& \sum_{j \in J \setminus \tilde{J}_1} t_j + \sum_{j \in \tilde{J}_1} \alpha_j + \sum_{j \in J \setminus \tilde{J}} \alpha_j z_j + |\tilde{J}_1| \alpha \\ \leq\;& \sum_{j \in J \setminus \tilde{J}} \max\{d_j, \alpha_j\} + d(\tilde{J}_0) + \sum_{j \in \tilde{J}_1} \alpha_j + |\tilde{J}_1| \alpha \\ =\;& \alpha_0 - \alpha - d(M) - \alpha(\tilde{J} \setminus M) + d(\tilde{J}_0) + \alpha(\tilde{J}_1) + |\tilde{J}_1| \alpha \\ =\;& \alpha_0 - \Big[(d(M) - \alpha(M)) - (d(\tilde{J}_0) - \alpha(\tilde{J}_0)) \Big] + (|\tilde{J}_1| - 1)\alpha,\end{aligned}$$

where the first inequality holds since

$$\begin{aligned}&\sum_{j \in J \setminus \tilde{J}_1} t_j + \sum_{j \in J \setminus \tilde{J}} \alpha_j z_j \\ =\;& \left(\sum_{j \in J \setminus \tilde{J}} t_j + \sum_{j \in J \setminus \tilde{J}} \alpha_j z_j \right) + \sum_{j \in \tilde{J}_0} t_j \leq \sum_{j \in J \setminus \tilde{J}} \max\{d_j, \alpha_j\} + d(\tilde{J}_0),\end{aligned}$$

and the second equality holds because $\sum_{j \in J \setminus \tilde{J}} \max\{d_j, \alpha_j\} = \alpha_0 - \alpha - d(M) - \alpha(\tilde{J} \setminus M)$. Furthermore, due to the choice of index m, $0 < \alpha \leq d_{j_{m+1}} - \alpha_{j_{m+1}}$. Thus, we have

$$(m - |\tilde{J}_0|)\alpha \leq (m - |\tilde{J}_0|)(d_{j_{m+1}} - \alpha_{j_{m+1}}) \leq \sum_{k=|\tilde{J}_0|+1}^{m} (d_{j_k} - \alpha_{j_k}).$$

Moreover, $-\Big[(d(M) - \alpha(M)) - (d(\tilde{J}_0) - \alpha(\tilde{J}_0)) \Big] \leq -\Big[\sum_{k=|\tilde{J}_0|+1}^{m} (d_{j_k} - \alpha_{j_k}) \Big]$. Thus we have

$$\begin{aligned}&\alpha_0 + (|\tilde{J}_1| - 1)\alpha - \Big[(d(M) - \alpha(M)) - (d(\tilde{J}_0) - \alpha(\tilde{J}_0)) \Big] \\ &\quad \leq \alpha_0 + (|\tilde{J}_1| - 1)\alpha - (m - |\tilde{J}_0|)\alpha = \alpha_0 + (u - m - 1)\alpha,\end{aligned}$$

completing the proof.

□

As in Proposition 6, Proposition 7 can be applied recursively to obtain new nontrivial valid inequalities for TPMC.

Next we give an example illustrating the valid inequalities introduced in this section.

Example 4. *Consider an instance of TPMC with a complete bipartite graph, $V_1 = \{1, 2\}$, $V_2 = \{1, 2, 3, 4\}$, $s = (31, 20)$ and $d = (11, 19, 8, 13)$. A valid inequality for X for this instance is*

$$x_{21} + x_{22} + x_{23} + x_{24} + 11z_1 + 19z_2 + 8z_3 + 13z_4 \geq 20, \tag{28}$$

which corresponds to inequality (15) with $I = \{2\}$ and $J = \{1, 2, 3, 4\}$. Note that $d(J) - s(V_1 \setminus I) = 20 \geq d_j$ for all $j \in J$.

Using (28) as the type-I base inequality, we apply the coefficient update in Proposition 6 and let $\tilde{J} = \{1, 4\}$, $J^* = \{1, 2, 4\}$. Then $\beta_1 + \beta_4 = 11 + 13 = 24$ and $(\beta_1 + \beta_4) - \beta_0 = 24 - 20 = 4 = \beta$, and we obtain the corresponding inequality (19)

$$x_{21} + x_{22} + x_{23} + x_{24} + 7z_1 + 15z_2 + 8z_3 + 9z_4 \geq 16, \tag{29}$$

which is valid for X.

Using (29) as the type-I base inequality, we apply the coefficient update in Proposition 6 and let $\tilde{J} = \{3, 4\}$, $J^* = \{2, 3, 4\}$. Then $\beta_3 + \beta_4 = 8 + 9 = 17$ and $(\beta_3 + \beta_4) - \beta_0 = 17 - 16 = 1 = \beta$ and again we obtain the corresponding inequality (19)

$$x_{21} + x_{22} + x_{23} + x_{24} + 7z_1 + 14z_2 + 7z_3 + 8z_4 \geq 15, \tag{30}$$

which is valid for X.

Now, consider the supply constraint (1c) for supplier 2

$$x_{21} + x_{22} + x_{23} + x_{24} \leq 20. \tag{31}$$

Then using (31) as the type-II base inequality with $I = \{2\}$ and $J = \{1, 2, 3, 4\}$, we apply the coefficient update in Proposition 7, where we let $\tilde{J} = \{2, 4\}$. Then $\alpha_0 - \sum_{j \in J \setminus \tilde{J}} \max\{d_j, \alpha_j\} = \alpha_0 - (d_1 + d_3) = 20 - (11 + 8) = 1$. However, all demand values in set $\tilde{J}$ are greater than 1 so $m = 0$ and $\alpha = \alpha_0 - (d_1 + d_3) - \alpha_2 - \alpha_4 = 20 - (11 + 8) - 0 - 0 = 1$. Then we obtain the corresponding inequality (27)

$$x_{21} + x_{22} + x_{23} + x_{24} + z_2 + z_4 \leq 21, \tag{32}$$

which is valid for X.

4.3 Strength of the Proposed Inequalities

Next we give several facet conditions for inequalities (15). Let V_2' be the set of markets. Observe that if $s(V_1) < d_j$ for some $j \in V_2'$ then the demand of market j can never be met in any feasible solution to TPMC. Therefore, we can set $z_j = 1$ for such markets and let $V_2 = \{j \in V_2' : s(V_1) \geq d_j\}$. In other words, we remove the markets that can never be satisfied from the given set of markets. Therefore, throughout we make the assumption that

$$s(V_1) \geq \max_{j \in V_2} d_j. \tag{33}$$

Let $J^< = \{j \in J : d_j < d(J) - s(V_1 \setminus I)\}$.

Theorem 2. *Inequality (15) defines a nontrivial facet of conv(X) only if the following conditions hold:*

1. *$d(J) > s(V_1 \setminus I)$.*
2. *There exists $j \in J$ such that $d_j > d(J) - s(V_1 \setminus I)$.*
3. *$s(V_1) \geq d(J) - \max_{j \in J}\{d_j\} + \max_{j \in V_2 \setminus J}\{d_j\}$.*
4. *If $s(V_1) < d(J)$ and $I \neq \emptyset$, then $|J^<| \geq 2$ and the sum of the smallest two demands in set $J^<$ is not greater than $d(J) - s(V_1 \setminus I)$.*

5. *$I \neq V_1$.*

6. *If $|J| = 1$, then $|V_1 \setminus I| = 1$.*

7. *$s(V_1) \geq d(J \setminus J^<) + \max_{j \in J^<}\{d_j\}$.*

8. *If $s(V_1) = d(J)$ and $d_j \geq d(J) - s(V_1 \setminus I)$ for all $j \in J$ then $|I| \leq 1$.*

In addition, if the following conditions hold, then (15) *is a facet of conv$(TPMC)$:*

9. *$s(V_1) > d(J) - \max_{j \in J}\{d_j\} + \max_{j \in V_2 \setminus J}\{d_j\}$.*

10. *There exists $\hat{J} \subsetneq J^<$ such that $d(J \setminus \hat{J}) > s(V_1 \setminus I)$ and $d(J \setminus \hat{J}') > s(V_1 \setminus I)$ where $\hat{J}' = \hat{J} \cup \{k_1\}$, for all $k_1 \in J^< \setminus \hat{J}$.*

11. *$s(V_1) > \max_{j \in V_2} d_j$.*

Proof. Necessity.

1. Assume that $d(J) - s(V_1 \setminus I) \leq 0$.

   From validity of inequality (15) we have $d(J) - s(V_1 \setminus I) \geq 0$ and combined with the assumption we get $d(J) - s(V_1 \setminus I) = 0$. The resulting inequality is implied by the nonnegativity of x_{ij} and z_j for $i \in I$, $j \in J$, $(i,j) \in E$.

2. Assume that $d_j \leq d(J) - s(V_1 \setminus I)$ for all $j \in J$. Under this assumption inequality (15) reduces to

$$\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} + \sum_{j \in J} d_j z_j \geq d(J) - s(V_1 \setminus I). \tag{34}$$

   We add all the demand constraints (1b) in J,

$$\sum_{i \in V_1, j \in J:(i,j) \in E} x_{ij} + \sum_{j \in J} d_j z_j = d(J). \tag{35}$$

   When we subtract (35) from (34) we obtain

$$\sum_{i \in V_1 \setminus I, j \in J:(i,j) \in E} x_{ij} \leq s(V_1 \setminus I). \tag{36}$$

   If $J \subsetneq V_2$ then inequality (36) is weaker than all the supply inequalities (1c) in $V_1 \setminus I$ combined, because $x_{ij} \geq 0$ for all $i \in I, j \in V_2 \setminus J$, $(i,j) \in E$. If $J = V_2$ then inequality (36) is dominated by the supply inequalities $\sum_{j \in V_2:(i,j) \in E} x_{ij} \leq s_i$ for all $i \in V_1 \setminus I$ unless $|V_1 \setminus I| = 1$. However, when $J = V_2$, $|V_1 \setminus I| = 1$ and $d_j \leq d(J) - s(V_1 \setminus I)$ for all $j \in J$ inequality (15) reduces to a trivial facet.

3. Assume that $s(V_1) < d(J) - \max_{j \in J}\{d_j\} + \max_{j \in V_2 \setminus J}\{d_j\}$. Because we have showed that there exists $j \in J$ such that $d_j > d(J) - s(V_1 \setminus I)$ we can conclude that $s(V_1 \setminus I) > d(J) - d_j \geq d(J) - \max_{j \in J}\{d_j\}$. Note that we have to have $s(I) < \max_{j \in V_2 \setminus J}\{d_j\}$ for $s(V_1) < d(J) - \max_{j \in J}\{d_j\} + \max_{j \in V_2 \setminus J}\{d_j\}$ to hold because if $s(I) \geq \max_{j \in V_2 \setminus J}\{d_j\}$, then $s(V_1) = s(V_1 \setminus I) + s(I) > d(J) - \max_{j \in J}\{d_j\} + \max_{j \in V_2 \setminus J}\{d_j\}$ which would contradict our assumption. Let $r^* = \arg\max_{j \in V_2 \setminus J}\{d_j\}$. Because (15) is a non-trivial facet, it is different from $z_{r^*} \leq 1$ and there exists solutions on the face defined by (15) with $z_{r^*} = 0$. Note that $\sum_{j \in J \setminus J^<} z_j \leq 1$ for any point to be on the face defined by inequality (15). We consider the following cases:

(a) $\sum_{j \in J \setminus J^<} z_j = 1 = z_l$ for some $l \in J \setminus J^<$.
In this case, left-hand side of inequality (15) reduces to

$$\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} + \sum_{j \in J \setminus \{l\}} (\min\{d(J) - s(V_1 \setminus I), d_j\}) z_j + d(J) - s(V_1 \setminus I)$$

since $l \in J \setminus J^<$, $\min\{d(J) - s(V_1 \setminus I), d_l\} = d(J) - s(V_1 \setminus I)$. Thus to satisfy inequality (15) at equality we must have $\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} = 0$, $z_j = 0$ for all $j \in J \setminus \{l\}$ and

$$\sum_{i \in V_1 \setminus I, j \in J \setminus \{l\} : (i,j) \in E} x_{ij} = d(J \setminus \{l\}) \leq s(V_1 \setminus I) - (d_{r^*} - s(I)) = s(V_1) - d_{r*} \tag{37}$$

where $d_{r^*} - s(I)$ is the amount of demand of market r^* that cannot be satisfied by the suppliers in set I. We obtain a contradiction because (37) implies that $s(V_1) \geq d(J) - d_l + d_{r*} \geq d(J) - \max_{j \in J}\{d_j\} + \max_{j \in V_2 \setminus J}\{d_j\}$, since $d_l \leq \max_{j \in J}\{d_j\}$.

(b) $\sum_{j \in J \setminus J^<} z_j = 0$.
Let $\hat{J} = \{j \in J^< : z_j = 1\}$. Then a point on the face defined by inequality (15) satisfies

$$\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} + \sum_{j \in \hat{J}} d_j = d(J) - s(V_1 \setminus I).$$

This implies that $\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} = d(J \setminus \hat{J}) - s(V_1 \setminus I) \geq 0$ because otherwise we would not have a feasible solution. Furthermore, $\sum_{i \in V_1 \setminus I, j \in J \setminus \hat{J} : (i,j) \in E} x_{ij} = s(V_1 \setminus I)$. Combining the results we observe that because $s(I) < d_{r^*}$ we cannot send all the demand of d_{r^*} from $s(I)$ so some of the supply from $s(V_1 \setminus I)$ should be sent to d_{r^*} but all the supply $s(V_1 \setminus I)$ is sent to markets in $J \setminus \hat{J}$. We reach a contradiction, we cannot have $z_{r^*} = 0$.

4. Suppose that $s(V_1) < d(J)$ and $I \neq \emptyset$, then not all demand in set J can be met, hence $\sum_{j \in J} z_j \geq 1$. Consider the following cases:

(a) $J^< = \emptyset$. Then inequality $\sum_{j \in J}(d(J) - s(V_1 \setminus I))z_j \geq d(J) - s(V_1 \setminus I)$ dominates inequality (15) since inequality (15) has the additional term $\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} \geq 0$.

(b) $|J^<| = 1$. Let $J^< = \{k\}$. We apply the coefficient update in Proposition 6 using inequality (15) as the type-I base inequality. Let $\tilde{J} = \{j, k\}$ where $j \in J \setminus \{k\}$. Therefore, $\beta = \beta_j + d_k - \beta_0 = d(J) - s(V_1 \setminus I) + d_k - (d(J) - s(V_1 \setminus I)) = d_k$ and the corresponding inequality (19) is

$$\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} + \sum_{j \in J \setminus \{k\}} (d(J) - s(V_1 \setminus I) - d_k) z_j + (d_k - d_k) z_k \geq d(J) - s(V_1 \setminus I) - d_k. \tag{38}$$

If we add $\sum_{j \in J} d_k z_j \geq d_k$ to inequality (38) we obtain (15). Hence, (15) cannot be a facet.

(c) $|J^<| \geq 2$ and $d_{j_1} + d_{j_2} > d(J) - s(V_1 \setminus I)$ where d_{j_1} and d_{j_2} are the two smallest demands in set $J^<$. We use the coefficient update in Proposition 6 using inequality (15) as the type-I base inequality. Let $\tilde{J} = \{j_1, j_2\}$. Therefore, $\beta = d_{j_1} + d_{j_2} - (d(J) - s(V_1 \setminus I))$ and the corresponding inequality (19) is

$$\begin{aligned} &\textstyle\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} + \sum_{j \in J \setminus J^<} (2(d(J) - s(V_1 \setminus I)) - d_{j_1} - d_{j_2}) z_j \\ &\quad + \textstyle\sum_{j \in J^< \setminus \{j_1, j_2\}} (d_j - (d_{j_1} + d_{j_2} - (d(J) - s(V_1 \setminus I)))) z_j \\ &\quad + (d(J) - s(V_1 \setminus I) - d_{j_2}) z_{j_1} + (d(J) - s(V_1 \setminus I) - d_{j_1}) z_{j_2} \\ &\qquad \geq 2(d(J) - s(V_1 \setminus I)) - d_{j_1} - d_{j_2}. \end{aligned} \tag{39}$$

Because d_{j_1} and d_{j_2} are the two smallest demands we have $J^* = J$ in Proposition 6. Note that if we add $\sum_{j \in J}(d_{j_1} + d_{j_2} - (d(J) - s(V_1 \setminus I)))z_j \geq d_{j_1} + d_{j_2} - (d(J) - s(V_1 \setminus I))$ to inequality (39) we obtain (15). Hence, (15) cannot be a facet.

5. Assume that $I = V_1$. Then inequality (15) reduces to

$$\sum_{i \in V_1, j \in J:(i,j) \in E} x_{ij} + \sum_{j \in J} d_j z_j \geq d(J). \tag{40}$$

Inequality (40) is a relaxation of the demand equalities (1b) in TPMC. Therefore, if $I = V_1$ then all points in TPMC are on the face defined by inequality (15), therefore this inequality does not define a proper face.

6. Suppose that $J = \{j\}$, but $|V_1 \setminus I| > 1$. Then inequality (15) is

$$\sum_{i \in I:(i,j) \in E} x_{ij} + (d_j - s(V_1 \setminus I)) z_j \geq d_j - s(V_1 \setminus I), \tag{41}$$

where $d_j > s(V_1 \setminus I)$ from facet condition 1. Subtracting the original demand equality (1b) for j from inequality (41), we get

$$\sum_{i \in V_1 \setminus I:(i,j) \in E} x_{ij} \leq s(V_1 \setminus I)(1 - z_j),$$

which is dominated by VUB inequalities (14) for $i \in V_1 \setminus I$.

7. Assume that $s(V_1) < d(J \setminus J^<) + \max_{j \in J^<}\{d_j\}$. Then not all demand for markets in set $J \setminus J^<$ and the largest demand in set $J^<$ can be met at the same time. Hence, $\sum_{j \in J \setminus J^<} z_j + z_m \geq 1$ where $m = \arg\max_{j \in J^<}\{d_j\}$. We use Proposition 6 and inequality (15) as the type-I base inequality. Let $\tilde{J} = \{l, m\}$ where $l \in J \setminus J^<$ then $\beta = d(J) - s(V_1 \setminus I) + d_m - (d(J) - s(V_1 \setminus I)) = d_m$. We obtain

$$\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} + \sum_{j \in J \setminus J^<} (d(J) - s(V_1 \setminus I) - d_m) z_j \quad + \textstyle\sum_{j \in J^< \setminus \{m\}} d_j z_j + (d_m - d_m) z_m \tag{42}$$
$$\geq d(J) - s(V_1 \setminus I) - d_m.$$

If we add $\sum_{j \in J \setminus J^<} d_m z_j + d_m z_m \geq d_m$ to inequality (42) we obtain (15). Hence, (15) cannot be a facet.

8. Assume that $s(V_1) = d(J)$, $d_j \geq d(J) - s(V_1 \setminus I)$ for all $j \in J$ and for contradiction $|I| \geq 2$. Because of assumption $s(V_1) = d(J)$ we have $d_j \geq d(J) - s(V_1 \setminus I) = s(V_1) - s(V_1 \setminus I) = s(I)$ for all $j \in J$. Under these assumptions inequality (15) reduces to $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} + \sum_{j \in J} s(I) z_j \geq s(I)$. Let $I' = I \setminus \{i'\}$ and $I^{''} = \{i'\}$ where $i' \in I$ ($I' \neq \emptyset$ and $I^{''} \neq \emptyset$ because $|I| \geq 2$ by assumption). Consider the following inequalities in the form of inequality (15) with set I replaced with sets I' and I'', respectively

$$\sum_{i \in I \setminus \{i'\}, j \in J:(i,j) \in E} x_{ij} + \sum_{j \in J} (s(I) - s_{i'}) z_j \geq s(I) - s_{i'}, \tag{43}$$

$$\sum_{j \in J:(i',j) \in E} x_{i'j} + \sum_{j \in J} s_{i'} z_j \geq s_{i'}. \tag{44}$$

Inequality (43) is valid because $d(J) - s(V_1 \setminus I') = d(J) - s(V_1 \setminus I) - s_{i'} = s(I) - s_{i'} > 0$. Furthermore, the coefficient of z_j is $\min\{d_j, s(I) - s_{i'}\} = s(I) - s_{i'}$ because of the assumption $d_j \geq d(J) - s(V_1 \setminus I) = s(I)$ for all $j \in J$. Inequality (44) is valid because $d(J) - s(V_1 \setminus I^{''}) = s(V_1) - s(V_1 \setminus I^{''}) = s(I^{''}) = s_{i'} > 0$ and similarly the coefficient of z_j is $\min\{s_{i'}, d_j\} = s_{i'}$, because $d_j \geq s(I) \geq s_{i'}$ for all $j \in J$ by assumption. By adding inequalities (43) and (44) we obtain inequality (15) with set I. Hence, (15) cannot be a facet.

Sufficiency. We use the technique in §I.4.3 Theorem 3.6 [19]. We show that inequality (15), plus any linear combination of the demand constraints $\sum_{i \in V_1 : (i,j) \in E} x_{ij} + d_j z_j = d_j$ for all $j \in V_2$ is the only inequality that is satisfied at equality by all points (x, z) feasible to TPMC that are tight at (15), i.e., we show that if all points of TPMC at which (15) is tight satisfy

$$\sum_{(i,j) \in E} \alpha_{ij} x_{ij} + \sum_{j \in V_2} \psi_j z_j = \hat{\alpha}, \tag{45}$$

then

1. $\alpha_{ij} = u_j$, $j \in V_2 \setminus J$, $i \in V_1$, $(i, j) \in E$,
2. $\alpha_{ij} = u_j$, $j \in J$, $i \in V_1 \setminus I$, $(i, j) \in E$,
3. $\alpha_{ij} = \bar{\alpha} + u_j$, $j \in J$, $i \in I$, $(i, j) \in E$,
4. $\psi_j = u_j d_j$, $j \in V_2 \setminus J$
5. $\psi_j = \bar{\alpha}\left(\min\left\{d(J) - s(V_1 \setminus I), d_j\right\}\right) + u_j d_j$, $j \in J$,
6. $\hat{\alpha} = \bar{\alpha}\left(d(J) - s(V_1 \setminus I)\right) + \sum_{j \in V_2} u_j d_j$.

In the proof we consider three different types of points at which (15) is tight. These points are solutions to TPMC but are subject to additional systems of constraints. Throughout, let ϵ be a very small number greater than zero unless noted otherwise.

1. Suppose that $d_l > d(J) - s(V_1 \setminus I)$ for $l = \arg\max_{j \in J}\{d_j\}$. Consider a point where only markets $j \in \{r\} \cup J \setminus \{l\}$ are satisfied for some $r \in V_2 \setminus J$ and constraints

$$\begin{aligned}
\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} &= 0 && \\
\sum_{i \in V_1 \setminus I, j \in J : (i,j) \in E} x_{ij} &= d(J) - d_l && \\
\sum_{i \in V_1 : (i,r) \in E} x_{ir} &= d_r && \\
x_{ij} &= 0, && i \in V_1, j \in \{l\} \cup V_2 \setminus (J \cup \{r\}) \\
x_{ij} &\geq \epsilon, && i \in V_1 \setminus I, j \in J \setminus \{l\} \\
x_{ir} &\geq \epsilon, && i \in V_1 \\
\sum_{j \in V_2 : (i,j) \in E} x_{ij} &\leq s_i - \epsilon, && i \in V_1 \\
z_j &= 1, && j \in \{l\} \cup V_2 \setminus (J \cup \{r\}) \\
z_j &= 0, && j \in \{r\} \cup J \setminus \{l\}
\end{aligned}$$

in addition to the original constraints are satisfied, which we refer to as System 1. We know that a solution to System 1 exists from facet conditions 9 and 11. For a solution to be feasible to System 1 the demand of markets $j \in \{r\} \cup J \setminus \{l\}$ have to be met, i.e., $s(V_1) \geq d(J) - \max_{j \in J}\{d_j\} + \max_{j \in V_2 \setminus J}\{d_j\}$. Additionally, we would like to change a given solution by increasing and decreasing the x values by ϵ hence the need for $>$ relationship in facet condition 9.

2. Suppose that $d_l > d(J) - s(V_1 \setminus I)$ for some $l \in J$. Consider a point where only markets $j \in J \setminus \{l\}$ are satisfied and constraints

$$\begin{aligned}
\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} &= 0 && \\
\sum_{i \in V_1 \setminus I, j \in J:(i,j) \in E} x_{ij} &= d(J) - d_l && \\
x_{ij} &= 0, && i \in V_1, j \in \{l\} \cup V_2 \setminus J \\
x_{ij} &\geq \epsilon, && i \in V_1 \setminus I, j \in J \setminus \{l\} \\
\sum_{j \in V_2:(i,j) \in E} x_{ij} &\leq s_i - \epsilon, && i \in V_1 \setminus I \\
z_j &= 1, && j \in \{l\} \cup V_2 \setminus J \\
z_j &= 0, && j \in J \setminus \{l\}
\end{aligned}$$

in addition to the original constraints are satisfied, which we refer to as System 2. We know that a solution to System 2 exists from facet condition 2 since there exists at least one $j \in J$ such that $s(V_1) \geq s(V_1 \setminus I) > d(J) - d_j$, and from facet condition 11.

3. We define $\hat{J} \subset J$ such that $d(J \setminus \hat{J}) > s(V_1 \setminus I)$. Due to the choice of $\hat{J}$ we have $d_j < d(J) - s(V_1 \setminus I)$ for all $j \in \hat{J}$ so $\hat{J} \subseteq J^<$ (if $d_{j'} \geq d(J) - s(V_1 \setminus I)$ and $j' \in \hat{J}$ then we cannot have $d(J \setminus \hat{J}) > s(V_1 \setminus I)$). In this point, markets in set $\hat{J} \cup V_2 \setminus J$ are rejected and constraints

$$\begin{aligned}
\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} &= d(J \setminus \hat{J}) - s(V_1 \setminus I) && \\
\sum_{i \in V_1 \setminus I, j \in J:(i,j) \in E} x_{ij} &= s(V_1 \setminus I) && \\
x_{ij} &= 0, && i \in V_1, j \in \hat{J} \cup V_2 \setminus J \\
x_{ij} &\geq \epsilon, && i \in V_1, j \in J \setminus \hat{J} \\
\sum_{j \in J \setminus \hat{J}:(i,j) \in E} x_{ij} &\leq s_i - \epsilon, && i \in I \\
z_j &= 1, && j \in \hat{J} \cup V_2 \setminus J \\
z_j &= 0, && j \in J \setminus \hat{J}
\end{aligned}$$

in addition to the original constraints are satisfied, which we refer to as System 3. We consider a set $\hat{J}$ such that all demand in set $J \setminus J^<$ is satisfied and $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} > 0$. This is possible due to facet conditions 7, 11, and non-negativity of x variables.

In order to establish the values of the coefficients α_{ij}, ψ_j and $\hat{\alpha}$, we construct a feasible solution to the given systems 1, 2 and 3. Then a small change in the solution is made. By evaluating (45) at both solutions, which are on the face defined by (15) and comparing the resulting expressions, the possible values of a set of coefficients are obtained.

We start by showing that

1. $\alpha_{ij} = u_j$, $j \in V_2 \setminus J$, $i \in V_1$, $(i,j) \in E$.

   Consider any solution to system 1 with any market $r \in V_2 \setminus J$ that is satisfied. Choose arbitrary suppliers $i, i' \in V_1$ such that $(i,r), (i',r) \in E$. Construct a new point by decreasing the flow on edge (i,r) by ϵ and increasing the flow on edge (i',r) by ϵ. Note that this point is also on the face defined by inequality (15). Thus,

$$\alpha_{ij} = u_j, j \in V_2 \setminus J, i \in V_1, (i,j) \in E.$$

2. $\alpha_{ij} = u_j$, $j \in J$, $i \in V_1 \setminus I$, $(i,j) \in E$. Note that if $|V_1 \setminus I| = 1$, then $\alpha_{ij} = u_j$, $j \in J$ trivially holds. We condition on the number of markets in set J.

   (a) $J = \{k\}$. Note that, from facet condition 6, we have $|V_1 \setminus I| = 1$, so the result holds.

   (b) $|J| \geq 2$. By assumption, $|V_1 \setminus I| > 1$. Due to facet condition 2 there exists $k \in J$ such that $d_k > d(J) - s(V_1 \setminus I)$. We consider a solution to system 2 with $l = k$. Choose any market $j \in J \setminus \{k\}$, any suppliers $i, i' \in V_1 \setminus I$ such that $(i,j), (i',j) \in E$. Make an ϵ-change of flow between the two suppliers i, i' and market j. Thus,

   $$\alpha_{ij} = u_j, j \in J \setminus \{k\}, i \in V_1 \setminus I, (i,j) \in E.$$

   Next we show that $\alpha_{ik} = u_k$ for all $i \in V_1 \setminus I$. If there exists another j^* such that $d_{j^*} > d(J) - s(V_1 \setminus I)$, $j^* \neq k$ then we consider a point satisfying System 2 with $l = j^*$, and use the same argument as before to show that $\alpha_{ik} = u_k$ for all $i \in V_1 \setminus I$. If no such j^* exists then $d_j \leq d(J) - s(V_1 \setminus I)$ for all $j \in J \setminus \{k\}$. In this case k is the only market in J with $d_k > d(J) - s(V_1 \setminus I)$. Then from facet condition 7 we know that there exists a solution to a variant of System 3 with $\hat{J} \subseteq J^< \setminus \{j\}$ for some $j \in J \setminus \{k\}$ (in which we set $\epsilon = 0$ in case facet condition 7 is satisfied at equality), where along with market k we can satisfy at least one more market, j. Choose suppliers $i, i' \in V_1 \setminus I$ such that $(i,k), (i',k), (i,j), (i',j) \in E$. Decrease flow on edges (i,j), (i',k) by ϵ and increase flow on edges (i,k), (i',j) by ϵ. Note that since we are using a solution to a variant of system 3 in which we set $\epsilon = 0$ inequality (15) is still tight. Thus,

   $$\alpha_{ik} - \alpha_{ij} - \alpha_{i'k} + \alpha_{i'j} = \alpha_{ik} - u_j - \alpha_{i'k} + u_j = \alpha_{ik} - \alpha_{i'k} = 0.$$

   Therefore, $\alpha_{ik} = u_k$ for all $i \in V_1 \setminus I$.

3. $\alpha_{ij} = \bar{\alpha} + u_j$, $j \in J$, $i \in I$, $(i,j) \in E$.

   Consider a solution to system 3 with $\hat{J} \subseteq J^<$. Choose any market $j \in J \setminus \hat{J}$, any two suppliers $i, i' \in I$ such that $(i,j), (i',j) \in E$. Make an ϵ-change of flow between the two suppliers i, i' and market j. Thus,

   $$\alpha_{ij} = \alpha^1_j, j \in J \setminus \hat{J}, i \in I, (i,j) \in E.$$

   Let $\alpha^1_j = \bar{\alpha}_j + u_j$, $j \in J \setminus \hat{J}$. Facet condition 10 and definition of $\hat{J}$ (i.e. $\hat{J} \subseteq J^<$) implies that for any $k_1 \in J^<$ we can redefine $\hat{J}$ to either include k_1 or not. More specifically, if $k_1 \in \hat{J}$ then market k_1 is rejected. To show that $\alpha_{ik_1} = \alpha^1_{k_1}$ for all $i \in I$, $(i,k_1) \in E$ we choose another $\hat{J}$ such that $k_1 \notin \hat{J}$. Using the same argument as before we obtain $\alpha_{ik_1} = \alpha^1_{k_1}$ for all $i \in I$, $(i,k_1) \in E$. As a result, we have shown that $\alpha_{ij} = \alpha^1_j, j \in \hat{J}, i \in I, (i,j) \in E$. Next we show that $\bar{\alpha}_j = \bar{\alpha}$, $j \in J \setminus \hat{J}$. Choose any markets $j, j' \in J \setminus \hat{J}$, any suppliers $i \in V_1 \setminus I$, $i' \in I$ such that $(i,j), (i',j), (i,j'), (i',j') \in E$. Decrease flow on edges (i,j'), (i',j) by ϵ and increase flow on edges (i,j), (i',j') by ϵ. Thus,

   $$\alpha_{ij} - \alpha_{ij'} - \alpha_{i'j} + \alpha_{i'j'} = u_j - u_{j'} - \alpha^1_j + \alpha^1_{j'} = 0.$$

   By again using $\alpha^1_j = \bar{\alpha}_j + u_j$ and $\alpha^1_{j'} = \bar{\alpha}_{j'} + u_{j'}$, we obtain

   $$\bar{\alpha}_j = \bar{\alpha}_{j'}.$$

   Since j and j' can be chosen as any market in $J \setminus \hat{J}$ we conclude that $\bar{\alpha}_j = \bar{\alpha}$, $j \in J \setminus \hat{J}$. Furthermore, since as before we can rearrange set $\hat{J}$ to include or not include any $k_1 \in J^<$ we get $\bar{\alpha}_j = \bar{\alpha}$, $j \in \hat{J}$.

4. $\psi_j = u_j d_j$, $j \in V_2 \setminus J$. We rewrite (45) using the information obtained until now and get

   $$\bar{\alpha} \sum_{i \in I, j \in J : (i,j) \in E} x_{ij} + \sum_{(i,j) \in E} u_j x_{ij} + \sum_{j \in V_2} \psi_j z_j = \hat{\alpha}. \tag{46}$$

Consider any solution to system 1 with any market $r \in V_2 \setminus J$ that is satisfied. Then we construct a new solution based on this solution where we set $z_r = 1$ and $x_{ir} = 0$ for all $i \in V_1$, $(i,r) \in E$ and all other variables remain the same. Note that this solution is also on the face defined by (15) since $r \in V_2 \setminus J$ and the new solution is a solution to system 2. We compare face (45) evaluated at these two solutions. Thus,

$$u_r \sum_{i \in V_1:(i,r) \in E} x_{ir} - \psi_r = 0.$$

Because $\sum_{i \in V_1:(i,r) \in E} x_{ir} = d_r$ we have $\psi_r = u_r d_r$.

5. $\psi_j = \bar{\alpha} \left(\min \{ d(J) - s(V_1 \setminus I), d_j \} \right) + u_j d_j$, $j \in J$.

   We consider 2 cases.

   (a) $d_{j'} < d(J) - s(V_1 \setminus I)$ for some $j' \in J$.

   We consider a solution to system 3 with $\hat{J}$ such that $d(\hat{J}) + d_{j'} \leq d(J) - s(V_1 \setminus I)$. This is a feasible solution due to facet condition 10 where $k_1 = j'$. We evaluate (46) at this solution and obtain

$$\bar{\alpha}(d(J \setminus \hat{J}) - s(V_1 \setminus I)) + \sum_{i \in V_1, j \in J \setminus \hat{J}:(i,j) \in E} u_j x_{ij} + \sum_{j \in \hat{J} \cup V_2 \setminus J} \psi_j = \hat{\alpha}.$$

   Then we use the same solution except now we set $z_{j'} = 1$, $x_{ij'} = 0$, $i \in V_1$, $(i,j') \in E$ (so we redefine $\hat{J}$ as $\hat{J}' = \hat{J} \cup \{j'\}$) and $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} = d(J \setminus \hat{J}) - s(V_1 \setminus I) - d_{j'}$ and evaluate (46) again. Note that this solution is also on the face defined by (15) because we had $z_{j'} = 0$, $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} = d(J \setminus \hat{J}) - s(V_1 \setminus I)$ and we changed it with $z_{j'} = 1$, $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} = d(J \setminus \hat{J}) - s(V_1 \setminus I) - d_{j'}$ and the coefficient of $z_{j'}$ is $d_{j'}$ in inequality (15). Thus,

$$\begin{aligned} \bar{\alpha}(d(J \setminus \hat{J}) - s(V_1 \setminus I) - d_{j'}) + \textstyle\sum_{i \in V_1, j \in J \setminus \hat{J}':(i,j) \in E} u_j x_{ij} \\ + \textstyle\sum_{j \in \hat{J} \cup V_2 \setminus J} \psi_j + \psi_{j'} = \hat{\alpha}. \end{aligned}$$

   Taking the difference between (46) evaluated at these two solutions, we obtain

$$\psi_{j'} = \bar{\alpha} d_{j'} + u_{j'} \sum_{i \in V_1:(i,j') \in E} x_{ij'} = \bar{\alpha} d_{j'} + u_{j'} d_{j'}.$$

   (b) $d_{j'} \geq d(J) - s(V_1 \setminus I)$ for some $j' \in J$.

   We consider a solution to system 3 with any feasible $\hat{J}$ such that the right hand side of inequality $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} = d(J \setminus \hat{J}) - s(V_1 \setminus I)$ is nonnegative and market j' is satisfied. In the solution we can set $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} = \sum_{i \in I:(i,j') \in E} x_{ij'}$. This is a feasible solution since $d_{j'} \geq d(J) - s(V_1 \setminus I)$ by assumption and we know that for inequality (15) to be tight we cannot have $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} > d(J) - s(V_1 \setminus I)$. Hence, $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} \leq d(J) - s(V_1 \setminus I)$ and we can choose a solution in which a part (or all) of the demand of market j' is met by suppliers in set I. We use $\psi_j = \bar{\alpha} d_j + u_j d_j$ for all $j \in J^<$ and recall that markets in set $\hat{J} \subseteq J^<$ are rejected. We evaluate (46) at this solution and obtain

$$\begin{aligned} \bar{\alpha}(d(J \setminus \hat{J}) - s(V_1 \setminus I) + d(\hat{J})) + u_{j'} \sum_{i \in I:(i,j') \in E} x_{ij'} + \sum_{i \in V_1 \setminus I, j \in J \setminus \hat{J}:(i,j) \in E} u_j x_{ij} \\ + \sum_{j \in \hat{J} \cup V_2 \setminus J} u_j d_j = \hat{\alpha}. \end{aligned}$$

Then we use the same solution except now we set $z_{j'} = 1$, $z_q = 0$, $q \in \hat{J}$ (this is still a feasible solution since $s(V_1) \geq s(V_1 \setminus I) \geq d(J) - d_{j'}$ by assumption, i.e., once market j' is rejected all other markets in set J can be satisfied) and $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} = 0$ (implying that $\sum_{i \in I:(i,j') \in E} x_{ij'} = 0$) and reevaluate (46). Note that this solution is also on the face defined by (15) because we had $z_{j'} = 0$, $z_q = 1$, $q \in \hat{J}$, $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} = d(J \setminus \hat{J}) - s(V_1 \setminus I)$ and we changed it with $z_{j'} = 1$, $z_q = 0$, $q \in \hat{J}$, $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} = 0$ and the coefficient of $z_{j'}$ is $d(J) - s(V_1 \setminus I)$. Thus,

$$\bar{\alpha}(0) + 0 + \sum_{i \in V_1 \setminus I, j \in J \setminus \{j'\}:(i,j) \in E} u_j x_{ij} + \sum_{j \in V_2 \setminus J} u_j d_j + \psi_{j'} = \hat{\alpha}.$$

Taking the difference between (46) evaluated at these two solutions, we get $\bar{\alpha}(d(J) - s(V_1 \setminus I)) + u_{j'} \sum_{i \in V_1:(i,j') \in E} x_{ij'} - \sum_{i \in V_1, j \in \hat{J}} u_j x_{ij} + \sum_{j \in \hat{J}} u_j d_j - \psi_{j'} = 0$. Because $\sum_{i \in V_1:(i,j') \in E} x_{ij} = d_{j'}$ and $\sum_{i \in V_1, j \in \hat{J}} u_j x_{ij} = \sum_{j \in \hat{J}} u_j d_j$ we have $\psi_{j'} = \bar{\alpha}(d(J) - s(V_1 \setminus I)) + u_{j'} d_{j'}$.

6. $\hat{\alpha} = \bar{\alpha}\left(d(J) - s(V_1 \setminus I)\right) + \sum_{j \in V_2} u_j d_j$.
   Rewriting equality (45), we get

$$\begin{aligned} \bar{\alpha}\left(\textstyle\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} + \sum_{j \in J} \min\left\{d(J) - s(V_1 \setminus I), d_j\right\} z_j\right) \\ + \textstyle\sum_{(i,j) \in E: j \in V_2} u_j x_{ij} + \sum_{j \in V_2} u_j d_j z_j = \hat{\alpha}. \end{aligned} \tag{47}$$

   Evaluating (47) at any point (x, z) feasible to TPMC that is tight at inequality (15) gives

$$\bar{\alpha}\left(d(J) - s(V_1 \setminus I)\right) + \sum_{j \in V_2} u_j \left(\sum_{i \in V_1:(i,j) \in E} x_{ij} + d_j z_j \right) = \hat{\alpha}.$$

   From equality (1b) in the definition of TPMC we have $\sum_{i \in V_1:(i,j) \in E} x_{ij} + d_j z_j = d_j$ for all $j \in V_2$. Thus, $\hat{\alpha} = \bar{\alpha}\left(d(J) - s(V_1 \setminus I)\right) + \sum_{j \in V_2} u_j d_j$.

□

Our next result shows that the coefficient update scheme in Proposition 6 is neither lifting nor coefficient strengthening. We show that both a type-I base inequality (16) and the corresponding inequality (19) can be facets of $\mathcal{T}_1$ under certain conditions.

Proposition 8. *If the following conditions hold, then type-I base inequality* (16) *and the corresponding inequality* (19) *are facets of* $\mathcal{T}_1$.

