AFRL-AFOSR-VA-TR-2015-0283

Stochastic Dynamic Mixed-Integer Programming

Suvrajeet Sen
UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES

05/05/2015
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)* 28/04/2015	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From - To)* Feb. 2014 - Dec. 2015

4. TITLE AND SUBTITLE Stochastic Dynamic Mixed-Integer Programming (SD-MIP)	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER AFOSR Grant FA9550-13-1-0015
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Suvrajeet Sen	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Southern California	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Mixed-Integer Programming has traditionally been restricted to deterministic models. Recent research has opened the door to stochastic optimization models, which are typically dynamic in nature. This project lays the foundation for stochastic dynamic mixed-integer and linear programming (SD-MIP). This project has produced several new ideas in connection with a) convexification of two-stage mixed-integer sets and b) multi-stage (including two-stage) stochastic linear programming. Together a) and b) provide the foundations for SD-MIP problems. From new concepts and algorithms to applications and software, this project has made significant breakthroughs in all aspects. This report provides a synopsis of both theoretical and computational results. As a preview, we mention that currently available deterministic MIP solvers, as powerful as they are known to be, are unable to solve SD-MIP models of modest size within an hour of computing. In contrast, our decomposition approach provides provably optimal solutions within the hour time-limit.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**		21	Suvrajeet Sen
					19b. TELEPHONE NUMBER *(Include area code)* 213 740 6810

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.

Stochastic Dynamic Mixed-Integer Programming (SD-MIP)

AFOSR Grant FA9550-13-1-0015

Suvrajeet Sen
Epstein Department of Industrial and Systems Engineering,
University of Southern California, Los Angeles, CA 90089

April 2015

Abstract

Mixed-Integer Programming has traditionally been restricted to deterministic models. Recent research has opened the door to stochastic optimization models, which are typically dynamic in nature. This project lays the foundation for stochastic dynamic mixed-integer and linear programming (SD-MIP). This project has produced several new ideas in connection with a) convexification of two-stage mixed-integer sets and b) multi-stage (including two-stage) stochastic linear programming. Together a) and b) provide the foundations for SD-MIP problems. From new concepts and algorithms to applications and software, this project has made significant breakthroughs in all aspects. This report provides a synopsis of both theoretical and computational results. As a preview, we mention that currently available deterministic MIP solvers, as powerful as they are known to be, are unable to solve SD-MIP models of modest size within an hour of computing. In contrast, our decomposition approach provides provably optimal solutions within the hour time-limit.

Introduction

This project deals with the solution of (SD-MIP) problems which arise in several application domains of interest to the Air Force. For instance, multi-aircraft trajectory planning problems with aircraft dynamics lead to such SD-MIP problems. Our project focuses on the mathematical structure of, and algorithms for SD-MIP problems. We begin this report by first presenting the general mathematical structure of such models.

In stating the SD-MIP model, we borrow the notion of a scenario tree, commonly used in stochastic programming (SP). This construct (i.e., a scenario tree) provides the probabilistic structure of information that may be expected to evolve over the planning horizon. A scenario tree $\mathcal{N}$ consists of nodes (n) and with each node one associates a time index $t(n)$. The links of a scenario tree represent the flow of information from one scenario node (n) to a child node (n_+), and one has $t(n) < t(n_+)$, and the conditional probability $p(n + |n)$ is denoted p_{n+}.

An important concept in our formulation is the description of state variables, which consists of two components, that is, $\underline{s}_n := (x_n, \underline{\omega}_n)$ includes: $x_n \in \mathbb{R}^{q_n}$, an endogenous part of the state vector (or simply "endogenous state")[1], and $\underline{\omega}_n$ denoting an outcome of the exogenous random variable (or simply "exogenous state"). Here the term "endogenous" refers to the fact that the algorithm may be used to exercise some control on the trajectory of $\{x_n\}$, while the term "exogenous" refers to the fact that the algorithm does not exercise any control on the outcomes $\underline{\omega}_n$. For example, in a stochastic inventory (or reservoir) control model x_n will represent inventory (or water-level) while $\underline{\omega}_n$ will represent demand (or load/precipitation resp.). In situations where we need to refer to an entire arbitrary sample path (outcome), we simply denote it as $\underline{\omega}$. Finally, we adopt following notational convention: The root node is indexed by node 0, and an initial vector x_0 (as well as the information s_0) is assumed to be given (fixed).