1. *If there exists* $j \in J^* \setminus \tilde{J}$ *with* $\beta_j < \beta_0$ *then* $\beta_j - \beta < \beta_0^{'}$ *and* $\beta(\tilde{J} \setminus \{j_p, j_{p-1}\}) + \beta_j \leq \beta_0$ *where* $\tilde{J} = \{j_1, j_2, \ldots, j_p\}$ *and* $\beta_{j_1} \leq \beta_{j_2} \leq \cdots \leq \beta_{j_p}$.
2. *For all* $j \in J \setminus J^*$, $\beta_j < \beta_0^{'}$ *and* $\beta(\tilde{J} \setminus \{j_p\}) + \beta_j \leq \beta_0$.

Proof. We first show that there exists $dim(\mathcal{T}_1) = |J| + 1$ many affinely independent points that satisfy inequality (19) at equality. Consider the following points:

- Let $t = 0$, $z_j = 1$ for all $j \in \tilde{J}$, $z_j = 0$ for all $j \in J \setminus \tilde{J}$. In this case, the left-hand side of inequality (19) is $\beta(\tilde{J}) - p\beta = \beta_0 + \beta - p\beta = \beta_0 - (p-1)\beta = \beta_0^{'}$.
- For each $j' \in \tilde{J}$, $t = \beta_{j'} - \beta$, $z_{j'} = 0$, $z_j = 1$ for all $j \in \tilde{J} \setminus \{j'\}$, $z_j = 0$ for all $j \in J \setminus \tilde{J}$. In this case, the left-hand side of inequality (19) is $\beta_{j'} - \beta + \beta(\tilde{J} \setminus \{j'\}) - (p-1)\beta = \beta(\tilde{J}) - p\beta = \beta_0 + \beta - p\beta = \beta_0 - (p-1)\beta = \beta_0^{'}$. This point also satisfies type-I base inequality (16) at equality.
- For each $j' \in J^* \setminus \tilde{J}$ we consider two cases:

1. $\beta_{j'} = \beta_0$.
   Let $t = 0$, $z_{j'} = 1$, $z_j = 0$ for all $j \in J \setminus \{j'\}$. The left-hand side of inequality (19) is $\min\{(\beta_{j'} - \beta), \beta_0^{'}\} = \min\{(\beta_0 - \beta), \beta_0 - (p-1)\beta\} = \beta_0 - (p-1)\beta$ since p is the number of elements in set $\tilde{J}$ and $p \geq 2$, for $\tilde{J}$ to be a minimal cover. This point also satisfies type-I base inequality (16) at equality.
2. $\beta_{j'} < \beta_0$.
   Let $t = \beta_0 - \beta(\tilde{J} \setminus \{j_p, j_{p-1}\}) - \beta_{j'}$, $z_j = 1$, for all $j \in \tilde{J} \setminus \{j_p, j_{p-1}\}$, $z_{j_p} = 0$, $z_{j_{p-1}} = 0$, $z_{j'} = 1$, $z_j = 0$ for all $J \setminus (\tilde{J} \cup \{j'\})$. From facet condition 1 we have $\beta_{j'} - \beta < \beta_0^{'}$ hence the left-hand side of inequality (19) is $\beta_0 - \beta(\tilde{J} \setminus \{j_p, j_{p-1}\}) - \beta_{j'} + \beta(\tilde{J} \setminus \{j_p, j_{p-1}\}) - (p-2)\beta + \beta_{j'} - \beta = \beta_0 - (p-1)\beta = \beta_0^{'}$. Note that due to facet condition 1, $t \geq 0$. Furthermore, this point also satisfies type-I base inequality (16) at equality.

- For each $j' \in J \setminus J^*$ first observe that $\beta_{j'} < \beta_0$ since by definition of J^*, $\beta_{j'} < \beta_{j_p} \leq \beta_0$. Let $t = \beta_0 - \beta(\tilde{J} \setminus \{j_p\}) - \beta_{j'}$, $z_j = 1$, for all $j \in \tilde{J} \setminus \{j_p\}$, $z_{j_p} = 0$, $z_{j'} = 1$, $z_j = 0$ for all $J \setminus (\tilde{J} \cup \{j'\})$. From facet condition 2 we have $\beta_{j'} < \beta_0^{'}$ hence the left-hand side of inequality (19) is $\beta_0 - \beta(\tilde{J} \setminus \{j_p\}) - \beta_{j'} + \beta(\tilde{J} \setminus \{j_p\}) - (p-1)\beta + \beta_{j'} = \beta_0 - (p-1)\beta = \beta_0^{'}$. Note that due to facet condition 2, $t \geq 0$. Furthermore, this point also satisfies type-I base inequality (16) at equality.

In total we have described $1 + |\tilde{J}| + |J^* \setminus \tilde{J}| + |J \setminus J^*| = |J| + 1$ many points. It is easy to see that these points are affinely independent. Furthermore, except for the first described point ($t = 0$, $z_j = 1$ for all $j \in \tilde{J}$, $z_j = 0$ for all $j \in J \setminus \tilde{J}$) all the other $|J|$ many points also satisfy type-I base inequality (16) at equality. If we replace the first point with the point $t = \beta_0$, $z_j = 0$ for all $j \in J$, which satisfies the type-I base inequality at equality, then we still get $|J| + 1$ many affinely independent points. Hence, both the type-I base inequality (16) and the corresponding inequality (19) are facets of $\mathcal{T}_1$ under conditions 1 and 2. □

Suppose that inequality $\sum_{j \in J} t_j \leq \alpha_0$ is given as a type-II base inequality in the form of (24) for set $\mathcal{S}_2$, where $\alpha_j = 0$ for all $j \in J$. Assume that there exists $\tilde{J}$ and m such that $\alpha_0 > d(J \setminus \tilde{J})$ and $\alpha_0 - d(J \setminus \tilde{J}) < \max_{j \in \tilde{J}}\{d_j\}$. These conditions imply that $m = 0$ and $\alpha = \alpha_0 - d(J \setminus \tilde{J})$. Then we obtain the corresponding inequality (27)

$$\sum_{j \in J} t_j + \sum_{j \in \tilde{J}} \alpha z_j \leq \alpha_0 + (|\tilde{J}| - 1)\alpha, \tag{48}$$

which is valid for $\mathcal{S}_2$, under these assumptions.

Proposition 9. *Inequality* (48)*, valid for $\mathcal{S}_2$, defines a facet of $\mathcal{T}_2$ only if*

1. $\tilde{J} \neq \emptyset$.

In addition, if the following conditions hold then (48) *is a facet of $\mathcal{T}_2$:*

2. $\alpha_0 < d(J \setminus \tilde{J}) + \min_{j \in \tilde{J}}\{d_j\}$,

3. $\alpha_0 < d(J \setminus \tilde{J}) + \max_{j \in \tilde{J}}\{d_j\} - \max_{j \in J \setminus \tilde{J}}\{d_j\}$,

4. $|J \setminus \tilde{J}| \geq 2$.

Proof. Necessity.

1. Assume that $\tilde{J} = \emptyset$. Then inequality (48) reduces to

$$\sum_{j \in J} t_j \leq \alpha_0 - \alpha. \tag{49}$$

This case implies that $\alpha = \alpha_0 - d(J \setminus \tilde{J}) = \alpha_0 - d(J)$. Thus, inequality (49) becomes $\sum_{j \in J} t_j \leq d(J)$ which is dominated by $t_j + d_j z_j \leq d_j$ for all $j \in J$.

Sufficiency. We show that there exists $dim(\mathcal{T}_2) = 2|J|$ many affinely independent points that satisfy inequality (48) at equality. Let $\epsilon > 0$ be a very small number and $j^* = \arg\max_{j \in \tilde{J}}\{d_j\}$ (j^* exists due to facet condition 1). Consider the following points:

- For each $j' \in \tilde{J}$, let $z_{j'} = 0$, $t_{j'} = \alpha_0 - d(J \setminus \tilde{J})$, $z_j = 1$, $j \in \tilde{J} \setminus \{j'\}$, $t_j = 0$, $j \in \tilde{J} \setminus \{j'\}$, $z_j = 0$, $j \in J \setminus \tilde{J}$, $t_j = d_j$, $j \in J \setminus \tilde{J}$. Note that this is a feasible solution due to the assumption that $\alpha_0 > d(J \setminus \tilde{J})$ and facet condition 2. Furthermore, for each such point we construct another point by increasing $t_{j'}$ by ϵ and decreasing any t_j, $j \in J \setminus \tilde{J}$ by ϵ (j exists due to facet condition 4). This gives $2|\tilde{J}|$ many points.
- For each $j'' \in J \setminus \tilde{J}$, let $z_{j''} = 1$, $t_{j''} = 0$, $z_j = 0$, $j \in (J \setminus (\tilde{J} \cup \{j''\})) \cup \{j^*\}$, $t_j = d_j$, $j \in J \setminus (\tilde{J} \cup \{j''\})$, $t_{j^*} = \alpha_0 - d(J \setminus \tilde{J}) + d_{j''}$, $z_j = 1$, $j \in \tilde{J} \setminus \{j^*\}$, $t_j = 0$, $j \in \tilde{J} \setminus \{j^*\}$. Note that this is feasible due to facet condition 3. For each such point we construct another point by increasing t_{j^*} by ϵ and decreasing any t_j, $j \in J \setminus (\tilde{J} \cup \{j''\})$ by ϵ (j exists due to facet condition 4). This gives $2|J \setminus \tilde{J}|$ many points.

It is easy to see that these points are affinely independent.

□

Now, suppose that we start with a type-II base inequality $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} \leq s(I)$ in Proposition 7. Note that inequality $\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} \leq s(I)$ is a relaxation of the supply constraints (1c). Let $t_j = \sum_{i \in I:(i,j) \in E} x_{ij}$ and $\alpha_j = 0$ for all $j \in J$ in inequality (24). Suppose that there exists $\tilde{J}$ and m such that $s(I) > d(J \setminus \tilde{J})$ and $s(I) - d(J \setminus \tilde{J}) < \max_{j \in \tilde{J}}\{d_j\}$. These conditions imply that $m = 0$ and $\alpha = s(I) - d(J \setminus \tilde{J})$. Then we obtain the inequality

$$\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} + \sum_{j \in \tilde{J}} \alpha z_j \leq s(I) + (|\tilde{J}| - 1)\alpha, \tag{50}$$

which is valid for X.

Proposition 10. *Inequality* (50)*, valid for X, defines a facet of conv(X) only if*

1. $\tilde{J} \neq \emptyset$.

In addition, if the following conditions hold then (50) *is a facet of conv*(X)*:*

2. $s(V_1) > d(J \setminus \tilde{J}) + \max_{j \in (V_2 \setminus J) \cup \tilde{J}}\{d_j\}$,

3. $s(I) < d(J \setminus \tilde{J}) + \min_{j \in \tilde{J}}\{d_j\}$,

4. $s(I) \leq d(J \setminus \tilde{J}) + \max_{j \in \tilde{J}}\{d_j\} - \max_{j \in J \setminus \tilde{J}}\{d_j\}$.

Proof. Necessity.

1. If we replace t_j by $\sum_{i \in I:(i,j) \in E} x_{ij}$ for all $j \in J$ and α_0 by $s(I)$ we can use the same argument as in the necessity of facet condition 1 in Proposition 9.

Sufficiency. For the proof we use §I.4.3 Theorem 3.6 [19]. We show that inequality (50), plus any linear combination of the demand constraints $\sum_{i \in V_1:(i,j) \in E} x_{ij} + d_j z_j = d_j$ for all $j \in V_2$ is the only inequality that is satisfied at equality by all points (x, z) feasible to TPMC that are tight at (50), i.e., we show that if all points of TPMC at which (50) is tight satisfy

$$\sum_{(i,j) \in E} \lambda_{ij} x_{ij} + \sum_{j \in V_2} \omega_j z_j = \hat{\lambda}, \tag{51}$$

then

1. $\lambda_{ij} = u_j$, $j \in V_2 \setminus J$, $i \in V_1$, $(i,j) \in E$,
2. $\lambda_{ij} = u_j$, $j \in J$, $i \in V_1 \setminus I$, $(i,j) \in E$,
3. $\lambda_{ij} = \bar{\lambda} + u_j$, $j \in J$, $i \in I$, $(i,j) \in E$,
4. $\omega_j = u_j d_j$, $j \in V_2 \setminus \tilde{J}$,
5. $\omega_j = \bar{\lambda}\alpha + u_j d_j$, $j \in \tilde{J}$,
6. $\hat{\lambda} = \bar{\lambda}\left(s(I) + (|\tilde{J}| - 1)\alpha\right) + \sum_{j \in V_2} u_j d_j$.

In the proof we consider four different types of points at which (50) is tight that make use of the facet conditions. Throughout, let ϵ be a very small number greater than zero unless noted otherwise.

1. Consider a point where only markets $j \in J \setminus \tilde{J} \cup \{r\}$ are satisfied for some $r \in V_2 \setminus J$, and constraints

$$\begin{aligned}
\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} &= d(J \setminus \tilde{J}) && \\
\sum_{i \in V_1 : (i,r) \in E} x_{ir} &= d_r && \\
x_{ij} &= 0, && i \in V_1, j \in \tilde{J} \cup V_2 \setminus (J \cup \{r\}) \\
x_{ij} &\geq \epsilon, && i \in I, j \in J \setminus \tilde{J} \\
x_{ir} &\geq \epsilon, && i \in V_1 \\
\sum_{j \in V_2 : (i,j) \in E} x_{ij} &\leq s_i - \epsilon, && i \in V_1 \\
z_j &= 1, && j \in \tilde{J} \cup V_2 \setminus (J \cup \{r\}) \\
z_j &= 0, && j \in \{r\} \cup J \setminus \tilde{J}
\end{aligned}$$

in addition to the original constraints are satisfied, which we refer to as System 1. We know that a solution to System 1 exists from assumption $s(I) > d(J \setminus \tilde{J})$ and facet condition 2.

2. Consider a point where only markets $j \in J \setminus \tilde{J}$ are satisfied, and constraints

$$\begin{aligned}
\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} &= d(J \setminus \tilde{J}) && \\
x_{ij} &= 0, && i \in V_1, j \in \tilde{J} \cup V_2 \setminus J \\
x_{ij} &\geq \epsilon, && i \in I, j \in J \setminus \tilde{J} \\
\sum_{j \in V_2 : (i,j) \in E} x_{ij} &\leq s_i - \epsilon, && i \in I \\
z_j &= 1, && j \in \tilde{J} \cup V_2 \setminus J \\
z_j &= 0, && j \in J \setminus \tilde{J}
\end{aligned}$$

in addition to the original constraints are satisfied, which we refer to as System 2. We know that a solution to System 2 exists from assumption $s(I) > d(J \setminus \tilde{J})$.

3. Consider a point where only markets $j \in J \setminus \tilde{J} \cup \{l\}$ are satisfied for some $l \in \tilde{J}$, and constraints

$$\begin{aligned}
\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} &= s(I) \\
\sum_{i \in V_1 \setminus I, j \in J:(i,j) \in E} x_{ij} &= d(J \setminus \tilde{J}) + d_l - s(I) \\
x_{ij} &= 0, && i \in V_1, j \in \tilde{J} \setminus \{l\} \cup V_2 \setminus J \\
x_{ij} &\geq \epsilon, && i \in V_1, j \in J \setminus \tilde{J} \cup \{l\} \\
\sum_{j \in V_2:(i,j) \in E} x_{ij} &\leq s_i - \epsilon, && i \in V_1 \setminus I \\
z_j &= 1, && j \in \tilde{J} \setminus \{l\} \cup V_2 \setminus J \\
z_j &= 0, && j \in J \setminus \tilde{J} \cup \{l\}
\end{aligned}$$

in addition to the original constraints are satisfied, which we refer to as System 3. We know that a solution to System 3 exists from facet conditions 2 and 3.

4. Consider a point where only markets $j \in J \setminus (\tilde{J} \cup \{j'\}) \cup \{l^*\}$ are satisfied for $l^* = \arg\max_{j \in \tilde{J}}\{d_j\}$ and some $j' \in J \setminus \tilde{J}$, and constraints

$$\begin{aligned}
\sum_{i \in I, j \in J:(i,j) \in E} x_{ij} &= s(I) \\
\sum_{i \in V_1 \setminus I, j \in J:(i,j) \in E} x_{ij} &= d(J \setminus \tilde{J}) + d_{l^*} - d_{j'} - s(I) \\
x_{ij} &= 0, && i \in V_1, j \in \{j'\} \cup \tilde{J} \setminus \{l^*\} \cup V_2 \setminus J \\
z_j &= 1, && j \in \{j'\} \cup \tilde{J} \setminus \{l^*\} \cup V_2 \setminus J \\
z_j &= 0, && j \in J \setminus (\tilde{J} \cup \{j'\}) \cup \{l^*\}
\end{aligned}$$

in addition to the original constraints are satisfied, which we refer to as System 4. We know that a solution to system 4 exists from facet conditions 2 and 4.

1. $\lambda_{ij} = u_j$, $j \in V_2 \setminus J$, $i \in V_1$, $(i,j) \in E$.

   Consider any solution to system 1 with any market $j = r \in V_2 \setminus J$ that is satisfied. Choose arbitrary suppliers $i, i' \in V_1$ such that $(i,j), (i',j) \in E$. Construct a new point by decreasing the flow on edge (i,j) by ϵ and increasing the flow on edge (i',j) by ϵ. Note that this point is also on the face defined by inequality (50). Thus,

$$\lambda_{ij} = u_j, j \in V_2 \setminus J, i \in V_1, (i,j) \in E.$$

2. $\lambda_{ij} = u_j$, $j \in J$, $i \in V_1 \setminus I$, $(i,j) \in E$.

   Consider any solution to system 3 with market $j \in J \setminus \tilde{J} \cup \{l\}$ satisfied for some $l \in \tilde{J}$. Choose arbitrary suppliers $i, i' \in V_1 \setminus I$ such that $(i,j), (i',j) \in E$. Construct a new point by decreasing the flow on edge (i,j) by ϵ and increasing the flow on edge (i',j) by ϵ. Note that this point is also on the face defined by inequality (50) since $i, i' \in V_1 \setminus I$. Thus,

$$\lambda_{ij} = u_j, j \in J \setminus \tilde{J} \cup \{l\}, i \in V_1 \setminus I, (i,j) \in E.$$

   Note that since we can use the above argument for any $l \in \tilde{J}$, we have $\lambda_{il} = u_l$ for all $l \in \tilde{J}$, $i \in V_1 \setminus I$, $(i,l) \in E$.

3. $\lambda_{ij} = \bar{\lambda} + u_j$, $j \in J$, $i \in I$, $(i,j) \in E$.

   Consider any solution to system 2. Choose arbitrary suppliers $i, i' \in I$ such that $(i,j), (i',j) \in E$ for $j \in J \setminus \tilde{J}$. Construct a new point by decreasing the flow on edge (i,j) by ϵ and increasing the flow on edge (i',j) by ϵ. Note that this point is also on the face defined by inequality (50). Thus,

   $$\lambda_{ij} = \lambda^1_j, j \in J \setminus \tilde{J}, i \in I, (i,j) \in E.$$

   Next we consider a solution to system 3 with $\epsilon = 0$. Choose arbitrary suppliers $i, i' \in I$ and market $j \in J \setminus \tilde{J}$ such that $(i,j), (i',j), (i,l), (i',l) \in E$. Construct a new point by decreasing the flow on edges (i,j), (i',l) by ϵ and increasing the flow on edges (i',j), (i,l) by ϵ. Note that this point is also on the face defined by inequality (50). Thus,

   $$-\lambda_{ij} + \lambda_{il} + \lambda_{i'j} - \lambda_{i'l} = -\lambda^1_j + \lambda_{il} + \lambda^1_j - \lambda_{i'l} = \lambda_{il} - \lambda_{i'l} = 0.$$

   Because l is any market in set $\tilde{J}$, $\lambda_{ij} = \lambda^1_j$, $j \in \tilde{J}$, $i \in I$, $(i,j) \in E$.

   Let $\lambda^1_j = \bar{\lambda}_j + u_j$, $j \in J$. Next we show that $\bar{\lambda}_j = \bar{\lambda}$, $j \in J$. We consider a solution to system 3 with $\epsilon = 0$. Choose any markets $j, j' \in J$, any suppliers $i \in V_1 \setminus I$, $i' \in I$ such that $(i,j), (i',j), (i,j'), (i',j') \in E$. Decrease flow on edges (i,j'), (i',j) by ϵ and increase flow on edges (i,j), (i',j') by ϵ. Thus,

   $$\lambda_{ij} - \lambda_{ij'} - \lambda_{i'j} + \lambda_{i'j'} = u_j - u_{j'} - \lambda^1_j + \lambda^1_{j'} = 0.$$

   By again using $\lambda^1_j = \bar{\lambda}_j + u_j$ and $\lambda^1_{j'} = \bar{\lambda}_{j'} + u_{j'}$, we obtain

   $$\bar{\lambda}_j = \bar{\lambda}_{j'} = \bar{\lambda}.$$

4. $\omega_j = u_j d_j$, $j \in V_2 \setminus \tilde{J}$.

   We rewrite (51) using the information obtained until now, and get

   $$\bar{\lambda} \sum_{i \in I, j \in J : (i,j) \in E} x_{ij} + \sum_{(i,j) \in E} u_j x_{ij} + \sum_{j \in V_2} \omega_j z_j = \hat{\lambda}. \tag{52}$$

   Consider any solution to system 1 with market $r \in V_2 \setminus J$ that is satisfied. Then we construct a new solution based on this solution where we set $z_r = 1$ and $x_{ir} = 0$ for all $i \in V_1$, $(i,r) \in E$ and all other variables remain the same. This is a solution to System 2. Thus this solution is also on the face defined by (50). We compare inequality (51) evaluated at these two solutions. Thus,

   $$u_r \sum_{i \in V_1 : (i,r) \in E} x_{ir} - \omega_r = 0.$$

   Because $\sum_{i \in V_1 : (i,r) \in E} x_{ir} = d_r$ we have $\omega_r = u_r d_r$, $r \in V_2 \setminus J$.

   Next we show that $\omega_j = u_j d_j$, $j \in J \setminus \tilde{J}$. First we consider a solution to system 3 where we choose $l = l^* = \arg\max_{j \in \tilde{J}} \{d_j\}$. This is a feasible choice due to facet condition 2. We evaluate (52) at this solution, and get

   $$\bar{\lambda}(s(I)) + \sum_{i \in V_1, j \in J \setminus \tilde{J} \cup \{l^*\} : (i,j) \in E} u_j x_{ij} + \sum_{j \in V_2 \setminus J \cup \tilde{J} \setminus \{l^*\}} \omega_j = \hat{\lambda}. \tag{53}$$

   Next we consider a solution to system 4 where some market $j' \in J \setminus \tilde{J}$ is rejected. We evaluate (52) at this solution, and obtain

   $$\bar{\lambda}(s(I)) + \sum_{i \in V_1, j \in J \setminus (\tilde{J} \cup \{j'\}) \cup \{l^*\} : (i,j) \in E} u_j x_{ij} + \sum_{j \in V_2 \setminus J \cup \tilde{J} \setminus \{l^*\}} \omega_j + w_{j'} = \hat{\lambda}. \tag{54}$$

   We subtract (54) from (53) and obtain $u_{j'} \sum_{i \in V_1 : (i,j') \in E} x_{ij'} - \omega_{j'} = 0$. Because $\sum_{i \in V_1 : (i,j') \in E} x_{ij'} = d_{j'}$ we have $\omega_{j'} = u_{j'} d_{j'}$, $j' \in J \setminus \tilde{J}$.

5. $\omega_j = \bar{\lambda}\alpha + u_j d_j$, $j \in \tilde{J}$.

   Consider any solution to system 3 with any market $l \in \tilde{J}$ that is satisfied. Then (51) reduces to

$$\bar{\lambda}(s(I)) + \sum_{i \in V_1, j \in J \setminus \tilde{J} \cup \{l\} : (i,j) \in E} u_j x_{ij} + \sum_{j \in V_2 \setminus J \cup \tilde{J} \setminus \{l\}} \omega_j = \hat{\lambda}. \tag{55}$$

   We also consider a solution to system 2 where market $l \in \tilde{J}$ is rejected. Then (51) reduces to

$$\bar{\lambda}(d(J \setminus \tilde{J}) + \sum_{i \in V_1, j \in J \setminus \tilde{J} : (i,j) \in E} u_j x_{ij} + \sum_{j \in V_2 \setminus J \cup \tilde{J}} \omega_j = \hat{\lambda}. \tag{56}$$

   We subtract (56) from (55) and obtain, $\bar{\lambda}(s(I) - d(J \setminus \tilde{J})) + u_l \sum_{i \in V_1 : (i,l) \in E} x_{il} - \omega_l = 0$. Since $s(I) - d(J \setminus \tilde{J}) = \alpha$ and $\sum_{i \in V_1 : (i,l) \in E} x_{il} = d_l$ we conclude that $\omega_l = \bar{\lambda}\alpha + u_l d_l$ for $l \in \tilde{J}$.

6. $\hat{\lambda} = \bar{\lambda}\left(s(I) + (|\tilde{J}| - 1)\alpha\right) + \sum_{j \in V_2} u_j d_j$.

   We rewrite (51), and get

$$\bar{\lambda}\left(\sum_{i \in I, j \in J : (i,j) \in E} x_{ij} + \sum_{j \in \tilde{J}} \alpha z_j\right) + \sum_{(i,j) \in E} u_j x_{ij} + \sum_{j \in V_2} u_j d_j z_j = \hat{\lambda}. \tag{57}$$

   Evaluating (57) at any point (x, z) feasible to $TPMC$ that is tight at inequality (50) gives

$$\bar{\lambda}\left(s(I) + (|\tilde{J}| - 1)\alpha\right) + \sum_{j \in V_2} u_j \left(\sum_{i \in V_1 : (i,j) \in E} x_{ij} + d_j z_j\right) = \hat{\lambda}.$$

   From the definition of $TPMC$ we have $\sum_{i \in V_1 : (i,j) \in E} x_{ij} + d_j z_j = d_j$ for all $j \in V_2$. Thus, $\hat{\lambda} = \bar{\lambda}\left(s(I) + (|\tilde{J}| - 1)\alpha\right) + \sum_{j \in V_2} u_j d_j$.

□

Even though Propositions 6 and 7 are general results for mixed-integer cover and knapsack sets $\mathcal{S}_1$ and $\mathcal{S}_2$, we observed that many of the facets for TPMC can be derived from the recursive application of these results.

Example 4. *(Continued.) Observe that inequalities* (28)*,* (29) *and* (30) *satisfy all the conditions given in Proposition 8 and inequality* (32) *satisfies all the conditions given in Proposition 10, and hence they are facets of conv(X).*

Finally, while the blossom inequalities (3) are strong for the case that $d_j \leq 2$ for all $j \in V_2$, they are not facet-defining for the general case of TPMC based on our experience with PORTA [7].

5 Computational Results

In this section we present our computational results for the TPMC problem. We conduct the experiments on an Intel Xeon x5650 Processor at 2.67GHz with 4GB RAM. We use IBM ILOG CPLEX 12.4 as the MIP solver. We test the TPMC problem for various settings of V_1 and V_2. There are 12 combinations of V_1 and V_2 as shown in Tables 1 and 2, in the first column. For each combination, we create 3 instances and report the averages. We observed that most instances of the TPMC problem are solved under a minute for each setting of V_1 and V_2. Therefore, we found "hard" instances by continually generating and solving instances until we were able to find 3 that were solved in at least 15 minutes under default CPLEX settings. Problem parameters are generated using a discrete uniform distribution with supply values $s_i \in [10, 20]$, demand values $d_j \in [10, 20]$, weights $w_{ij} \in [20, 50]$ and lost revenues $r_j \in [5000, 6000]$. In our computations, we impose a time limit of half an hour, and consider the following four algorithms:

(1) BB (Branch and Bound): TPMC formulation, (1a)-(1e) with no cuts,

(2) UC (User Cuts): TPMC formulation, (1a)-(1e) with only user cuts,

(3) CD (CPLEX Default Settings): TPMC formulation, (1a)-(1e) with default CPLEX cuts,

(4) UCD (User Cuts and CPLEX Default Settings): TPMC formulation, (1a)-(1e) with user cuts and default CPLEX cuts.

Table 1: Comparison of Algorithms BB and UC

$\|V_1\|, \|V_2\|$	**RGap**		**RCuts**		**EGap**		**ECuts**		**Time (unslvd)**		**B&C Nodes**	
	BB	UC	BB	UC	BB	UC	BB	UC	BB	UC	BB	UC
200,230	86.7%	86.7%	-	u8	0.8%	1.0%	-	u189	1800(3)	1800(3)	176305.3	107412
200,240	1.7%	1.7%	-	u4.3	0.7%	0.8%	-	u186	1800(3)	1800(3)	168384.7	91004.7
200,250	28.1%	28.1%	-	u4.7	0.2%	0.2%	-	u103.3	1800(3)	1800(3)	141624.3	89966.7
300,330	54.3%	54.3%	-	u4	0.5%	0.5%	-	u125.7	1800(3)	1800(3)	75466.3	56085.3
300,340	1.6%	1.6%	-	u6	0.9%	0.9%	-	u123.3	1800(3)	1800(3)	60125.7	50122.7
300,350	58%	29.5%	-	u4.3	0.4%	0.4%	-	u98.3	1800(3)	1800(3)	51141.3	44166.7
400,430	0.8%	0.7%	-	u4	0.4%	0.5%	-	u74.7	1800(3)	1800(3)	29565	32466.7
400,440	81%	54.3%	-	u4.7	0.3%	0.3%	-	u72.7	1800(3)	1800(3)	23177.7	26600
400,450	0.4%	0.4%	-	u4.7	0.2%	0.2%	-	u57.7	1800(3)	1800(3)	22593	23733.3
500,530	83.2%	83.2%	-	u7	0.5%	0.2%	-	u52	1800(3)	1690.5(2)	18152.3	16380
500,540	81.4%	81.4%	-	u4	0.4%	0.4%	-	u61.3	1800(3)	1800(3)	16115	17573.3
500,550	0.6%	0.6%	-	u6.3	0.4%	0.4%	-	u21.7	1800(3)	1800(3)	14767.7	18100
Average	43.5%	35.2%	-	u5.2	0.5%	0.5%	-	u96.4	1800(3)	1790.9(2.9)	66451.5	47801

Table 2: Comparison of Algorithms CD and UCD

$\|V_1\|, \|V_2\|$	**RGap**		**RCuts**		**EGap**		**ECuts**		**Time (unslvd)**		**B&C Nodes**	
	CD	UCD	CD	UCD	CD	UCD	CD	UCD	CD	UCD	CD	UCD
200,230	58.4%	58.1%	10.7	6,u2.3	0.2%	0.2%	568.7	569,u37.3	1342.3(1)	1249.8(1)	64767	57478.7
200,240	1.6%	1.6%	10	9.3,u2.7	0.4%	0.6%	307.3	219.7,u101.7	1420.9(2)	1333.9(2)	93360.7	74344.3
200,250	28.1%	1%	7.3	4,u3.3	0.1%	0.1%	573.3	412,u17.3	1265.4(1)	815.6(1)	53962.7	39750
300,330	0.8%	0.7%	12	8,u2	0.3%	0.3%	164.7	178.3,u50.3	1800(3)	1227.1(2)	72057.7	33551.3
300,340	1.6%	1.5%	13.3	7.3,u1.3	0.2%	0%	334.7	239,u16.3	1678.3(1)	1067.6(1)	49950	31914.3
300,350	29.5%	29.5%	4.7	5.7,u2	0.2%	0.2%	161	139.3,u53.3	1025.9(1)	901.7(1)	29653.7	23473.7
400,430	0.7%	0.7%	10.7	8.3,u2.7	0.2%	0.3%	114.7	105.7,u41	1800(3)	1234.9(2)	34852	21395.7
400,440	27.5%	27.8%	12	8,u3.3	0.2%	0.1%	128.7	167.7,u25	1800(3)	1216.6(2)	24729.3	16442.7
400,450	0.4%	0.4%	6	6.3,u3.7	0.1%	0.1%	133	173.3,u23	1800(3)	1800(3)	19398.7	22166.7
500,530	42.3%	42.3%	7.7	5.3,u3	0.2%	0.2%	76.3	86.3,u55	1800(3)	1661.7(2)	34852	21089.7
500,540	27.9%	27.9%	7	8.7,u1	0.4%	0.2%	58.3	151.3,u32.7	1800(3)	1800(3)	20709	18757.3
500,550	0.6%	0.6%	7	8.7,u3.7	0.4%	0.3%	26.3	68,u34.7	1800(3)	1800(3)	17482	18707.3
Average	18.3%	16%	9	7.1,u2.6	0.2%	0.2%	220.6	209.1,u40.6	1611.1(2.3)	1342.4(1.9)	42981.2	31589.3

In Tables 1 and 2, column **RGap** reports the average percentage integrality gap at the root node just before branching, which is $100 \times (zub - zrb)/zub$, where zub is the objective function value of the best integer solution obtained within time limit and zrb is the best lower bound obtained at the root node. Column **RCuts** reports the average number of cuts added at the root node. In column **EGap**, we report the average percentage end gap at termination output by CPLEX, which is $100 \times (zub - zbest)/zub$, where $zbest$ is the best lower bound available within time limit. Column **ECuts** reports the average number of cuts added after the problem is solved to optimality within the time limit. Column **Time (unslvd)** reports the average solution time in seconds and the number of unsolved instances in parentheses in cases where not all three instances are solved to optimality within time limit. We denote the user cuts by **u** and for the other cuts, i.e., cuts added by CPLEX we do not use a prefix. In column **B&C Nodes** we report the average number of branch-and-cut tree nodes explored. At the end of Tables 1 and 2 we give the averages of **RGap**, **RCuts**, **EGap**, **ECuts**, **Time (unslvd)** and **B&C Nodes**, respectively. For each value in the tables we report the numbers rounded to the first decimal place.

User cuts are generated every 10000 B&C nodes. For the variable upper bound inequalities (14) we add a violated inequality if $s_i < d_j$, $i \in V_1$, $j \in V_2$, $(i,j) \in E$ and $\bar{x}_{ij} > s_i(1 - \bar{z}_j)$. Recall that inequalities (19) are related to the weight inequalities for 0/1 knapsack problems. The exact separation of weight inequalities involves solving 0/1 knapsack problems. Weismantel, Kaparis and Letchford give exact pseudo-polynomial separation algorithms for weight inequalities [15, 26]. The optimization problems for finding the most violated inequalities (15) and (27) involve nonlinear objectives and constraints that resemble knapsack constraints. Thus, we use a heuristic separation for inequalities (15), (19) and (27). Let $(\bar{x}, \bar{z})$ be a fractional point. The heuristic for finding a violated inequality (15) takes $(\bar{x}, \bar{z})$ and selects sets I and J simultaneously. Set J includes a market with fractional $\bar{z}$ value, and other markets that receive demand from the same suppliers as the market with fractional $\bar{z}$. All the suppliers that do not send demand to markets in set J are placed in set I. More details for this heuristic can be found in Algorithm 2. The heuristic for finding a violated inequality (19) uses the type-I base inequalities (15), and adds the smallest p coefficients of the z variables that exceed the right-hand side, β_0 to obtain the cover $\tilde{J}$. For all the instances in Tables 1 and 2 the violated inequality (15) (i.e. type-I base inequality) found by the heuristic separation has the coefficients of all the z variables equal to the right-hand side, β_0. It is easy to see that if at least two coefficients of z variables are not strictly less than the right-hand side, β_0 in a given type-I base inequality, the new inequality of type (19) cannot be a facet of conv(X). Therefore, for the given instances no violated inequality of type (19) is generated. Note that our separation heuristic for inequality (19) is different than that of [14, 15, 26] because our choice of set J also impacts the continuous term $t = \sum_{i \in I, j \in J:(i,j) \in E} x_{ij}$, which is not present in their setting. We have three heuristics for finding a violated inequality (27). Two of them uses the supply constraints as a base inequality for a certain choice of J (i.e. $\sum_{j \in J:(i,j) \in E} x_{ij} \leq s_i$ for $i \in V_1$ and $J \subseteq V_2$), one of which finds an inequality with $|\tilde{J}| = 1$ and the other finds an inequality with $|\tilde{J}| = |J| - 1$. The details for these heuristics are given by Algorithms 3 and 4, respectively. The third heuristic uses $\sum_{i \in V_1, j \in V_2:(i,j) \in E} x_{ij} \leq s(V_1)$ as a base inequality and finds a violated inequality with $\tilde{J}$ that includes the rejected markets and markets that have fractional $\bar{z}$ values. More details on this heuristic is given in Algorithm 5.

Table 1 compares the performance of algorithms BB and UC, to isolate the reduction in the root gap (8.3%) using our inequalities. Similarly, Table 2 compares the performance of the algorithms CD and UCD, to illustrate the marginal benefit of incorporating our inequalities into default CPLEX, where we observe a reduction in the root gap of 2.3%. Due to the reduction in the integrality gap the number of branch-and-cut nodes is almost always lower for UC and UCD compared to BB and CD, respectively. The solution times and the number of unsolved instances are slightly lower for algorithms that include our proposed inequalities. However, the end gap is not lower for algorithms UC and UCD compared to BB and CD, respectively. In conclusion, our preliminary computational results show that our proposed inequalities does have some positive effects, but the separation heuristics need to be significantly improved.

Algorithm 2 Heuristic separation for inequalities (15)

```
                         Input: $(\bar{x}, \bar{z})$
    Output: Sets $I$ and $J$ and the corresponding cut for each fractional $\bar{z}$
$I \leftarrow V_1$
$s(V_1 \setminus I) = 0$
$d(J) = 0$
$\mathit{tempSupplies} \leftarrow \emptyset$
$\mathit{tempDemand} \leftarrow \emptyset$
$\mathit{switch} = 0$
for all the fractional variables $\bar{z}_j$ do
    $\mathit{tempDemand} = \{j\}$
    $J = \{j\}$
    while $|\mathit{tempDemand}| \geq 1$ or $|\mathit{tempSupplies}| \geq 1$ do
        if $\mathit{switch} = 0$ then
            for all the supplies $i$ that have an edge to all nodes $j$ in $\mathit{tempDemand}$ do
                if $\bar{x}_{ij} > 0$ then
                    $I \leftarrow I \setminus \{i\}$
                    $s(V_1 \setminus I) \leftarrow s(V_1 \setminus I) + s_i$
                    $\mathit{tempSupplies} \leftarrow \mathit{tempSupplies} \cup \{i\}$
                end if
            end for
            $\mathit{switch} = 1$
            $\mathit{tempDemand} \leftarrow \emptyset$
        end if
        if $\mathit{switch} = 1$ then
            for all demand $j$ that have an edge to all nodes $i$ in $\mathit{tempSupplies}$ do
                if $\bar{x}_{ij} > 0$ then
                    $J \leftarrow J \cup \{j\}$
                    $d(J) \leftarrow d(J) + d_j$
                    $\mathit{tempDemand} \leftarrow \mathit{tempDemand} \cup \{j\}$
                end if
            end for
            $\mathit{switch} = 0$
            $\mathit{tempSupplies} \leftarrow \emptyset$
        end if
    end while
    if $d(J) > s(V_1 \setminus I)$ and $|J| \geq 2$ and $\max_{j \in J}\{d_j\} > d(J) - s(V_1 \setminus I)$ then
        if $\sum_{i \in I, j \in J:(i,j) \in E} \bar{x}_{ij} + \sum_{j \in J}(\min\{d(J) - s(V_1 \setminus I), d_j\})\bar{z}_j < d(J) - s(V_1 \setminus I)$ then
            add inequality (15) with $I$ and $J$
        end if
    end if
    $I \leftarrow V_1$
    $s(V_1 \setminus I) = 0$
    $d(J) = 0$
    $\mathit{switch} = 0$
end for
```

Algorithm 3 Heuristic separation for inequalities (27) that finds $|\tilde{J}| = 1$

Input: $(\bar{x}, \bar{z})$

Output: Sets I, J, $\tilde{J}$ and the corresponding cut for each fractional $\bar{z}$

```
$I, J, \tilde{J} \leftarrow \emptyset$
$d(J \setminus \tilde{J}) = 0$
$\alpha = 0$
for all the fractional variables $\bar{z}_j$ do
    $J \leftarrow \{j\}$, $\tilde{J} \leftarrow \{j\}$
    for all i such that $\bar{x}_{ij} > 0$ do
        $I \leftarrow \{i\}$
        for all $j^* \neq j$ do
            if $\bar{x}_{ij^*} = d_{j^*}$ then
                $J \leftarrow J \cup \{j^*\}$
                $d(J \setminus \tilde{J}) = d(J \setminus \tilde{J}) + d_{j^*}$
            end if
        end for
        $\alpha = s_i - d(J \setminus \tilde{J})$
        if $|J| \geq 2$ and $\sum_{j \in J:(i,j) \in E} \bar{x}_{ij} + \alpha \bar{z}_j > s_i$ then
            add inequality (27) with $I$, $J$, $\tilde{J}$ and $\alpha$
        end if
        $I \leftarrow \emptyset$, $J \leftarrow \{j\}$, $d(J \setminus \tilde{J}) = 0$
    end for
end for
```

Algorithm 4 Heuristic separation for inequalities (27) that finds $|\tilde{J}| = |J| - 1$

Input: $(\bar{x}, \bar{z})$

Output: Sets I, J, $\tilde{J}$ and the corresponding cut for each fractional $\bar{z}$

```
$J_0 \leftarrow \{j \in V_2 : \bar{z}_j = 0\}$
$J_1 \leftarrow \{j \in V_2 : \bar{z}_j = 1\}$
$I \leftarrow \emptyset$
$\alpha = 0$
$maxdj\tilde{J} = \max_{j \in J_1}\{d_j\}$
for all the fractional variables $\bar{z}_j$ do
    $\tilde{J} \leftarrow J_1 \cup \{j\}$
    if $maxdj\tilde{J} < d_j$ then
        $maxdj\tilde{J} = d_j$
    end if
    for all $i \in V_1$ do
        for all $j' \in J_0$ do
            if $\bar{x}_{ij'} > 0$ and $s_i > d_{j'}$ and $s_i - d_{j'} < maxdj\tilde{J}$ then
                $\alpha = s_i - d_{j'}$
                $I \leftarrow \{i\}$, $J \leftarrow \tilde{J} \cup \{j'\}$
                if $\sum_{j \in J:(i,j) \in E} \bar{x}_{ij} + \alpha \sum_{j \in \tilde{J}} \bar{z}_j > s_i + (|\tilde{J}| - 1)\alpha$ then
                    add inequality (27) with $I$, $J$, $\tilde{J}$ and $\alpha$
                end if
            end if
        end for
    end for
end for
```

Algorithm 5 Heuristic separation for inequalities (27) that finds general $\tilde{J}$

Input: $(\bar{x}, \bar{z})$

Output: Sets I, J, $\tilde{J}$ and the corresponding cut

```
$J_f \leftarrow \{j \in V_2 : 0 < \bar{z}_j < 1\}$
$J_1 \leftarrow \{j \in V_2 : \bar{z}_j = 1\}$
$I \leftarrow V_1$
$J \leftarrow V_2$
$\tilde{J} \leftarrow J_f \cup J_1$
$\alpha = 0$
$maxdj\tilde{J} = \max_{j \in J_1 \cup J_f}\{d_j\}$
if $s(V_1) - d(V_2 \setminus \tilde{J}) > 0$ and $s(V_1) - d(V_2 \setminus \tilde{J}) < maxdj\tilde{J}$ then
    $\alpha = s(V_1) - d(V_2 \setminus \tilde{J})$
    if $\sum_{i \in V_1, j \in V_2:(i,j) \in E} \bar{x}_{ij} + \alpha \sum_{j \in \tilde{J}} \bar{z}_j > s(V_1) + (|\tilde{J}| - 1)\alpha$ then
        add inequality (27) with $I$, $J$, $\tilde{J}$ and $\alpha$
    end if
end if
```

Acknowledgements. We thank László Végh for his suggestions for the reduction used in the proof of Proposition 2. Santanu S. Dey gratefully acknowledges the support of the AIR Force Office of Scientific Research grant FA9550-12-1-0154. Pelin Damcı-Kurt and Simge Küçükyavuz are supported, in part, by NSF-CMMI grant 1055668, and an allocation of computing time from the Ohio Supercomputer Center.