Given a scenario tree, the SD-MIP model may be stated as follows.

$$\text{Min } \{d_0^\top u_0 + E\left[h_{0+}\left(\underline{\tilde{s}}_{0+}\right)\right] : u_0 \in U_0,\ x_{0+} = a_{0+} + A_{0+}x_0 + B_{0+}u_0 \} \tag{0}$$

where $\{h_n\}$ are defined recursively for $n \geq 1$ as

$$h_n\left(\underline{s}_n\right) = c_n^\top x_n + \text{ Min } \left\{ \begin{array}{c} d_n^\top u_n + E[h_{n+}\left(\underline{\tilde{s}}_{n+}\right)] \\ \text{s.t. } u_n \in U_n(\, x_n), \\ U_n(x_n) = \left\{u_n \in \mathbb{R}^{m_n} \times \mathbb{Z}^{\ell_n} \middle| \, D_n u_n \leq b_n - C_n x_n\right\} \end{array} \right\} \tag{1}$$

and,

$$x_{n+} = a_{n+} + A_{n+}x_n + B_{n+}u_n. \tag{2}$$

This is a very general setting for stochastic dynamic mixed-integer programming because it not only allows a multi-stage decision process, but it also allows both continuous and discrete variables $U_n(x_n) \subseteq \mathbb{R}^{m_n} \times \mathbb{Z}^{\ell_n}$ which evolve over time.

In our applications (e.g. mission planning for UAVs and other aircrafts), discrete choices (e.g. choosing which "way-points" to use) will be made at the start, and revised at an intermediate point of the mission. In total, this constitutes two points in time where discrete choices (way-points, targets etc.) are made, and several time points at which continuous choices are enacted (namely, speed, altitude, etc.) The cost associated with carrying out such missions is unknown at the start, and the total cost is typically revealed only at the end of the mission. Thus decisions (both discrete and continuous) are made under uncertainty. We should emphasize that our models are best suited for a collection

[1] As with $\underline{\omega}_n$ we use a uniform size of the vector x_n so as to keep the notation manageable.

of aircrafts, so that the synergy can be utilized to get the most out of the available resources. In this sense, we are interested in vector-valued decisions and constraint spaces.

This report outlines the main results produced to solve problems whose models may be looked upon as Stochastic – Dynamic Mixed-Integer Programming (MIP) problems. While the presence of discrete choices leads to MIP, it is important to recognize that the discrete and continuous decisions must work hand-in-hand. As a result, we will study both stochastic mixed-integer programming (SMIP) as well as stochastic linear programming (SLP) problems. By using a combination of ideas from cutting plane theory of deterministic MIP (especially disjunctive programming of Balas 1979) and nested decomposition (e.g., Birge 1985, Pereira and Pinto 1991), the algorithms developed in this study lead to several important breakthroughs.

a) In the SMIP thrust, we have developed the first general purpose algorithm for general stochastic mixed-integer programming problems. Here the term general is intended to convey the notion that randomness in the model is allowed in all data elements, and the discrete decisions are allowed to be general integers, rather than simply binary. These capabilities are particularly important because decisions like the size of the payload to deliver matters, and they are typically discrete. In addition, uncertainty in data is allowed to appears in every type of data element, be they objective function coefficients, constraint right-hand sides, or even within the constraint coefficients themselves. Prior studies of this class of problems have usually been restricted to special cases (e.g. binary). While others have attempted such generality in the past, there have not been any practical algorithms with which to conduct computational studies in the case of general SMIP models considered in this project. The algorithm which we have developed, referred to as the Ancestral Benders' Cutting plane (ABC) algorithm is by far the most general approach developed to date.

b) As part of this project, we have also developed tools for very large scale Stochastic Linear Programming (SLP). There are several reasons for this. First, SLP models continue to challenge many of the fastest computers to date, and many applications within the DoD (e.g. planning airlift operations) lead to some of the largest optimization models to date. Towards this end, we have developed a new notion of optimality that is especially geared towards very large scale SP models. This new concept, which we refer to as statistical optimality, allows one to predict the quality of a proposed decision on the basis of statistical tests which measure the likelihood that a proposed decision is within a certain tolerance level from an optimal solution. With this new concept, we have been able to show the validity of our algorithms for certain instances which have been identified by the Defense Science Board Report (2011) as one of the major challenges in tradestudies conducted for DoD acquisitions. We also observe that just as LP-technology is critical in the solution of large deterministic MIPs, the ability to solve large scale SLP problems is just as important in the solution of SD-MIP.

We begin this report with a presentation of the ABC algorithm for the two-stage case, and then show how such an algorithm may be applied to the kind of mission planning examples that were mentioned earlier. Following the summary of the discrete case, we will present the ideas underlying our SLP advances. In all cases, we provided a illustrative computational results, although our publications, and dissemination material include much more extensive computational results.

Ancestral Benders' Cutting-plane (ABC) Algorithm for Discrete Choices

The discussion presented in this section is based on material that is available in Qi and Sen (2015), although the notation used in that paper is more in keeping with two-stage models. Using the notation introduced in (0-2) a two-stage SMIP model may be stated as follows:

$$\text{Min}\ \{d_0^\top u_0 + E[h_n(s_n)] : u_0 \in U_0,\ x_n = a_n + A_n x_0 + B_n u_0 , u_0 \in \mathbb{R}^{m_0} \times \mathbb{Z}^{\ell_0} \} \qquad (3)$$

where the functions $\{h_n\}$ are defined for $n \geq 1$ as

$$h_n(s_n) = c_n^\top x_n + \ \text{Min} \left\{ \begin{matrix} d_n^\top u_n \\ \text{s.t. } u_n \in U_n(x_n), \\ U_n(x_n) = \left\{ u_n \in \mathbb{R}^{m_n} \times \mathbb{Z}^{\ell_n} \middle| D_n u_n \leq b_n - C_n x_n \right\} \end{matrix} \right\} \qquad (4)$$

where the probability $p_n > 0$ to be used in (3) is assumed to be given, and the number of outcomes indexed by n is finite. As in (0-2), the initial state x_0 and correspondingly s_0 are given. However, in contrast to the statement in (0-2), we have dropped the notation "0+" in the objective function in favor of using scenario nodes with index "0" in stage 1, and $n = 1, \dots, N$ in stage 2. One more observation about the second stage problem in (4) is that there is no future cost because the second stage is the end of the horizon. This simpler statement in the objective function allows us to focus on the main difficulty at hand, which are the mixed-integer requirements in the constraints for u_0 in the first-stage, and u_n in the second-stage. However, we will return to the case of terminal (third-stage) costs in the sub-section which extends the ABC algorithm.

The algorithm consists of two major elements: a first stage branch-and-bound (B&B) algorithm in which the objective function is approximated in a sequential manner. This B&B scheme is an extension of Benders' decomposition to cases where the objective function is difficult to approximate, as is the case in SMIP models. The next piece of the ABC algorithm is to provide set-convexification (of the second-stage), and value-function approximations which are easily managed within the B&B process. These two elements are described first, and we follow that up with how one might extend this idea to the class of DoD applications which motivated this study. We also provide a summary of our computational experience with SMIP instances, many of which are unsolvable by one of the most advanced commercial code.

First Stage Branch-and-Bound

The ABC algorithm starts out the branch-and-bound process (B&B) process at node 0 of the tree, with the objective function $d_0^\top u_0 + E[h_n(s_n)]$ replaced by a value function approximation as is common in dynamic programming; namely, the approximation has the following form.

$$\text{Min}\, \{d_0^\top u_0 + h_0^k(u_0) : u_0 \in U_0, u_0 \in \mathbb{R}^{m_0} \times \mathbb{Z}^{\ell_0}\,\} \tag{5}$$

As is common in the MIP literature, we assume that inputs to the instance include upper and lower bounds for all integer variables, and these box-constraints will be denoted Q_0.

While value functions h_0^k of MIP problems are notoriously complex to discover, and their use within (5) is even more complicated because of the presence of mixed-integer variables in (5). The value functions h_0^k have a close kinship with MIP duality, and Caroe and Tind (1998) were the first to propose using MIP duality to create these value function for use in (5). However, MIP value functions are themselves non-convex, and discontinuous in general, and solving (5) directly is impractical at best.

Our approach is to replace $h_0^k(u_0)$ and the set $u_0 \in \mathbb{R}^{m_0} \times \mathbb{Z}^{\ell_0}$ by creating functional approximations $h_{0,j}^k$ that are valid over a family of subsets $\{Q_{0,j}^k\}$, which cover all integers in Q_0. Hence the lower bounding approximations will replace (5) by the following family of problems, one for each $Q_{0,j}^k$.

$$\text{Min}\, \{d_0^\top u_0 + h_{0,j}^k(u_0) : u_0 \in U_0, u_0 \in Q_{0,j}^k\,\}. \tag{6}$$

The functions $\{h_{0,j}^k\}$ will be required to satisfy certain properties as specified below. To do so, recall that $x_n = a_n + A_n x_0 + B_n u_0$. Hence, for a given u_0 one can associate an MIP value denoted $h_n(s_n(u_0))$. Hence we will require the approximations $h_{0,j}^k(u_0)$ to satisfy

$$h_{0,j}^k(u_0) \leq \sum_n p_n h_n\big(s_n(u_0)\big) \ \ \forall\, u_0 \in Q_{0,j}^k\,. \tag{7}$$