A Proofs of Section 2

In this section, we assume that all data are integral.

Proposition 1. *The decision version of TPMC is NP-complete even when:*

1. $s_i = 1$ *for all* $i \in V_1$*,* $d_j = d \geq 3$ *for all* $j \in V_2$*,* $w_{ij} = 0$ *for all* $(i,j) \in E$ *and* $r_j = 1$ *for all* $j \in V_2$*.*
2. $|V_1| = 1$ *and* $w_{ij} = 0$ *for all* $(i,j) \in E$*.*

Proof. Since TPMC is a mixed integer linear problem with rational data, it is in NP. We present two reductions to verify the two parts of this result.

1. We reduce every instance of the Exact 3-Cover (E3C) problem to an instance of TPMC. An instance of E3C is given as: Let B be a base set where $|B| = 3q$ for some $q \in \mathbb{N}$. Let C be a collection of subsets of B where each subset is of cardinality 3. Does there exist $D \subseteq C$ such that $|D| = q$ and the union of sets in D covers every element of B?

   It is well-known that E3C is strongly NP-complete [10]. Given an instance of E3C, we construct an instance of TPMC as follows: For every element in B, we construct a node in V_1 and for every element in C we construct a node in V_2. For $i \in V_1$, we use the notation $B(i)$ to denote the element of B corresponding to node i. Similarly, for $j \in V_2$, we let $C(j)$ denote the element of C corresponding to node j. We add an edge between $i \in V_1$ and $j \in V_2$ if $B(i) \in C(j)$. Let $s_i = 1$ for all $i \in V_1$. Let $d_j = 3$ for all $j \in V_2$. Let $w_{ij} = 0$ for all $(i,j) \in E$. Let $r_j = 1$ for all $j \in V_2$.

   Next, we verify that there exists $D \subseteq C$ such that $|D| = q$ and D covers every element of B if and only if there exists a feasible solution to TPMC with a cost at most $|C| - q$. Note that the size of the TPMC instance is polynomially bounded by the size of the E3C instance.

   (⇒) Assume that there exists $\{D_1, \ldots, D_q\} =: D \subseteq C$ such that D covers every element of B. Let $D(u)$ represent the element of D (and therefore of C) that contains $u \in B$. Now construct the solution

   $$\begin{aligned} \hat{x}_{ij} &= \begin{cases} 1 & \text{if } B(i) = u \text{ and } C(j) = D(u) \\ 0 & \text{otherwise.} \end{cases} \\ \hat{z}_j &= \begin{cases} 1 & \text{if } C(j) \notin \{D_1, \ldots, D_q\} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

   It is straightforward to verify that $(\hat{x}, \hat{z})$ satisfies all the constraints of TPMC and $\sum_{(i,j)\in E} w_{ij}\hat{x}_{ij} + \sum_{j \in V_2} r_j \hat{z}_j = |C| - q$.
   (⇐) Consider a solution $(\hat{x}, \hat{z})$ of TPMC such that

   $$\sum_{(i,j)\in E} w_{ij}\hat{x}_{ij} + \sum_{j\in V_2} r_j \hat{z}_j = \sum_{j \in V_2} \hat{z}_j \leq |C| - q. \tag{58}$$

   Since there are $3q$ supply nodes, each with a capacity of 1, the demand of at most q nodes can be satisfied. Therefore, from (58), we conclude that there are exactly q nodes whose demands are satisfied. Let $D = \{C(j) \mid \sum_{i \in V_1} \hat{z}_j = 0\}$. Clearly, $|D| = q$ and D covers every element of B. As a result TPMC is strongly NP-complete.

2. We reduce every instance of the Subset Sum (SS) problem to an instance of TPMC. An instance of SS is given as: Let A be a finite set, $a_n \in \mathbb{Z}^+$ be the size of each element $n \in A$ and B be a positive integer. Does there exist a subset $A^{'} \subseteq A$ such that the sum of the sizes of the elements in $A^{'}$ is exactly B?

   It is well-known that SS is NP-complete [10]. Given an instance of SS, we construct an instance of TPMC as follows: We construct a single node $V_1 = \{1\}$ and for every element in A we construct a node in V_2. We add all the edges between the nodes in V_1 and V_2. Let the single supply be $s_1 = B$. Let demand of market j be $d_j = a_j$ for all $j \in V_2 = A$. Finally, let the unit shipping costs and lost revenues be $w_{1j} = 0$ and $r_j = d_j$, for $j \in V_2$.

   Next, we verify that there exists subset $A^{'} \subseteq A$ such that the sum of the sizes of the elements in $A^{'}$ is exactly B if and only if there exists a feasible solution to TPMC with a cost of at most $\sum_{n \in A} a_n - B$. Note that the size of the TPMC instance is polynomially bounded by the size of the SS instance.

   (⇒) Assume that there exists a subset $A^{'} \subseteq A$ such that the sum of the sizes of the elements in $A^{'}$ is exactly B. Now construct the solution

$$\hat{x}_{1j} = \begin{cases} a_j & \text{if } j \in A^{'} \\ 0 & \text{otherwise,} \end{cases}$$

$$\hat{z}_j = \begin{cases} 1 & \text{if } j \notin A^{'} \\ 0 & \text{otherwise.} \end{cases}$$

   It is straightforward to verify that $(\hat{x}, \hat{z})$ satisfies all the constraints of TPMC and $\sum_{(i,j) \in E} w_{ij}\hat{x}_{ij} + \sum_{j \in V_2} r_j \hat{z}_j = \sum_{n \in A} a_n - B$.
   (⇐) Consider a solution $(\hat{x}, \hat{z})$ of TPMC such that

$$\sum_{(i,j) \in E} w_{ij}\hat{x}_{ij} + \sum_{j \in V_2} r_j \hat{z}_j = \sum_{j \in V_2} a_j \hat{z}_j \leq \sum_{n \in A} a_n - B. \tag{59}$$

   The total demand satisfied by any feasible solution is at most B since we cannot satisfy more than the supply. Furthermore, since each edge has a cost per unit flow of 0, we have that $\sum_{(i,j) \in E} w_{ij}\hat{x}_{ij} = 0$. Therefore, from (59), the total demand satisfied must equal B. Let the set of satisfied demand nodes be $A^{'} = \{j \in A : \hat{z}_j = 0\}$, so we have $\sum_{n \in A^{'}} a_n = B$.

□

Proposition 2. *Suppose that $d_j \leq 2$ for all $j \in V_2$. Then there exists a polynomial-time algorithm to solve TPMC.*

Proof. We can convert a given instance of TPMC with $d_j \leq 2$ for all $j \in V_2$ and arbitrary supplies into an equivalent instance with all supplies equal to 1. Observe that in any feasible solution since $d_j \leq 2$ for all $j \in V_2$, no supply can send more than $2|V_2|$ units. Therefore, if $s_i > 1$ for some $i \in V_1$, then we construct an updated instance by replacing supply node $i \in V_1$ with $\min\{s_i, 2|V_2|\}$ supply nodes with a capacity of 1 and unit shipping cost to demand node j of w_{ij} for $(i, j) \in E$. Notice that the resulting instance is polynomial in the size of the original problem. Therefore from now on we assume that $s_i = 1$ for all $i \in V_1$.

We show that TPMC with $d_j \leq 2$ for all $j \in V_2$ is equivalent to the problem of finding a minimum weight perfect matching on a suitably constructed general graph $G' = (V', E')$.

1. For each $i \in V_1$, we add a corresponding $i \in V'$ and similarly for each $j \in V_2$ we add $j \in V'$. (When we use notation $V_1 \subseteq V'$, V_1 represents the vertices of V' corresponding to the vertices V_1 of G; similary for V_2.)

2. Let $M_1 = \{j \in V_2 : d_j = 1\}$ and $M_2 = \{j \in V_2 : d_j = 2\}$.

3. For each demand node $j \in V_2$, add a node $j' \in V'$ (note that this is in addition to $j \in V'$ for $j \in V_2$ as described in 1.). Add an edge $(j, j') \in E'$ with a cost of r_j. We refer to the set of nodes $j' \in V'$ corresponding to $j \in M_1$ as M_1'. (We define M_2' similarly.)

4. For each $i \in V_1$ such that $(i, j) \in E$ and $j \in M_1$, add the edge $(i, j) \in E'$ with cost of w_{ij}.

5. For each $i \in V_1$ such that $(i, j) \in E$ and $j \in M_2$, add two nodes, $ij1, ij2 \in V'$. Add edges $(i, ij1), (ij1, ij2), (ij2, j), (ij2, j') \in E'$ with costs $\frac{w_{ij}}{2}$, 0, $\frac{w_{ij}}{2}$, $\frac{w_{ij}}{2}$ respectively.

6. If $|V_1|$ is odd, we add an additional artificial node $\{0\}$ to V'. Let $V_1' \subseteq V'$ be defined as $V_1' := V_1 \cup M_1'$ if $|V_1|$ is even and $V_1' := V_1 \cup M_1' \cup \{0\}$ if $|V_1|$ is odd.

7. For all $u, v \in V_1'$ such that $u \neq v$, add an edge $(u, v) \in E'$ with a cost of 0. Therefore, the subgraph induced by the nodes in V_1' is a complete graph/clique.

Note that the size of the resulting minimum weight perfect matching problem is polynomial in the size of the TPMC problem. Figure 2 illustrates the original graph of a TPMC instance, where the demand of market A is 2 and that of market B is 1. Figure 3 illustrates the new graph. (The clique induced by $V_1 \cup \{B'\} \cup \{0\}$ is not shown.)

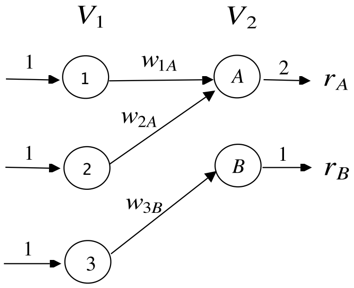


Figure 2: A TPMC instance

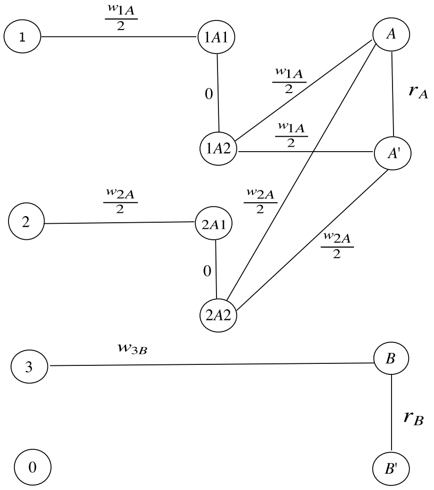


Figure 3: Construction of G'

We next show that any solution to the TPMC problem corresponds to a perfect matching in $G' = (V', E')$. Consider a feasible solution (x, z) to the TPMC problem. If $z_j = 0$ for $j \in M_1$, then there exists exactly one supply node i such that $x_{ij} = 1$. For constructing a matching in G', we choose edge (i, j), where $i \in V_1$ and $j \in M_1$, thereby covering nodes i and j in V'. If $z_j = 0$ for $j \in M_2$, then there exists two supply nodes i_1 and $i_2 \in V_1$ such that $x_{i_1 j} = x_{i_2 j} = 1$. For constructing a matching in G', without loss of generality, we choose edges $(i_1, i_1 j1)$, $(i_1 j2, j)$, $(i_2, i_2 j1)$ and $(i_2 j2, j')$, thereby covering nodes $i_1, i_2, i_1 j1, i_1 j2, i_2 j1, i_2 j2, j, j'$. If $z_j = 1$ for $j \in V_2$, then no supply node i sends demand to j and for the matching we choose edge (j, j'), hence covering nodes j and j' in V'. Moreover if $j \in M_2$, we choose edges $(ij1, ij2)$ for all $(i, j) \in E, i \in V_1$ in the matching and therefore the nodes $ij1$, $ij2$, j, j' are also covered. Hence whether $z_j = 1$ or $z_j = 0$, and whether $j \in M_1$ or $j \in M_2$, the nodes in V_2, M_2', and the nodes $ij1, ij2$ for all $(i, j) \in E, j \in M_2$ are always covered by the edges in the matching we have selected thus far. To complete the proof we show how nodes $i \in V_1'$ are also covered in all cases by extending the matching we have until now.

Let $\bar{M}_1 = \{j \in M_1 : z_j = 0\}$, $\bar{M}_2 = \{j \in M_2 : z_j = 0\}$ and $\bar{V}_1 = \{i \in V_1 : x_{ij} = 1\}$. In other words, set $\bar{M}_1$ represents the nodes $j \in M_1$ whose unit demands are satisfied, set $\bar{M}_2$ represents the nodes $j \in M_2$ whose demands, $d_j = 2$, are satisfied, and set $\bar{V}_1$ represents the set of supply nodes that send demand. Observe that the nodes in $\bar{V}_1$ are also covered in the matching constructed thus far. However, the nodes $j \in V_1 \setminus \bar{V}_1$, and $j' \in M_1'$ for $j \in \bar{M}_1$ and $\{0\}$ (if it exists) are not yet covered. Note that $|\bar{V}_1| = |\bar{M}_1| + 2|\bar{M}_2|$. We consider two cases.

1. $|V_1|$ is even. If $|\bar{V}_1|$ is even, then $|V_1| - |\bar{V}_1|$ and $|\bar{M}_1|$ are even. If $|\bar{V}_1|$ is odd, then $|V_1| - |\bar{V}_1|$ and $|\bar{M}_1|$ are odd. Therefore, $|V_1| - |\bar{V}_1| + |\bar{M}_1|$ is always even. Thus, we can cover all nodes $i \in V_1 \setminus \bar{V}_1$ and $j' \in M_1'$ for $j \in \bar{M}_1$ using $\frac{|V_1|-|\bar{V}_1|+|\bar{M}_1|}{2}$ many disjoint edges that exist between them (recall that the subgraph induced by the nodes $i \in V_1'$ form a complete graph).

2. $|V_1|$ is odd. If $|\bar{V}_1|$ is even, then $|V_1| - |\bar{V}_1|$ is odd and $|\bar{M}_1|$ is even. If $|\bar{V}_1|$ is odd, then $|V_1| - |\bar{V}_1|$ is even and $|\bar{M}_1|$ is odd. Therefore, $|V_1| - |\bar{V}_1| + |\bar{M}_1|$ is always odd. Recall that when $|V_1|$ is odd we have an additional dummy node $\{0\}$ that forms a fully connected graph with nodes $i \in V_1$ and $j \in M_1'$. Therefore, we obtain an even number of nodes that need to be covered by choosing $\frac{|V_1|-|\bar{V}_1|+|\bar{M}_1|+1}{2}$ disjoint edges.

So we have verified that given any solution to the TPMC problem we can find a perfect matching in $G' = (V', E')$. Moreover, it is straightforward to check that the cost of this matching is equal to the cost of the given solution to TPMC.

Next we show that any solution to the perfect matching in $G' = (V', E')$ corresponds to a feasible solution of the TPMC problem. Let P be the set of edges that are in the perfect matching. If edge $(j', j) \in P$ for $j' \in M_1'$, $j \in M_1$ (or $j \in M_2$, $j' \in M_2'$), then set $z_j = 1$. Set all remaining $z_j = 0$. If edge $(i, j) \in P$ for $j \in M_1$, then we set $x_{ij} = 1$. If edge $(i, ij1) \in P$, then set $x_{ij} = 1$. Set all remaining $x_{ij} = 0$. Note that due to the construction of graph G', a supply node $i \in V_1$ can send at most 1 unit of demand. Similarly for $j \in M_1$ a single edge that has j as one of its endpoints will be selected. For $j' \in M_2'$, $j \in M_2$ if edge $(j, j') \in P$, then for any $i \in V_1$ edges $(ij2, j), (ij2, j') \notin P$. However, if edge $(j, j') \notin P$ then for a perfect matching there must exist exactly two $i_1, i_2 \in V_1$ such that $(i_1 j2, j), (i_2 j2, j') \in P$. Therefore, for any $j \in M_2$ either the demand is fully satisfied or it is rejected altogether. Finally, it is easy to see that the cost of the solution to the TPMC problem is equivalent to the cost of the corresponding perfect matching in G', completing the proof.

□

References

[1] K. Aardal, Y. Pochet, and L. A. Wolsey. Capacitated facility location: Valid inequalities and facets. *Math. of Oper. Res.*, 20:185–197, 1992.

[2] E. H. Aghezzaf and L. A. Wolsey. Lot-sizing polyhedra with a cardinality constraint. *Oper. Res. Lett.*, 11:13–18, 1992.

[3] J. Aráoz, W. H. Cunningham, J. Edmonds, and J. Green-Krótki. Reductions to 1-matching polyhedra. *Networks*, 13:455–473, 1983.

[4] A. Atamtürk. Flow pack facets of the single node fixed–charge flow polytope. *Oper. Res. Lett.*, 29:107–114, 2001.

[5] A. Atamtürk. Cover and pack inequalities for (mixed) integer programming. *Annals OR*, 139(1):21–38, 2005.

[6] A. Berger, V. Bonifaci, F. Grandoni, and G. Schäfer. Budgeted matching and budgeted matroid intersection via the gasoline puzzle. *Math. Program.*, 128:355–372, 2011.

[7] T. Christof and A. Löbel. PORTA - a polyhedron transformation algorithm. *Version 1.4.1*, 2008.

[8] F. A. Chudak and D. P. Williamson. Improved approximation algorithms for capacitated facility location problems. *Math. Program.*, 102(2):207–222, Mar. 2005.

[9] J. Edmonds and E. L. Johnson. Matching: a well-solved class of integer linear programs. *Combinatorial structures and their applications*, pages 89–92, 1970.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York, 1979.

[11] J. Geunes, R. Levi, H. E. Romeijn, and D. B. Shmoys. Approximation algorithms for supply chain planning and logistics problems with market choice. *Math. Program.*, 130:85–106, 2009.

[12] M. Grötschel and O. Holland. A cutting plane algorithm for minimum perfect 2-matching. *Computing*, 39:327–344, 1987.

[13] Z. Gu, G. L. Nemhauser, and M. W. P. Savelsbergh. Lifted flow cover inequalities for mixed 0–1 integer programs. *Math. Program.*, 85:439–467, 1999.

[14] C. Helmberg and R. Weismantel. Cutting plane algorithms for semidefinite relaxations. In P. Pardalos and H. Wolkowicz, editors, *Topics in Semidefinite and Interior-Point Methods. Fields Institute Communications Series Vol. 18, AMS*, pages 197–213, 1997.

[15] K. Kaparis and A. N. Letchford. Separation algorithms for 0-1 knapsack polytopes. *Math. Program.*, 124(1-2):69–91, 2010.

[16] P. Kleinschmidt and H. Schannath. A strongly polynomial algorithm for the transportation problem. *Math. Program.*, 68:1–13, 1995.

[17] A. N. Letchford, G. Reinelt, and D. O. Theis. Odd minimum cut sets and b-matchings revisited. *SIAM J. Discrete Math.*, 22(4):1480–1487, 2008.

[18] R. Levi, J. Geunes, H. E. Romeijn, and D. B. Shmoys. Inventory and facility location models with market selection. *Proceedings of the 12th IPCO*, pages 111–124, 2005.

[19] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, Inc., New York, NY, 1988.

[20] M. W. Padberg and M. R. Rao. Odd minimum cut-sets and b-matchings. *Math. Oper. Res.*, 7:67–80, 1982.

[21] M. W. Padberg, T. J. Van Roy, and L. A. Wolsey. Valid linear inequalities for fixed charge problems. *Oper. Res.*, 33:842–861, 1985.

[22] J. Plesník. Constrained weighted matchings and edge coverings in graphs. *Discrete Applied Mathematics*, 92:229–241, 1999.

[23] A. Schrijver. *Combinatorial Optimization.* Springer, 2003.

[24] J. I. A. Stallaert. The complementary class of generalized flow cover inequalities. *Disc. Appl. Math.*, 77:73–80, 1997.

[25] W. Van den Heuvel, O. E. Kundakçıoğlu, J. Geunes, H. E. Romeijn, T. C. Sharkey, and A. P. M. Wagelmans. Integrated market selection and production planning: complexity and solution approaches. *Math. Program.*, 134:395–424, 2012.

[26] R. Weismantel. On the 0/1 knapsack polytope. *Math. Program.*, 77:49–68, 1997.

[27] L. A. Wolsey. Submodularity and valid inequalities in capacitated fixed charge networks. *Oper. Res. Lett.*, 8:119–124, 1989.

A scenario decomposition algorithm for 0-1 stochastic programs

Shabbir Ahmed
School of Industrial & Systems Engineering,
Georgia Institute of Technology, 765 Ferst Drive, Atlanta, GA 30332.

June 26, 2013

Abstract

We propose a scenario decomposition algorithm for stochastic 0-1 programs. The algorithm recovers an optimal solution by iteratively exploring and cutting-off candidate solutions obtained from solving scenario subproblems. The scheme is applicable to quite general problem structures and can be implemented in a distributed framework. Illustrative computational results on standard two-stage stochastic integer programming and nonlinear stochastic integer programming test problems are presented.

1 Introduction

We consider stochastic programs of the following form

$$\min\{\mathbb{E}[f(x,\boldsymbol{\xi})] : \; x \in X \subseteq \{0,1\}^n\}, \tag{1}$$

where $\boldsymbol{\xi}$ is a random vector with support Ξ and known distribution P, and the expectation in (1) is with respect to P. An important example of (1) is the class of two-stage stochastic programs with 0-1 first stage variables with

$$f(x,\xi) = c^\top x + \min\{\phi(y(\xi),\xi) : \; y(\xi) \in Y(\xi,x)\}$$

where, for realization ξ of $\boldsymbol{\xi}$, $y(\xi)$ is the second stage decision vector, $\phi(\cdot,\xi)$ is the second stage objective function, and $Y(\xi,x)$ is the second stage constraint system depending on the first stage decision vector x. We assume that the random vector $\boldsymbol{\xi}$ has a finite support, i.e. $\Xi = \{\xi^1,\ldots,\xi^N\}$, where each ξ^i for $i \in \{1,\ldots,N\}$ is referred to as a scenario. We can then rewrite (1) as

$$\min\left\{\sum_{i=1}^N f_i(x) : \; x \in X \subseteq \{0,1\}^n\right\}, \tag{2}$$

where $f_i(x) = p_i f(x,\xi^i)$ and p_i is the probability mass associated with scenario i.

A popular approach for solving (2) is the so-called scenario or dual decomposition method. By making copies of the decision variables x, problem (2) can be reformulated as

$$\min\left\{\sum_{i=1}^N f_i(x^i) : \; x^i \in X \; \forall\, i, \sum_{i=1}^N A_i x^i = h\right\}$$

where the equations $\sum_{i=1}^N A_i x^i = h$ enforce the nonanticipativity constraints $x^1 = \cdots = x^N$. The Lagrangian dual problem by dualizing these nonanticipativity constraints take the form

$$\max_\lambda \left\{ v(\lambda) := \sum_{i=1}^N \min\left\{f_i(x^i) + \lambda^\top A_i x^i : \; x^i \in X\right\} - \lambda^\top h \right\}, \tag{3}$$

where λ is the dual vector. For any λ the value $v(\lambda)$ provides a lower bound to (2) which can be evaluated by separately solving subproblems corresponding to each scenario. The best possible such bound is given by the optimal value of the dual problem (3) which is a large-scale and nonsmooth, albeit convex, optimization problem and is in general quite challenging. Even with an optimal dual solution, owing to the presence of integer decision variables, there typically remains a duality gap, and moreover a primal feasible solution (one that satisfies the original and the nonanticipativity constraints) is not readily available. Caroe and Schultz [5] proposed a branch and bound algorithm where the dual problem (3) is used to generate lower bounds, the integer feasible solutions to the scenario subproblems are "averaged" to generate a possibly fractional primal solutions, and the feasible region is successively partitioned by branching on the fractional variables to enforce integrality. Alonso-Ayuso et.al. [2] propose solving scenario subproblems (with $\lambda = 0$) with a branch-and-fix approach that coordinates the selection of the branching nodes and branching variables in order to enforce the nonanticipativity constraints. A number of heuristics [6, 9, 15] have also been designed based on the scenario-wise decomposability of the dual problem (3). In these approaches the dual problem is augmented by an additional penalty to achieve consensus among the scenario subproblem solutions, and the penalty parameters are iteratively updated to try to achieve primal feasible solutions. No convergence guarantees are available for such approaches, but they are reported to have good performance in various applications.

In this paper we propose a scenario decomposition algorithm for (2) that proceeds by solving scenario subproblems to generate candidate solutions and lower bounds, evaluating candidate solutions to get upper bounds, and cutting off candidate solutions from the scenario subproblems to get improved lower bounds and new candidate solutions. The scheme is applicable for quite general problem structures and, since unlike existing exact scenario decomposition methods requires little coordination among scenario subproblems, can be implemented easily in a distributed framework. The remainder of the paper is organized as follows: in Section 2 we describe the proposed scenario decomposition method along with some implementation issues, in Section 3 we discuss some justification for exploring solutions to scenario subproblems as candidate solutions for the overall problem, and finally in Section 4 we present some computational results using a distributed implementation of the proposed method on standard two-stage stochastic integer programming and nonlinear stochastic integer programming test problems.

2 A Scenario Decomposition Algorithm

We propose a simple scenario decomposition algorithm that exploits the scenario-wise decomposability of (3) and the 0-1 nature of the variables. The algorithm explores solutions to the scenario subproblems as candidate primal feasible solutions to the overall problem. The explored solutions are then cut-off from future consideration in all subproblems. This allows for an improvement in the lower bound by restricting the feasible region and eventually closing the duality gap.

The overall scheme is outlined in Algorithm 1. In the lower bounding step (lines 3-12), the scheme uses the decomposable dual problem (3) to generate lower bounds, and a set of candidate solutions. Note that we have not exactly specified how to update the dual solutions λ since the general scheme is independent of this. In fact, the scheme is valid without updating λ at all. In the upper bounding step (lines 13-22) each of the candidate solutions is evaluated and the best of these provide an upper bound and becomes the incumbent solution x^*. A key feature of the scheme is that the evaluated set of solutions is discarded from the set of feasible solutions (note the set $X \setminus S$ in line 7). Because of the 0-1 nature of the solutions this can be easily accomplished by adding "integer cuts" of the form

$$\sum_{j:\widehat{x}_j=1}(1-x_j)+\sum_{j:\widehat{x}_j=0}x_j \geq 1 \ \forall\, \widehat{x} \in S$$

to the original constraints X of the problem. Recently stronger formulations for discarding a given set of 0-1 solutions have been proposed which can also be used in this context [3].

Proposition 1 *Assuming that solving a scenario subproblem (step 7) and evaluating the objective function (step 17) requires finite time, Algorithm 1 (scenario decomposition) terminates in a finite number of iterations returning an optimal solution or detecting infeasibility.*

Algorithm 1 Scenario Decomposition

```
$UB \leftarrow +\infty$, $LB \leftarrow -\infty$, $S = \emptyset$, $x^* \leftarrow \emptyset$
while $UB > LB$ do
    Lower bounding:
    $\lambda \leftarrow 0$
    while some termination criteria is not met do
        for $i = 1$ to $N$ do
            solve $\min\{f_i(x) + \lambda^\top A_i x : \ x \in X \setminus S\}$
            let $v_i$ be the optimal value and $x^i$ be an optimal solution
        end for
        update $\lambda$
    end while
    $LB \leftarrow \sum_{i=1}^N v_i - \lambda^\top h$, $\widehat{S} = \cup_{i=1}^N \{x^i\}$ and $S \leftarrow S \cup \widehat{S}$
    Upper bounding:
    for $x \in \widehat{S}$ do
        $u \leftarrow 0$
        for $i = 1$ to $N$ do
            compute $f_i(x)$ and set $u \leftarrow u + f_i(x)$
        end for
        if $UB > u$ then
            $UB \leftarrow u$ and $x^* \leftarrow x$
        end if
    end for
end while
```

Proof: If the problem is infeasible the lower bound is set to $+\infty$ and the algorithm terminates after the first iteration. Otherwise, finite termination of the algorithm follows from the fact that the feasible region gets strictly smaller in each iteration and so the lower bound is nondecreasing (the lower bound is $+\infty$ when the feasible region becomes empty). Moreover since no solution is ever discarded without evaluation, it follows from the finiteness of the solution set that the algorithm returns an optimal solution in a finite number of iterations. □

The proposed scenario decomposition algorithm presumes that we have exact oracles to solve scenario subproblems of the form:

$$\min\{f_i(x) + \lambda^\top A_i x : \ x \in X \setminus S\}$$

for any λ and S, and to evaluate the objective function $f_i(x)$ for a given solution. These are similar to the assumptions made for other decomposition schemes based on the dual problem (3). For two-stage stochastic integer programs these steps require solving many single scenario integer programs and can present significant bottleneck. Of course instead of exact computations, we can compute the scenario subproblems and objective function to some pre-specified tolerance, use the best lower bound in the lower bounding step and the best upper bound in the upper bounding step, and terminate the algorithm after the lower and upper bounds are within tolerance. Also, note that in the upper bounding step when evaluating a solution $\widehat{x}$ we do not need to consider the scenarios for which this solution was optimal in the lower bounding step since the corresponding objective function values are already available. Significant computational benefits can be obtained by using relaxations of the scenario subproblems. Let $\underline{f}_i(x)$ be a relaxation of $f_i(x)$, i.e. $\underline{f}_i(x) \leq f_i(x)$ for all $x \in X$. For two-stage stochastic integer programs such a relaxation could be obtained by relaxing the integrality requirements on the second-stage variables. It is valid to use $\underline{f}_i(x)$ in the lower bounding step. The upper bounding step can be modified as follows: for a given candidate solution $\widehat{x}$ first evaluate $\underline{u} := \sum_{i=1}^N \underline{f}_i(\widehat{x})$ and if $\underline{u} \geq UB$ then we do not need to further consider $\widehat{x}$ since a better solution is already known; on the other hand if $\underline{u} < UB$ then we need to evaluate $u = \sum_{i=1}^N f_i(\widehat{x})$ as usual before

$\widehat{x}$ can be cut-off from further consideration. We could also evaluate and discard only a small subset of the solutions found rather than considering all the scenario subproblem solutions $\widehat{S}$. This would reduce the effort within a main iteration, perhaps at the expense increasing the number of main iterations.

An attractive feature of the proposed scheme is that there is very little interaction between the scenario subproblems in the lower and upper bounding steps. Accordingly the scheme can be easily implemented in a distributed framework. Each processor can be assigned a subset of scenarios and would essentially run through Algorithm 1 independently except for broadcasting and collecting solutions and objective values after solving scenario subproblems in step 7 and after evaluating objective functions in step 17. Note that if a subgradient algorithm (cf. [4]) is used to update the dual solutions λ where the update only depends on the scenario subproblem solutions and objective values, then each processor can locally update their copies of the dual solutions after exchanging solutions to the scenario subproblems in step 7, and these local dual solutions will be identical. In Section 4 we provide computational results using such a distributed implementation.

3 Optimality of scenario solutions

The algorithm proposed in the previous section explores solutions to the scenario subproblems as candidates for solutions to the overall problem. This is a departure from other methods based on solving the dual problem (3) where candidate solutions are generated by aggregating solutions to the scenario subproblems. In this section we attempt to provide some rationale why the solutions to the scenario subproblems themselves can be good solutions to the overall problem in the case the problem of interest is a sample average approximation problem.

Consider a 0-1 stochastic program

$$(P): \ \min\left\{\mathbb{E}[f(x,\boldsymbol{\xi})]:\ x \in X \subseteq \{0,1\}^n\right\},$$

and its sample average approximation

$$(SAA_N): \ \min\left\{(1/N)\sum_{i=1}^{N} f(x,\xi^i):\ x \in X \subseteq \{0,1\}^n\right\},$$

corresponding to an iid sample $\{\xi^i\}_{i=1}^N$. The scenario subproblem corresponding to the i-th scenario/sample is

$$(P_i): \ \min\left\{f(x,\xi^i):\ x \in X \subseteq \{0,1\}^n\right\}.$$

Let S_N be the set of optimal solutions of (SAA_N), and S_i be the set of optimal solutions of (P_i) for $i = 1,\ldots,N$. Note that these are random sets since they depend on the sample drawn. We investigate the following question:

> What is the probability that the set of solutions to one of the scenario problems (P_i) contain a solution to (SAA_N)?

More precisely we would like to estimate $\Pr[S_N \cap (\bigcup_{i=1}^N S_i) \neq \emptyset]$ and understand its dependence on X and N.

Trivially, if $|X| \leq 2$ and $N \geq 2$ then the above probability is 1. On the other hand, it is not hard to construct examples where this probability is arbitrarily small. For example, suppose $X = \{x^1, x^2, x^3\}$, $\boldsymbol{\xi}$ has two realizations ξ^1 and ξ^2 with equal probability, and the values of $f(x,\xi)$ are as in Table 1. Then for N reasonably large $S_N = \{x^1\}$ with very high probability while $\bigcup_{i=1}^N S_i = \{x^2, x^3\}$ with probability one.

In the following we show that if the collection of random objective functions $\{f(x,\boldsymbol{\xi})\}_{x\in X}$ of (P) is jointly normal with a nonnegative and strictly diagonally dominant covariance matrix then

$$\lim_{N\to\infty} \Pr\left[S_N \cap (\bigcup_{i=1}^{N} S_i) \neq \emptyset\right] = 1$$

Solution	ξ^1	ξ^2
x^1	0	0
x^2	-1	2
x^3	2	-1

Table 1: Values of $f(x, \xi)$ for an example

exponentially fast. Before proceeding with a proof of the above result we make some remarks. The condition of nonnegative covariance matrix implies that the objective functions at different solutions are nonnegatively correlated, and therefore, perhaps it is not very surprising that a scenario solution is likely to be optimal for the sample average problem. Recall that a matrix is strictly diagonally dominant if the diagonal element in a row is greater than the sum of the absolute values of all other elements in that row. This condition can be ensured by adding independent zero-mean normal noise to the objective function value for each solution without changing the optimal solution or optimal value of (P). We will need the following Lemma.

Lemma 1 *Let W be a $K+1$ dimensional Gaussian random vector, $W \sim N(\mu, \Sigma)$ with the following properties:*

1. *For some $\epsilon \geq 0$, $\mu_0 \leq \mu_k + \epsilon$ for all $k = 1, \dots, K$ and*
2. *the covariance matrix Σ is nonnegative and strictly diagonally dominant.*

Condition 2 implies that there exists δ such that $0 < \delta^2 := \min_{k \neq l}\{\sigma_k^2 - \sigma_{kl}, \sigma_l^2 - \sigma_{kl}\}$. Then

$$\Pr[W_0 \leq W_k \; \forall k = 1, \dots, K] \geq (\Phi(-\sqrt{2}\epsilon/\delta))^K,$$

where Φ is the standard normal cdf.

Proof: Since $\mu_0 \leq \mu_k + \epsilon$ for all $k = 1, \dots, K$, we have

$$\Pr[W_0 \leq W_k \; \forall k = 1, \dots, K] \quad \geq \quad \Pr[(W_0 - \mu_0) - (W_k - \mu_k) \leq -\epsilon \; \forall k = 1, \dots, K].$$

Let $U_k := (W_0 - \mu_0) - (W_k - \mu_k)$ for $k = 1, \dots, K$, and note that U is a K dimensional Gaussian random vector with $\mathbb{E}[U_k] = 0$, $\mathbb{E}[U_k^2] = \sigma_0^2 + \sigma_k^2 - 2\sigma_{0k} \geq 2\delta^2$, and $\mathbb{E}[U_k U_l] = \sigma_0^2 - \sigma_{0l} - \sigma_{0k} + \sigma_{kl} \geq 0$ for all $k, l = 1, \dots, K$. Consider another K dimensional Gaussian random vector V with $\mathbb{E}[V_k] = 0$, $\mathbb{E}[V_k^2] = \mathbb{E}[U_k^2]$ and $\mathbb{E}[V_k V_l] = 0$ for all $k, l = 1, \dots, K$. Since $\mathbb{E}[V_k V_l] \leq \mathbb{E}[U_k U_l]$ for all $k, l = 1, \dots, K$, by Slepian's inequality [14]

$$\Pr[W_0 \leq W_k \; \forall k] \geq \Pr[U_k \leq -\epsilon \; \forall k] \geq \Pr[V_k \leq -\epsilon \; \forall k] = \prod_{k=1}^{K} \Pr[V_k \leq -\epsilon],$$

where the last identity follows from independence. Since $\mathbb{E}[V_k^2] \geq 2\delta^2$. Hence for any k,

$$\Pr[V_k \leq -\epsilon] = \Pr\left[V_k/\sqrt{\mathbb{E}[V_k^2]} \leq -\epsilon/\sqrt{\mathbb{E}[V_k^2]}\right] \geq \Pr\left[V_k/\sqrt{\mathbb{E}[V_k^2]} \leq -\sqrt{2}\epsilon/\delta\right] = \Phi(-\sqrt{2}\epsilon/\delta)$$

and the result follows. □

Proposition 2 *If the collection of random objective functions $\{f(x, \boldsymbol{\xi})\}_{x \in X}$ of (P) is jointly normal with a nonnegative and strictly diagonally dominant covariance matrix then*

$$\lim_{N \to \infty} \Pr\left[S_N \cap (\bigcup_{i=1}^{N} S_i) \neq \emptyset\right] = 1$$

exponentially fast.