Because of the above minorizing property, (6) will provide lower bounds on (5). In the B&B phase, each subset $Q_{0,j}^k$ will be associated with a first stage decision $u_{0,j}^k$, and a lower bound $v_{0,j}^k$ defined as

$$v_{0,j}^k \coloneqq\ d_0^\top u_{0,j}^k + h_{0,j}^k\big(u_{0,j}^k\big) \coloneqq\ \text{Min}\,\{d_0^\top u_0 + h_{0,j}^k(u_0) : u_0 \in U_0, u_0 \in Q_{0,j}^k\,\}. \tag{8}$$

Then, the B&B process of the ABC algorithm will estimate the most optimistic lower bound as

$$\bar{v}_0^k \coloneqq\ \text{Min}_j\,\big\{d_0^\top u_{0,j}^k + h_{0,j}^k\big(u_{0,j}^k\big)\big\}, \text{and } \bar{u}_0^k \in \text{argmin}\,\big\{d_0^\top u_{0,j}^k + h_{0,j}^k\big(u_{0,j}^k\big)\big\}, \tag{9}$$

If the solution $\bar{u}_0^k$ satisfies the mixed-integer requirement, then we update the incumbent and the upper bound, as in any B&B algorithm. On the other hand, if $\bar{u}_0^k$ does not satisfy the mixed-integer requirements, we identify which of the remaining boxes need further exploration, and depending on the exploration rule, one would refine one of the boxes in the collection $\{Q_{0,j}^k\}$. We should bear in mind however, that further branching should also be accompanied by obtaining improved approximations of the collection of functions $h_{0,j}^k$.

Improving Second-Stage Approximations

One of the main insights underlying the ABC algorithm is value functions to be used in the first stage can be improved by using parametric cutting planes similar to the ones that have been developed in the context of stochastic combinatorial optimization algorithms as in Sen and Higle (2005) and Sen and Sherali (2006). One of the major advantages of using parametric cutting planes in the second stage is that they are designed to adapt to adjust to changes in the first-stage decision which is a parameter for the second-stage in the context of decomposition algorithms. As a result, these cuts automatically provide a memory of past visits to fractional solutions. In addition, the strengthened relaxations provide stronger value function approximations (i.e. Benders' cuts) which are used to update $\{h_{0,j}^k\}$ as the iteration counter k increases. These second-stage approximations are described next.

In the notation used below, each node n will have variables u_n, and these will be influenced by decisions made by the parent node, designated as node 0 in the two stage case. Because we are interested in generalizing the two-stage algorithm to the multi-stage case, it will be convenient to refer to the parent node of n, as the node $n-$. Thus, in the following development, the variables u_0 and those referred to as u_{n-} will be the same. This shift in notation will facilitate our development for dynamics after stage 2 in our multi-stage application.

Because multi-term disjunctive sets have been shown to provide the convex hull of mixed-integer sets, we adopt the use of multi-term disjunctions to obtain cutting planes to convexify the second-stage problem. These cuts can be generated in one of two ways: a) by generating disjunctions based on the cutting plane tree (CPT) algorithm (Chen et al 2011) which was one of the fundamental results obtained from the 2011-12 portion of our AFOSR-funded research, or b) using a truncated B&B tree where the second-stage approximation for scenario node n is solved until either a maximum time or B&B node limit is reached. At that point in the algorithmic process for the second stage will typically have a fractional solution which can be cut away by using a parametric cutting plane. The resulting cuts will then be added to improve the approximation used in the second-stage constraints. The details of this strategy are summarized next.

Let u^k_{n-} be given, and suppose that we proceed to a scenario indexed by n. We then find an approximate solution u^k_n, and if this solution satisfies the mixed-integer requirements, then, no additional cutting planes will be added for the current iteration. On the other hand, if some integer restrictions are violated, then we generate a lifted cut in the space (u_{n-}, u_n). The cut will delete the point (u^k_{n-}, u^k_n), but will not delete any mixed-integer feasible pair (u_{n-}, u_n). In order to accomplish this task, let us first examine the following set.