Proof: Pick $\epsilon \in (0,1)$ and let S_*^ϵ be the set of ϵ-optimal solutions to (P), i.e. $S_*^\epsilon = \{x \in X : \mathbb{E}[f(x,\boldsymbol{\xi})] \leq \mathbb{E}[f(x,\boldsymbol{\xi})] + \epsilon\, \forall y \in X\}$. Note that,

$$\Pr\left[S_N \cap (\bigcup_{i=1}^N S_i) \neq \emptyset\right] \geq \Pr\left[[S_*^\epsilon \subseteq (\bigcup_{i=1}^N S_i)] \wedge [S_N \subseteq S_*^\epsilon]\right] \geq \Pr[S_*^\epsilon \subseteq (\bigcup_{i=1}^N S_i)] + \Pr[S_N \subseteq S_*^\epsilon] - 1,$$

where the last inequality is using Boole's inequality. From SAA theory [8] we already know that

$$\Pr[S_N \subseteq S_*^\epsilon] \geq (1 - |X| e^{-N\frac{\epsilon^2}{4\sigma^2_{\max}}}).$$

where $\sigma^2_{\max} = \max_{x \in X}[\mathbb{V}[f(x,\boldsymbol{\xi})]$. Consider $x^* \in S_*^\epsilon$, then

$$\begin{array}{rcl}
\Pr[S_*^\epsilon \subseteq (\bigcup_{i=1}^N S_i) \neq \emptyset] & \geq & \Pr[x^* \in (\bigcup_{i=1}^N S_i)] \\
& = & 1 - \prod_{i=1}^N \Pr[x^* \notin S_i] \\
& = & 1 - \prod_{i=1}^N (1 - \Pr[x^* \in S_i]) \\
& = & 1 - \prod_{i=1}^N (1 - \Pr[f(x^*,\boldsymbol{\xi}) \leq f(y,\boldsymbol{\xi}) \;\forall\, y \in X \setminus \{x^*\}]).
\end{array}$$

Applying Lemma 1 with $K = |X \setminus \{x^*\}|$,$W_0 = f(x^*,\boldsymbol{\xi})$ and $W_k = f(y,\boldsymbol{\xi})$ for $y \in X \setminus \{x^*\}$, we get

$$\Pr\left[f(x^*,\boldsymbol{\xi}) \leq f(y,\boldsymbol{\xi}) \;\forall\, y \in X \setminus \{x^*\}\right] \geq (\Phi(-\sqrt{2}\epsilon/2\delta))^{|X|},$$

for some $\delta > 0$ depending on the covariance matrix of $\{f(x,\boldsymbol{\xi})\}_{x \in X}$. Thus

$$\Pr\left[S_N \cap (\bigcup_{i=1}^N S_i) \neq \emptyset\right] \geq (1 - (1 - (\Phi(-\sqrt{2}\epsilon/\delta))^{|X|})^N) - |X| e^{-N\frac{\epsilon^2}{4\sigma^2_{\max}}}. \tag{4}$$

Since $\Phi(-\sqrt{2}\epsilon/\delta) > 0$ for $\delta > 0$, taking limits with respect to N on both sides of (4) we get the desired result. □

We close this section with some remarks on Proposition 2. Note that when ϵ is very small relative to δ, $\Phi(-\sqrt{2}\epsilon/\delta) \approx 1/2$ and so the right hand side of (4) is approximately $(1 - (1 - (1/2)^{|X|})^N) - |X|e^{-NC}$ where C is a small constant. This indicates that smaller the solution set,$|X|$, relative to the sample size N, higher is the probability that one of the scenario solutions is optimal. If $|X|$ is large relative to N, then the bound in (4) is very loose, even negative. Consequently inequality (4) does not lead to any meaningful finite sample size estimates, but implies asymptotic behavior with respect to N for fixed X. Proposition 2 can be extended to more general distributions under different assumptions. For example, if the random vector $\{f(x^*,\boldsymbol{\xi}) - f(y,\boldsymbol{\xi})\}_{y \in X \setminus \{x^*\}}$ (where x^* is an ϵ-optimal solution of (P)) has a density and contains a ball centered at 0 of radius at least ϵ in its support, then there exists $p > 0$ such that $\Pr\left[f(x^*,\boldsymbol{\xi}) \leq f(y,\boldsymbol{\xi}) \;\forall\, y \in X \setminus \{x^*\}\right] \geq p$. We can then replace $(\Phi(-\sqrt{2}\epsilon/\delta))^{|X|}$ with p in (4) and the conclusion of Proposition 2 holds.

4 Computational Results

In this section we report some computational results using a distributed implementation of the proposed scenario decomposition to solve a class of two-stage stochastic integer programs and a class of nonlinear stochastic programs. Our implementation uses CPLEX 12.5 as the MIP solver (in single thread mode) and openMPI [7] for parallelization. We use a barrier synchronization step after solving the scenario subproblems and evaluating objective function across the processors before coordinating the scenario solutions and optimal values. Preliminary experiments with the subgradient method to update the dual solutions λ did not provide significant improvement in the lower bounds. Consequently, in the current implementation we use $\lambda = 0$ and do not update the dual solutions. We use an absolute optimality tolerance of 10^{-6} in the overall algorithm and in solving the MIP subproblems, and impose a time limit of 5000 seconds. All computations are done using the `boyle` cluster in the ISyE High Performance Computing Facility (`http://www.isye.gatech.edu/computers/hpc/`). These constitute a set of 16 identical machines each with 8 Intel Xeon 2.66GHz chips and 8 GB RAM running Linux. We limit our parallelization to 32 processors since this is the current limit on the cluster.

4.1 Two-stage stochastic integer programming instances

Our first set of results correspond to the sslp problem instances available in the SIPLIB library at http://www2.isye.gatech.edu/~sahmed/siplib/. These are a set of 10 two-stage stochastic integer programming instances from a telecommunications application [11]. The deterministic equivalent of the largest of these problems has over 1000000 binary variables. In our implementation we use continuous relaxation of the second stage variables in the lower bounding step as explained in Section 2 .

Table 2 presents the computational results. The first column is the name of the instance in the form sslp_n_1_n_2_N where n_1 is the number of first stage variables (all binary), n_2 is the number of second stage variables (mixed), and N is the number of scenarios. Columns 2-6 present the optimal value, the number of main iterations, total number of solutions explored, and the total solution time using 1 and 32 processors (except for the three sslp_15_45 instances where the number of scenarios are less than 32; in this case the number of processors used is equal to the number of scenarios). Each reported solution time is the average of 3 independent runs to account for the load variability in the computing cluster. In each of the 10 instances an optimal solution is found in the first iteration, i.e. one of the solutions to the scenario subproblems was optimal. In Figure 1 we present speedup of the proposed distributed scenario decomposition algorithm with respect to the number of processors for the instance sslp_10_50_500. The appearance of super-linear speedup is due to the fact that the run times are averages of 3 independent runs, and perhaps also due to the cache effects. The parallel solution times are significantly smaller and serial solution times are competitive in comparison to the times reported in the literature for these instances [10, 11, 12, 13]. In particular, comparing with the parallel implementation of [10] for solving the dual problem (without recovering primal feasible solutions) we note that [10] reports 2900+ seconds on 1 processor and 800+ seconds on 32 processors for sslp_10_50_500.

				Time (secs)	
Problem	Optimal Value	Iterations	Solutions	1 processor	32 processors
sslp_5_25_50	-121.600	2	16	2.3	0.7
sslp_5_25_100	-127.370	2	17	6.0	1.0
sslp_10_50_50	-364.640	4	123	65.7	6.0
sslp_10_50_100	-354.190	4	187	153.3	11.0
sslp_10_50_500	-349.136	3	241	1296.0	46.3
sslp_10_50_1000	-351.711	2	191	2691.0	60.7
sslp_10_50_2000	-347.262	2	226	4952.0	143.3
sslp_15_45_5	-262.400	6	27	4.3	2.0*
sslp_15_45_10	-260.500	13	116	35.3	7.0*
sslp_15_45_15	-253.600	17	233	92.7	13.3*

Table 2: Solutions times for the sslp instances (*the number of processors is equal to the number of scenarios.)

4.2 Nonlinear stochastic integer programming instances

Our second set of instances is a class of expected utility knapsack problems from [1]. These problems are of the form

$$\min\left\{-(1/N)\sum_{i=1}^{N}\mathcal{U}(\sum_{j=1}^{n}v_{ij}x_j):\ \sum_{j=1}^{n}a_jx_j\leq 1,\ x_j\in\{0,1\}^n\right\},$$

where $\mathcal{U}(t) = 1 - \exp(-t/c)$, i.e. a negative exponential utility function with risk aversion coefficient c. We generated instances with number of variables $n \in \{25, 50, 100\}$ and number of scenarios $N \in \{100, 500, 1000\}$. The coefficients v_{ij} and a_j for each instance were generated according to [1] and $c = 4$. The data for all instances are available in the SIPLIB library at http://www2.isye.gatech.edu/~sahmed/siplib/.

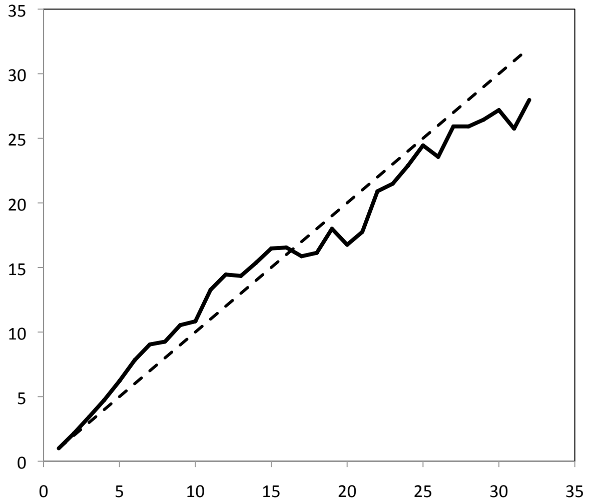


Figure 1: Speedup for `sslp_10_50_500`.

Note that for these problem, due to the monotonicity of the utility function, solving the i-th scenario subproblem in the lower bounding step (when $\lambda = 0$) is equivalent to solving the (linear) knapsack problem

$$\min \left\{ \sum_{j=1}^{n} v_{ij} x_j : \sum_{j=1}^{n} a_j x_j \leq 1, \; x_j \in \{0,1\}^n \right\}.$$

The lower bound is then obtained by averaging the negative of the utility function evaluated at the optimal values of such knapsack problems corresponding to each scenario. The upper bounding step constitute simply evaluating the scenario solutions.

Table 3 presents the computational results. The first column is the name of the instance in the form `exputil`_n_N where n is the number of variables and N is the number of scenarios. As before, columns 2-6 present the optimal value, the number of main iterations, total number of solutions explored, and the total solution time or the % optimality gap in case the 5000 sec time limit is exceeded. Each reported solution time is the average of 3 independent runs to account for the load variability in the computing cluster. Once again, in all of the instances an optimal solution is found in the first iteration.

				Time (secs) / gap (%)	
Problem	Optimal Value	Iterations	Solutions	1 processor	32 processors
`exputil_25_100`	-0.242304	2	49	4.7	1.7
`exputil_25_500`	-0.246211	2	68	24.0	6.3
`exputil_25_1000`	-0.242750	2	261	122.0	15.7
`exputil_50_100`	-0.246343	2	78	3.3	2.3
`exputil_50_500`	-0.247751	3	462	100.0	20.7
`exputil_50_1000`	-0.247643	3	323	225.3	41.7
`exputil_100_100`	-0.247542	22	2072	4443.0	690.7
`exputil_100_500`	-0.248989	>22	>10096	0.025%	0.009%
`exputil_100_1000`	-0.249317	9	5418	0.004%	721.0

Table 3: Solution times for the `exputil` instances

Acknowledgements

This research has been supported in part by the National Science Foundation (Grant # 1129871) and the Air Force Office of Scientific Research (Grant #FA9550-12-1-0154). The author thanks Dave Goldsman and Alexander Shapiro for valuable discussions.

References

[1] Shabbir Ahmed and Alper Atamtürk. Maximizing a class of submodular utility functions. *Mathematical Programming*, 128(1-2):149–169, 2011.

[2] Antonio Alonso-Ayuso, Laureano F. Escudero, and M. Teresa Ortunso. Bfc, a branch-and-fix coordination algorithmic framework for solving some types of stochastic pure and mixed 01 programs. *European Journal of Operational Research*, 151(3):503 – 519, 2003.

[3] Gustavo Angulo, Shabbir Ahmed, Santanu S. Dey, and Volker Kaibel. Forbidding vertices. *Working paper*, 2013.

[4] Dmitri Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[5] Claus C. Caroe and Ruediger Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1-2):37–45, 1999.

[6] Teodor Gabriel Crainic, Xiaorui Fu, Michel Gendreau, Walter Rei, and Stein W. Wallace. Progressive hedging-based metaheuristics for stochastic network design. *Networks*, 58(2):114–124, 2011.

[7] Edgar Gabriel, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, Ralph H. Castain, David J. Daniel, Richard L. Graham, and Timothy S. Woodall. Open MPI: Goals, concept, and design of a next generation MPI implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary, 2004.

[8] Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[9] Arne Lokketangen and David L. Woodruff. Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming. *Journal of Heuristics*, 2(2):111–128, 1996.

[10] Miles Lubin, Kipp Martin, Cosmin Petraa, and Burhaneddin Sandikci. On parallelizing dual decomposition in stochastic integer programming. Technical Report ANL/MCS-P3037-0912, Argonne National Labs, 2012.

[11] Lewis Ntaimo and Suvrajeet Sen. The million-variable ”march” for stochastic combinatorial optimization. *J. of Global Optimization*, 32(3):385–400, 2005.

[12] Lewis Ntaimo and Suvrajeet Sen. A comparative study of decomposition algorithms for stochastic combinatorial optimization. *Computational Optimization and Applications*, 40(3):299–319, 2008.

[13] Lewis Ntaimo and Matthew W. Tanner. Computations with disjunctive cuts for two-stage stochastic mixed 0-1 integer programs. *Journal of Global Optimization*, 41(3):365–384, 2008.

[14] David Slepian. The one-sided barrier problem for gaussian noise. *Bell System Technical Journal*, pages 463–501, 1962.

[15] Jean-Paul Watson and David Woodruff. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, pages 1–16, 2010.

Strengthened Bounds for the Probability of k-Out-Of-n Events

Feng Qiu[*], Shabbir Ahmed, and Santanu S. Dey

August 17, 2013

Abstract

Given a set of n random events in a probability space, represented by n Bernoulli variables (not necessarily independent,) we consider the probability that at least k out of n events occur. When partial distribution information, i.e., individual probabilities and all joint probabilities of up to m ($m < n$) events, are provided, only an upper or lower bound can be computed for this probability. Recently Prèkopa and Gao (Discrete Appl. Math. 145 (2005) 444) proposed a polynomial-size linear program to obtain strong bounds for the probability of union of events, i.e., $k = 1$. In this work, we propose inequalities that can be added to this linear program to strengthen the bounds. We also show that with a slight modification of the objective function this linear program and the inequalities can be used for the more general case where k is any positive integer less than or equal to n. We use the strengthened linear program to compute probability bounds for the examples used by Prèkopa and Gao, and the comparison shows significant improvement in the bound quality.

Keywords: linear programming, probability bound, k-of-n event

1 Introduction

Let $\{A_j : \ j \in N\}$ be a set of events, where N is the index set $\{1, \dots, n\}$. Define random variable $X_j : A_j \to \{0, 1\}$ as $X_j = 1$ if A_j occurs, and $X_j = 0$, otherwise. Define μ as a random variable that represents the number of events, A_js, that occur, i.e., $\mu = \sum_j X_j$. Let k be a positive integer between 1 and n; then the probability that (at least) k out of n events in $\{A_1, ..., A_n\}$ occur is denoted by $\mathbb{P}(\mu \geq k)$.

Computation of $\mathbb{P}(\mu \geq k)$ is often needed in applications. For example, in a maximum availability location problem [3], the probability that the population in a subregion is covered by at least k facilities is used to calculate the expected

[*] Tel:+1-(608)-354-5638, Fax: +1-(630)-252-9868, Email: fqiu@gatech.edu

coverage. Another example is the reliability problem of communication networks, where each arc fails with a certain probability and we want to compute or approximate the node-to-node reliability of the system.

Accurate computation of $\mathbb{P}(\mu \geq k)$ is not easy. A complete distribution function for a system of Bernoulli events involves an exponential size of data, which is difficult to handle when n is large. Furthermore, in practice, the complete distribution function for $(X_1, ..., X_n)$ is often not available, unless the Bernoulli random numbers X_j are independent from each other. The available information is often the marginal distributions and joint distributions up to level m ($m \ll n$). In this situation, it is desirable/preferable to compute a lower or upper bound using only a limited amount of information.

A number of bounding results that utilize different amounts of information under different settings have been proposed. The classic Boole inequality yields a lower bound for the union of events with only individual marginal probabilities, i.e., $k = 1$ and $m = 1$. Dawson and Sanko [4] provide a sharp lower bound for the probability of the union of events using marginal and pairwise joint probabilities, i.e., $m = 2$; Kwerel [5, 6] developed bounds that can utilize one more degree of joint probabilities, i.e., $m = 3$; Prèkopa, Boros and other researchers [7, 8, 2] employed certain linear programs to derive lower and upper bounds for a general case where m can be any positive integer less than or equal to n. Most of these works use aggregation of the individual joint probabilities of the same degree in their formulations, called binomial moments. As a consequence of summation, individual probability information is lost, and the given information is not fully utilized. To make better use of the available information, Prèkopa and Gao [9] derived LP-based bounds for a union of events using partially aggregated information. We call this LP model as the *partially aggregated model* (PAM). The numerical examples showed that their bounds were at least as strong as other results using binomial moments.

In this paper, we first observe that the results in [9] which is stated for the case of $k = 1$, generalizes in a straightforward way to the case where k is any positive integer less than or equal to n. Our key contribution is to identify inequalities that can be appended to PAM to strengthen the bounds. We call the resulting LP model as *strengthened partially aggregated model* (SPAM). These results are presented in Section 2. We test the strength of the extra inequalities using the instances in [9], and the results show significant improvement of the bound quality of SPAM over PAM. We also computationally identify a family of probability distributions for which the bounds provided by SPAM has the largest improvement over the bounds provided by PAM model. We report these computational results in Section 3.

2 Strengthening the Partially Aggregated Model

2.1 The Partially Aggregated Model for $k \in \{1, \ldots, n\}$

Let S be a subset of N. Denote the joint probability that all events in the set S occur as $p_S := \mathbb{P}(\bigcap_{\ell \in S} A_\ell)$. Let $s_t^j := \sum_{\substack{S:\ |S|=t \\ j \in S}} p_S$. We first state the main result of [9] using our notation.

Theorem 1 ([9])**.** *Given the joint distributions up to level m, i.e., $|S| \leq m$, a lower (or upper) bound for the probability of the union of these events, i.e., $\mathbb{P}(\mu \geq 1)$, can be calculated by solving the following linear program (PAM):*

$$\begin{aligned} \min(\max) & \sum_{i=1}^{n} \sum_{j=1}^{n} v_{ij} & & \qquad (1a) \\ \text{s.t.} & \sum_{i=0}^{n} \sum_{j=1}^{n} v_{ij} & = 1 & \qquad (1b) \\ & \sum_{i=t}^{n} \binom{i}{t} v_{ij} & = \tfrac{1}{t} s_t^j & \qquad (1c) \\ & v_{ij} & \geq 0. & \qquad (1d) \end{aligned}$$

□

The LP model above and those in [7, 8, 2, 9] are derived using probabilistic reasonings (See Appendix 1 for a brief review.)

We next show that the bounds on $\mathbb{P}(\mu \geq k)$ for $k \in \{1, \ldots, n\}$ can be obtained by solving (1) by changing the objective function (1a) to $\sum_{i=k}^{n} \sum_{j=1}^{n} v_{ij}$.

As pointed out in [9], any bound that uses p_S can be recovered as the value of the objective function of the Boolean LP model [1], which consists of the probability of each possible outcome. Let C be a subset of N, and let the probability of the outcome associated with C be denoted as

$$w_C := \mathbb{P}\left((\bigcap_{i \in C} A_i) \bigcap (\bigcap_{j \in N \setminus C} \bar{A}_j) \right),$$

where $\bar{A}_j$ represents that event A_j does not occur. Note that these outcomes are mutually exclusive. The probability of any event can be represented by using the probabilities of an appropriate set of outcomes. For example, the probability that at least k out of n events occur can be expressed as $\sum_{C \subseteq N, |C| \geq k} w_C$. Given joint distributions up to level m, i.e., p_S, $|S| \leq m$, a lower (or upper) bound of $\mathbb{P}(\mu \geq k)$ can be obtained by the following Boolean linear program [1]:

$$\min(\max) \quad \sum_{C:|C| \geq k} w_C$$

$$\text{s.t.} \sum_{C \subseteq N} w_C = 1 \qquad (2)$$
$$\sum_{C: S \subseteq C} w_C = p_S, \quad \forall S \subseteq N,\ |S| \leq m$$
$$w_C \geq 0\ \forall C \subseteq N.$$

Formulation (2) consists of an exponential number of variables and may not be useful in practice. In order to obtain the PAM LP model, we first duplicate each row with right-hand side p_S in (2) $|S|$-1 times, add up rows with $|S| = t$ and $j \in S$ for each t and j, and then arrive at

$$\min \sum_{i=k}^{n} \sum_{j=1}^{n} \frac{1}{i} \sum_{C: \substack{|C|=i \\ j \in C}} w_C \qquad (3a)$$

$$\text{s.t.} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{i} \sum_{C: \substack{|C|=i \\ j \in C}} w_C = 1 \qquad (3b)$$

$$\sum_{i=t}^{n} \binom{i-1}{t-1} \sum_{C: \substack{|C|=i \\ j \in C}} w_C = \sum_{S: \substack{|S|=t \\ j \in S}} p_S \quad t = 1, ..., m\ j = 1, ..., n \qquad (3c)$$

$$w_C \geq 0 \quad \forall C \subseteq N. \qquad (3d)$$

Equations (3a)-(3d) are the resulting rows by duplicating and aggregating the rows in (2). Notice that the following variables share the same coefficient in each row: w_C with $|C| = i$ and $j \in C$. Therefore, we aggregate these variables into a single variable and, for notational simplicity, scale the resulting aggregrated variable as

$$v_{ij} := \frac{1}{i} \sum_{C: \substack{|C|=i \\ j \in C}} w_C \quad i = 1, ..., n\ j = 1, ...n. \qquad (4)$$

Thus we obtain the following linear program:

$$\min \sum_{i=k}^{n} \sum_{j=1}^{n} v_{ij} \qquad (5a)$$

$$\text{s.t.} \sum_{j=1}^{n} \sum_{i=0}^{n} v_{ij} = 1 \qquad (5b)$$

$$\sum_{i=t}^{n} \binom{i}{t} v_{ij} = \frac{1}{t} s_t^j \quad t = 1, ..., m\ j = 1, ..., n \qquad (5c)$$

$$v_{ij} = \frac{1}{i} \sum_{C: \substack{|C|=i \\ j \in C}} w_C \quad i = 1, ..., n\ j = 1, ...n \qquad (5d)$$

$$v \geq 0 \; i = 1, ..., n \; j = 1, ..., n \tag{5e}$$

$$w_C \geq 0 \; \forall C \subseteq N. \tag{5f}$$

Now observing that the variables w_C do not appear in the objective function (5a), we may *relax* (5), by *dropping* the variables w_C and the constraints (5d) and (5f), to obtain the PAM model.

2.2 Strengthened PAM

As discussed in the previous section, PAM is obtained by relaxing (5). In order to strengthen PAM, we project out the variables w_C from the set defined by the constraints (5d) and (5f) to obtain valid inequalities which we append to PAM. Again we are able to accomplish this using the fact that the w_C variables do not appear in the objective function.

We need the following lemma.

Lemma 1. *Let* $S^i := \{x \in \{0,1\}^n \mid \sum_{j=1}^n x_j = i\}$*. For* $1 \leq i \leq n-1$*,* $\dim(S^i) = n$*.*

Proof. We list n vectors from S^i that are linearly independent. Set vectors $v^1, \ldots, v^{i+1}$ as

$$v_k^j = \begin{cases} 1 & k \in \{1, \ldots, i+1\} \setminus \{j\} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Set vectors $v^{i+2}, \ldots, v^n$ as

$$v_k^j = \begin{cases} 1 & k \in \{1, \ldots, i-1\} \cup \{j\} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

It is straighforward to check that $v^1, \ldots, v^n$ belong to S^i and are linearly independent. □

Proposition 2. *Let*

$$W_i = \{(..., w_C, ..., v_{ij}, ...) \in \mathbb{R}_+^{\binom{n}{i}} \times \mathbb{R}^n : v_{ij} = \frac{1}{i} \sum_{\substack{C: |C|=i \\ j \in C}} w_C \; j = 1, ...n\}. \tag{8}$$

Then the projections of W_i*s onto* v *space are the following sets:*

$$Proj_v(W_i) = \{(v_{i1}, ..., v_{in}) \in \mathbb{R}^n \mid -(i-1)v_{ij} + \sum_{t \neq j} v_{it} \geq 0, \; v_{ij} \geq 0 \; j \in N\} \qquad i = 2, ..., n-2, \tag{9}$$

$$Proj_v(W_{n-1}) = \{(v_{(n-1)1}, ..., v_{(n-1)n}) \in \mathbb{R}^n \mid -(n-2)v_{(n-1)j} + \sum_{t \neq j} v_{(n-1)t} \geq 0, \; j \in N\} \tag{10}$$

and

$$Proj_v(W_n) = \{(v_{n1}, ..., v_{nn}) \in \mathbb{R}^n_+ \mid v_{n1} = v_{nt}\ \ 2 \leq t \leq n,\ \ v_{n1} \geq 0\}. \qquad (11)$$

Proof. Note that $\mathrm{Proj}_v(W_i)$ is a cone. Therefore, for simplicity of analysis, we may scale the v variables and thus without loss of generality assume that

$$W_i := \{(..., w_C, ..., v_{ij}, ...) \in \mathbb{R}_+^{\binom{n}{i}} \times \mathbb{R}^n : v_{ij} = \sum_{C:\substack{|C|=i \\ j \in C}} w_C \ \ j = 1, ...n\}. \qquad (12)$$

We show that (9) holds for an arbitrary i between 2 and $n-1$, where $|C| = i$. To simplify notations, for now, we drop subscript i in the constraint. Let $w = (..., w_C, ...) \in \mathbb{R}_+^{\binom{n}{i}}$ and $v = (..., v_{ij}, ...) \in \mathbb{R}^n$. We then rewrite (12) as follows

$$W_i = \{(w, v) \in \mathbb{R}_+^{\binom{n}{i}} \times \mathbb{R}^n : v = Gw\},$$

where $G = (..., g_C, ...) \in \mathbb{R}^{n \times \binom{n}{i}}$ is the coefficient matrix and $g_C = (g_C^1, ..., g_C^j, ..., g_C^n)^\top$ is the column corresponding to the variable w_C. $g_C^j = 1$ if $j \in C$; $g_C^j = 0$ if $j \notin C$. G consists of all permutations of the 0-1 vector with i ones and $n - i$ zeros. Rewrite (9) as follows:

$$V_i := \{v \in \mathbb{R}^n : (e - ie_j)^\top v \geq 0,\ e_j^\top v \geq 0\ \ j = 1, ..., n\},$$

where $e \in \mathbb{R}^n$ is a vector with all components equal to one and $e_j \in \mathbb{R}^n$ is the j-th unit vector.

To show that V_i is the projection of W_i into v-space, we need to show

$$\bar{v} \in V_i \Leftrightarrow \text{There is a } \bar{w} \in \mathbb{R}_+^{\binom{n}{i}} \text{ such that } G\bar{w} = \bar{v}.$$

By Farkas' lemma, we have that

$$\text{There is a } \bar{w} \in \mathbb{R}_+^{\binom{n}{i}} \text{ such that } G\bar{w} = \bar{v} \Leftrightarrow u \in \mathbb{R}^n \text{ such that } u^\top G \geq 0 \Rightarrow u^\top \bar{v} \geq 0.$$

Let $\{u_\ell\}$ be the set of extreme rays of the cone$\{u : u^\top G \geq 0\}$. It is sufficient to show that the set of constraint vectors in V_i, i.e., $(e - ie_j)$ and e_j, is exactly $\{u_\ell\}$.

We first show $(e - ie_j)$ and e_j are extreme rays.

$(e - ie_j)^\top$ is a feasible solution for the cone $\{u : u^\top G \geq 0\}$:

$$(e - ie_j)^\top g_C = \begin{cases} 0 & C : j \in C \\ i - 1 & C : j \notin C. \end{cases}$$

Furthermore, the products above have $\binom{n-1}{i-1}$ zeros, which means that $\binom{n-1}{i-1}$ constraints in $u^\top G \geq 0$ are tight. Notice that the binding constraint vectors g_C are all the permutations of vectors with the j-th position fixed to one. Therefore, by Lemma 1, they span $\mathbb{R}^{n-1}$ and we can find $n-1$ linearly independent vectors among them which have zero products with $(e - ie_j)$. Thus, $(e - ie_j)$ is an extreme ray of $\{u : u^\top G \geq 0\}$.

As for e_j, we have $e_j^\top G$ as follows

$$e_j^\top g_C = \begin{cases} 1 & C : j \in C \\ 0 & C : j \notin C. \end{cases}$$

Therefore, e_j is a feasible solution for the cone $\{u : u^\top G \geq 0\}$ and the product above has $\binom{n-1}{i}$ zeros that correspond to g_C with $j \notin C$. Among those columns that have zero products with e_j, by using Lemma 1 we can find $n-1$ linearly independent columns. Therefore, e_j is an extreme ray of $\{u : u^\top G \geq 0\}$.

Now we show by contradiction that there are no other extreme rays. Let $\lambda \neq 0$ be a new distinct extreme ray; and let $g_{S_\ell} = (g^1_{S_\ell}, ..., g^t_{S_\ell}, ...g^n_{S_\ell})^\top$ $\ell = 1, ..., n-1$ be the set of linear independent columns of G with $\lambda^\top g_{S_\ell} = 0$ $\ell = 1, ..., n-1$. Since $\lambda \neq e_j$ for all $j = 1, ..., n$, we have

$$\nexists t \text{ such that } g^t_{S_\ell} = 0 \quad \forall \ell = 1, ..., n-1 \tag{13}$$

since otherwise, e_t would be the extreme ray formed by the half planes $\{u^\top g_{S_\ell} \geq 0 \ \ell = 1, ..., n-1\}$. Similarly, since $\lambda \neq (e - ie_j)$ for all $j = 1, ..., n$, we have

$$\nexists t \text{ such that } g^t_{S_\ell} = 1 \quad \forall \ell = 1, ..., n-1. \tag{14}$$

Using (13) and $g_{S_\ell} \geq 0$, we obtain that $\lambda_{t^*} < 0$ for some t^* (Otherwise, $\lambda^\top g_{S_\ell} > 0$ for some ℓ by (13)). By (14), there is a ℓ^* such that $g^{t^*}_{S_{\ell^*}} = 0$. Since $\lambda^\top g_{S_{\ell^*}} = \sum_{t \neq t^*} \lambda_t g^t_{S_{\ell^*}} = 0$ and $g_{S_{\ell^*}} \neq 0$, we can find an index $\bar{t}$ such that $\lambda_{\bar{t}} \geq 0$ and $g^{\bar{t}}_{S_{\ell^*}} = 1$. Let g^* be a vector obtained by switching the components at position t^* and $\bar{t}$ in vector $g_{S_{\ell^*}}$. Note that g^* is also a vector of G. Then $\lambda^\top g^* = \lambda^\top g_{S_{\ell^*}} + \lambda_{t^*} - \lambda_{\bar{t}} < 0$. Thus, λ is not a feasible ray of the cone $\{u : u^\top G \geq 0\}$. Therefore, there are no extreme rays of $\{u : u^\top G \geq 0\}$ other than $\{(e - ie_j),\ e_i\ j = 1, ..., n\}$ and $\text{Proj}(W_i) = V_i$ for $i = 2, ..., n-2$.

For $\text{Proj}(W_{n-1})$, we can show that $\{(e - ie_j)\ j = 1, ..., n\}$ are extreme rays of $\{u : u^\top G \geq 0\}$. However $\{e_j\}$ are not extreme rays in this case since each row of G has only one zero. Using a similar argument as that in (2), we can show that $\{(e - ie_j)\ j = 1, ..., n\}$ are the only extreme rays and that $\text{Proj}(W_n) = V_n$.

The equations in (10) hold since $v_{nj} = v_N$ for all $j = 1, ...n$. □

Notice that the projection of W_1 yields only non-negativity constraints on $v_{1j},\ j = 1, ..., n$ and the non-negativity of $v_{(n-1),j}$ for $j = 1, ..., n$ is implied by $-(i-1)v_{(n-1)j} + \sum_{t \neq j} v_{it} \geq 0$ $j = 1, ..., n$.

The inequalities in Proposition 2,

$$-(i-1)v_{ij} + \sum_{t \neq j} v_{it} \geq 0,\ v_{ij} \geq 0 \quad j = 1, ..., n,\ i = 2, ..., n-1 \tag{15}$$

and

$$v_{n1} = v_{nt} \; 2 \leq t \leq n, \tag{16}$$

are valid for model (5) and they are not implied by the constraints in PAM. We call the linear program with these additional constraints as the *Strengthened Partially Aggregated Model* or *SPAM.*

3 Numerical Examples

In the following, we calculate lower bounds with the strengthened model for the examples in [9], and compare the new bounds with those presented in [9]. Note that the lower bounds in [9] are only for the probability of the union of events, i.e., $k = 1$. Examples 1, 2, and 3 have 20 Bernoulli variables each, i.e., $X_1, ..., X_{20}$. All the outcomes and their probabilities are presented in Table 1, 2, and 3, respectively. With these tables, we can obtain the marginal probabilities, pair-wise joint probabilities, and higher-order joint probabilities by adding up appropriate rows.

Table 1: Probability Distributions in Example 1

Outcome	Bernoulli Random Variables																				Probability
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}	X_{20}	
1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0.012214
2	0	1	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0.022231
3	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0.023287
4	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0.033976
5	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	0	0.034761
6	0	1	0	0	1	0	1	0	1	0	1	1	0	0	1	0	1	0	0	1	0.044582
7	1	0	0	1	0	1	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0.045943
8	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0.055185
9	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0.056404
10	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0.066317
11	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0.067685
12	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0.077376
13	1	0	0	1	0	1	0	0	0	1	0	0	1	0	1	0	0	1	0	1	0.078648
14	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0.088878
15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.292514

We compare the lower bounds corresponding to $k = 1$ produced by SPAM with the results in [9] in Table 4. Note that all of the bounds in Table 4 are calculated with only marginal probabilities and pair-wise joint probabilities, except those in the fourth column.

The second column cites the results obtained by the formula derived in [4], which can be obtained by further aggregating the Boolean LP model. We call this the *Fully Aggregated Model* or *FAM* (See (18) in Appendix 1). The third column cites the results obtained by PAM derived in [9]. The fourth column cites the results obtained by PAM but with three binomial moments, including the triple-wise joint probabilities. Prèkopa and Gao also developed heuristics in [9] to strengthen the lower bounds by PAM, but the best results obtained are

Table 2: Probability Distributions in Example 2

Outcome	Bernoulli Random Variables																				Probability
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}	X_{20}	
1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	0	1	0	0	1	0	0.00896463
2	1	1	0	1	0	0	0	0	0	1	0	1	0	1	1	0	0	1	0	1	0.02492217
3	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0.02109813
4	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	0	0	1	0	0	0.03779353
5	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0.04632610
6	1	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0	0	0	0	1	0.04284324
7	1	1	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0.07804262
8	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0.02536991
9	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0.01916672
10	1	1	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0.06340085
11	0	1	0	1	1	0	0	0	1	1	0	1	0	0	1	1	0	0	0	1	0.07315289
12	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0.07732742
13	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0.02248020
14	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0.09164494
15	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0.36745660

Table 3: Probability Distributions in Example 3

Outcome	Bernoulli Random Variables																				Probability
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}	X_{20}	
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.10176880
2	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0.11299200
3	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0.01514044
4	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0.05684733
5	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0.03270125
6	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0.10050750
7	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0.07306695
8	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0.01743922
9	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0.06284498
10	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0.05830101
11	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0.06833096
12	1	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	1	0.07153743
13	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0.04503293
14	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0.03487869
15	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0.14860240

Table 4: Comparison with results in [9] for $k = 1$

Example	FAM	PAM	PAM(3)	SPAM
4	0.8275266	0.8580833	0.8864460	**0.9394167**
5	0.8658182	0.9100646	0.9354100	**0.9482229**
6	0.8985498	0.9435812	0.9587778	**0.9715460**

no better than those obtained by involving triple-wise probabilities, which are listed in the fourth column. Therefore, we ignore the results of the heuristics. The fifth column gives the results obtained by SPAM. We observe in Table 4 that the extra inequalities derived in this study significantly improve the lower bounds calculated by PAM.

Now, we use the same instances to calculate the lower bounds for the probability of 3-coverage, i.e., $k = 3$, and summarize the results in Table 5, which again shows that SPAM produces significantly tighter lower bounds than both FAM and PAM. We also calculate the upper bounds and compare the results with the results in [9]. However, we do not observe improvement in the new upper bounds.

Table 5: Lower Bounds for 3-Coverage ($k = 3$)

Example	FAM	PAM	PAM(3)	SPAM
1	0.6643058	0.665249	**0.6768353**	0.6745504
2	0.7298830	0.7528989	0.7847514	**0.8005835**
3	0.7387907	0.7819380	0.8482093	**0.8614323**

In Table 6 we provide examples to show that for certain probability distributions, the improvement by the inequalities in Proposition 2 can be much more significant than the improvement observed in Examples 1, 2, and 3. The specific distrobution under which the lower bounds in Table 6 are calculated are obtained computationally by an optimization model provided in Appendix 2. We run this optimization model with n fixed at 10, m fixed at 3, and k varying from 2 to 4. The optimization problems are solved by the nonlinear solver BARON and the optimal solutions produce probability distributions under which the gaps between PAM and SPAM are maximal. In our experiments we enforce a solution time limit of ten hours. Columns 3 and 4 of Table 6 list the lower bounds yield by PAM and SPAM, respectively. For comparison we also calculate the lower bounds using FAM and the Boolean model (BM) and list them in Columns 2 and 5, respectively.

Table 6: Comparison of Lower Bounds

k	FAM	PAM	SPAM	BM
2	0.6296296	0.6296296	1.0000000	1.0000000
3	0.3412896	0.3432099	0.8065844	0.8065844
4	0.2108243	0.2143992	0.8599802	0.8599802

References

[1] G. Boole. An investigation into the laws of thought. *Walton and Maberly, London.[Reprinted (1951) Dover, New York.]*, 1854.

[2] E. Boros and A. Prekopa. Closed form two-sided bounds for probabilities that at least r and exactly r out of n events occur. *Mathematics of Operations Research*, pages 317–342, 1989.

[3] M.S. Daskin. A maximum expected covering location model: formulation, properties and heuristic solution. *Transportation Science*, 17(1):48–70, 1983.

[4] D. A. Dawson and D. Sankoff. An inequality for probabilities. *Proc. Amer. Math. Soc.*, 18:504–507, 1967.

[5] S. M. Kwerel. Bounds on the probability of the union and intersection of m events. *Advances in Appl. Probability*, 7:431–448, 1975.

[6] S. M. Kwerel. Most stringent bounds on the probability of the union and intersection of m events for systems partially specified by $S_1, S_2, \cdots, S_k$, $2 \leq k < m$. *J. Appl. Probability*, 12(3):612–619, 1975.

[7] A. Prékopa. Boole-bonferroni inequalities and linear programming. *Operations Research*, pages 145–162, 1988.

[8] A. Prékopa. Sharp bounds on probabilities using linear programming. *Operations Research*, pages 227–239, 1990.

[9] A. Prékopa and L. Gao. Bounding the probability of the union of events by aggregation and disaggregation in linear programs. *Discrete applied mathematics*, 145(3):444–454, 2005.

Appendix 1

We briefly motivate the partially aggregated model introduced in [9]. Let $\binom{\mu}{i}$ represent the combination of choosing i out v. We observe that [7]

$$\binom{\mu}{i} = \sum_{S\subseteq N:|S|=i} \prod_{\ell\in S} X_\ell. \tag{17}$$

Notice that both sides of above equation are random numbers. After taking expectation, we have the following equation

$$\sum_{t=1}^{n} \binom{t}{i} v_i = \sum_{S\subseteq N:|S|=i} p_S, \tag{18}$$

where $p_S = \mathbb{P}(\bigcap_{\ell\in S} A_\ell)$ and the right-hand side is the i-th binomial moment. Note that (18) are the constraints of the LP model in [7, 8, 2], which we call *Fully Aggregated Model* or FAM.

A simple consequence of (17) is as follows [9]

$$X_j \binom{\mu-1}{i-1} = X_j \sum_{S\subseteq N:|S|=i} \prod_{\ell\in S} X_\ell. \tag{19}$$

Taking expectation of both sides in above equation, we have the following

$$\sum_{t=1}^{n} \binom{t-1}{i-1} v_{tj} = \sum_{S \subseteq N:|S|=i, j \in S} p_S, \tag{20}$$

where $v_{tj} = \mathbb{P}(\mu = t \cap X_j = 1)$. Note that the right-hand side of (20) only consists of the probabilities of those events that A_j occurs. Since the probability of the union of $A_1, \ldots, A_n$ can be expressed as follows

Theorem 3 ([9]).

$$\mathbb{P}(\mu \geq 1) = \sum_{j=1}^{n} \sum_{i=1}^{n} \frac{1}{i} v_{ij}, \tag{21}$$

an upper bound or a lower bound on the probability of union of events can be obtained by minimizing or maximizing the right-hand side of (21) over the set of constraints in (20) and the non-negativity constraints [9].

Appendix 2

The following optimization model identifies the probability distributions for which the gap between SPAM and PAM is maximal. Let z^*_{SPAM} and z^*_{PAM} be the lower bounds for the probability that at least k-out-of-n events occur, obtained by $SPAM$ model and PAM model, respectively. Let variables p_S, for all $S \subseteq N$and $|S| \leq m$, represent the probability distributions we aim to obtain. Note that s_i^j is now a function of p_S. Let π_0 and π_i^j be the dual variables for constraint (5b) and (5c), respectively; μ_i^j be the dual variable corresponding to the constraint (15) for $i = 2, .., n-1$ and $j = 1, ..., n$, and μ_n^j be the dual variable corresponding to the constraint (16) for $j = 2, ..., n$.

$$\begin{aligned}
\text{gap} = \quad & \max_{p_S} \, (z^*_{SPAM} - z^*_{PAM}) \\
= \quad & \max_{p_S} \, (\min_{v_{ij}} \{ \sum_{j=1}^{n} \sum_{i=k}^{n} v_{ij} : \text{(5b), (5c), (5e), (15), and (16)} \} \\
& \quad - \min_{v_i} \{ \sum_{i=k}^{n} v_i : \text{(5b), (5c), and (5e)} \}) \\
= \quad & \max_{p_S} \, (\max_{(\pi,\mu)} \{ \pi_0 + \sum_{j=1}^{n} \sum_{i=1}^{m} \frac{1}{i} \pi_i^j s_i^j : \\
& \pi_0 + \pi_1^j \leq e_1 \;\; j = 1, ..., n \\
& \pi_0 + \sum_{i=1}^{m} \binom{t}{i} \pi_i^j - (t-1)\mu_j^t + \sum_{\ell \neq j} \mu_t^\ell \leq e_t \;\; t = 2, ..., n-1; \;\; j = 1, ..., n \\
& \pi_0 + \sum_{i=1}^{m} \binom{n}{i} \pi_i^1 - \sum_{j=2}^{n} \mu_n^j \leq e_n
\end{aligned}$$

$$
\begin{aligned}
& \pi_0 + \sum_{i=1}^{m} \binom{n}{i} \pi_i^j + \mu_n^j \leq e_n \ j = 2, ..., n \\
& \pi_0, \pi_i^j, \mu_n^j \text{ free, } , \mu_i^j \geq 0 \ i = 2, ..., n-1\} \\
& + \max\{\sum_{i=k}^{n} \sum_{j=1}^{n} -v_{ij} : \sum_{i,j} v_{ij} = 1; \sum_{i=t}^{n} \binom{i}{t} v_{ij} = \frac{1}{t} s_t^j(x) \quad t = 1, ..., m \ j = 1, ..., n, \\
& v_i^j \geq 0\}) \\
= \ & \max_{p_S, \pi, \mu, v_i^j} \{\pi_0 + \sum_{j=1}^{n} \sum_{i=1}^{m} \frac{1}{i} \pi_i^j s_i^j - \sum_{i=k}^{n} \sum_{j=1}^{n} v_{ij} : \\
& \pi_0 + \pi_1^j \leq e_1 \ j = 1, ..., n \\
& \pi_0 + \sum_{i=1}^{m} \binom{t}{i} \pi_i^j - (t-1)\mu_j^t + \sum_{\ell \neq j} \mu_t^\ell \leq e_t \ t = 2, ..., n-1; \ j = 1, ..., n \\
& \pi_0 + \sum_{i=1}^{m} \binom{n}{i} \pi_i^1 - \sum_{j=2}^{n} \mu_n^j \leq e_n \\
& \pi_0 + \sum_{i=1}^{m} \binom{n}{i} \pi_i^j + \mu_n^j \leq e_n \ j = 2, ..., n \\
& \sum_{i,j} v_{ij} = 1; \sum_{i=t}^{n} \binom{i}{t} v_{ij} = \frac{1}{t} s_t^j(x) \quad t = 1, ..., m \ j = 1, ..., n, \\
& \pi_0, \pi_i^j, \mu_n^j \text{ free, } , \mu_i^j \geq 0 \ i = 2, ..., n-1, \ v_i^j \geq 0\},
\end{aligned}
$$

where $e_t = 0$ if $t < k$; $e_t = 1$, otherwise.

The feasible set for $(..., p_S, ...)$ is as follows

$$
\mathcal{P}^m = \{(..., p_S, ...) \in \mathbb{R}_+^{\binom{n}{1} + ... + \binom{n}{m}} : \exists v \in \mathbb{R}_+^{2^n} \text{ s.t. } \sum_{C \subseteq N} w_C \leq 1, p_S = \sum_{C : S \subseteq C} w_C \ \forall S \subseteq N, |S| \leq m\}.
$$

Forbidden vertices

Gustavo Angulo[*], Shabbir Ahmed[†], Santanu S. Dey[†], Volker Kaibel[‡]

March 1, 2014

Abstract

In this work, we introduce and study the forbidden-vertices problem. Given a polytope P and a subset X of its vertices, we study the complexity of linear optimization over the subset of vertices of P that are not contained in X. This problem is closely related to finding the k-best basic solutions to a linear problem. We show that the complexity of the problem changes significantly depending on the encoding of both P and X. We provide additional tractability results and extended formulations when P has binary vertices only. Some applications and extensions to integral polytopes are discussed.

1 Introduction

Given a nonempty rational polytope $P \subseteq \mathbb{R}^n$, we denote by $\operatorname{vert}(P)$, $\operatorname{faces}(P)$, and $\operatorname{facets}(P)$ the sets of vertices, faces, and facets of P, respectively, and we write $f(P) := |\operatorname{facets}(P)|$. We also denote by $\operatorname{xc}(P)$ the extension complexity of P, that is, the minimum number of inequalities in any linear extended formulation of P, i.e., a description of a polyhedron whose image under a linear map is P (see for instance [6].) Finally, given a set $X \subseteq \operatorname{vert}(P)$, we define $\operatorname{forb}(P, X) := \operatorname{conv}(\operatorname{vert}(P) \setminus X)$, where $\operatorname{conv}(S)$ denotes the convex hull of $S \subseteq \mathbb{R}^n$. This work is devoted to understanding the complexity of the forbidden-vertices problem defined below.

Definition 1. *Given a polytope $P \subseteq \mathbb{R}^n$, a set $X \subseteq \operatorname{vert}(P)$, and a vector $c \in \mathbb{R}^n$, the forbidden-vertices problem is to either assert $\operatorname{vert}(P) \setminus X = \varnothing$, or to return a minimizer of $c^\top x$ over $\operatorname{vert}(P) \setminus X$ otherwise.*

Our work is motivated by enumerative schemes for stochastic integer programs [9], where a series of potential solutions are evaluated and discarded from the search space. As we will see later, the problem is also related to finding different basic solutions to a linear program.

To address the complexity of the forbidden-vertices problem, it is crucial to distinguish between different encodings of a polytope.

Definition 2. *An explicit description of a polytope $P \subseteq \mathbb{R}^n$ is a system $Ax \leq b$ defining P. An implicit description of P is a separation oracle which, given a rational vector $x \in \mathbb{R}^n$, either asserts $x \in P$, or returns a valid inequality for P that is violated by x.*

[*] Georgia Institute of Technology and Pontificia Universidad Católica de Chile: gangulo@gatech.edu

[†] Georgia Institute of Technology: sahmed@isye.gatech.edu, santanu.dey@isye.gatech.edu

[‡] Otto-von-Guericke-Universität Magdeburg: kaibel@ovgu.de

Note that an extended formulation for P is a particular case of an implicit description. When P admits a separation oracle that runs in time bounded polynomially in the facet complexity of P and the encoding size of the point to separate, we say that P is tractable. We refer the reader to [19, Section 14] for a deeper treatment of the complexity of linear programming.

We also distinguish different encodings of a set of vertices.

Definition 3. *An explicit description of $X \subseteq \text{vert}(P)$ is the list of the elements in X. If $X = \text{vert}(F)$ for some face F of P, then an implicit description of X is an encoding of P and some valid inequality for P defining F.*

Below we summarize our main contributions.

- In Section 2, we show that the complexity of optimizing over $\text{vert}(P) \setminus X$ or describing $\text{forb}(P, X)$ changes significantly depending on the encoding of P and/or X. In most situations, however, the problem is hard.
- In Section 3 we consider the case of removing a list X of binary vectors from a 0-1 polytope P. When P is the unit cube, we present two compact extended formulations describing $\text{forb}([0,1]^n, X)$. We further extend this result and show that the forbidden-vertices problem is polynomially solvable for tractable 0-1 polytopes.
- Then in Section 4 we apply our results to the k-best problem and to binary all-different polytopes, showing the tractability of both. Finally, in Section 5, we also provide extensions to integral polytopes.

The complexity results of Sections 2 and 3 lead to the classification shown in Table 1, depending on the encoding of P and X, and whether P has 0-1 vertices only or not. Note that $(*)$ is implied, for instance, by Theorem 18. Although we were not able to establish the complexity of $(**)$, Proposition 19 presents a tractable subclass.

		P			
		General		0-1	
		Explicit	Implicit	Explicit	Implicit
X	Explicit	$\mathcal{NP}$-hard (Thm. 11) Polynomial for fixed $\|X\|$ (Prop. 6)	$\mathcal{NP}$-hard for $\|X\| = 1$ (Thm. 9)	Polynomial	Polynomial (Thm. 16)
	Implicit	$\mathcal{NP}$-hard (Prop. 10)	$\mathcal{NP}$-hard $(*)$	$(**)$	$\mathcal{NP}$-hard (Thm. 18)

Table 1: Complexity classification.

In constructing linear extended formulations, disjunctive programming emerges as a practical powerful tool. The lemma below follows directly from [2] and the definition of extension complexity. We will frequently refer to it.

Lemma 4. *Let $P_1, \ldots, P_k$ be nonempty polytopes in $\mathbb{R}^n$. If $P_i = \{x \in \mathbb{R}^n | \; \exists y_i \in \mathbb{R}^{m_i} : \; E_i x + F_i y_i = h_i, \; y_i \geq 0\}$, then $\text{conv}(\cup_{i=1}^k P_i) = \{x \in \mathbb{R}^n | \; \exists x_i \in \mathbb{R}^n, \; y_i \in \mathbb{R}^{m_i}, \; \lambda \in \mathbb{R}^k : \; x = \sum_{i=1}^k x_i, \; E_i x_i + F_i y_i = \lambda_i h_i, \; \sum_{i=1}^k \lambda_i = 1, \; y_i \geq 0, \; \lambda \geq 0\}$. In particular, we have $\text{xc}\left(\text{conv}(\cup_{i=1}^k P_i)\right) \leq \sum_{i=1}^k (\text{xc}(P_i) + 1)$.*

2 General polytopes

We begin with some general results when $P \subseteq \mathbb{R}^n$ is an arbitrary polytope. The first question is how complicated $\text{forb}(P, X)$ is with respect to P.

Proposition 5. *For each n, there exists a polytope $P_n \subseteq \mathbb{R}^n$ and a vertex $v_n \in \text{vert}(P_n)$ such that P_n has $2n+1$ vertices and n^2+1 facets, while $\text{forb}(P_n, \{v_n\})$ has 2^n facets.*

Proof. Let $Q_n := [0,1]^n \cap L$, where $L := \left\{x \in \mathbb{R}^n \middle| \, \mathbf{1}^\top x \leq \frac{3}{2}\right\}$ and $\mathbf{1}$ is the vector of ones. It has been observed [1] that Q_n has $2n+1$ facets and n^2+1 vertices. We translate Q_n and define $Q'_n := Q_n - \frac{1}{n}\mathbf{1} = \left[-\frac{1}{n}, 1-\frac{1}{n}\right]^n \cap L'$, where $L' := \left\{x \in \mathbb{R}^n \middle| \, \mathbf{1}^\top x \leq \frac{1}{2}\right\}$. Since Q'_n is a full-dimensional polytope having the origin in its interior, there is a one-to-one correspondence between the facets of Q'_n and the vertices of its polar $P_n := (Q'_n)^*$ and vice versa. In particular, P_n has n^2+1 facets and $2n+1$ vertices. Let $v \in \text{vert}(P_n)$ be the vertex associated with the facet of Q'_n defined by L'. From polarity, we have $\text{forb}(P_n, \{v\})^* = \left[-\frac{1}{n}, 1-\frac{1}{n}\right]^n$. Thus $\text{forb}(P_n, \{v\})^*$ is a full-dimensional polytope with the origin in its interior and 2^n vertices. By polarity, we obtain that $\text{forb}(P_n, \{v\})$ has 2^n facets. □

Note that the above result only states that $\text{forb}(P, X)$ may need exponentially many inequalities to be described, which does not constitute a proof of hardness. Such a result is provided by Theorem 11 at the end of this section. We first show that $\text{forb}(P, X)$ has an extended formulation of polynomial size in $f(P)$ when both P and X are given explicitly and the cardinality of X is fixed.

Proposition 6. *Suppose $P = \{x \in \mathbb{R}^n | \, Ax \leq b\}$. Using this description of P, and an explicit list of vertices X, we can construct an extended formulation of $\text{forb}(P, X)$ that requires at most $f(P)^{|X|+1}$ inequalities, i.e., $\text{xc}(\text{forb}(P, X)) \leq f(P)^{|X|+1}$.*

Proof. Let $X = \{v_1, \ldots, v_{|X|}\}$ and define $\mathcal{F}_X := \{F_1 \cap \cdots \cap F_{|X|} | \, F_i \in \text{facets}(P), \, v_i \notin F_i, \, i = 1, \ldots, |X|\}$. We claim

$$\text{forb}(P, X) = \text{conv}\left(\cup_{F \in \mathcal{F}_X} F\right).$$

Indeed, let $w \in \text{vert}(P) \setminus X$. For each $i = 1, \ldots, |X|$, there exists $F_i \in \text{facets}(P)$ such that $w \in F_i$ and $v_i \notin F_i$. Therefore, letting $F := F_1 \cap \cdots \cap F_{|X|}$, we have $F \in \mathcal{F}_X$ and $w \in F$, proving the forward inclusion. For the reverse inclusion, consider $F \in \mathcal{F}_X$. By definition, F is a face of P that does not intersect X, and hence $F \subseteq \text{forb}(P, X)$.

By Lemma 4, we have $\text{xc}(\text{forb}(P, X)) \leq \sum_{F \in \mathcal{F}_X} (\text{xc}(F) + 1)$. Since $\text{xc}(F) \leq f(F) \leq f(P) - 1$ for each proper face F of P and $|\mathcal{F}_X| \leq f(P)^{|X|}$, the result follows. □

Note that when $X = \{v\}$, the above result reduces $\text{forb}(P, \{v\})$ to the convex hull of the union of the facets of P that are not incident to v, which is a more intuitive result. Actually, we can expect describing $\text{forb}(P, X)$ to be easier when the vertices in X are "far" thus can be removed "independently", and more complicated when they are "close". Proposition 6 can be refined as follows.

The graph of a polytope P, or the 1-skeleton of P, is a graph G with vertex set $\text{vert}(P)$ such that two vertices are adjacent in G if and only if they are adjacent in P.

Proposition 7. *Let G be the graph of P. Let $X \subseteq \text{vert}(P)$ and let $(X_1, \ldots, X_m)$ be a partition of X such that X_i and X_j are independent in G, i.e., there is no edge connecting X_i to X_j, for all $1 \leq i < j \leq m$. Then*

$$\text{forb}(P, X) = \bigcap_{i=1}^{m} \text{forb}(P, X_i).$$

Proof. We only need to show $\text{forb}(P, X) \supseteq \bigcap_{i=1}^m \text{forb}(P, X_i)$. For this, it is enough to show that for each c we have $\max\{c^\top x : \, x \in \text{forb}(P, X)\} \geq \max\left\{c^\top x : \, x \in \bigcap_{i=1}^m \text{forb}(P, X_i)\right\}$. Given c, let v be an optimal solution to the maximization problem in the right-hand side, and let $W \subseteq \text{vert}(P)$ be the set of vertices

w of P such that $c^\top w \geq c^\top v$. Observe that W induces a connected subgraph of the graph G of P since the simplex method applied to $\max\{c^\top x : \, x \in P\}$ starting from a vertex in W visits elements in W only. Hence, due to the independence of $X_1, \ldots, X_m$, either there is some $w \in W$ with $w \notin X_1 \cup \cdots \cup X_m$, in which case we have $w \in \text{forb}(P, X)$ and $c^\top w \geq c^\top v$ as desired, or $W \subseteq X_i$ for some i, which yields the contradiction $v \in \text{forb}(P, X_i) \subseteq \text{forb}(P, W)$ with $c^\top x < c^\top v$ for all $x \in \text{vert}(P) \setminus W$. □

Conversely, we may be tempted to argue that if $\text{forb}(P, X) = \text{forb}(P, X_1) \cap \text{forb}(P, X_2)$, then X_1 and X_2 are "far". However, this is not true in general. For instance, consider P being a simplex. Then any $X \subseteq \text{vert}(P)$ is a clique in the graph of P, and yet $\text{forb}(P, X) = \text{forb}(P, X_1) \cap \text{forb}(P, X_2)$ for any partition (X_1, X_2) of X.

Proposition 7 generalizes the main result of [12] regarding cropped cubes. Moreover, the definition of being "croppable" in [12] in the case of the unit cube coincides with the independence property of Proposition 7.

Recall that a vertex of an n-dimensional polytope is simple if it is contained in exactly n facets. Proposition 7 also implies the following well-known fact.

Corollary 8. *If X is independent in the graph of P and all its elements are simple, then*

$$\text{forb}(P, X) = P \cap \bigcap_{v \in X} H_v,$$

where H_v is the half-space defined by the n neighbors of v that does not contain v.

Proof. The result follows from Proposition 7 since, as X is simple, we have $\text{forb}(P, \{v\}) = P \cap H_v$ for any $v \in X$. □

Observe that when P is given by an extended formulation or a separation oracle, $f(P)$ may be exponentially large with respect to the size of the encoding, and the bound given in Proposition 6 is not interesting. In fact, in this setting and using recent results on the extension complexity of the cut polytope [5], we show that removing a single vertex can render an easy problem hard.

Let $K_n = (V_n, E_n)$ denote the complete graph on n nodes. We denote by $\text{CUT}(n)$, $\text{CUT}^0(n)$, and st-$\text{CUT}(n)$ the convex hull of the characteristic vectors of cuts, nonempty cuts, and st-cuts of K_n, respectively.

Theorem 9. *For each n, there exists a set $S_n \subseteq \mathbb{R}^{n(n-1)/2}$ with $|S_n| = 2^{n-1} + n - 1$ and a point $v_n \in S_n$ such that linear optimization over S_n can be done in polynomial time and $\text{xc}(\text{conv}(S_n))$ is polynomially bounded, but linear optimization over $S_n \setminus \{v_n\}$ is $\mathcal{NP}$-hard and $\text{xc}(\text{conv}(S_n \setminus \{v_n\}))$ grows exponentially.*

Proof. Let $T_n := \{n^2 \mathbf{1}_e | \, e \in E_n\}$, where $\mathbf{1}_e$ is the e-th unit vector, and define $S_n := \text{vert}\left(\text{CUT}^0(n)\right) \cup T_n$.

We have that linear optimization over S_n can be done in polynomial time. To see this, suppose we are minimizing $c^\top x$ over S_n. Let x^T and x^C be the best solution in T_n and $\text{CUT}^0(n)$, respectively. Note that computing x^T is trivial, and if c has a negative component, then x^T is optimal. Otherwise, c is nonnegative and x^C can be found with a max-flow/min-cut algorithm. Then the best solution among x^T and x^C is optimal. Now, consider the dominant of $\text{CUT}^0(n)$ defined as $\text{CUT}^0(n)_+ := \text{CUT}^0(n) + \mathbb{R}_+^{n(n-1)/2}$. From [4], we have that $\text{CUT}^0(n)_+$ is an unbounded polyhedron having the same vertices as $\text{CUT}^0(n)$, and moreover, it has an extended formulation of polynomial size in n. Let $L := \{x \in \mathbb{R}^{n(n-1)/2} | \, \sum_{e \in E_n} x_e \leq n^2\}$. Then $\text{CUT}^0(n)_+ \cap L$ is a polytope having two classes of vertices: those corresponding to vert $\left(\text{CUT}^0(n)\right)$ and those belonging to the hyperplane defining L. Let W be the latter set. Since conv$(W) \subseteq$ conv(T_n), we obtain conv(S_n) = conv $\left(\text{CUT}^0(n) \cup T_n\right)$ = conv $\left((\text{CUT}^0(n) \cup W) \cup T_n)\right)$ = conv $\left((\text{CUT}^0(n)_+ \cap L) \cup T_n\right)$. Applying disjunctive programming in the last expression yields a compact extended formulation for conv(S_n).

Now, let v_n be any point from T_n, say the one corresponding to $\{s,t\} \in E$. We claim that linear optimization over $S_n \setminus \{v_n\}$ is $\mathcal{NP}$-hard. To prove this, consider an instance of $\max\{c^\top x |\ x \in st\text{-CUT}(n)\}$, where c is a positive vector. Let $\bar{c} := \max\{c_e |\ e \in E\}$. Let d be obtained from c as

$$d_e = \begin{cases} c_e & e \neq \{s,t\} \\ c_e + \bar{c}n^2 & e = \{s,t\} \end{cases}$$

and consider the problem $\max\{d^\top x |\ x \in S_n \setminus \{v_n\}\}$. We have that every optimal solution to this problem must satisfy $x_{st} = 1$. Indeed, if $x \in T_n \setminus \{v_n\}$, then for some $e \in E_n \setminus \{\{s,t\}\}$ we have $d^\top x = d_e x_e = c_e n^2$. If $x \in$ vert(CUT$^0(n)$) is not an st-cut, then $x_{st} = 0$ and thus $d^\top x \leq \bar{c}n^2$. On the other hand, if x is an st-cut, then $x_{st} = 1$ and thus $d^\top x \geq d_{st}x_{st} = c_{st} + \bar{c}n^2$. Therefore $x_{st} = 1$ in any optimal solution, and in particular, such a solution must define an st-cut of maximum weight. Finally, since $x_{st} \leq 1$ defines a face of conv$(S_n \setminus \{v_n\})$ and conv$(S_n \setminus \{v_n\}) \cap \{x \in \mathbb{R}^{n(n-1)/2} |\ x_{st} = 1\} = st$-CUT$(n)$, we conclude that xc(conv$(S_n \setminus \{v_n\})$) is exponential in n, for otherwise applying disjunctive programming over all pairs of nodes s and t would yield an extended formulation for CUT(n) of polynomial size, contradicting the results in [5]. □

Contrasting Proposition 6 and Theorem 9 shows that the complexity of forb(P, X) depends on the encoding of P. On the other hand, in all cases analyzed so far, X has been explicitly given as a list. Now we consider the case where X = vert(F) for some face F of P.

Proposition 10. *Given a polytope $P \subseteq \mathbb{R}^n$ and a face F, both described in terms of the linear inequalities defining them, optimizing a linear function over* vert$(P) \setminus$ vert(F) *is $\mathcal{NP}$-hard. Moreover,* xc(conv(vert$(P) \setminus$ vert(F))) *cannot be polynomially bounded in the encoding length of the inequality description of P and thus not in n.*

Proof. Let $a \in \mathbb{Z}^n_+$ and $b \in \mathbb{Z}_+$, and consider the binary knapsack set $S := \{x \in \{0,1\}^n |\ a^\top x \leq b\}$. Let $P := \{x \in [0,1]^n |\ 2a^\top x \leq 2b + 1\}$ and note that $S = P \cap \mathbb{Z}^n$. It is straightforward to verify that $x \in$ vert(P) is fractional if and only if $2a^\top x = 2b+1$. Then, if F is the facet of P defined by the previous constraint, we have S = vert$(P) \setminus$ vert(F). The second part of the statement is a direct consequence of [17] using multipliers 4^i as discussed after Remark 3.4 of that reference. □

It follows from Theorem 9 and Proposition 10 that only when P and X are explicitly given there is hope for efficient optimization over forb(P, X).

In a similar vein, when the linear description of P is provided, we can consider the vertex-enumeration problem, which consists of listing all the vertices of P. We say that such a problem is solvable in polynomial time if there exists an algorithm that returns the list in time bounded by a polynomial of n, $f(P)$, and the output size |vert(P)|. In [8] it is shown that given a partial list of vertices, the decision problem "is there another vertex?" is $\mathcal{NP}$-hard for (unbounded) polyhedra, and in [3] this result is strengthened to polyhedra having 0-1 vertices only. Building on these results, we show hardness of the forbidden-vertices problem (Def. 1) for general polytopes.

Theorem 11. *The forbidden-vertices problem is $\mathcal{NP}$-hard, even if both P and X are explicitly given.*

Proof. Let $Q = \{x \in \mathbb{R}^n :\ Ax = b,\ x \geq 0\}$ be an unbounded polyhedron such that vert$(Q) \subseteq \{0,1\}^n$. In [3], it is shown that given the linear description of Q and a list $X \subseteq$ vert(Q), it is $\mathcal{NP}$-hard to decide whether $X \neq \text{vert}(Q)$. Let P be the polytope obtained by intersecting Q with the half-space defined by $\sum_{i=1}^{n} x_i \leq n+1$, and let F be the facet of P associated with this constraint. Then we have $\text{vert}(P) = \text{vert}(Q) \cup \text{vert}(F)$, $\sum_{i=1}^{n} x_i \leq n$ for $x \in \text{vert}(Q)$, and $\sum_{i=1}^{n} x_i = n+1$ for $x \in \text{vert}(F)$. Now, given the description of P and a list $X \subseteq \text{vert}(Q) \subseteq \text{vert}(P)$, consider the instance of the forbidden-vertices problem $\min\{\sum_{i=1}^{n} x_i : \ x \in \text{vert}(P) \setminus X\}$. The optimal value is equal to $n+1$ if and only if $X = \text{vert}(Q)$. Since the reduction is clearly polynomial, the result follows. □

In fact, it also follows from [3] that the forbidden-vertices problem for general polytopes becomes hard already for $|X| = n$. Fortunately, the case of 0-1 polytopes is amenable to good characterizations.

3 0-1 polytopes

We consider polytopes having binary vertices only. We show that $\text{forb}(P, X)$ is tractable as long as P is and X is explicitly given. Our results for $P = [0,1]^n$ allow us to obtain tractability in the case of general 0-1 polytopes.

3.1 The 0-1 cube

In this subsection we have $P = [0,1]^n$, and therefore $\text{vert}(P) = \{0,1\}^n$. We show the following result.

Theorem 12. *Let X be a list of n-dimensional binary vectors. Then* $\text{xc}(\text{forb}([0,1]^n, X)) \leq \mathcal{O}(n|X|)$.

For this, we present two extended formulations involving $\mathcal{O}(n|X|)$ variables and constraints. The first one is based on an identification between nonnegative integers and binary vectors. The second one is built by recursion and lays ground for a simple combinatorial algorithm to optimize over $\text{forb}([0,1]^n, X)$ and for an extension to remove vertices from general 0-1 polytopes.

3.1.1 First extended formulation

Let $N := \{1, \ldots, n\}$ and $\mathcal{N} := \{0, \ldots, 2^n - 1\}$. There exists a bijection between $\{0,1\}^n$ and $\mathcal{N}$ given by the mapping $\sigma(v) := \sum_{i \in N} 2^{i-1} v_i$ for all $v \in \{0,1\}^n$. Therefore, we can write $\{0,1\}^n = \{v^0, \ldots, v^{2^n-1}\}$, where v^k gives the binary expansion of k for each $k \in \mathcal{N}$, that is, $v^k = \sigma^{-1}(k)$. Let $X = \{v^{k_1}, \ldots, v^{k_m}\}$, where without loss of generality we assume $k_l < k_{l+1}$ for all $l = 1, \ldots, m-1$. Also, let $\mathcal{N}_X := \{k \in \mathcal{N} |\ v^k \in X\}$. Then we have

$$\{0,1\}^n \setminus X = \left\{ x \in \{0,1\}^n |\ \sum_{i \in N} 2^{i-1} x_i \notin \mathcal{N}_X \right\}.$$

Now, for integers a and b, let

$$K(a,b) = \left\{ x \in \{0,1\}^n |\ a \leq \sum_{i \in N} 2^{i-1} x_i \leq b \right\}.$$

If $b < a$, then $K(a,b)$ is empty. Set $k_0 = -1$ and $k_{m+1} = 2^n$. Then we can write

$$\{0,1\}^n \setminus X = \bigcup_{l=0}^{m} K(k_l + 1, k_{l+1} - 1).$$

Thus

$$\text{forb}([0,1]^n, X) = \text{conv}\left(\bigcup_{l=0}^{m} K(k_l+1, k_{l+1}-1)\right) = \text{conv}\left(\bigcup_{l=0}^{m} \text{conv}(K(k_l+1, k_{l+1}-1))\right). \tag{1}$$

For $k \in \mathcal{N}$, let $N^k := \{i \in N |\ v_i^k = 1\}$. From [15] we have

$$\text{conv}(K(a,b)) = \left\{ x \in [0,1]^n : \begin{array}{ll} \displaystyle\sum_{j \notin N^a |\ j>i} x_j \geq 1 - x_i & \forall i \in N^a \\ \displaystyle\sum_{j \in N^b |\ j>i} (1 - x_j) \geq x_i & \forall i \notin N^b \end{array} \right\},$$

thus $\text{conv}(K(a,b))$ has $\mathcal{O}(n)$ facets. Finally, combining this and (1), by Lemma 4, we have that $\text{forb}([0,1]^n, X)$ can be described by an extended formulation having $\mathcal{O}(n|X|)$ variables and constraints.

3.1.2 Second extended formulation

Given $X \subseteq \{0,1\}^n$, let X' denote the projection of X onto the first $n-1$ coordinates. Also, let $\widehat{X} := \widetilde{X} \setminus X$, where $\widetilde{X}$ is constructed from X by flipping the last coordinate of each of its elements. The result below is key in giving a recursive construction of $\text{forb}([0,1]^n, X)$.

Proposition 13. $\{0,1\}^n \setminus X = \left[\left(\{0,1\}^{n-1} \setminus X'\right) \times \{0,1\}\right] \cup \widehat{X}$.

Proof. Given $v \in \{0,1\}^n$, let $v' \in \{0,1\}^{n-1}$ and $\widetilde{v} \in \{0,1\}^n$ be the vectors obtained from v by removing and by flipping its last coordinate, respectively.

Let $v \in \{0,1\}^n \setminus X$. If $\widetilde{v} \in X$, since $v \notin X$, we have $v \in \widehat{X}$. Otherwise $v' \notin X'$, and thus $v \in (\{0,1\}^{n-1} \setminus X') \times \{0,1\}$.

For the converse, note that $\widehat{X} \subseteq \{0,1\}^n \setminus X$. Finally, if $v \in (\{0,1\}^{n-1} \setminus X') \times \{0,1\}$, then $v' \notin X'$ and thus $v \notin X$. □

The second proof of Theorem 12 follows from Proposition 13 by induction. Suppose that $\text{forb}([0,1]^{n-1}, X')$ has an extended formulation with at most $(n-1)(|X'|+4)$ inequalities, which holds for $n = 2$. Then we can describe $\text{forb}([0,1]^{n-1}, X') \times \{0,1\}$ using at most $(n-1)(|X'|+4)+2$ inequalities. Since the polytope $\text{conv}(\widehat{X})$ requires at most $|\widehat{X}|$ inequalities in an extended formulation, we obtain an extended formulation for $\text{forb}([0,1]^n, X)$ of size no more than $[(n-1)(|X'|+4)+2+1] + [|\widehat{X}|+1] \leq n(|X|+4)$.

3.2 General 0-1 polytopes

In this subsection we analyze the general 0-1 case. We show that the encoding of X plays an important role in the complexity of the problem.

3.2.1 Explicit X

In order to prove tractability of the forbidden vertices problem corresponding to general 0-1 tractable polytopes, we introduce the notion of X-separating faces for the 0-1 cube.

Definition 14. *Given $X \subseteq \{0,1\}^n$, we say that $\mathcal{F} \subseteq \text{faces}([0,1]^n)$ is X-separating if $\{0,1\}^n \setminus X = \cup_{F\in\mathcal{F}} F \cap \{0,1\}^n$. We denote by $\mu(X)$ the minimal cardinality of an X-separating set.*

Clearly, if $\mathcal{F}$ is X-separating, then

$$\min\left\{c^\top x \middle| \, x \in \{0,1\}^n \setminus X\right\} = \min_{F\in\mathcal{F}} \min\left\{c^\top x \middle| \, x \in F \cap \{0,1\}^n\right\}.$$

Thus, if we can find an X-separating family of cardinality bounded by a polynomial on n and $|X|$, then we can optimize in polynomial time over $\{0,1\}^n \setminus X$ by solving the inner minimization problem for each $F \in \mathcal{F}$ and then picking the smallest value.

Proposition 15. *For every nonempty set $X \subseteq \{0,1\}^n$, we have $\mu(X) \leq n|X|$.*

Proof. For each $y \in \{0,1\}^n \setminus X$, let $0 \leq k \leq n-1$ be the size of the longest common prefix between y and any element of X, and consider the face $F = F(y) := \{x \in [0,1]^n | \, x_i = y_i \; \forall 1 \leq i \leq k+1\} = (y_1, \ldots, y_k, y_{k+1}) \times [0,1]^{n-k-1}$. Then the collection $\mathcal{F} := \{F(y) | \, y \in \{0,1\}^n \setminus X\}$ is X-separating since any $y \in \{0,1\}^n \setminus X$ belongs to $F(y)$ and no element of X lies in any $F(y)$ by maximality of k. Clearly, $|\mathcal{F}| \leq n|X|$ since each face in $\mathcal{F}$ is of the form $(v_1, \ldots, v_k, 1 - v_{k+1}) \times [0,1]^{n-k-1}$ for some $v \in X$. □

In other words, letting X^i be the projection of X onto the first i components and $\widehat{X}^i := (X^{i-1} \times \{0,1\}) \setminus X^i$, where $\widehat{X}^1 := \{0,1\} \setminus X^1$, we have

$$\{0,1\}^n \setminus X = \bigcup_{i=1}^{n} \left[\widehat{X}^i \times \{0,1\}^{n-i}\right].$$

Moreover, it also follows from the proof of Proposition 15 that $\mu(X)$ is at most the number of neighbors of X since if $(v_1, \ldots, v_k, 1 - v_{k+1}, v_{k+2}, \ldots, v_n)$ is a neighbor of $v \in X$ that also lies in X, then the face $\{(v_1, \ldots, v_k, 1 - v_{k+1})\} \times [0,1]^{n-k-1}$ in not included in $\mathcal{F}$ in the construction above.

Now, let $P \subseteq \mathbb{R}^n$ be an arbitrary 0-1 polytope. Note that $\text{vert}(P) \setminus X = \text{vert}(P) \cap (\{0,1\}^n \setminus X)$. On the other hand, if $\mathcal{F} \subseteq \text{faces}([0,1]^n)$ is X-separating, then $\{0,1\}^n \setminus X = \cup_{F\in\mathcal{F}} F \cap \{0,1\}^n$. Combining these two expressions, we get

$$\text{vert}(P) \setminus X = \bigcup_{F\in\mathcal{F}} \text{vert}(P) \cap F \cap \{0,1\}^n = \bigcup_{F\in\mathcal{F}} P \cap F \cap \{0,1\}^n.$$

Note that since P has 0-1 vertices and F is a face of the unit cube, then $P \cap F$ is a 0-1 polytope. Moreover, if P is tractable, so is $P \cap F$. Recalling that $\mu(X) \leq n|X|$ from Proposition 15, we obtain

Theorem 16. *If $P \subseteq \mathbb{R}^n$ is a tractable 0-1 polytope, then the forbidden-vertices problem is polynomially solvable.*

In fact, a compact extended formulation for $\text{vert}(P) \setminus X$ is available when P has one.

Proposition 17. *For every 0-1 polytope P and for every nonempty set $X \subseteq \text{vert}(P)$, we have*

$$\text{xc}(\text{forb}(P, X)) \leq \mu(X)(\text{xc}(P) + 1).$$

Proof. The result follows from

$$\operatorname{forb}(P, X) = \operatorname{conv}\left(\bigcup_{F\in\mathcal{F}} P \cap F \cap \{0,1\}^n\right) = \operatorname{conv}\left(\bigcup_{F\in\mathcal{F}} F\right),$$

Lemma 4, and $\operatorname{xc}(F) \leq \operatorname{xc}(P)$ for any face F of P. □

Observe that when P is tractable but its facet description is not provided, Theorem 16 is in contrast to Theorem 9. Having all vertices with at most two possible values for each component is crucial to retain tractability when X is given as a list. However, when X is given by a face of P, the forbidden-vertices problem can become intractable even in the 0-1 case.

3.2.2 Implicit X

Let TSP(n) denote the convex hull of the characteristic vectors of Hamiltonian cycles in the complete graph K_n. Also, let SUB(n) denote the subtour-elimination polytope for K_n with edge set E_n.

Theorem 18. *For each n, there exists a 0-1 polytope $P_n \subseteq \mathbb{R}^{n(n-1)/2}$ and a facet $F_n \in \operatorname{facets}(P_n)$ such that linear optimization over P_n can be done in polynomial time and $\operatorname{xc}(P_n)$ is polynomially bounded, but linear optimization over $\operatorname{vert}(P_n) \setminus \operatorname{vert}(F_n)$ is $\mathcal{NP}$-hard and $\operatorname{xc}(\operatorname{forb}(P_n, \operatorname{vert}(F_n)))$ grows exponentially.*

Proof. Given a positive integer n, consider $T_n^+ := \{x \in \{0,1\}^{E_n} | \sum_{e\in E_n} x_e = n+1\}$, $T_n^- := \{x \in \{0,1\}^{E_n} | \sum_{e\in E_n} x_e = n-1\}$, and $H_n :=$ TSP(n) $\cap \{0,1\}^{E_n}$. The idea is to "sandwich" H_n between T_n^- and T_n^+ to obtain tractability, and then remove T_n^- to obtain hardness.

We first show that linear optimization over $T_n^- \cup H_n \cup T_n^+$ is polynomially solvable. Given $c \in \mathbb{R}^{n(n-1)/2}$, consider $\max\{c^\top x | \, x \in T_n^- \cup H_n \cup T_n^+\}$. Let x^- and x^+ be the best solution in T_n^- and T_n^+, respectively, and note that x^- and x^+ are trivial to find. Let m be the number of nonnegative components of c. If $m \geq n+1$, then x^+ is optimal. If $m \leq n-1$, then x^- is optimal. If $m = n$, let $x^n \in \{0,1\}^{E_n}$ have a 1 at position e if and only if $c_e \geq 0$. If x^n belongs to H_n, which is easy to verify, then it is optimal. Otherwise either x^- or x^+ is an optimal solution.

Now we show that linear optimization over $H_n \cup T_n^+$ is $\mathcal{NP}$-hard. Given $c \in \mathbb{R}^{n(n-1)/2}$ with $c > 0$, consider $\min\{c^\top x | \, x \in H_n\}$. Let $\bar{c} := \max\{c_e | \, e \in E_n\}$ and define $d_e := c_e + n\bar{c}$. Consider $\min\{d^\top x | \, x \in H_n \cup T_n^+\}$. For any $x \in T_n^+$, we have $d^\top x = (n+1)n\bar{c} + c^\top x > (n+1)n\bar{c}$. For any $x \in H_n$, we have $d^\top x = n^2\bar{c} + c^\top x \leq n^2\bar{c} + n\bar{c} = (n+1)n\bar{c}$. Hence, the optimal solution to the latter problem belongs to H_n and defines a tour of minimal length with respect to c.

Letting $P_n := \operatorname{conv}(T_n^- \cup H_n \cup T_n^+)$, we have that P_n is a tractable 0-1 polytope, $\sum_{e\in E_n} x_e \geq n-1$ defines a facet F_n of P_n, and $\operatorname{vert}(P_n) \setminus \operatorname{vert}(F_n) = H_n \cup T_n^+$, which is an intractable set. Now, since $\operatorname{forb}(P_n, \operatorname{vert}(F_n)) = \operatorname{conv}(H_n \cup T_n^+)$, we have that $\sum_{e\in E_n} x_e \geq n$ defines a facet of $\operatorname{forb}(P_n, \operatorname{vert}(F_n))$ and $\operatorname{forb}(P_n, \operatorname{vert}(F_n)) \cap \{x \in \mathbb{R}^{n(n-1)/2} | \sum_{e\in E_n} x_e = n\} =$ TSP(n). Therefore, $\operatorname{xc}(\operatorname{forb}(P_n, \operatorname{vert}(F_n)))$ is exponential in n [18]. It remains to show that $\operatorname{xc}(P_n)$ is polynomial in n.

Let $T_n := \{x \in \{0,1\}^{E_n} | \sum_{e\in E_n} x_e = n\}$ and let $\overline{H}_n := T_n \setminus H_n$ be the set of incidence vectors of n-subsets of E_n that do not define a Hamiltonian cycle. Given $x \in \{0,1\}^{E_n}$, let $N(x)$ be the set of neighbors of x in $[0,1]^{E_n}$, let $L(x)$ be the half-space spanned by $N(x)$ that does not contain x, and let $C(x) := [0,1]^{E_n} \setminus L(x)$. Finally, let $\Delta_n := \operatorname{conv}(T_n^- \cup T_n \cup T_n^+)= \{x \in [0,1]^{E_n} | \, n-1 \leq \sum_{e\in E_n} x_e \leq n+1\}$.

We claim that $P_n = \operatorname{conv}(T_n^- \cup \text{SUB}(n) \cup T_n^+)$. By definition, we have $P_n \subseteq \operatorname{conv}(T_n^- \cup \text{SUB}(n) \cup T_n^-)$. To show the reverse inclusion, it suffices to show SUB(n) $\subseteq P_n$. Note that any two distinct elements in

T_n can have at most $|E_n| - 2$ tight inequalities in common from those defining Δ_n. Thus, T_n defines an independent set in the graph of Δ_n. Moreover, for each $x \in T_n$ the set of neighbors in Δ_n is $N(x)$ and thus all vertices in T_n are simple. As $\overline{H}_n \subseteq T_n$, we have that $\overline{H}_n$ is simple and independent, and by Corollary 8 we have

$$P_n = \Delta_n \cap \bigcap_{x \in \overline{H}_n} L(x) = \Delta_n \setminus \bigcup_{x \in \overline{H}_n} C(x).$$

Since $\text{SUB}(n) \subseteq \Delta_n$, from the second equation above, it suffices to show $C(x) \cap \text{SUB}(n) = \varnothing$ for all $x \in \overline{H}_n$. For this, note that for any $x \in \overline{H}_n$, there exists a set $\varnothing \neq S \subsetneq V_n$ such that $x(\delta(S)) \leq 1$, which implies $y(\delta(S)) \leq 2$ for all $y \in N(x)$. Thus $C(x) \cap \text{SUB}(n) = \varnothing$ as $x(\delta(S)) \geq 2$ is valid for $\text{SUB}(n)$.

Finally, applying disjunctive programming and since $\text{xc}(\text{SUB}(n))$ is polynomial in n [20], we conclude that P_n has an extended formulation of polynomial size. □

To conclude this section, consider the case where P is explicitly given and X is given as a facet of P. Although we are unable to establish the complexity of the forbidden-vertices problem in this setting, we present a tractable case and discuss an extension.

Proposition 19. *Let $P = \{x \in \mathbb{R}^n | \ Ax \leq b\}$ be a 0-1 polytope, where A is TU and b is integral. Let F be the face of P defined by $a_i^\top x = b_i$. Then*

$$\text{forb}(P, \text{vert}(F)) = P \cap \{x \in \mathbb{R}^n | \ a_i^\top x \leq b_i - 1\}.$$

Proof. We have

$$\text{vert}(P) \setminus \text{vert}(F) = P \cap \{x \in \{0,1\}^n | \ a_i^\top x \leq b_i - 1\}.$$

Since A is TU and b in integral, the set $P \cap \{x \in \mathbb{R}^n | \ a_i^\top x \leq b_i - 1\}$ is an integral polyhedron contained in P, which is a 0-1 polytope. □

Since any face is the intersection of a subset of facets, the above result implies that removing a single face can be efficiently done by disjunctive programming in the context of Proposition 19. Also, if we want to remove a list of facets, that is, $X = \cup_{F \in \mathcal{F}} \text{vert}(F)$ and $\mathcal{F}$ is a subset of the facets of P, then we can solve the problem by removing one facet at a time. However, if $\mathcal{F}$ is a list of faces, then the problem becomes hard in general.

Proposition 20. *If $\mathcal{F}$ is a list of faces of $[0,1]^n$, then optimizing a linear function over $\{0,1\}^n \setminus \cup_{F \in \mathcal{F}} \text{vert}(F)$ is $\mathcal{NP}$-hard.*

Proof. Let $G = (V, E)$ be a graph. Consider the problem of finding a minimum cardinality vertex cover of G, which can be formulated as

$$\begin{array}{rll} \min & \sum_{i \in V} x_i & \\ \text{s.t.} & x_i + x_j \geq 1 & \forall \{i,j\} \in E \\ & x_i \in \{0,1\} & \forall i \in V. \end{array}$$

Construct $\mathcal{F}$ by adding a face of the form $F = \{x \in [0,1]^n | \ x_i = 0, \ x_j = 0\}$ for each $\{i,j\} \in E$. Then the vertex cover problem, which is $\mathcal{NP}$-hard, reduces to optimization of a linear function over $\{0,1\}^n \setminus \cup_{F \in \mathcal{F}} \text{vert}(F)$. □

4 Applications

4.1 k-best solutions

The k-best problem defined below is closely related to removing vertices.

Definition 21. *Given a nonempty 0-1 polytope $P \subseteq \mathbb{R}^n$, a vector $c \in \mathbb{R}^n$, and a positive integer k, the k-best problem is to either assert $|\text{vert}(P)| \leq k$ and return $\text{vert}(P)$, or to return $v_1, \ldots, v_k \in \text{vert}(P)$, all distinct, such that $\max\{c^\top v_i |\ i = 1, \ldots, k\} \leq \min\{c^\top v |\ v \in \text{vert}(P) \setminus \{v_1, \ldots, v_k\}\}$.*

Since we can sequentially remove vertices from 0-1 polytopes, we can prove the following.

Proposition 22. *Let $P \subseteq [0,1]^n$ be a tractable 0-1 polytope. Then, for any $c \in \mathbb{R}^n$, the k-best problem can be solved in polynomial time on k and n.*

Proof. For each $i = 1, \ldots, k$, solve the problem

$$
\begin{array}{lll}
(\mathcal{P}_i) & \min & c^\top x \\
& \text{s.t.} & x \in P_i,
\end{array}
$$

where $P_1 := P$, $P_i := \text{forb}(P_{i-1}, \{v_{i-1}\}) = \text{forb}(P, \{v_1, \ldots, v_{i-1}\})$ for $i = 2, \ldots, k$, and $v_i \in \text{vert}(P_i)$ is an optimal solution to $(\mathcal{P}_i)$, if one exists, for $i = 1, \ldots, k$. From Theorem 16, we can solve each of these problems in polynomial time. In particular, if $(\mathcal{P}_i)$ is infeasible, we return $v_1, \ldots, v_{i-1}$. Otherwise, by construction, $v_1, \ldots, v_k$ satisfy the required properties. Clearly, the construction is done in polynomial time. □

The above complexity result was originally obtained in [10] building on ideas from [16] by applying a branch-and-fix scheme.

4.2 Binary all-different polytopes

With edge-coloring of graphs in mind, the binary all-different polytope has been introduced in [11]. It was furthermore studied in [14] and [13]. We consider a more general setting.

Definition 23. *Given a positive integer k, nonempty 0-1 polytopes $P_1, \ldots, P_k$ in $\mathbb{R}^n$, and vectors $c_1, \ldots, c_k \in \mathbb{R}^n$, the binary all-different problem is to solve*

$$
\begin{array}{llll}
(\mathcal{P}) & \min & \sum_{i=1}^k c_i^\top x_i & \\
& \text{s.t.} & x_i \in \text{vert}(P_i) & i = 1, \ldots, k \\
& & x_i \neq x_j & 1 \leq i < j \leq k.
\end{array}
$$

In [11], it was asked whether the above problem is polynomially solvable in the case $P_i = [0,1]^n$ for all $i = 1, \ldots, k$. Using the tractability of the k-best problem, we give a positive answer even for the general case of distinct polytopes.

Given a graph $G = (V, E)$ and $U \subseteq V$, a U-matching in G is a matching $M \subseteq E$ such that each vertex in U is contained in some element of M.

Theorem 24. *If $P_i \subseteq \mathbb{R}^n$ is a tractable nonempty 0-1 polytope for $i = 1, \ldots, k$, then the binary all-different problem is polynomially solvable.*

Proof. For each $i = 1, \ldots, k$, let S_i be the solution set of the k-best problem (Def. 21) for P_i and c_i. Observe that $|S_i| \leq k$. Now, consider the bipartite graph $G = (S \cup R, E)$, where $S := \cup_{i=1}^{k} S_i$ and $R := \{1, \ldots, k\}$. For each $v \in S$ and $i \in R$, we include the arc $\{v, i\}$ in E if and only if $v \in S_i$. Finally, for each $\{v, i\} \in E$, we set $w_{vi} := c_i^\top v$.

We claim that $(\mathcal{P})$ reduces to finding an R-matching in G of minimum weight with respect to w. It is straightforward to verify that an R-matching in G defines a feasible solution to $(\mathcal{P})$ of equal value. Thus, it is enough to show that if $(\mathcal{P})$ is feasible, then there exists an R-matching with the same optimal value. Indeed, let $(x_1, \ldots, x_k)$ be an optimal solution to $(\mathcal{P})$ that does not define an R-matching, that is, such that $x_i \notin S_i$ for some $i = 1, \ldots, k$. Then, we must have $|\text{vert}(P_i)| > k$ and $|S_i| = k$. This latter condition and $x_i \notin S_i$ imply the existence of $v \in S_i$ such that $v \neq x_j$ for all $j = 1, \ldots, k$. Furthermore, by the definition of S_i, we also have $c_i^\top v \leq c_i^\top x_i$. Therefore, the vector $(x_1, \ldots, x_{i-1}, v, x_{i+1}, \ldots, x_k)$ is an optimal solution to $(\mathcal{P})$ having its i-th subvector in S_i. Iteratively applying the above reasoning to all components, we obtain an optimal solution to $(\mathcal{P})$ given by an R-matching as desired. □

5 Extension to integral polytopes

In this section, we generalize the forbidden-vertices problem to integral polytopes, that is, to polytopes having integral extreme points, even allowing the removal of points that are not vertices. We show that for an important class of integral polytopes the resulting problem is tractable.

For an integral polytope $P \subseteq \mathbb{R}^n$ and $X \subseteq P \cap \mathbb{Z}^n$, we define $\text{forb}_I(P, X) := \text{conv}((P \cap \mathbb{Z}^n) \setminus X)$.

Definition 25. *Given an integral polytope $P \subseteq \mathbb{R}^n$, a set $X \subseteq P \cap \mathbb{Z}^n$ of integral vectors, and a vector $c \in \mathbb{R}^n$, the forbidden-vectors problem asks to either assert $(P \cap \mathbb{Z}^n) \setminus X = \varnothing$, or to return a minimizer of $c^\top x$ over $(P \cap \mathbb{Z}^n) \setminus X$ otherwise.*

Given vectors $l, u \in \mathbb{R}^n$ with $l \leq u$, we denote $[l, u] := \{x \in \mathbb{R}^n | \; l_i \leq x_i \leq u_i, i = 1, \ldots, n\}$. We term these sets as boxes.

Definition 26. *An integral polytope $P \subseteq \mathbb{R}^n$ is box-integral if for any pair of vectors $l, u \in \mathbb{Z}^n$ with $l \leq u$, the polytope $P \cap [l, u]$ is integral.*

Polytopes defined by a TU matrix and an integral right-hand-side, or by a box-TDI system, are examples of box-integral polytopes. Further note that if P is tractable and box-integral, so is $P \cap [l, u]$. When both conditions are met, we say that P is box-tractable.

With arguments analogous to that of the 0-1 case, we can verify the following result.

Theorem 27. *If $P \subseteq \mathbb{R}^n$ is a box-tractable polytope, then, given a list $X \subseteq P \cap \mathbb{Z}^n$, the forbidden-vectors problem is polynomially solvable. Moreover,*

$$\text{xc}(\text{forb}_I(P, X)) \leq 2n|X|(\text{xc}(P) + 1).$$

Proof. Since P is bounded, it is contained in a box. Without lost of generality and to simplify the exposition, we may assume that $P \subseteq [0, r-1]^n$ for some $r \geq 2$. As in the 0-1 case, we first address the case $P = [0, r-1]^n$, for which we provide two extended formulations for $\text{forb}_I(P, X)$ involving $\mathcal{O}(n|X|)$ variables and constraints.

The first extended formulation relies on the mapping $\phi(x) := \sum_{i=1}^{n} r^{i-1}x_i$ for $x \in [0, r-1]^n$, which defines a bijection with $\{0, \dots, r^n - 1\}$. Letting $K_r(a,b) := \{x \in \{0, \dots, r-1\}^n |\ a \leq \phi(x) \leq b\}$, we have that $\text{forb}_I(P, X)$ is the convex hull of the union of at most $|X| + 1$ sets of the form $K_r(a,b)$. Since $\text{conv}(K_r(a,b))$ has $\mathcal{O}(n)$ facets [7], by disjunctive programming we obtain an extended formulation for $\text{forb}_I(P, X)$ having $\mathcal{O}(n|X|)$ inequalities.

For the second extended formulation, let X' denote the projection of X onto the first $n-1$ coordinates and set $\widehat{X} := (X' \times \{0, \dots, r-1\}) \setminus X$. Along the lines of Proposition 13, we have

$$\{0, \dots, r-1\}^n \setminus X = \left[\left(\{0, \dots, r-1\}^{n-1} \setminus X'\right) \times \{0, \dots, r-1\}\right] \cup \widehat{X}.$$

Although $\widehat{X}$ can have up to $r|X|$ elements, we also see that $\widehat{X}$ is the union of at most $2|X|$ sets of the form $v \times \{\alpha, \dots, \beta\}$ for $v \in X'$ and integers $0 \leq \alpha \leq \beta \leq r-1$. More precisely, for each $v \in X'$, there exist integers $0 \leq \alpha_1^v \leq \beta_1^v < \alpha_2^v \leq \beta_2^v < \cdots < \alpha_{q_v}^v \leq \beta_{q_v}^v \leq r-1$ such that

$$\widehat{X} = \bigcup_{v \in X'} \bigcup_{l=1}^{q_v} v \times \{\alpha_l^v, \dots, \beta_l^v\}$$

and $\sum_{v \in X'} q_v \leq 2|X|$. Therefore, $\text{conv}(\widehat{X})$ can be described with $\mathcal{O}(|X|)$ inequalities. Then a recursive construction of an extended formulation for $\text{forb}_I(P, X)$ is analogous to the binary case and involves $\mathcal{O}(n|X|)$ variables and constraints.

In order to address the general case, we first show how to cover $\{0, \dots, r-1\}^n \setminus X$ with boxes. For each $i = 1, \dots, n$, let X^i be the projection of X onto the first i components and let $\widehat{X}^i := (X^{i-1} \times \{0, \dots, r-1\}) \setminus X^i$, where $\widehat{X}^1 := \{0, \dots, r-1\} \setminus X^1$. Working the recursion backwards yields

$$\{0, \dots, r-1\}^n \setminus X = \bigcup_{i=1}^{n} \left[\widehat{X}^i \times \{0, \dots, r-1\}^{n-i}\right].$$

Combining the last two expressions, we arrive at

$$\{0, \dots, r-1\}^n \setminus X = \bigcup_{i=1}^{n} \bigcup_{v \in X^{i-1}} \bigcup_{l=1}^{q_v} v \times \{\alpha_l^v, \dots, \beta_l^v\} \times \{0, \dots, r-1\}^{n-i}.$$

The right-hand-side defines a family $\mathcal{B}$ of at most $2n|X|$ boxes in $\mathbb{R}^n$, yielding

$$\{0, \dots, r-1\}^n \setminus X = \bigcup_{[l,u] \in \mathcal{B}} [l, u] \cap \mathbb{Z}^n.$$

Finally, if $P \subseteq [0, r-1]^n$, then

$$(P \cap \mathbb{Z}^n) \setminus X = (P \cap \mathbb{Z}^n) \cap (\{0, \dots, r-1\}^n \setminus X) = \bigcup_{[l,u] \in \mathcal{B}} P \cap [l, u] \cap \mathbb{Z}^n.$$

Moreover, if P is box-tractable, then

$$\text{forb}_I(P, X) = \text{conv}\left(\bigcup_{[l,u] \in \mathcal{B}} \text{conv}\left(P \cap [l, u] \cap \mathbb{Z}^n\right)\right) = \text{conv}\left(\bigcup_{[l,u] \in \mathcal{B}} P \cap [l, u]\right),$$

where each term within the union is a tractable set. □

The k-best problem and the binary all-different problem can be extended to the case of integral vectors as follows.

Definition 28. *Given a nonempty integral polytope $P \subseteq \mathbb{R}^n$, a vector $c \in \mathbb{R}^n$, and a positive integer k, the integral k-best problem is to either assert $|P \cap \mathbb{Z}^n| \leq k$ and return $P \cap \mathbb{Z}^n$, or to return $v_1, \dots, v_k \in P \cap \mathbb{Z}^n$, all distinct, such that* $\max\{c^\top v_i | \; i = 1, \dots, k\} \leq \min\{c^\top v | \; v \in \; (P \cap \mathbb{Z}^n) \setminus \{v_1, \dots, v_k\}\}$.

Definition 29. *Given a positive integer k, nonempty integral polytopes $P_1, \dots, P_k$ in $\mathbb{R}^n$, and vectors $c_1, \dots, c_k \in \mathbb{R}^n$, the integral all-different problem is to solve*

$$\begin{array}{lll} (\mathcal{P}) \;\min & \sum_{i=1}^k c_i^\top x_i & \\ \quad\quad \text{s.t.} & x_i \in P \cap \mathbb{Z}^n & i = 1, \dots, k \\ & \quad x_i \neq x_j & 1 \leq i < j \leq k. \end{array}$$

The above problems can be shown to be polynomially solvable if the underlying polytopes are box-tractable.

Acknowledgments

We thank Marc Pfetsch for pointing out the example in Proposition 5. This research has been supported in part by the Air Force Office of Scientific Research (Grant #FA9550-12-1-0154) and by Deutsche Forschungsgemeinschaft (KA 1616/4-1).

References

[1] D. Avis, D. Bremner, and R. Seidel, *How good are convex hull algorithms?*, Computational Geometry **7** (1997), 265–301.

[2] E. Balas, *Disjunctive Programming*, Discrete Optimization II (E.L. Johnson, P.L. Hammer, and B.H. Korte, eds.), Annals of Discrete Mathematics, vol. 5, Elsevier, 1979, pp. 3–51.

[3] E. Boros, K. Elbassioni, V. Gurvich, and H.R. Tiwary, *The negative cycles polyhedron and hardness of checking some polyhedral properties*, Annals of Operations Research **188** (2011), 63–76.

[4] M. Conforti, G. Cornuéjols, and G. Zambelli, *Extended formulations in combinatorial optimization*, 4OR **8** (2010), 1–48.

[5] S. Fiorini, S. Massar, S. Pokutta, H.R. Tiwary, and R. de Wolf, *Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds*, Proceedings of the 44th symposium on Theory of Computing (New York, NY, USA), STOC '12, ACM, 2012, pp. 95–106.

[6] ______, *Exponential Lower Bounds for Polytopes in Combinatorial Optimization*, arXiv:1111.0837v4 (2013).

[7] A. Gupte, *n-dimensional superincreasing knapsack polytopes have $\mathcal{O}(n)$ facets*, Optimization Online (2013).

[8] L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, and V. Gurvich, *Generating all vertices of a polyhedron is hard*, Discrete & Computational Geometry **39** (2008), 174–190.

[9] G. Laporte and F.V. Louveaux, *The integer L-shaped method for stochastic integer programs with complete recourse*, Operations Research Letters **13** (1993), 133–142.

[10] E.L. Lawler, *A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem*, Management Science **18** (1972), 401–405.

[11] J. Lee, *All-Different Polytopes*, Journal of Combinatorial Optimization **6** (2002), 335–352.

[12] ______, *Cropped cubes*, Journal of Combinatorial Optimization **7** (2003), 169–178.

[13] J. Lee, J. Leung, and S. de Vries, *Separating type-I odd-cycle inequalities for a binary-encoded edge-coloring formulation*, Journal of Combinatorial Optimization **9** (2005), 59–67.

[14] J. Lee and F. Margot, *On a binary-encoded ILP coloring formulation*, INFORMS Journal on Computing **19** (2007), 406–415.

[15] F.M. Muldoon, W.P. Adams, and H.D. Sherali, *Ideal representations of lexicographic orderings and base-2 expansions of integer variables*, Operations Research Letters **41** (2013), 32–39.

[16] K.G. Murty, *Letter to the Editor—An Algorithm for Ranking all the Assignments in Order of Increasing Cost*, Operations Research **16** (1968), 682–687.

[17] S. Pokutta and M. Van Vyve, *A note on the extension complexity of the knapsack polytope*, Operations Research Letters **41** (2013), 347–350.

[18] T. Rothvoß, *The matching polytope has exponential extension complexity*, arXiv:1311.2369 (2013).

[19] A. Schrijver, *Theory of linear and integer programming*, Wiley, 1998.

[20] M. Yannakakis, *Expressing combinatorial optimization problems by linear programs*, Journal of Computer and System Sciences **43** (1991), 441–466.

Improving the integer L-shaped method

Gustavo Angulo, Shabbir Ahmed, Santanu S. Dey
Georgia Institute of Technology
gangulo@gatech.edu, sahmed@isye.gatech.edu, santanu.dey@isye.gatech.edu

April 15, 2014

Abstract

We consider the integer L-shaped method for two-stage stochastic integer programs. To improve the performance of the algorithm, we present and combine two strategies that deal with different aspects of the algorithm. First, to avoid time-consuming exact evaluations of the second-stage cost function, we propose a simple modification that alternates between linear and mixed-integer subproblems. Then, to better approximate the shape of the second-stage cost function, we present a general framework to generate optimality cuts via a cut-generating linear program which considers information from all solutions found up to any given stage of the method. In order to address the impact of the proposed approaches, we report computational results on two classes of stochastic integer problems.

1 Introduction

In this work we consider mixed-integer programs of the form

$$\begin{aligned} \text{(IP)} \quad \min_{x,z,\theta} \quad & cx + dz + \theta \\ \text{s.t.} \quad & Ax + Cz \le b && (1) \\ & Q(x) - \theta \le 0 && (2) \\ & x \in \{0,1\}^n && (3) \\ & z \ge 0,\ z \in Z, && (4) \end{aligned}$$

where Z is a mixed-integer set and $Q(x)$ is a real-valued function taking a binary vector x as argument. We say that (x^*, z^*, θ^*) is a candidate solution if (x^*, z^*) satisfies (1), (3), and (4). If in addition (2) holds, then we say (x^*, z^*, θ^*) is a feasible (candidate) solution. Constraint (2) together with the presence of θ in the objective function ensures $\theta = Q(x)$ is satisfied by any optimal solution to (IP). A fundamental assumption is that given x, $Q(x)$ can be computed with a reasonable amount of effort.

In the context of two-stage stochastic integer programming, we usually have

$$Q(x) := \mathbb{E}_{\xi}\left[\min_{y}\{qy : \ Wy = h - Tx,\ y \in Y\}\right],$$

which denotes the expected second-stage cost of x with respect to the random data $\xi = (q, W, T, h)$. We assume that Y imposes some integrality requirements on y. When ξ has a finite set of possible outcomes, we have $Q(x) = \sum_\xi p_\xi Q_\xi(x)$, where $Q_\xi(x)$ denotes the optimal second-stage value of the scenario associated to ξ, and p_ξ is the probability of occurrence of ξ. Thus, (IP) can be cast as a large-scale mixed integer program. When the burden of solving (IP) is mainly due to the presence of a large number of scenarios, schemes similar to Benders' decomposition [4] and the L-shaped method [16] can be effective. The idea is to relax (2) and consider θ as an underestimator of $Q(x)$, and successively add cuts in the (x, θ)-space to better approximate the shape of $Q(x)$. This is done until an optimal solution (x^*, z^*, θ^*) satisfying $\theta^* = Q(x^*)$ is found. When the second-stage problem is a linear program, $Q(x)$ is convex in x and thus can be approximated by subgradients using optimal dual solutions. In contrast, when the second-stage problem is a mixed-integer program, such a nice property does not hold, and moreover, $Q(x)$ can even be discontinuous. Thus, the decomposition approaches of the linear case have to be modified to accommodate integer variables in the second stage. In [8], such a modification, the integer L-shaped method, is introduced. It is designed for two-stage stochastic integer problems having binary first-stage variables as it exploits the facial property of 0-1 sets. More generally, the integer L-shaped method can be applied to any mixed-integer problem having the form of (IP) as long as $Q(x)$ is computable from binary x. In particular, it also fits situations where $Q(x)$ can be evaluated with a closed-form analytical formula, but it does not have an amenable mixed-integer formulation. Applications of this method include vehicle routing [11], [6], probabilistic traveling salesman problems [9], location problems[10], and generalized assignment [1], among others.

Next we describe the integer L-shaped method. Let X be the projection of the feasible region of (IP) onto the x-space, and let $L \in \mathbb{R}$ be a lower bound on $Q(x)$ over X. Then (IP) can be equivalently formulated as

$$\begin{array}{lll} \text{(MP)} & \min & cx + dz + \theta \\ & \text{s.t.} & Ax + Cz \leq b \\ & & \Pi x - \mathbf{1}\theta \leq \pi_0 \qquad (5) \\ & & x \in \{0,1\}^n \\ & & z \geq 0,\ z \in Z \\ & & \theta \geq L, \end{array}$$

where $\mathbf{1}$ denotes a vector of ones of appropriate size, as long as for each $x^* \in X$ constraints (5) include a cut of the form $\pi^k x - \theta \leq \pi_0^k$ such that $\pi^k x - \pi_0^k \leq Q(x)$ for all $x \in X$ and $\pi^k x^* - \pi_0^k = Q(x^*)$. In other words, the affine function $\pi^k x - \pi_0^k$ underestimates $Q(x)$ on X, and the estimate is tight at x^*. The optimality cuts of Laporte and Louveaux [8] define such a cut family and form the basis of the integer L-shaped method.

Given $x^* \in \{0,1\}^n$, let $S(x^*) := \{i : x_i^* = 1\}$. In [8], the (standard) integer optimality cut at x^* is defined as

$$\theta \geq (Q(x^*) - L)\left(\sum_{i \in S(x^*)} x_i - \sum_{i \notin S(x^*)} x_i - |S(x^*)|\right) + Q(x^*). \qquad (6)$$

Let $\Delta_{x^*}(x) := |S(x^*)| - \sum_{i \in S(x^*)} x_i + \sum_{i \notin S(x^*)} x_i$ be the Hamming distance between x and x^*, and note that $0 \leq \Delta_{x^*}(x) \leq n$ with $\Delta_{x^*}(x) = 0$ if and only if $x = x^*$. Thus, if $x = x^*$, then the right-hand side of (6) attains its maximum value $Q(x^*)$. If $x \in \{0,1\}^n \setminus \{x^*\}$, then it takes a value less than or equal to L. Since $\theta \geq L$, combining both cases, we have that (6) models the implication $x = x^* \Rightarrow \theta \geq Q(x)$. Observe also that as we have one cut per element in X, (5) might have exponentially many constraints. Thus, (5) is omitted from the initial formulation (MP) and cuts (6) are added on-the-fly as new solutions are discovered.

It is important to keep in mind that given the enumerative nature of (6), in practice these cuts are complemented with other inequalities that, albeit not tight, help to improve the global lower bound on $Q(x)$. When $Q(x)$ is the expected second-stage value of x given by the value function of a mixed-integer program, the most obvious inequalities to add are the subgradient cuts given by the continuous relaxation $Q_{LP}(x)$ of $Q(x)$. They have the form

$$\theta \geq s(x - x^*) + Q_{LP}(x^*), \tag{7}$$

where s is a subgradient of $Q_{LP}(x)$ at x^*.

An implementation of the integer L-shaped method with a current state-of-the-art solver works as follows. Having computed a lower bound L on $Q(x)$ and solved the continuous relaxation of (IP) with Benders' decomposition, we end up with a linear master problem that includes subgradient cuts of the form (7). Then we reinforce the binary requirements on x and any integrality restrictions on z, leading to a mixed-integer master problem of the form (MP), but where the system (5) is a relaxation of (IP), so that an optimal solution to the current problem may not be feasible to (IP). The idea now is to solve the mixed-integer master problem in a way such that all integer solutions are checked against feasibility with respect to (IP) before being accepted as an incumbent. For this, the solver proceeds in a similar fashion to branch-and-cut, that is, it generates a search tree by solving linear subproblems, branching, and adding cutting planes. The main difference is that when a candidate integer solution (x^*, z^*, θ^*) satisfying (1), (3) and (4) is found at a node of the search tree, an additional routine, the so-called optimality cut function, is called in order to assert feasibility and add optimality cuts. If the solution is infeasible to the true problem (IP), i.e., $\theta^* < Q(x^*)$, this function generates an optimality cut that is applied to all pending nodes in the master problem tree, ensuring that this solution is discarded. Then the solver continues exploring the tree with the guarantee that any discarded, and thus infeasible, solution will not appear again. If the solution is actually feasible to (IP), then it is accepted by the optimality cut function and the current incumbent is updated accordingly. A modern implementation of the (standard) integer L-shaped method is presented in Algorithm 1 below.

Algorithm 1 Integer L-shaped method

```
Input: $A, C, b, c, d, Q : X \to \mathbb{R}, Q_{LP} : X \to \mathbb{R}$
Output: Optimal solution $x^*$ to (IP) and optimal value
1: Compute a lower bound $L$ of $Q(x)$
2: Solve the LP relaxation of (IP) with Benders' decomposition
3: Declare $x$ variables as binary in master problem
4: Initialize the optimality cut function
5: Solve the integer master problem using the optimality cut function to assert feasibility of solutions
   and add optimality cuts
6: return $x^*$ and optimal value
```

In line 4 of Algorithm 1 we initialize any additional structures that may be needed by the optimality cut function before invoking the solver in line 5. In particular, as there may be several solutions sharing the same x subvector, we keep a list V of first-stage x for which $Q(x)$ has been computed to avoid duplicate evaluations. In a standard implementation, the optimality cut function has the form shown in Algorithm 2

The optimality cut function returns `TRUE` if the candidate integer solution is indeed feasible to (IP). Otherwise it returns `FALSE` to reject the solution and apply the optimality cut. Note that the steps in lines 4 and 5 in Algorithm 2 are not needed for convergence of the method, but help to improve the global lower bound on $Q(x)$.

The optimality cut (6) relies on exact evaluations of $Q(x)$, which can be very time-consuming in the case where $Q(x)$ is given by a complicated mixed-integer program. Also, observe that (6) depends on x^* and $Q(x^*)$ only, i.e., it only depends on the point to be cut-off. In particular, it does not take into account the information provided by other solutions that we may have found while exploring the first-stage set. To improve the performance of the integer L-shaped method, we propose two approaches to

Algorithm 2 Standard optimality cut function

```
Input: $(x^*, z^*, \theta^*)$ candidate integer solution, $Q : X \to \mathbb{R}$, $Q_{LP} : X \to \mathbb{R}$, $V$
Output: true if solution is feasible, false otherwise
if $x^* \in V$ then // We know $\theta^* \geq Q(x^*)$
    return true
end if
Compute $Q_{LP}(x^*)$
Add the subgradient cut (7)
Compute $Q(x^*)$
$V \leftarrow V \cup \{x^*\}$
if $\theta^* < Q(x^*)$ then
    Add the integer optimality cut (6)
    return false
else
    return true
end if
```

deal with the above issues. First, in Section 2, we present a simple modification that alternates between exact and approximate evaluations of $Q(x)$. Then, in Section 3, we introduce of a new type of optimality cut that includes information obtained from different solutions; in particular, evaluations and estimates of $Q(x)$ at different points. These new cuts are obtained through a cut-generating linear program which is constructed based on ideas from disjunctive programming and the forbidden-vertices problem [2]. Then, in Section 4, we outline an implementation that combines both modifications in a single method. Finally, in Section 5, we present computational results of the proposed variants on two classes of stochastic integer programs.

2 Alternating cuts

In this section we present a simple cut strategy to decrease the overall effort incurred in computing the function $Q(x)$.

Suppose that while solving (IP) with the integer L-shaped method, a candidate solution (x^*, z^*, θ^*) has been found along the search tree of (MP). Recall that we reject the solution if $\theta^* < Q(x^*)$. Since $Q_{LP}(x) \leq Q(x)$, a sufficient condition to reject (x^*, z^*, θ^*) is $\theta^* < Q_{LP}(x^*)$. Given that $Q_{LP}(x)$ is convex, we have that the subgradient cut (7) is a valid inequality that cuts off the pair (x^*, θ^*) in the (x, θ)-space. Therefore, instead of evaluating $Q(x^*)$ exactly, we first evaluate $Q_{LP}(x^*)$ and check whether $\theta^* < Q_{LP}(x^*)$. If so, we add the subgradient cut (7) to remove (x^*, θ^*). Otherwise, we compute $Q(x^*)$ and check whether $\theta^* < Q(x^*)$. If so, we add the integer optimality cut (6). Otherwise, we accept the solution. The key idea is to use $Q_{LP}(x)$ as a proxy for $Q(x)$ to check feasibility of a candidate solution, preventing unnecessary, and more costly, computations of $Q(x)$.

The modification just described is similar in spirit to sequential approximation schemes such as [15], [5], [7], and [14], where the second-stage cost function $Q(x)$ is approximated by linear programs which, starting with $Q_{LP}(x)$, are iteratively strengthened with additional cuts. Although these methods are shown to converge after a finite number of steps, the convergence can be very slow and in practice exact evaluations of $Q(x)$ may be required. In contrast, in the context of the integer L-shaped method, we propose to use $Q_{LP}(x)$ as the unique intermediate approximation for $Q(x)$, which is a simple yet useful modification whose implementation is rather straightforward and, to the best of our knowledge, has not been reported in the literature.

To implement the approach presented above, in addition to V, we also keep a list V_{LP} of visited first-stage solutions x for which the continuous relaxation $Q_{LP}(x)$ has been computed. The modified strategy, which we call alternating cuts, proceeds as shown in Algorithm 3.

Algorithm 3 Optimality cut function with alternating cut strategy

```
Input: $(x^*, z^*, \theta^*)$ candidate integer solution, $Q : X \to \mathbb{R}$, $Q_{LP} : X \to \mathbb{R}$, $V$, $V_{LP}$
Output: true if solution is feasible, false otherwise
 1: if $x^* \in V$ then // We know $\theta^* \geq Q(x^*)$
 2:     return true
 3: end if
 4: if $x^* \notin V_{LP}$ then
 5:     Compute $Q_{LP}(x^*)$
 6:     $V_{LP} \leftarrow V_{LP} \cup \{x^*\}$
 7:     if $\theta^* < Q_{LP}(x^*)$ then
 8:         Add the subgradient cut (7).
 9:         return false
10:     end if
11: end if
    // Now we have $x^* \in V_{LP}$ and $\theta^* \geq Q_{LP}(x^*)$
12: Compute $Q(x^*)$.
13: $V \leftarrow V \cup \{x^*\}$
14: if $\theta^* < Q(x^*)$ then
15:     Add the integer optimality cut (6)
16:     return false
17: else
18:     return true
19: end if
```

Note that if $x^* \notin V_{LP}$ satisfies (7), then x^* is included into V_{LP} and thus the steps in lines 12–19 of Algorithm 3 are applied to check whether (x^*, z^*, θ^*) is a feasible solution or not. As we shall see in Section 5, this simple modification yields speedups of one order of magnitude on instances from the literature.

3 New optimality cuts

In this section, we present a new class of integer optimality cuts that can be used as an alternative to the standard cut (6). After providing an overview of the approach, we show how to construct a cut-generating linear program to separate these new inequalities and then we discuss some implementation details. In this section we denote conv (T) the convex hull of a set T of real vectors.

Let S be the projection of the feasible set of (MP) onto the (x, θ)-space, which corresponds to the epigraph of $Q(x)$ over X, i.e.,

$$S = \{(x, \theta) \in X \times \mathbb{R} : \ \theta \geq Q(x)\}.$$

Let $V \subseteq X$ be such that $Q(x)$ is known for all $x \in V$. We have

$$S \subseteq S(X, V) := \bigcup_{x \in V} \{(x, \theta) : \ \theta \geq Q(x)\} \cup (X \setminus V) \times \{\theta : \ \theta \geq L\}.$$

In some sense, $S(X, V)$ is the best approximation of S when only the values of $Q(x)$ for $x \in V$ are known and only the trivial lower bound L is available over $X \setminus V$. We consider the relaxation $S(V)$ of $S(X, V)$ given by

$$S(X,V) \subseteq S(V) := \bigcup_{x \in V} \{(x,\theta) : \theta \geq Q(x)\} \cup (\{0,1\}^n \setminus V) \times \{\theta : \theta \geq L\}.$$

Figure 1 illustrates an example with $x \in \{0,1\}^2$ and $V = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$. The bold dots represent the values of $Q(x)$ that are known depending on whether x belongs to V or not. Then $S(V)$ is given by the union of the vertical rays.

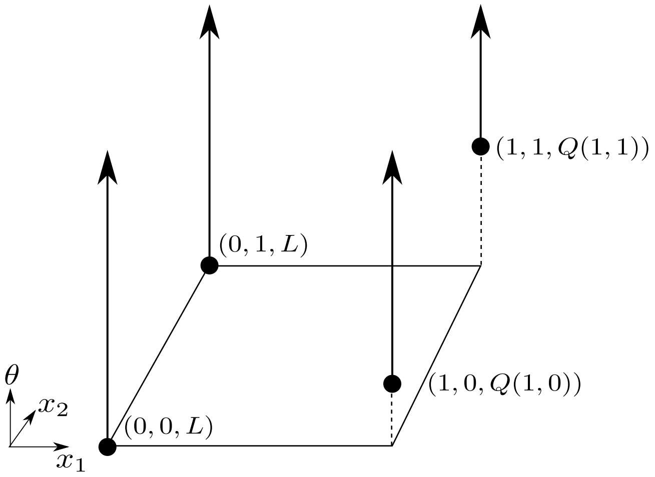
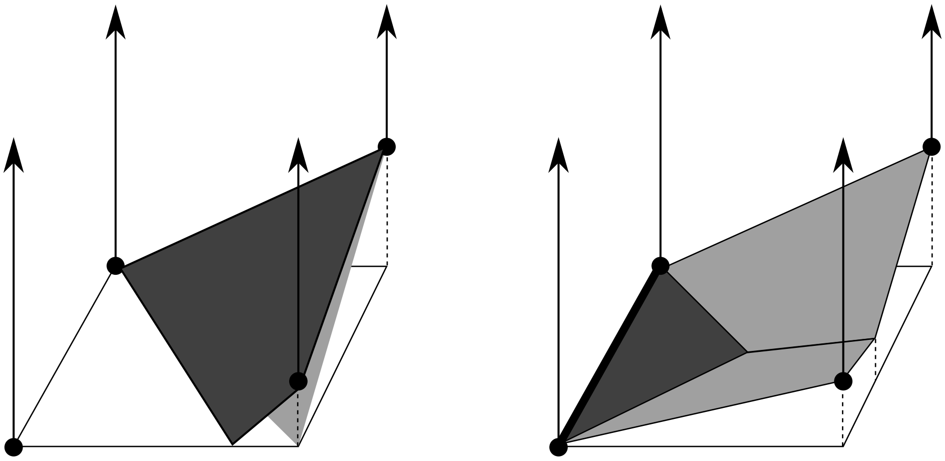


Figure 1: $S(V)$.

Observe that $S(V) \subseteq S(U)$ for any $U \subseteq V$, and in particular, $S(V) \subseteq S(\{x\})$ for $x \in V$. Moreover, $S(V) = \bigcap_{x \in V} S(\{x\})$. Since (6) is a valid inequality for conv $(S(\{x\}))$, it is also valid for conv $(S(V))$. Actually, (6) is the only nontrivial cut needed to describe conv $(S(\{x\}))$. However, in general, conv $(S(V)) \subseteq \bigcap_{x \in V}$ conv $(S(\{x\}))$ holds with strict containment, i.e., adding (6) for all $x \in V$ does not yield conv $(S(V))$. Our goal is to derive a compact extended formulation for conv $(S(V))$ and use it to generate optimality cuts for a point (x^*, θ^*) in the (x, θ)-space that take into account the values of $Q(x)$ for $x \in V$. Figure 2 shows how this information can improve our approximation of the convex hull of the epigraph of $Q(x)$.

Several steps of the construction of our cut-generating linear program rely on Lemma 1 below, which follows from disjunctive programming [3] applied in the context of linear extended formulations of polyhedra.

Lemma 1. *Let $P_1, \ldots, P_k$ be nonempty polyhedra in $\mathbb{R}^n$ having the same recession cone. If $P_i = \{x \in \mathbb{R}^n | \, \exists y_i \in \mathbb{R}^{m_i} : \; E_i x + F_i y_i \geq h_i\}$, then* conv $\left(\cup_{i=1}^k P_i\right) = \{x \in \mathbb{R}^n | \; \exists x_i \in \mathbb{R}^n, \; y_i \in \mathbb{R}^{m_i}, \; \lambda \in \mathbb{R}^k : \; x = \sum_{i=1}^k x_i, \; E_i x_i + F_i y_i \geq \lambda_i h_i, \; \sum_{i=1}^k \lambda_i = 1, \; \lambda \geq 0\}$.



(a) The standard cut at $x = (1,1)$ (light gray) can be strengthened if $Q(1,0)$ is known (dark gray).

(b) The description of the solutions that are not in V given by standard cuts (gray triangle) can be made exact (bold segment).

Figure 2: Improving the description of conv $(S(V))$.

3.1 Construction of CGLP

Clearly, we have conv $(S(V)) =$ conv $\left(P_Q(V) \cup P_L(V)\right)$, where

$$P_Q(V) := \operatorname{conv}\left(\bigcup_{x \in V} \{(x,\theta) : \theta \geq Q(x)\}\right)$$

and

$$P_L(V) := \operatorname{conv}\left(\{0,1\}^n \setminus V\right) \times \{\theta : \theta \geq L\}.$$

Thus to describe conv $(S(V))$ it suffices to provide compact extended formulations for $P_Q(V)$ and $P_L(V)$ and then apply disjunctive programming to their union as illustrated in Figure 3.

Describing $P_Q(V)$ is trivial: letting $V = \{x^1, \ldots, x^t\}$, then $P_Q(V)$ is the set of vectors $(x^Q, \theta^Q) \in \mathbb{R}^n \times \mathbb{R}$ for which there exists $\phi \in \mathbb{R}^t$ satisfying

$$\begin{aligned}
&-x^Q + \sum_{s=1}^{t} \phi_s x^s = 0 \\
&-\theta^Q + \sum_{s=1}^{t} \phi_s Q(x^s) \leq 0 \\
&\sum_{s=1}^{t} \phi_s = 1 \\
&\phi \geq 0.
\end{aligned}$$

To describe $P_L(V)$, it is enough to describe conv $(\{0,1\}^n \setminus V)$ and then take the Cartesian product with $\{\theta : \theta \geq L\}$. We build on results from the forbidden-vertices problem [2] to do this.

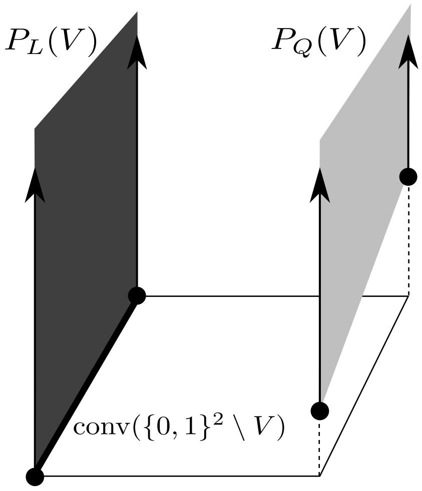


Figure 3: $\mathrm{conv}\,(S(V)) = \mathrm{conv}\,\big(P_Q(V) \cup P_L(V)\big)$.

Let V^i be the projection of V onto the first i coordinates. Define $\hat{V}^1 := \{0,1\} \setminus V^1$, $\hat{V}^i := [V^{i-1} \times \{0,1\}] \setminus V^i \subseteq \{0,1\}^i$ for $i \geq 2$, and write $\hat{V}^i = \{v^i_1, \ldots, v^i_{k_i}\}$. Finally, for all i, let $W^{ij} := \hat{V}^i \times \{0\}^{j-i} = \{w^{ij}_1, \ldots, w^{ij}_{k_i}\} \subseteq \{0,1\}^j$ for all $j \geq i$ and define $W^i := W^{in} = \{w^i_1, \ldots, w^i_{k_i}\} \subseteq \{0,1\}^n$.

From [2], for all $1 \leq j \leq n-1$ we have

$$\{0,1\}^{j+1} \setminus V^{j+1} = \Big[\Big(\{0,1\}^j \setminus V^j\Big) \times \{0,1\}\Big] \cup \hat{V}^{j+1}. \tag{8}$$

The idea behind (8) is that any vector in $\{0,1\}^{j+1} \setminus V^{j+1}$ is such that either its projection onto $\{0,1\}^j$ does not lie in V^j or it is obtained by flipping the value of the last component of a vector in V^{j+1} otherwise.

Example 2. *Consider $n = 3$ and $V = \left\{ \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \right\}$. For $j = 2$, we have $V^2 = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$ and therefore $\{0,1\}^2 \setminus V^2 = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$. Clearly, any vector in $\{0,1\}^3$ whose projection onto $\{0,1\}^2$ lies outside V^2 must belong to $\{0,1\}^3 \setminus V$. Hence $[\{0,1\}^2 \setminus V^2] \times \{0,1\} = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\} \subseteq \{0,1\}^3 \setminus V$. On the other hand, we have $V \subseteq V^2 \times \{0,1\} = \left\{ \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \right\}$, and thus $\hat{V}^3 = [V^2 \times \{0,1\}] \setminus V = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\} \subseteq \{0,1\}^3 \setminus V$. Then we can verify that (8) holds for $j = 2$, i.e., $\{0,1\}^3 \setminus V = [(\{0,1\}^2 \setminus V^2) \times \{0,1\}] \cup \hat{V}^3$.*

We use the recursion (8) to derive an explicit extended formulation for $\mathrm{conv}\,(\{0,1\}^n \setminus V)$ having $\mathcal{O}(n|V|)$ variables and constraints.

Proposition 3. *For all $2 \leq j \leq n$, $\mathrm{conv}\,\big(\{0,1\}^j \setminus V^j\big)$ is given by all $x \in \mathbb{R}^j$ for which there exist vectors y, λ,* *and μ satisfying*

$$\begin{aligned}
&-x+y+\sum_{i=1}^{j}\sum_{l=1}^{k_i}\mu_l^i w_l^{ij}=0\\
&\sum_{l=1}^{k_1}\mu_l^1-\lambda_1=0\\
&\sum_{l=1}^{k_i}\mu_l^i+\lambda_{i-1}-\lambda_i=0 \quad \forall 2\le i\le j-1\\
&\sum_{l=1}^{k_j}\mu_l^j+\lambda_{j-1}=1\\
&y_1=0\\
&y_i-\lambda_{i-1}\le 0 \quad \forall 2\le i\le j\\
&y\ge 0,\ \lambda\ge 0,\ \mu\ge 0.
\end{aligned}$$

Proof. We apply induction on $2\le j\le n$. The base case reduces to proving that $\operatorname{conv}\left(\{0,1\}^2\setminus V^2\right)$ is given by

$$\begin{aligned}
&-x+y+\sum_{i=1}^{2}\sum_{l=1}^{k_i}\mu_l^i w_l^{i2}=0\\
&\sum_{l=1}^{k_1}\mu_l^1-\lambda_1=0\\
&\sum_{l=1}^{k_2}\mu_l^2+\lambda_1=1\\
&y_1=0\\
&y_2-\lambda_1\le 0\\
&y\ge 0,\ \lambda\ge 0,\ \mu\ge 0.
\end{aligned}\tag{9}$$

Indeed, from (8), we have

$$\{0,1\}^2\setminus V^2=\left[\left(\{0,1\}^1\setminus V^1\right)\times\{0,1\}\right]\cup\hat{V}^2.\tag{10}$$

By definition, we have $W^{12}=\hat{V}^1\times\{0\}=(\{0,1\}\setminus V^1)\times\{0\}$. Then observe that

$$\left(\{0,1\}^1\setminus V^1\right)\times\{0,1\}=W^{12}+\left\{\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix}\right\},$$

and thus

$$\operatorname{conv}\left(\left(\{0,1\}^1\setminus V^1\right)\times\{0,1\}\right)=\operatorname{conv}\left(W^{12}\right)+\left\{y\in\mathbb{R}^2:\ y_1=0,\ 0\le y_2\le 1\right\}.$$

Writing $W^{12}=\{w_1^{12},\dots,w_{k_1}^{12}\}$, it follows that $\operatorname{conv}\left(\left(\{0,1\}^1\setminus V^1\right)\times\{0,1\}\right)$ is given by $p\in\mathbb{R}^2$ such that

$$\begin{aligned}
& -p + y + \sum_{l=1}^{k_1} \mu_l^1 w_l^{12} = 0 \\
& \sum_{l=1}^{k_1} \mu_l^1 = 1 \\
& y_1 = 0 \\
& y_2 \leq 1 \\
& y \geq 0,\ \mu^1 \geq 0.
\end{aligned}$$

We also have by definition $\hat{V}^2 = W^{22} = \{w_1^{22}, \ldots, w_{k_2}^{22}\}$, and thus conv $(\hat{V}^2)$ is given by $q \in \mathbb{R}^2$ such that

$$\begin{aligned}
& -q + \sum_{l=1}^{k_2} \mu_l^2 w_l^{22} = 0 \\
& \sum_{l=1}^{k_2} \mu_l^2 = 1 \\
& \mu^2 \geq 0.
\end{aligned}$$

From (10), we apply Lemma 1 to the above polytopes: we introduce a multiplier $0 \leq \lambda_1 \leq 1$, we include the equation $x = p + q$, and we multiply the right-hand-side vectors of the first and second systems by λ_1 and $1 - \lambda_1$, respectively. After eliminating p and q, we immediately obtain the desired system (9) for conv $\left(\{0,1\}^2 \setminus V^2\right)$.

Now, assuming that the claim holds for some $2 \leq j \leq n-1$, we will prove that it also holds for $j+1$. Since conv $\left((\{0,1\}^j \setminus V^j) \times \{0,1\}\right) =$ conv $\left((\{0,1\}^j \setminus V^j)\right) \times [0,1]$, by the inductive hypothesis, we have that conv $\left((\{0,1\}^j \setminus V^j) \times \{0,1\}\right)$ is given by $p \in \mathbb{R}^{j+1}$ satisfying

$$\begin{aligned}
& -p + y + \sum_{i=1}^{j} \sum_{l=1}^{k_i} \mu_l^i w_l^{ij+1} = 0 \\
& \sum_{l=1}^{k_1} \mu_l^1 - \lambda_1 = 0 \\
& \sum_{l=1}^{k_i} \mu_l^i + \lambda_{i-1} - \lambda_i = 0 \quad \forall 2 \leq i \leq j-1 \\
& \sum_{l=1}^{k_j} \mu_l^j + \lambda_{j-1} = 1 \\
& y_1 = 0 \\
& y_i - \lambda_{i-1} \leq 0 \quad \forall 2 \leq i \leq j \\
& y_{j+1} \leq 1 \\
& y \geq 0,\ \lambda \geq 0,\ \mu \geq 0,
\end{aligned}$$

where we have appended a new variable $0 \leq y_{j+1} \leq 1$ and vectors w_l^{ij} have been extended to w_l^{ij+1} by appending another component with value 0.

We also have that conv $\left(\hat{V}^{j+1}\right)$ is given by $q \in \mathbb{R}^{j+1}$ satisfying

$$\begin{aligned}
&-q + \sum_{l=1}^{k_{j+1}} \mu_l^{j+1} w_l^{j+1j+1} = 0\\
&\sum_{l=1}^{k_{j+1}} \mu_l^{j+1} = 1\\
&\mu^{j+1} \geq 0.
\end{aligned}$$

From (8), it is enough to apply Lemma 1 to the above polytopes to find an extended formulation for conv $\left(\{0,1\}^{j+1} \setminus V^{j+1}\right)$. Analogously to the base case, we introduce a multiplier $0 \leq \lambda_j \leq 1$, we include the equation $x = p + q$, and we multiply the right-hand-side vectors of the first and second systems by λ_j and $1 - \lambda_j$, respectively. After eliminating p and q, we immediately obtain the desired system for conv $\left(\{0,1\}^{j+1} \setminus \ V^{j+1}\right)$. □

From Proposition 3, we obtain that conv $(\{0,1\}^n \setminus V)$ is given by the vectors $x^L \in \mathbb{R}^n$ such that

$$\begin{aligned}
&-x^L + y + \sum_{i=1}^{n} \sum_{l=1}^{k_i} \mu_l^i w_l^i = 0\\
&\sum_{l=1}^{k_1} \mu_l^1 - \lambda_1 = 0\\
&\sum_{l=1}^{k_i} \mu_l^i + \lambda_{i-1} - \lambda_i = 0 \quad \forall 2 \leq i \leq n-1\\
&\sum_{l=1}^{k_n} \mu_l^n + \lambda_{n-1} = 1\\
&y_1 = 0\\
&y_i - \lambda_{i-1} \leq 0 \quad \forall 2 \leq i \leq n\\
&y \geq 0,\ \lambda \geq 0,\ \mu \geq 0.
\end{aligned}$$

Appending the variable θ^L and the constraint $\theta^L \geq L$ to the above system gives an extended formulation for $P_L(V)$. Note that excluding the nonnegativity restrictions, the constraint matrix has $3n$ rows and $3n + \mathcal{O}(n|V|)$ columns, i.e, only its width changes with V. In particular, updating the formulation can be done columnwise, which is a desirable property from the computational point of view.

Once again, we apply disjunctive programming, but this time to $P_L(V)$ and $P_Q(V)$ to derive an extended formulation for conv $(S(V))$. Note that both $P_L(V)$ and $P_Q(V)$ have $\{(0,\theta) \in \mathbb{R}^n \times \mathbb{R} : \theta \geq 0\}$ as their recession cone and thus Lemma 1 applies. We introduce a multiplier $0 \leq \delta \leq 1$, we include the equations $x = x^L + x^Q$ and $\theta = \theta^L + \theta^Q$, and we multiply the right-hand-side vectors of the systems defining $P_L(V)$ and $P_Q(V)$ by δ and $1 - \delta$, respectively.

Recall that in the definition of $S(V)$ we have dropped the dependence on X. To recover part of that information, we can describe a polyhedron that lies between conv (S) and conv $(S(V))$. For that, $P_L(V)$ can be coupled with any valid inequality for (MP). In particular, including variables $z \geq 0$ and the system $Ax^L + Cz \leq b$ tightens the formulation. Lower bounds of the form $\Pi x^L - \mathbf{1}\theta^L \leq \pi_0$ can be useful too to better approximate the shape of the epigraph S of $Q(x)$. Thus we may assume that both types of constraints are added to the formulation of $P_L(V)$, and that $\theta^L \geq L$ is absorbed in $\Pi x^L - \mathbf{1}\theta^L \leq \pi_0$.

Finally, we obtain that if (x^*, θ^*) does not belong to $\operatorname{conv}(S(V))$, and thus not to $\operatorname{conv}(S)$, then the following system is infeasible:

$$
\begin{array}{ll}
(\alpha) & x^L + x^Q = x^* \\
(\beta) & \theta^L + \theta^Q = \theta^* \\
(\sigma) & -x^L + y + \displaystyle\sum_{i=1}^{n} \sum_{l=1}^{k_i} \mu_l^i w_l^i = 0 \\
(\rho_1) & \displaystyle\sum_{l=1}^{k_1} \mu_l^1 - \lambda_1 = 0 \\
(\rho_i) & \displaystyle\sum_{l=1}^{k_i} \mu_l^i + \lambda_{i-1} - \lambda_i = 0 \quad \forall 2 \le i \le n-1 \\
(\rho_n) & \displaystyle\sum_{l=1}^{k_n} \mu_l^n + \lambda_{n-1} - \delta = 0 \\
(\varphi_1) & y_1 = 0 \\
(\varphi_i) & y_i - \lambda_{i-1} \le 0 \quad \forall 2 \le i \le n \\
(\psi) & \Pi x^L - \mathbf{1}\theta^L - \pi_0 \delta \le 0 \\
(\nu) & Ax^L + Cz - b\delta \le 0 \\
(\gamma) & -x^Q + \displaystyle\sum_{s=1}^{t} \phi_s x^s = 0 \\
(\tau) & -\theta^Q + \displaystyle\sum_{s=1}^{t} \phi_s Q(x_s) \le 0 \\
(\eta) & \displaystyle\sum_{s=1}^{t} \phi_s + \delta = 1 \\
 & y \ge 0,\ \lambda \ge 0,\ \mu \ge 0 \\
 & \phi \ge 0 \\
 & \delta \ge 0.
\end{array}
$$

By Farkas' Lemma, and after removing redundancies, we arrive at the alternative system

$$
\begin{array}{rcll}
x^*\alpha + \theta^*\beta + \eta & < & 0 & \\
\alpha - \sigma + A^\top \nu + \Pi^\top \psi & = & 0 & \\
\beta - \mathbf{1}\psi & = & 0 & \\
-\rho_n + \eta - b\nu - \pi_0 \psi & \ge & 0 & \\
C^\top \nu & \ge & 0 & \\
\sigma_i + \varphi_i & \ge & 0 & 2 \le i \le n \\
-\rho_i + \rho_{i+1} + \varphi_{i+1} & \ge & 0 & 1 \le i \le n-1 \\
w_l^i \sigma + \rho_i & \ge & 0 & 1 \le n,\ 1 \le l \le k_i \\
x^s \alpha + Q(x^s)\beta + \eta & \ge & 0 & 1 \le s \le t \\
\beta \ge 0,\ \varphi \ge 0,\ \nu \ge 0,\ \psi \ge 0. & & &
\end{array}
$$

Thus, any feasible solution to the above system yields an inequality $\alpha x + \beta\theta \ge -\eta$ that is valid for $\operatorname{conv}(S)$, but is violated by (x^*, θ^*).

For finite termination of the integer L-shaped method, we need a tightness condition at the current solution, namely $\alpha x^* + \beta Q(x^*) = -\eta$. Including this condition yields $0 > x^*\alpha + \theta^*\beta + \eta = x^*\alpha + \beta Q(x^*) + \eta - \beta Q(x^*) + \theta^*\beta = \beta(\theta^* - Q(x^*))$. Since $\theta^* < Q(x^*)$, we conclude that $\beta > 0$ in any feasible tight solution. Therefore, we replace the condition $x^*\alpha + \theta^*\beta + \eta < 0$ with $x^*\alpha + Q(x^*)\beta + \eta = 0$ and the normalization $\beta = 1$. Note that the objective function of the resulting linear program is fixed to zero, and we only need to find a feasible solution, which always exists by definition of the system; in particular, (6) is feasible. The final system, denoted CGLP, reads

$$
\begin{alignat}{3}
\alpha - \sigma + A^\top \nu + \Pi^\top \psi &\;=\;& 0 & \notag\\
\mathbf{1}\psi &\;=\;& 1 & \notag\\
-\rho_n + \eta - b\nu - \pi_0\psi &\;\geq\;& 0 & \notag\\
C^\top \nu &\;\geq\;& 0 & \notag\\
\sigma_i + \varphi_i &\;\geq\;& 0 &\quad 2 \leq i \leq n \notag\\
-\rho_i + \rho_{i+1} + \varphi_{i+1} &\;\geq\;& 0 &\quad 1 \leq i \leq n-1 \notag\\
w_l^i\sigma + \rho_i &\;\geq\;& 0 &\quad 1 \leq i \leq n,\ 1 \leq l \leq k_i \tag{11}\\
x^s\alpha + Q(x^s) + \eta &\;\geq\;& 0 &\quad 1 \leq s < t \tag{12}\\
x^t\alpha + Q(x^t) + \eta &\;=\;& 0 & \tag{13}\\
\varphi \geq 0,\ \nu \geq 0,\ \psi \geq 0. && & \notag
\end{alignat}
$$

Having set $x^t := x^*$, we solve CGLP to find a feasible solution to the system. In particular, we obtain α and η defining a CGLP-based optimality cut of the form

$$\alpha x + \theta \geq -\eta \tag{14}$$

which by construction cuts off (x^*, θ^*) with $\theta^* < Q(x^*)$.

3.2 Implementation

The main difference that we are proposing with the standard implementation is the use of the CGLP-based cut (14) in place of (6). This requires keeping a list V of first-stage solutions for which $Q(x)$ has been computed and updating CGLP accordingly. Algorithm 4 shows the procedure.

A key step is found in line 7 of Algorithm 4 as $\operatorname{conv}(S(V))$ has to be recomputed whenever a new vector x^* is added to V. Of course, we could derive CGLP from scratch every time. Doing so requires computing the sets W^i and thus creating $\mathcal{O}(n|V|)$ constraints in (11). Instead, we propose to perform marginal updates from an iteration to the next one using the fact that $W^i = \hat{V}^i \times \{0\}^{n-i}$.

Let $V_t = \{x^1, \ldots, x^t\}$ be the set of the first t solutions found along the master tree. Similarly, let V_t^i be the projection of V_t onto the first i components and set $\hat{V}_t^i := [V_t^{i-1} \times \{0,1\}] \setminus V_t^i$ with $\hat{V}_t^1 := \{0,1\} \setminus V_t^1$. Suppose a new vector $x^{t+1} = (x_1, \ldots, x_n)$ is to be included and let V_{t+1}, V_{t+1}^i, $\hat{V}_{t+1}^i$ be the updated sets. Let $\bar{x}^i := (x_1, \ldots, x_{i-1}, x_i)$ and $\hat{x}^i := (x_1, \ldots, x_{i-1}, 1 - x_i)$. Clearly, we have $V_{t+1} = V_t \cup \{x^{t+1}\}$

Algorithm 4 Optimality cut function with CGLP-based optimality cuts

```
Input: $(x^*, z^*, \theta^*)$ candidate integer solution, $Q : X \to \mathbb{R}$, $Q_{LP} : X \to \mathbb{R}$, $V$
Output: true if solution is feasible, false otherwise
if $x^* \in V$ then // We know $\theta^* \geq Q(x^*)$
    return true
end if
Compute $Q_{LP}(x^*)$
Add the subgradient cut (7)
Compute $Q(x^*)$
Update CGLP.
$V \leftarrow V \cup \{x^*\}$
if $\theta^* < Q(x^*)$ then
    Solve CGLP to obtain $\alpha$ and $\eta$
    Add the integer optimality cut (14)
    return false
else
    return true
end if
```

and $V^i_{t+1} = V^i_t \cup \{\bar{x}^i\}$. Now, to obtain $\hat{V}^i_{t+1}$, observe that

$$\begin{aligned}
\hat{V}^i_{t+1} &= \left[V^{i-1}_{t+1} \times \{0,1\}\right] \setminus V^i_{t+1} \\
&= \left[V^{i-1}_t \times \{0,1\} \cup \{\hat{x}^i, \bar{x}^i\}\right] \setminus \left[V^i_t \cup \{\bar{x}^i\}\right] \\
&= \left[V^{i-1}_t \times \{0,1\} \cup \{\hat{x}^i\}\right] \setminus \left[V^i_t \cup \{\bar{x}^i\}\right] \\
&= \left(\left[V^{i-1}_t \times \{0,1\}\right] \setminus \left[V^i_t \cup \{\bar{x}^i\}\right]\right) \cup \left(\{\hat{x}^i\} \setminus \left[V^i_t \cup \{\bar{x}^i\}\right]\right) \\
&= \left(\hat{V}^i_t \setminus \{\bar{x}^i\}\right) \cup \left(\{\hat{x}^i\} \setminus \ V^i_t\right).
\end{aligned}$$

Therefore, if $\hat{x}^i \notin \hat{V}^i_t$ and $\hat{x}^i \notin V^i_t$, then $\hat{x}^i$ is included in $\hat{V}^i_{t+1}$. Also, if $\bar{x}^i \in \hat{V}^i_t$, then $\bar{x}^i$ is removed to obtain $\hat{V}^i_{t+1}$. Further observe that both operations cannot occur at the same iteration since the equivalence

$$\bar{x}^i \in \hat{V}^i_t \iff \hat{x}^i \in V^i_t \ \wedge \ \bar{x}^i \notin V^i_t$$

implies that $\hat{x}^i \notin V^i_t$ and $\bar{x}^i \in \hat{V}^i_t$ cannot hold true at the same time.

It follows that updating V involves adding or removing at most one vector for each W^i, totaling at most n such operations. The system CGLP is updated accordingly by appending or dropping at most n rows in (11). Also, x^{t+1} takes the place of x^t in (13) and the cut corresponding to x^t now takes the form (12) by changing the equality sign into inequality. The procedure to update CGLP is shown in Algorithm 5.

4 Combined method

Now we outline an implementation of the integer L-shaped method that combines the alternating strategy discussed in Section 2 with the new optimality cuts presented in Section 3.

Algorithm 5 Updating CGLP

```
Input: CGLP, $V^i$, $\hat{V}^i$, $t$, $x^{t+1} = (x_1, \ldots, x_n)$
Output: Updated CGLP, $V^i$, $\hat{V}^i$
for $1 \leq i \leq n$ do
    $\bar{x}^i \leftarrow (x_1, \ldots, x_{i-1}, x_i)$
    $\hat{x}^i \leftarrow (x_1, \ldots, x_{i-1}, 1 - x_i)$
    if $\hat{x}^i \notin \hat{V}^i$ and $\hat{x}^i \notin V^i$ then
        $w \leftarrow \hat{x}^i \times \{0\}^{n-i}$
        Add $w\sigma + \rho_i \geq 0$ to (11)
        $\hat{V}^i \leftarrow \hat{V}^i \cup \{\hat{x}^i\}$
    end if
    if $\bar{x}^i \in \hat{V}^i$ then
        $w \leftarrow \bar{x}^i \times \{0\}^{n-i}$
        Remove $w\sigma + \rho_i \geq 0$ from (11)
        $\hat{V}^i \leftarrow \hat{V}^i \setminus \{\bar{x}^i\}$
    end if
    $V^i \leftarrow V^i \cup \{\bar{x}^i\}$
end for
Add $x^t\alpha + Q(x^t) + \eta \geq 0$ to (12)
Replace (13) with $x^{t+1}\alpha + Q(x^{t+1}) + \eta = 0$
```

We keep two disjoint lists of first-stage solutions: in V_{LP} we include solutions for which only $Q_{LP}(x)$ has been computed, while in V we keep solutions for which $Q(x)$ has been evaluated. At any given stage, we assume that for each $x \in V$ we have added an optimality cut that is tight at x. Now, when a candidate integer solution (x^*, z^*, θ^*) is found in the master tree, we check whether $x^* \in V$ or not. If so, we accept the solution as we already know $Q(x^*) \leq \theta^*$. Now, if $x^* \notin V_{LP}$, then we compute $Q_{LP}(x^*)$, we add x^* into V_{LP}, and in case $\theta < Q_{LP}(x^*)$, we add the subgradient cut (7). At this point, if (x^*, θ^*) has been neither accepted nor rejected, we have $x^* \in V_{LP}$ and $\theta^* \geq Q_{LP}(x^*)$. Thus we compute $Q(x^*)$, we move x^* from V_{LP} to V, and in case $\theta < Q(x^*)$, we add the CGLP-based cut (14) and accept the solution otherwise. Algorithm 6 below presents the method.

5 Results

In this section we address the performance of the variants of the integer L-shaped method discussed so far. Given that the implementations differ in the cut strategy used and in the type of optimality cut added, we consider the following combinations:

1. *Std-Std*: standard cut strategy and standard optimality cut (6); see Section 1.
2. *Alt-Std*: alternating cut strategy and standard optimality cut (6); see Section 2.
3. *Std-CGLP*: standard cut strategy and new optimality cut (14); see Section 3.
4. *Alt-CGLP*: alternating cut strategy and new optimality cut (14); see Section 4.

In other words, *Std-Std* corresponds to the usual implementation of the integer L-shaped method, on top of which the variants are built.

Our computational implementation uses CPLEX 12.5.0.1 as a solver and its Callable Library for advanced control routines. Since either optimality cuts (6) or (14) are part of the complete formulation

Algorithm 6 Optimality cut function with alternating cut strategy and CGLP-based optimality cuts

```
Input: $(x^*, z^*, \theta^*)$ candidate integer solution, $Q : X \to \mathbb{R}, Q_{LP} : X \to \mathbb{R}, V, V_{LP}$
Output: true if solution is feasible, false otherwise
if $x^* \in V$ then // We know $\theta^* \geq Q(x^*)$
    return true
end if
if $x^* \notin V_{LP}$ then
    Compute $Q_{LP}(x^*)$.
    $V_{LP} \leftarrow V_{LP} \cup \{x^*\}$.
    if $\theta^* < Q_{LP}(x^*)$ then
        Add the subgradient cut (7).
        return false
    end if
end if
// Now we have $x^* \in V_{LP}$ and $\theta^* \geq Q_{LP}(x^*)$
Compute $Q(x^*)$.
Update CGLP.
$V \leftarrow V \cup \{x^*\}$.
$V_{LP} \leftarrow V_{LP} \setminus \{x^*\}$
if $\theta^* < Q(x^*)$ then
    Solve CGLP to obtain $\alpha$ and $\eta$
    Add the integer optimality cut (14)
    return false
else
    return true
end if
```

(MP) but not included from the beginning, we have to add them on-the-fly through the optimality cut function. This routine is called every time the solver finds a candidate integer solution to the master problem and is in charge of generating an optimality cut if needed. In the case of CGLP, it calls additional subroutines to make the required updates to generate (14). We include the formulation of the first-stage set in CGLP, along the subgradients cuts derived from the linear relaxation of $Q(x)$ used in Benders' decomposition.

The experiments were carried out on a personal computer with 3.33 Ghz CPU, 4 Gb of RAM, and running Linux. A relative optimality gap of 0.01% was set as stopping criterion and a time limit of 7200 seconds was imposed. We do not report on the extensive form of the instances as solving them using an off-the-shelf solver is much slower than the decomposition approaches.

5.1 Stochastic server location problem

The stochastic server location problem is described in [12]. Given n locations, in the first stage we are asked to decide where to place servers so that the demand given by m potential customers is satisfied in the second stage. The uncertain data is the set of customers to be served in the second stage and the objective is to maximize the expected second-stage revenue minus the first-stage installation costs. In minimization form, the problem can be written as

$$\begin{aligned} \min \quad & cx + Q(x) \\ \text{s.t.} \quad & x \in \{0,1\}^n, \end{aligned}$$

where $Q(x) := \mathbb{E}_{\xi}[Q_{\xi}(x)]$ and

$$\begin{aligned} Q_{\xi}(x) := \quad \min \quad & q_1 y + q_2 s \\ \text{s.t.} \quad & W_1 y + W_2 s \geq h(\xi) - Tx \\ & y \in \{0,1\}^{nm} \\ & s \in \mathbb{R}^n_+ . \end{aligned}$$

The random right-hand-side vector $h(\xi)$ represents the set of active customers in a given scenario.

We tested our methods on the instances presented in [13]. Instances `SSLP.n.m.k` have n locations, m customers, and k scenarios, leading to n binary variables in the first stage and nm binary variables and n nonnegative variables per scenario in the second stage. For each n and m, five replications with k scenarios each are considered. We did not include instances having $n = 5$ as all of them took less than 1 second to solve with any method.

Tables 1 and 2 summarize our results. In both tables, column *Instance* indicates the combination of n, m and k as above. Headers *Std-Std*, *Alt-Std*, *Std-CGLP*, *Alt-CGLP* denote the type of implementation under consideration. Here we present the averages over the five replications of each instance. Detailed results are given in Tables 6 and 7 in the Appendix.

In Table 1 we present the overall results for all four methods. Columns *Nodes* show the average number of nodes explored in the master problem. Columns *Time* show the average total time spent to reach optimality, which includes computing an initial lower bound L, solving the LP relaxation with Benders' decomposition, and exploring and evaluating candidate solutions in the master problem.

From Table 1, we see that there is no significant variation in the number of explored nodes among the different methods. Now, the implementations that use the alternating cut strategy clearly outperform the other two methods, with speedups of one order of magnitude. On the other hand, with a few exceptions, the use of CGLP-based cuts does not cause major changes in the total running time, especially when combined with the alternating cut strategy. This can be explained by the fact that in these problems, the first-stage is very simple as $X = \{0,1\}^n$ with $n \leq 15$, which does not present a challenge for CPLEX.

Instance	Std-Std		Std-CGLP		Alt-Std		Alt-CGLP	
	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
SSLP.10.50.50	402.4	70.9	394.8	71.5	406.8	6.8	404.2	6.8
SSLP.10.50.100	370.2	91.1	373.0	90.5	371.8	13.2	371.0	13.6
SSLP.10.50.500	381.0	548.5	385.0	561.7	386.8	64.0	385.0	65.5
SSLP.10.50.1000	360.0	1294.1	357.8	1307.1	367.4	128.2	368.2	129.3
SSLP.10.50.2000	392.2	3298.0	371.4	3160.7	404.4	339.3	404.6	336.7
SSLP.15.45.5	772.6	81.5	750.2	89.0	763.4	2.7	764.6	2.8
SSLP.15.45.10	1408.0	400.9	1370.8	353.6	1450.8	6.1	1414.0	6.5
SSLP.15.45.15	1500.0	534.3	1498.4	539.1	1526.0	11.7	1523.6	11.9
SSLP.15.45.20	495.6	358.4	481.4	347.8	500.4	8.0	502.4	8.1
SSLP.15.45.25	733.0	708.4	698.8	704.9	737.8	16.7	732.2	17.4

Table 1: Stochastic server location: overall results.

To understand the effect of alternating cuts, in Table 2 we present details regarding subproblems. Recall that every time a candidate integer solution is found, we have to check whether it is feasible, by either solving a series of MIPs or LPs, one per scenario, and then add a cut to discard the solution if necessary. Headers *#LP* and *#MIP* denote the average number of times a candidate solution was checked using linear or mixed-integer subproblems, while headers *Time LP* and *Time MIP* indicate the average time spent in each case. We focus only on the implementations *Std-Std* and *Alt-Std* as the comparison for the remaining pair is similar.

From Table 2, we see that with the alternating cut strategy the number of MIP evaluations reduces considerably. This means that in the problems we tested, most of the time it is not necessary to compute the exact second-stage value of a given first-stage solution to reject it. Furthermore, only a small fraction of these solutions are visited twice, and only in those cases we have to solve MIP subproblems. The benefits are evident.

Instance	Std-Std				Alt-Std			
	#LP	#MIP	Time LP	Time MIP	#LP	#MIP	Time LP	Time MIP
SSLP.10.50.50	147.6	147.6	1.8	65.8	148.6	3.4	1.7	1.8
SSLP.10.50.100	131.6	131.6	3.3	81.0	130.8	3.8	3.0	3.5
SSLP.10.50.500	131.6	131.6	16.4	497.2	130.6	3.0	14.8	15.2
SSLP.10.50.1000	132.0	132.0	33.6	1193.3	127.2	3.0	30.2	32.8
SSLP.10.50.2000	142.6	142.6	72.4	3082.5	143.4	4.2	67.3	133.2
SSLP.15.45.5	143.0	143.0	0.3	80.6	143.2	5.8	0.3	1.9
SSLP.15.45.10	262.0	262.0	1.1	398.2	268.5	5.3	1.1	3.6
SSLP.15.45.15	310.6	310.6	1.9	530.1	317.4	6.0	1.9	7.9
SSLP.15.45.20	99.4	99.4	0.7	356.1	98.4	3.2	0.7	5.9
SSLP.15.45.25	162.4	162.4	1.5	704.3	163.0	5.4	1.4	12.8

Table 2: Stochastic server location: subproblems details.

5.2 Stochastic multiple binary knapsack problem

The second benchmark set corresponds to a class of stochastic multiple binary knapsack problems. They have the form

$$\begin{array}{ll} \min & cx + dz + Q(x) \\ \text{s.t.} & Ax + Cz \geq b \\ & x \in \{0,1\}^n \\ & z \in \{0,1\}^n, \end{array}$$

where $Q(x) := \mathbb{E}_\xi[Q_\xi(x)]$,

$$\begin{array}{lll} Q_\xi(x) := & \min & q(\xi)y \\ & \text{s.t.} & Wy \geq h - Tx \\ & & y \in \{0,1\}^n, \end{array}$$

and all data are nonnegative integers. In the second-stage problem, only the objective vector $q(\xi)$ is random, following a discrete distribution with finitely many scenarios.

We generated 30 instances of the above problem with $n = 120$ and 20 equiprobable scenarios. The systems $Ax + Cz \geq b$ and $Wy \geq h - Tx$ have 50 and 5 rows, respectively. The entries of A, C, T, W, c, d, and q are i.i.d. sampled from the uniform distribution over $\{1, \ldots, 100\}$. We set $b = \frac{3}{4}(A\mathbf{1} + C\mathbf{1})$ and $h = \frac{3}{4}(T\mathbf{1} + W\mathbf{1})$, with $\mathbf{1}$ denoting the n-dimensional vector of ones.

We divided the instances intro three groups depending on how much time the standard implementation took to solve each of them: *Easy* (less than 200 seconds, instances 1–6), *Medium* (between 200 and 1000 seconds, instances 7–18), and *Hard* (more than 1000 seconds, instances 19–29). We ommitted instance 30 since none of the methods was able to solve it to optimality within the time limit.

Tables 3, 4, and 5 below summarize the results. Column *Difficulty* denotes the instance class. The remaining headers and columns are as in Tables 1 and 2. Detailed results are given in Tables 8, 9, and 10 in the Appendix.

Difficulty	Std-Std		Std-CGLP		Alt-Std		Alt-CGLP	
	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
Easy	151531.5	87.1	127696.0	82.5	154611.7	86.0	133724.2	82.8
Medium	945487.8	520.8	714822.5	453.8	940249.3	516.1	748502.7	446.7
Hard	3356158.1	2125.7	2654448.1	1833.6	3371088.5	2065.2	2656526.5	1756.3

Table 3: Stochastic multiple knapsack: overall results.

From Table 3, we see that the application of the alternating cut strategy does not yield the time savings we saw with the stochastic server location problems. On the other hand, in most instances, adding CGLP-based cuts instead of standard cuts yields reductions in both the number of nodes and the total time, regardless of the cut strategy being used. We would like to conclude that these improvements are due to the fact that CGLP-based cuts help to explore the master tree. However, at this point, that is not completely clear, as for example, time reductions could be consequence of less evaluations of $Q(x)$ and not because of the strength of the new cuts.

To aid our analysis, in Table 4 we report the average number of candidate solutions for which $Q_{LP}(x)$ and $Q(x)$ were evaluated and the average time spent doing so. This time we compare *Std-Std* and *Std-CGLP*, and the notation is similar to that of Table 2.

Difficulty	Std-Std				Std-CGLP			
	#LP	#MIP	Time LP	Time MIP	#LP	#MIP	Time LP	Time MIP
Easy	13.7	13.7	0.0	25.7	14.7	14.7	0.0	28.1
Medium	49.2	49.2	0.1	99.1	54.1	54.1	0.1	107.2
Hard	112.3	112.3	0.2	231.6	114.9	114.9	0.2	235.8

Table 4: Stochastic multiple knapsack: subproblems details.

We observe that both implementations require roughly the same number of evaluations of both $Q_{LP}(x)$ and $Q(x)$, which explains why alternating cuts does not outperform the standard cut strategy. Moreover, the difference in the time solving subproblems is very small compared to the total running times presented in Table 3. Thus, the reductions observed in Table 3 can be attributed to the better approximation of the first-stage set given by the CGLP-based cuts and not to the variability of the evaluations. In this regard, it is important to stress that, in principle, having a better description of the first-stage set does not have a direct relationship with the number of candidates solutions found in the master tree, and actually, having more candidates could hurt the total running time if their evaluation is too costly. However, in situations where after decomposing the problem the burden of the computation lies on the master problem, our improved cuts may prove beneficial as exemplified by our results.

Finally, in Table 5 we present the overhead incurred by using CGLP to generate cuts, that is, the time spent in additional operations to maintain and solve CGLP through the method. For each class, column $|V|$ shows the average final size of V, which is the number of candidate solutions for which $Q(x)$ was evaluated exactly. Headers *Update* and *Generate* denote the average total time spent updating the formulation of CGLP and actually solving the system to find an optimality cut, respectively. This additional time is already included in the total running time presented in Table 3.

As expected, the overhead increases as more solutions are included in the extended formulation. Updating CGLP takes practically no time, whereas generating the cut takes a nonnegligible amount of

Difficulty	$\lvert V\rvert$	Update	Generate
Easy	14.7	0.0	0.5
Medium	54.1	0.2	7.2
Hard	114.9	0.4	24.7

Table 5: Stochastic multiple knapsack: CGLP overhead.

time. However, compared to the total running time, the overhead is very small and the effort of computing improved cuts pays off as shown in Table 3. For more complicated problems where the number of binary first-stage variables is too large or where too many candidate solutions are evaluated, the cost of maintaining CGLP is likely to be higher. In those cases, we can enforce rules to limit the number of calls to CGLP, such as using the standard optimality cuts as a baseline and applying the improved cuts only once in a while.

6 Concluding remarks

In this work, we have presented two modifications to the integer L-shaped method with the objective of reducing the running time of the algorithm. The first one, termed alternating cuts strategy, seeks to avoid expensive evaluations of the second-stage cost function, while the second, the use of CGLP-based optimality cuts, helps to better approximate the shape of the epigraph of the cost function when evaluations at different points are available. Our computational results suggest the following:

1. The alternating cuts strategy works better in problems where the computational bottleneck of (IP) is in evaluating $Q(x)$. Even when that is not the case, this modification does not seem to hurt the total running times and thus it could be considered as the base method on top of which more evolved algorithms can be built.
2. CGLP-based cuts are a viable alternative when the first-stage set is difficult to explore and computing $Q(x)$ is a relatively cheap operation. As the sole purpose of these new cuts is to have a better representation of the epigraph of the second-stage cost function within the master problem, there is no guarantee about the number or the sequence of solutions for which $Q(x)$ is evaluated, and thus, in general, this method performs well when the impact of this variability is small compared with the effort of solving the master problem.
3. We also point out that our overall computational experience indicates that CGLP-based cuts are particularly suitable for problems having additional integer variables in the set Z, since a deep cut discarding a point (x^*, θ^*) in the (x, θ)-space may also prove effective in discarding a large number of points of the form (x^*, z, θ^*) for $z \in Z$.
4. As favorable conditions for both modifications are unlikely to be attained at the same time, we observe that time reductions in a combined method are mainly consequence of one strategy or the other, but not because of the combination of both. That being said, it would be interesting to experiment with implementations where CGLP also incorporates approximations of $Q(x)$ such as subgradient cuts or ad-hoc lower bounds rather than exact evaluations only. That would require also keeping track of firt-stage vectors x for which estimates of $Q(x)$ have been computed.
5. Finally, in more general settings where $Q(x)$ is an easy-to-evaluate nonconvex function for which a tractable convex underestimator is not available, CGLP-based cuts may prove helpful in solving problems having the form (IP). Situations where $Q(x)$ is given by black-box computations remain a case study to be explored.

References

[1] M. Albareda-Sambola, M.H. van der Vlerk, and E. Fernández, *Exact solutions to a class of stochastic generalized assignment problems*, European journal of operational research **173** (2006), 465–487.

[2] G. Angulo, S. Ahmed, S.S. Dey, and V. Kaibel, *Forbidden vertices*, Optimization Online (2013).

[3] E. Balas, *Disjunctive Programming*, Discrete Optimization II (E.L. Johnson, P.L. Hammer, and B.H. Korte, eds.), Annals of Discrete Mathematics, vol. 5, Elsevier, 1979, pp. 3–51.

[4] J.F. Benders, *Partitioning procedures for solving mixed-variables programming problems*, Numerische Mathematik **4** (1962), 238–252.

[5] D. Gade, S. Küçükyavuz, and S. Sen, *Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs*, Mathematical Programming (2012), 1–26.

[6] M. Gendreau, G. Laporte, and R. Séguin, *An exact algorithm for the vehicle routing problem with stochastic demands and customers*, Transportation Science **29** (1995), 143–155.

[7] S. Küçükyavuz and M. Zhang, *Finitely convergent decomposition algorithms for two-stage stochastic pure integer programs*, Optimization Online (2013).

[8] G. Laporte and F.V. Louveaux, *The integer L-shaped method for stochastic integer programs with complete recourse*, Operations Research Letters **13** (1993), 133–142.

[9] G. Laporte, F.V. Louveaux, and H. Mercure, *A priori optimization of the probabilistic traveling salesman problem*, Operations research **42** (1994), 543–549.

[10] G. Laporte, F.V. Louveaux, and L. van Hamme, *Exact solution to a location problem with stochastic demands*, Transportation Science **28** (1994), 95–103.

[11] ———, *An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands*, Operations Research **50** (2002), 415–423.

[12] L. Ntaimo and S. Sen, *The million-variable march for stochastic combinatorial optimization*, Journal of Global Optimization **32** (2005), 385–400.

[13] L. Ntaimo and M.W. Tanner, *Computations with disjunctive cuts for two-stage stochastic mixed 0-1 integer programs*, Journal of Global Optimization **41** (2008), 365–384.

[14] Y. Qi and S. Sen, *Ancestral Benders' cuts and multi-term disjunctions for mixed-integer recourse decisions in stochastic programming*, Optimization Online (2013).

[15] S. Sen and J.L. Higle, *The C^3 theorem and a D^2 algorithm for large scale stochastic mixed-integer programming: set convexification*, Mathematical Programming **104** (2005), 1–20.

[16] R.M. Van Slyke and R. Wets, *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM Journal on Applied Mathematics **17** (1969), 638–663.

Appendix

Stochastic server location problem

Instance	Rep.	Std-Std		Std-CGLP		Alt-Std		Alt-CGLP	
		Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
SSLP.10.50.50	a	478	61.1	485	67.8	468	7.0	466	7.1
	b	452	91.3	434	85.5	472	6.7	466	6.7
	c	300	79.3	297	80.0	303	7.0	298	7.1
	d	237	25.3	224	26.9	230	5.0	230	5.0
	e	545	97.5	534	97.2	561	8.2	561	8.3
SSLP.10.50.100	a	452	109.7	434	106.2	462	17.1	470	18.3
	b	497	80.3	494	83.7	493	11.0	493	11.1
	c	313	95.3	291	98.0	302	12.8	289	13.6
	d	216	49.5	224	43.6	229	10.7	229	10.5
	e	373	120.5	422	121.0	373	14.2	374	14.4
SSLP.10.50.500	a	466	605.5	470	643.9	472	63.3	476	63.5
	b	441	482.6	447	492.9	449	57.7	449	57.8
	c	277	571.7	292	557.6	275	64.0	271	64.8
	d	235	348.8	239	353.5	247	57.5	247	57.6
	e	486	733.8	477	760.8	491	77.5	482	84.0
SSLP.10.50.1000	a	481	1542.1	473	1549.6	486	134.5	487	135.4
	b	473	1128.7	477	1142.2	460	114.5	466	116.8
	c	276	1509.3	261	1509.7	282	124.2	279	125.5
	d	225	752.8	227	782.2	229	113.2	229	113.7
	e	345	1537.6	351	1551.8	380	154.4	380	155.3
SSLP.10.50.2000	a	466	3777.1	467	3769.2	472	382.7	478	373.2
	b	472	2565.3	471	2751.8	483	246.7	478	251.0
	c	286	3189.4	286	3158.8	302	368.9	300	360.5
	d	219	1937.1	219	1994.5	223	249.0	225	249.4
	e	518	5021.2	414	4129.2	542	449.4	542	449.6
SSLP.15.45.5	a	230	11.3	233	11.6	244	0.7	244	0.7
	b	261	2.9	262	3.0	270	0.5	262	0.5
	c	2364	320.9	2288	354.9	2298	9.9	2294	10.2
	d	870	56.2	826	58.7	872	1.4	888	1.7
	e	138	16.4	142	16.8	133	1.0	135	1.0
SSLP.15.45.10	a	430	79.0	442	80.1	429	2.7	428	2.8
	b	284	189.0	251	190.9	256	6.2	278	7.6
	c	2384	245.0	2240	236.6	2512	7.4	2449	7.7
	d	2534	1090.7	2550	906.8	2606	7.9	2501	8.0
SSLP.15.45.15	a	1408	1646.1	1329	1594.5	1368	13.1	1358	13.3
	b	223	55.7	216	55.5	212	2.3	219	2.3
	c	2676	580.6	2718	611.0	2791	19.0	2785	18.9
	d	2986	359.1	2994	404.1	3038	22.3	3024	23.0
	e	207	30.1	235	30.2	221	1.9	232	1.9
SSLP.15.45.20	a	498	186.4	469	181.4	506	4.0	523	4.1
	b	351	87.2	335	87.5	341	7.6	331	7.6
	c	380	196.8	358	193.4	380	5.1	387	5.2
	d	552	873.0	548	898.1	560	20.7	562	20.9
	e	697	448.4	697	378.5	715	2.8	709	2.8
SSLP.15.45.25	a	658	554.1	629	532.0	662	18.4	633	18.5
	b	671	324.7	620	435.1	670	9.0	680	6.7
	c	433	165.2	399	160.7	447	11.8	422	11.9
	d	965	435.2	946	465.7	967	26.9	1001	32.3
	e	938	2062.7	900	1931.0	943	17.6	925	17.8

Table 6: Stochastic server location: overall results per instance.

Instance	Rep.	Std-Std				Alt-Std			
		#LP	#MIP	Time LP	Time MIP	#LP	#MIP	Time LP	Time MIP
SSLP.10.50.50	a	189	189	2.2	55.5	185	3	2.1	1.7
	b	154	154	1.6	86.2	166	3	1.6	1.6
	c	113	113	1.6	74.5	109	4	1.6	2.3
	d	44	44	0.6	21.3	45	2	0.6	0.9
	e	238	238	2.8	91.6	238	5	2.7	2.6
SSLP.10.50.100	a	181	181	4.3	98.4	183	6	3.9	6.4
	b	176	176	4.0	69.7	175	2	3.7	0.9
	c	112	112	3.1	86.2	109	5	2.9	4.1
	d	51	51	1.3	40.9	50	2	1.2	2.2
	e	138	138	3.7	109.8	137	4	3.3	4.0
SSLP.10.50.500	a	178	178	21.1	549.8	179	2	19.1	11.0
	b	152	152	17.8	428.5	150	2	15.2	7.4
	c	89	89	13.7	523.9	89	4	12.0	18.6
	d	56	56	8.1	303.3	56	2	8.1	12.4
	e	183	183	21.1	680.4	179	5	19.8	26.7
SSLP.10.50.1000	a	188	188	46.4	1429.6	185	3	41.9	29.4
	b	163	163	36.6	1028.5	156	2	32.4	21.2
	c	106	106	29.6	1410.1	95	3	25.8	30.2
	d	56	56	16.0	665.2	55	2	15.3	27.3
	e	147	147	39.4	1433.1	145	5	35.5	56.1
SSLP.10.50.2000	a	184	184	92.6	3548.0	181	5	82.4	169.9
	b	158	158	70.3	2352.7	156	2	65.4	44.2
	c	98	98	60.7	2980.4	103	5	56.6	167.0
	d	59	59	34.2	1746.0	58	2	31.2	62.1
	e	214	214	104.3	4785.4	219	7	101.0	222.7
SSLP.15.45.5	a	28	28	0.1	10.9	28	2	0.1	0.2
	b	42	42	0.1	2.5	41	4	0.1	0.2
	c	481	481	1.0	318.3	496	17	0.9	7.7
	d	154	154	0.3	55.2	141	4	0.3	0.6
	e	10	10	0.0	16.1	10	2	0.0	0.7
SSLP.15.45.10	a	93	93	0.3	77.8	90	2	0.3	1.6
	b	68	68	0.2	188.2	67	5	0.2	5.4
	c	501	501	2.2	240.1	538	9	2.3	2.9
	d	386	386	1.7	1086.8	379	5	1.7	4.4
SSLP.15.45.15	a	263	263	1.6	1642.3	262	4	1.5	9.7
	b	41	41	0.2	54.4	39	2	0.2	1.0
	c	623	623	4.3	572.8	645	16	4.4	11.8
	d	597	597	3.3	352.0	613	6	3.1	16.2
	e	29	29	0.2	29.0	28	2	0.2	0.8
SSLP.15.45.20	a	134	134	0.9	183.7	132	2	0.9	1.4
	b	63	63	0.4	85.1	61	2	0.4	5.6
	c	61	61	0.4	195.2	60	4	0.4	3.6
	d	148	148	1.1	870.2	145	6	1.0	18.0
	e	91	91	0.7	446.2	94	2	0.7	0.7
SSLP.15.45.25	a	156	156	1.3	550.0	147	4	1.2	14.4
	b	135	135	1.3	321.0	148	4	1.4	5.3
	c	73	73	0.6	162.1	74	4	0.6	8.8
	d	213	213	2.2	430.0	215	7	2.1	21.9
	e	235	235	2.0	2058.5	231	8	1.9	13.7

Table 7: Stochastic server location: subproblems details per instance.

Stochastic multiple binary knapsack problem

Instance	Std-Std		Std-CGLP		Alt-Std		Alt-CGLP	
	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
1	27705	26.4	27615	27.0	27837	25.5	26295	24.9
2	63528	41.1	55448	38.6	65213	41.1	57170	38.2
3	93185	59.9	87560	60.2	101121	57.8	81480	50.9
4	137303	101.1	121687	97.3	132782	89.1	127963	89.1
5	224063	107.2	183462	94.1	244755	112.1	251017	128.3
6	363405	186.6	290404	177.5	355962	190.1	258420	165.5
7	503998	245.8	401809	204.4	517313	250.4	397677	200.2
8	436738	267.5	310136	218.0	431356	249.3	334569	214.8
9	470356	273.3	451931	269.7	502174	280.5	450104	254.8
10	507120	315.6	320672	251.1	518329	333.1	342582	257.5
11	623424	379.4	675292	404.9	637749	422.5	615580	342.6
12	887595	468.7	672117	422.7	954211	502.8	741931	436.2
13	1099397	541.0	1024147	692.2	1172464	579.1	984003	527.4
14	1416129	686.6	880154	516.9	1484427	711.5	1057895	600.3
15	1650580	714.4	1120524	509.8	1692521	726.8	1148229	516.0
16	1322774	749.9	832266	533.7	1013473	572.2	956447	579.2
17	1197577	771.1	900476	652.4	1192205	753.2	974525	686.0
18	1230166	836.7	988346	769.7	1166769	811.6	978490	745.9
19	2189204	1158.0	1618305	950.0	2225393	1160.4	1713778	962.0
20	2395096	1460.9	1663945	1142.5	2383548	1404.2	1756720	1109.5
21	3277812	1488.2	2789613	1328.8	3563188	1603.1	3144784	1499.3
22	2702878	1664.7	2244862	1422.6	2816341	1714.0	2087732	1430.1
23	2309196	1825.3	1919811	1711.6	2306792	1715.5	1833302	1520.5
24	3301135	1998.1	2690441	1771.6	3101311	1816.8	2580654	1620.4
25	3346788	2310.7	2987190	2149.9	3541754	2346.8	2998747	2068.1
26	3024670	2319.8	2966064	2373.0	3087399	2258.1	2806757	2172.3
27	3890594	2344.4	3225433	2099.7	3787260	2210.1	3128508	1980.4
28	4762714	3223.2	3253202	2425.3	4449516	2890.2	3285741	2311.3
29	5717652	3589.2	3840063	2795.1	5819471	3597.8	3885068	2645.9

Table 8: Stochastic multiple knapsack: overall results per instance.

Instance	Std-Std				Std-CGLP			
	#LP	#MIP	Time LP	Time MIP	#LP	#MIP	Time LP	Time MIP
1	9	9	0.0	14.4	9	9	0.0	14.7
2	9	9	0.0	15.7	9	9	0.0	15.8
3	14	14	0.0	24.7	14	14	0.0	26.0
4	24	24	0.0	46.2	24	24	0.0	45.9
5	12	12	0.0	21.0	12	12	0.0	19.8
6	14	14	0.0	32.1	20	20	0.0	46.2
7	10	10	0.0	17.3	10	10	0.0	17.2
8	40	40	0.1	80.3	40	40	0.1	80.7
9	34	34	0.1	74.5	36	36	0.1	74.6
10	46	46	0.1	97.9	49	49	0.1	102.1
11	46	46	0.1	77.6	47	47	0.1	78.9
12	45	45	0.1	108.5	51	51	0.1	123.0
13	45	45	0.1	87.2	87	87	0.2	160.9
14	51	51	0.1	124.4	51	51	0.1	123.9
15	22	22	0.0	29.9	26	26	0.1	36.2
16	79	79	0.2	128.8	74	74	0.2	119.3
17	80	80	0.2	168.0	81	81	0.2	167.5
18	92	92	0.2	194.7	97	97	0.2	202.1
19	66	66	0.1	134.3	65	65	0.1	131.9
20	97	97	0.2	193.0	98	98	0.2	193.9
21	49	49	0.1	99.8	48	48	0.1	97.7
22	93	93	0.2	245.2	91	91	0.2	237.2
23	175	175	0.4	341.7	176	176	0.4	339.1
24	89	89	0.2	211.2	92	92	0.2	221.1
25	127	127	0.3	222.2	127	127	0.3	221.8
26	155	155	0.3	331.5	157	157	0.3	331.4
27	103	103	0.2	246.0	111	111	0.2	264.1
28	150	150	0.3	263.7	152	152	0.3	268.0
29	131	131	0.3	259.4	147	147	0.3	287.6

Table 9: Stochastic multiple knapsack: subproblems details per instance.

Instance	$\|V\|$	Update	Generate
1	9	0.0	0.2
2	9	0.0	0.2
3	14	0.0	0.4
4	24	0.0	1.0
5	12	0.0	0.3
6	20	0.0	0.8
7	10	0.0	0.2
8	40	0.1	2.7
9	36	0.1	2.5
10	49	0.1	3.4
11	47	0.1	4.7
12	51	0.1	4.5
13	87	0.2	13.2
14	51	0.1	5.3
15	26	0.1	1.3
16	74	0.2	11.9
17	81	0.3	15.7
18	97	0.4	21.2
19	65	0.2	6.9
20	98	0.3	21.7
21	48	0.1	4.0
22	91	0.3	18.8
23	176	0.9	41.2
24	92	0.3	13.8
25	127	0.4	29.8
26	157	0.6	37.0
27	111	0.4	25.1
28	152	0.6	39.6
29	147	0.6	33.3

Table 10: Stochastic multiple knapsack: CGLP overhead per instance.

On a Cardinality-Constrained Transportation Problem With Market Choice

Matthias Walter[a], Pelin Damcı-Kurt[b], Santanu S. Dey[c,*], Simge Küçükyavuz[b]

[a]*Institut für Mathematische Optimierung, Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany*
[b]*Department of Integrated Systems Engineering, The Ohio State University, Columbus, OH 43210, United States*
[c]*School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, United States*

Abstract

It is well-known that the intersection of the matching polytope with a cardinality constraint is integral [8]. In this note, we prove a similar result for the polytope corresponding to the transportation problem with market choice (TPMC) (introduced in [4]) when the demands are in the set $\{1,2\}$. This result generalizes the result regarding the matching polytope. The result in this note uses the fact that some special classes of minimum weight perfect matching problem with a cardinality constraint on a subset of edges can be solved in polynomial time.

Keywords: Transportation problem with market choice, cardinality constraint, integral polytope

1. Introduction and Main Result

1.1. Transportation Problem with Market Choice

The transportation problem with market choice (TPMC), introduced in the paper [4], is a transportation problem in which suppliers with limited capacities have a choice of which demands (markets) to satisfy. If a market is selected, then its demand must be satisfied fully through shipments from the suppliers. If a market is rejected, then the corresponding potential revenue is lost. The objective is to minimize the total cost of shipping and lost revenues. See [5, 7, 9] for approximation algorithms and heuristics for several other supply chain planning and logistics problems with market choice.

Formally, we are given a set of supply and demand nodes that form a bipartite graph $G = (V_1 \cup V_2, E)$. The nodes in set V_1 represent the supply nodes, where for $i \in V_1$, $s_i \in \mathbb{N}$ represents the capacity of supplier i. The nodes in set V_2 represent the potential markets, where for $j \in V_2$, $d_j \in \mathbb{N}$ represents the demand of market j. The edges between supply and demand nodes have weights that represent shipping costs w_e, where $e \in E$. For each $j \in V_2$, r_j is the revenue lost if the market j is rejected. Let $x_{\{i,j\}}$ be the amount of demand of market j satisfied by supplier i for $\{i,j\} \in E$, and let z_j be an indicator variable taking a value 1 if market j is rejected and 0 otherwise. A mixed-integer programming (MIP) formulation of the problem is given where the objective is to minimize the transportation costs and the lost revenues due to unchosen markets:

$$\min_{x \in \mathbb{R}_+^{|E|},\, z \in \{0,1\}^{|V_2|}} \quad \sum_{e \in E} w_e x_e + \sum_{j \in V_2} r_j z_j \tag{1}$$

$$\text{s.t.} \quad \sum_{i:\{i,j\} \in E} x_{\{i,j\}} = d_j(1 - z_j) \qquad \forall j \in V_2 \tag{2}$$

$$\sum_{j:\{i,j\} \in E} x_{\{i,j\}} \le s_i \qquad \forall i \in V_1. \tag{3}$$

[*] Corresponding author
Email addresses: matthias.walter@ovgu.de (Matthias Walter), damci-kurt.1@osu.edu (Pelin Damcı-Kurt), santanu.dey@isye.gatech.edu (Santanu S. Dey), kucukyavuz.2@osu.edu (Simge Küçükyavuz)

We refer to the formulation (1)-(3) as TPMC. The first set of constraints (2) ensures that if market $j \in V_2$ is selected (i.e., $z_j = 0$), then its demand must be fully satisfied. The second set of constraints (3) model the supply restrictions.

TPMC is strongly NP-complete in general [4]. Aardal and Le Bodic [1] give polynomial-time reductions from this problem to the capacitated facility location problem [6], thereby establishing approximation algorithms with constant factors for the metric case and a logarithmic factor for the general case.

1.2. TMPC with $d_j \in \{1, 2\}$ for all $j \in V_2$ and the Matching Polytope

When $d_j \in \{1, 2\}$ for each demand node $j \in V_2$, TPMC is polynomially solvable [4]. This is proven through a reduction to a minimum weight perfect matching problem on a general (non-bipartite) graph $G' = (V', E')$; see [4]. We call this special class of the problem, the *simple TPMC problem* in the rest of this note.

Observation 1 (Simple TPMC generalizes Matching on General Graphs)**.** *The matching problem can be seen as a special case of the simple TPMC problem. Let $G = (V, E)$ be a graph with n vertices and m edges. We construct a bipartite graph $\hat{G} = (\hat{V}^1 \cup \hat{V}^2, \hat{E})$ as follows: $\hat{V}^1$ is a set of n vertices corresponding to the n vertices in G, and $\hat{V}^2$ corresponds to the set of edges of G, i.e., $\hat{V}^2$ contains m vertices. We use $\{i, j\}$ to refer to the vertex in $\hat{V}^2$ corresponding to the edge $\{i, j\}$ in E. The set of edges in $\hat{E}$ are of the form $\{i, \{i, j\}\}$ and $\{j, \{i, j\}\}$ for every $i, j \in V$ such that $\{i, j\} \in E$. Now we can construct (the feasible region of) an instance of TPMC with respect to $\hat{G} = (\hat{V}^1 \cup \hat{V}^2, \hat{E})$ as follows:*

$$Q = \{(x, z) \in \mathbb{R}^{2m}_{+} \times \mathbb{R}^m \mid x_{\{i,e\}} + x_{\{j,e\}} + 2z_e = 2 \ \forall e = \{i, j\} \in \hat{V}^2 \tag{4}$$

$$\sum_{j:\{i,j\}\in E} x_{\{i,\{i,j\}\}} \leq 1 \ \forall i \in \hat{V}^1 \tag{5}$$

$$z_e \in \{0, 1\} \ \forall e \in \hat{V}^2\}. \tag{6}$$

Clearly there is a bijection between the set of matchings in G and the set of solutions in Q. Moreover, let

$$H := \{(x, z, y) \in \mathbb{R}^{2m} \times \mathbb{R}^m \times \mathbb{R}^m \mid (x, z) \in Q, y = e - z\},$$

where e is the all ones vector in $\mathbb{R}^m$. Then we have that the incidence vector of all the matchings in $G = (V, E)$ is precisely the set $\mathrm{proj}_y(H)$.

Note that the instances of the form of (4)-(6) are special cases of simple TPMC instances, since in these instances all s_i's are restricted to be exactly 1 *and all d_j's are restricted to be exactly* 2.

1.3. Simple TPMC with Cardinality Constraint: Main Result

An important and natural constraint that one may add to the TPMC problem is that of a service level, that is the number of rejected markets is restricted to be at most k. This restriction can be modelled using a *cardinality constraint*, $\sum_{j \in V_2} z_j \leq k$, appended to (1)-(3). We call the resulting problem cardinality-constrained TPMC (CCTPMC). If we are able to solve CCTPMC in polynomial-time, then we can solve TPMC in polynomial time by solving CCTPMC for all $k \in \{0, \dots, |V_2|\}$. Since TPMC is NP-hard, CCTPMC is NP-hard in general.

In this note, we examine the effect of appending a cardinality constraint to the simple TPMC problem.

Theorem 1. *Given an instance of TPMC with V_2, the set of demand nodes, and E, the set of edges, let $X \subseteq \mathbb{R}^{|E|}_{+} \times \{0, 1\}^{|V_2|}$ be the set of feasible solutions of this instance of TPMC. Let $k \in \mathbb{Z}_+$ and $k \leq |V_2|$. Let $X^k := \mathrm{conv}(X \cap \{(x, z) \in \mathbb{R}^{|E|}_{+} \times \{0, 1\}^{|V_2|} \mid \sum_{j \in V_2} z_j \leq k\})$. If $d_j \leq 2$ for all $j \in V_2$, then $X^k = \mathrm{conv}(X) \cap \{(x, z) \in \mathbb{R}^{|E|}_{+} \times [0, 1]^{|V_2|} \mid \sum_{j \in V_2} z_j \leq k\}$.*

Our proof of Theorem 1 is presented in Section 2. We note that the result of Theorem 1 holds even when X^k is defined as $\mathrm{conv}(X \cap \{(x, z) \in \mathbb{R}^{|E|}_{+} \times \{0, 1\}^{|V_2|} \mid \sum_{j \in V_2} z_j \geq k\})$ or $\mathrm{conv}(X \cap \{(x, z) \in \mathbb{R}^{|E|}_{+} \times \{0, 1\}^{|V_2|} \mid \sum_{j \in V_2} z_j = k\})$.

By invoking the ellipsoid algorithm and the use of Theorem 1 we obtain the following corollary.

Corollary 1. *Cardinality constrained simple TPMC is polynomially solvable.*

We note that, as a consequence of Theorem 1 (but also inherent in our proof), a special class of minimum weight perfect matching problem with a cardinality constraint on a subset of edges can be solved in polynomial time: Simple TPMC can be reduced to a minimum weight perfect matching problem on a general (non-bipartite) graph $G' = (V', E')$ [4]. Therefore, it is possible to reduce CCTPMC with $d_j \leq 2$ for all $j \in V_2$ to a *minimum weight perfect matching problem with a cardinality constraint on a subset of edges.* Hence, Corollary 1 implies that a special class of minimum weight perfect matching problems with a cardinality constraint on a subset of edges can be solved in polynomial time.

Note that the intersection of the perfect matching polytope with a cardinality constraint on a strict subset of edges is not always integral.

Example 1. *Consider the cycle C_4 of length 4 with edge set $E = \{\{1,2\},\{2,3\},\{3,4\},\{1,4\}\}$, and the cardinality constraint $x_{12} + x_{34} = 1$. The only perfect matchings are $\{\{1,2\},\{3,4\}\}$ and $\{\{1,4\},\{2,3\}\}$ for which the cardinality constraint has activity 2 and 0, respectively. Thus the perfect matching polytope is a line which is intersected by the hyperplane defined by the cardinality constraint in the (fractional) center.*

To the best of our knowledge, the complexity status of minimum weight perfect matching problem on a general graph with a cardinality constraint on a subset of edges is open. This can be seen by observing that if one can solve minimum weight perfect matching problem with a cardinality constraint on a subset of edges in polynomial time, then one can solve the exact perfect matching problem in polynomial time; see discussion in the last section in [2].

Finally we ask the natural question: Does the statement of Theorem 1 hold when $d_j \leq 2$ does not hold for every j? The next example illustrates that the statement does not hold in such case.

Example 2. *Consider an instance of TPMC where $G = (V_1 \cup V_2, E)$ is a bipartite graph with*

$$\begin{aligned}
V_1 &= \{i_1, i_2, i_3, i_4, i_5, i_6\}, \qquad V_2 = \{j_1, j_2, j_3, j_4\},\\
E &= \{\{i_1, j_1\}, \{i_2, j_2\}, \{i_3, j_3\}, \{i_4, j_1\}, \{i_4, j_4\}, \{i_5, j_2\}, \{i_5, j_4\}, \{i_6, j_3\}, \{i_6, j_4\}\},\\
s_i &= 1, i \in V_1, \qquad d_{j_1} = d_{j_2} = d_{j_3} = 2,\, d_{j_4} = 3.
\end{aligned}$$

For $k = 2$ it can be verified that we obtain a non-integer extreme point of $\operatorname{conv}(X) \cap \{(x,z) \in \mathbb{R}^p_+ \times [0,1]^n \mid \sum_{j=1}^n z_j \leq k\}$, given by $x_{\{i_1,j_1\}} = x_{\{i_2,j_2\}} = x_{\{i_3,j_3\}} = x_{\{i_4,j_1\}} = x_{\{i_4,j_4\}} = x_{\{i_5,j_2\}} = x_{\{i_5,j_4\}} = x_{\{i_6,j_3\}} = x_{\{i_6,j_4\}} = z_1 = z_2 = z_3 = z_4 = \frac{1}{2}$. To see this, consider the face defined by the supply constraints of nodes $\{i_4, i_5, i_6\}$ and observe that this face has precisely two solutions having 1 and 3 markets, respectively.

Therefore, $X^k \neq \operatorname{conv}(X) \cap \{(x,z) \in \mathbb{R}^p_+ \times [0,1]^n \mid \sum_{j=1}^n z_j \leq k\}$ in this example.

2. Proof of Theorem 1

To prove Theorem 1 we use an improved reduction to a minimum weight matching problem (compared to the reduction in [4]) and then use the well-known adjacency properties of the vertices of the perfect matching polytope. Since the integrality result does not hold for the perfect matching polytope on a general graph with a cardinality constraint on any subset of edges, as illustrated in Example 1, we need to refine the adjacency criterion.

We begin with some notation. For a graph $G = (V, E)$ with node set V and edge set E, and a node $v \in V$, we denote by $\delta(v) := \delta_G(v) := \{e \in E \mid v \in e\}$ the set of edges incident to v. For a vector $x \in \mathbb{R}^{|E|}$ and a subset $F \subseteq E$ of its ground set, we define $x(F) := \sum_{f \in F} x_f$.

We now describe the improved reduction to a minimum weight matching problem. Consider a simple TPMC instance on a graph $G = (V_1 \cup V_2, E)$ with supplies $s \in \mathbb{N}^{|V_1|}$, demands $d \in \{1,2\}^{|V_2|}$, edge weights $w \in \mathbb{R}^{|E|}$, and revenues $r \in \mathbb{R}^{|V_2|}$. Let $D_k = \{j \in V_2 \mid d_j = k\}$ be the partitioning of V_2 into two classes corresponding to the demands.

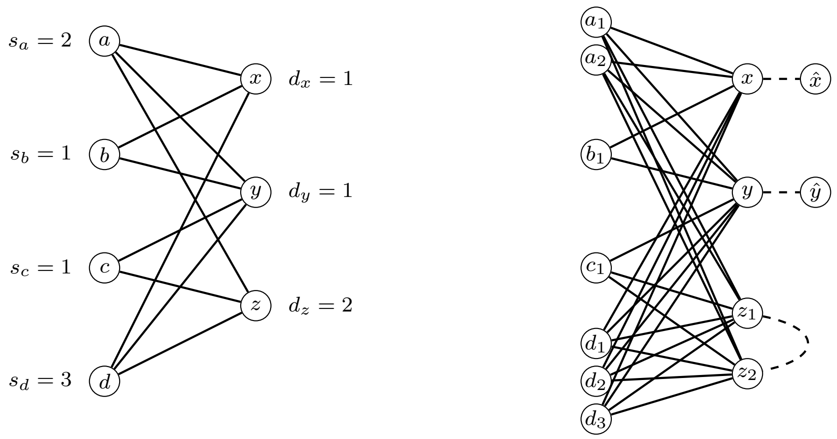


Figure 1: Improved Reduction to a Matching Problem

We create the auxiliary graph G^* (see Figure 1) with nodes $V_1^s \cup D_1 \cup \widehat{D}_1 \cup D_2^1 \cup D_2^2$ and edges $E_1 \cup E_2 \cup F_1 \cup F_2$ with

$$\begin{aligned}
V_1^s &= \{i_\ell \mid i \in V_1 \text{ and } \ell \in \{1,2,\ldots,s_i\}\},\\
\widehat{D}_1 &= \{\hat{j} \mid j \in D_1\},\\
D_2^k &= \{j_k \mid j \in D_2\} \text{ for } k = 1,2,\\
E_1 &= \{\{i_\ell, j\} \mid \{i,j\} \in E,\ \ell \in \{1,2,\ldots,s_i\} \text{ and } j \in D_1\},\\
E_2 &= \{\{i_\ell, j_k\} \mid \{i,j\} \in E,\ \ell \in \{1,2,\ldots,s_i\}, j \in D_2 \text{ and } k \in \{1,2\}\},\\
F_1 &= \{\{j, \hat{j}\} \mid j \in D_1\}, \text{ and}\\
F_2 &= \{\{j_1, j_2\} \mid j \in D_2\}.
\end{aligned}$$

In the construction every node $i \in V_1$ with supply s_i is split into s_i identical nodes with intended supply value of 1. Furthermore, to every node $j \in V_2$ with demand 1 we attach an edge with a dead end $\hat{j}$, and every node $j \in V_2$ with demand 2 is split into nodes j_1 and j_2 which are connected by an edge. Note that this is a polynomial construction, because the supply, s_i, is at most $2|V_2|$ for any $i \in V_1$.

Lemma 1. *Let $X \subseteq \mathbb{R}_+^{|E|} \times \{0,1\}^{|V_2|}$ be the set of feasible solutions of a simple TPMC instance on a graph $G = (V_1 \cup V_2, E)$ with supplies $s \in \mathbb{N}^{|V_1|}$ and demands $d \in \{1,2\}^{|V_2|}$. Let the sets D_k and the auxiliary graph G^* be defined as above.*

Then $P := \operatorname{conv}(X)$ is equal to the projection of the face of the matching polytope $P_{match}(G^*)$ of G^*

$$Q := \{y \in P_{match}(G^*) \mid y(\delta(v)) = 1 \text{ for all } v \in D_1 \cup D_2^1 \cup D_2^2\}$$

via the map π defined by $x_{\{i,j\}} = \sum_{\ell=1}^{s_i} y_{\{i_\ell,j\}}$ for $\{i,j\} \in E$ and $j \in D_1$, $x_{\{i,j\}} = \sum_{\ell=1}^{s_i} (y_{\{i_\ell,j_1\}} + y_{\{i_\ell,j_2\}})$ for $\{i,j\} \in E$ and $j \in D_2$, $z_j = y_{\{j,\hat{j}\}}$ for $j \in D_1$, and $z_j = y_{\{j_1,j_2\}}$ for $j \in D_2$.

Proof. We first show $\pi(Q) \subseteq P$. Let y be a vertex of Q and $(x,z) = \pi(y)$ be the projection.

Clearly, for all $i \in V_1$ we have $x(\delta_G(i)) = \sum_{\ell=1}^{s_i} y(\delta_{G^*}(i_\ell)) \leq s_i$, i.e., (x,z) satisfies (3). For every node $j \in D_1$ we have $x(\delta_G(j)) + z_j = y(\delta_{G^*}(j) \setminus \{\{j,\hat{j}\}\}) + y_{\{j,\hat{j}\}} = y(\delta_{G^*}(j)) = 1$. Furthermore, for every node $j \in D_2$ we have $x(\delta_G(j)) + 2z_j = y(\delta_{G^*}(j_1) \setminus \{\{j_1,j_2\}\}) + y(\delta_{G^*}(j_1) \setminus \{\{j_1,j_2\}\}) + 2y_{\{j_1,j_2\}} = y(\delta_{G^*}(j_1)) + y(\delta_{G^*}(j_2)) = 2$. Hence, (x,z) satisfies (2) proving $(x,z) \in \operatorname{conv}(X)$ since z is binary.

We now show $P \subseteq \pi(Q)$ for which it suffices to consider only integer points in P since both polytopes are integral. Note that P is integral since for integral z the remaining system is totally unimodular with integral right-hand side. Let $(x,z) \in P \cap (\mathbb{Z}_+^{|E|} \times \{0,1\}^{|V_2|})$ be an integral point in P. For $j \in D_1$, let $e_j \in E$ be the unique edge with $x_{\{i,j\}} > 0$, and for $j \in D_2$, let $\{e_j, f_j\}$ be the set of edges incident to j with positive x-value. Observe that if $e_j = f_j$ holds, then $x_{e_j} = 2$, and otherwise $x_{e_j} = x_{f_j} = 1$.

Construct a matching M satisfying

$$\begin{aligned} M = \{\{j,\hat{j}\} \mid j \in D_1 \text{ with } z_j = 1\} \cup \{\{j_1, j_2\} \mid j \in D_2 \text{ with } z_j = 1\} \quad (7)\\ \cup \{\{i_\ell, j\} \mid j \in D_1 \text{ and } i \in e_j \text{ with } z_j = 0\} \quad (8)\\ \cup \{\{i_\ell, j_1\} \mid j \in D_2 \text{ and } i \in e_j \text{ with } z_j = 0\} \quad (9)\\ \cup \{\{i_\ell, j_2\} \mid j \in D_2 \text{ and } i \in f_j \text{ with } z_j = 0\} \quad (10)\end{aligned}$$

choosing ℓ in (8)–(10) such that every node $i_\ell \in V_1^s$ has at most one incident edge in M. This is possible since for each $i \in V_1$, G^* has s_i identical copies $i_1, \ldots, i_{s_i}$ and M has to contain at most $x(\delta_G(i)) \leq s_i$ edges incident to one of the copies because x is integral.

We first prove that M is indeed a matching. A node $j \in D_1$ is matched either to $\hat{j}$ (if $z_j = 1$) or by e_j. Similarly, either j_1 and j_2 are matched by the edge $\{j_1, j_2\}$ (again if $z_j = 1$) or by e_j and f_j, respectively.

The fact that M projects to (x,z) is easy to check by the construction of M according to (7)–(10). This concludes the proof. □

We now turn to the proof of Theorem 1. By definition of the projection map π in Lemma 1, the equation $z(V_2) = k$ corresponds to the equation $y(F_1 \cup F_2) = k$ in Q, that is,

$$P \cap \{(x,z) \in \mathbb{R}_+^{|E|} \times [0,1]^{|V_2|} \mid z(V_2) = k\} = \{\pi(y) \mid y \in Q \text{ with } y(F_1 \cup F_2) = k\}$$

holds. Hence, in order to show that the former is integral (and since π projects integral vectors to integral vectors), it suffices to prove the following claim:

Claim 1. *Let $X \subseteq \mathbb{R}_+^{|E|} \times \{0,1\}^{|V_2|}$ be the set of feasible solutions of a simple TPMC instance on a graph $G = (V_1 \cup V_2, E)$ with supplies $s \in \mathbb{N}^{|V_1|}$ and demands $d \in \{1,2\}^{|V_2|}$. Let the sets D_k and the auxiliary graph G^* be defined as above and let Q be as in Lemma 1.*

Then $\{y \in Q \mid y(F_1 \cup F_2) = k\}$ is an integral polytope for any integer $k \in \mathbb{Z}_+$.

Proof. Let $H = \{y \mid y(F_1 \cup F_2) = k\}$ denote the intersecting hyperplane and assume, for the sake of contradiction, that $Q \cap H$ is not integral. Then there must exist two adjacent (in Q) matchings M_1 and M_2 defining an edge of Q that is intersected by H in its relative interior, i.e., $|M_1 \cap (F_1 \cup F_2)| < k$ and $|M_2 \cap (F_1 \cup F_2)| > k$.

By the adjacency characterization of the matching polytope [3], the symmetric difference $C := M_1 \Delta M_2$ must be a connected component (a cycle or a path) in G^* containing edges of M_1 and M_2 in an alternating fashion.

We now verify that there must exist a path e-P-f in C of odd length consisting of two edges $e, f \in C \cap (F_1 \cup F_2)$ and a subpath P in $C \setminus (F_1 \cup F_2)$: If for every choice of $e, f \in M_2 \cap C \cap (F_1 \cup F_2)$ there exists an edge belonging to $M_1 \cap (F_1 \cup F_2)$ in all subpath(s) e-P-f of C, then M_2 can have at most one more edge of $F_1 \cup F_2$ than M_1 in C. However since M_2 contains at least two more edges of $F_1 \cup F_2$ than M_1 does, we have that there exists a path e-P-f in C consisting of two edges $e, f \in M_2 \cap C \cap (F_1 \cup F_2)$ and a subpath P in $C \setminus (F_1 \cup F_2)$. Now since e-P-f is subpath of C and $e, f \in M_2$, we have that P is of odd length.

Clearly, since we have $P \cap (F_1 \cup F_2) = \emptyset$, all of P's edges must go between V_1^s and $D_1 \cup (D_2^1 \cup D_2^2)$. Since P also has odd length, one of its endpoints is in V_1^s. But no edge in $F_1 \cup F_2$ is incident to any node in V_1^s which yields a contradiction. □

Acknowledgements.

Pelin Damcı-Kurt and Simge Küçükyavuz are supported, in part, by NSF-CMMI grant 1055668. Santanu S. Dey gratefully acknowledges the support of the Air Force Office of Scientific Research grant FA9550-12-1-0154.

References

[1] Aardal, K. and Le Bodic, P. (2014). Approximation algorithms for the transportation problem with market choice and related models. *Operations Research Letters*, 42:549–552.

[2] Berger, A., Bonifaci, V., Grandoni, F., and Schäfer, G. (2011). Budgeted matching and budgeted matroid intersection via the gasoline puzzle. *Math. Program.*, 128:355–372.

[3] Chvátal, V. (1975). On certain polytopes associated with graphs. *J. Combin. Theory Ser. B*, 18(2):138 – 154.

[4] Damcı-Kurt, P., Dey, S., and Küçükyavuz, S. (2015). On the transportation problem with market choice. *Discrete Applied Mathematics*, 181:54–77.

[5] Geunes, J., Levi, R., Romeijn, H. E., and Shmoys, D. B. (2009). Approximation algorithms for supply chain planning and logistics problems with market choice. *Math. Program.*, 130:85–106.

[6] Kuehn, A. A. and Hamburger, M. J. (1963). A heuristic program for locating warehouses. *Management Science*, 9(4):643–666.

[7] Levi, R., Geunes, J., Romeijn, H. E., and Shmoys, D. B. (2005). Inventory and facility location models with market selection. *Proceedings of the 12th IPCO*, pages 111–124.

[8] Schrijver, A. (2003). *Combinatorial Optimization.* Springer.

[9] Van den Heuvel, W., Kundakçıoğlu, O. E., Geunes, J., Romeijn, H. E., Sharkey, T. C., and Wagelmans, A. P. M. (2012). Integrated market selection and production planning: complexity and solution approaches. *Math. Program.*, 134:395–424.

1.

1. Report Type

Final Report

Primary Contact E-mail
Contact email if there is a problem with the report.

sahmed@isye.gatech.edu

Primary Contact Phone Number
Contact phone number if there is a problem with the report

4048942320

Organization / Institution name

Georgia Institute of Technology

Grant/Contract Title
The full title of the funded effort.

Selective Optimization

Grant/Contract Number
AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".

FA9550-12-1-0154

Principal Investigator Name
The full name of the principal investigator on the grant or contract.

Shabbir Ahmed

Program Manager
The AFOSR Program Manager currently assigned to the award

Fariba Fahroo

Reporting Period Start Date

04/01/2012

Reporting Period End Date

04/30/2015

Abstract

This project focuses on developing algorithms for optimization problems that have intrinsic limitations preventing the utilization of all available decision alternatives (problem variables) and/or the satisfaction of all constraints. Part of the optimization decision in these problems is the selection of which variables to use and/or which subset of constraints to satisfy. We refer to these problems as selective optimization (SO) problems. The combinatorial aspects of selection make these problems extremely difficult. In this project we develop a set of generic tools applicable to a wide class of selective optimization problems. Our approach is based on standard mixed-integer programming (MIP) formulations of selective optimization problems. While such formulations can be attacked by commercial optimization solvers, they typically exhibit extremely poor performance. We develop a variety of effective model and algorithm enhancement techniques for the standard MIP formulations. These techniques are easily integrable into commercial MIP solvers, thereby making them readily usable in applications of selective optimization.

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.

AFD-070820-035.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.

AFOSRannualReport2015.pdf

Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.

Archival Publications (published) during reporting period:

1. F. Qiu, S. Ahmed, S.S. Dey, L. Wolsey. ‘‘Covering linear programming with violations,’’ INFORMS Journal on Computing, vol. 26, pp. 531-546, 2014.
2. P. Damci-Kurt, S.S. Dey, S. Kucukyavuz. "On the transportation problem with market choice” Discrete Applied Mathematics, vol.181, pp.54-77, 2015.
3. S. Ahmed. ‘‘A scenario decomposition algorithm for 0-1 stochastic programs,’’ Operations Research Letters, vol. 41, pp. 565-569, 2013.
4. F. Qiu, S. Ahmed, S.S. Dey. “Strengthened bounds for probability of k-out-of-n events,” accepted for publication in Discrete Applied Math.
5. G. Angulo, S. Ahmed, S.S. Dey, V. Kaibel. ‘‘Forbidden vertices,’’ Mathematics of Operations Research, vol.40, pp.350-360, 2015.
6. G. Angulo, S. Ahmed, S.S. Dey. “Improving the integer L-shaped method,” under review in the INFORMS J. on Computing.
7. M. Walter, P. Damci-Kurt, S.S. Dey, S. Kucukyavuz. “On a Cardinality-Constrained Transportation Problem With Market Choice” submitted for publication, 2015

Changes in research objectives (if any):

None

Change in AFOSR Program Manager, if any:

None

Extensions granted or milestones slipped, if any:

None

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

```
Jun 16, 2015 09:45:26 Success: Email Sent to: sahmed@isye.gatech.edu
```