$$\begin{gathered}\mathcal{Y}^k_{n,j} := \{(x_n, u_n) | C^k_n x_n + D^k_n u_n \le b^k_n,\ x_n = a_n + A_n x_0 + B_n u_{n-}, \\ u_{n-} \in Q^k_{n-,j}, u_n \in \left[\cup_\tau \left(Q^k_{n,\tau}\right)\right]\},\end{gathered} \tag{10}$$

where $u^k_n \notin \left[\cup_\tau \left(Q^k_{n,\tau}\right)\right]$. Note that the matrices C_n and D_n, and the vector b_n from (1) are subsumed within matrices C^k_n and D^k_n respectively, and the vector b^k_n. Also, x_0 is fixed. Thus, substituting the state variable x_n using the linear dynamics, we obtain a representation completely in terms of the variables (u_{n-}, u_n). This set has the following representation (with an appropriate matrix $\bar{C}^k_n$, and the right-hand side $\bar{b}^k_n$).

$$\mathcal{Z}^k_{n,j} := \{(u_{n-}, u_n) | \bar{C}^k_n u_{n-} + D^k_n u_n \le \bar{b}^k_n,\ u_{n-} \in Q^k_{n-,j}, u_n \in \left[\cup_\tau \left(Q^k_{n,\tau}\right)\right]\}. \tag{11}$$

The set in (11) is a disjunctive relaxation of the mixed-integer requirement, and can therefore be used to generate a valid inequality using disjunctive programming. Suppose that the resulting cut is denoted $\pi^k_{n-} u_{n-} + \pi^k_n u_n \le \rho^k_n$. Then, assuming B_n is invertible, and recalling that $x_{n-} = x_0$ (fixed), we obtain the cut coefficients in the space of state and decision variables (x_n, u_n), and the set (10) can now be updated as $\mathcal{Y}^{k+1}_{n,j}$, with appropriately updated matrices C^{k+1}_n, D^{k+1}_n and the vector b^{k+1}_n which include the new parametric cut.

Using disjunctive programming results (Balas 1979), and outcomes of our AFOSR-funded research (Chen et al 2011), we have shown that for the two stage case, this approach provides a finitely convergent algorithm as stated below.

Proposition 1. (Qi and Sen 2015) Suppose that the set of first and second stage decisions form a compact set of integers. Let the cuts be derived using multi-term disjunctive cuts as in Chen et al (2011), and the updates of the approximations in (10) and (11) are carried out accordingly. Finally, suppose that the first stage B&B method uses (6) and (7) as discussed above. Then the ABC algorithm terminates with an optimal solution in finitely many iterations. ■

Application of SMIP Models for Mission Planning

Consider a mission dealing with carrying out reconnaissance and/or addressing threats over possibly hostile territory. There are three stages of decision-making, with the first two involving the discrete choices: a) which target/threat/way-point to assign to a particular vehicle b) should we reassign this vehicle based on observed conditions, and c) identify the safest route back to base. The first decision will be denoted u_0, which will specify both a way-point, as well as the pay-load. While the way-point and pay-load combinations are important, they are not unique. In other words, if at some point in the future, the way-point changes, there may be a subset of locations which can be served with the pay-load decision chosen at the beginning. As in the presentation of the ABC algorithm, it will be more convenient to consider this variable as being denoted u_{n-}, with $n-$ representing node “0” of a scenario tree. Once this decision is made, and the mission begins, the aircrafts receive data, which allows them to make some adjustments (recourse decisions), which will be denoted u_n where, n is a label associated with the specific kind of data that is observed. After the decision u_n is implemented, the aircraft will carry out the mission, and will use a sequence of maneuvers which will deliver the pay-load, and bring the aircraft back to home base. These decisions will be represented as u_{n+} The decisions associated with returning to the base may be looked up as linear/quadratic programming problem, whereas the first two choices are based on discrete selections. This last aspect (delivering the pay-load and returning to home-base) is difficult to predict at the start of the mission, but will depend on which specific node of the scenario tree actually transpires. Hence decisions in this phase will be denoted u_{n+}. In our setup, the decisions u_{n-} and u_n are mixed-integer, whereas, u_{n+} is continuous. Thus, while the model is in fact a three stage model, only the first two stages have mixed-integer variables, and the setup of the ABC algorithm work is applicable to this situation. A summary of these choices are provided in Table 1.

Because such stochastic dynamic mixed-integer problems (SD-MIP) are computationally challenging, it is important to trade-off accuracy with tractability. We do so by using linear dynamics, although the parameters are all allowed to be governed by observations which could be stochastic. Solving such a model helps both planning and execution because optimal (risk minimization) strategies can be studied prior to the mission.

Table 1. Layout of Discrete and Continuous Decisions in Mission Planning

Stage	States	State Transition Equations	Decisions (Continuous/Integer)
0	$x_0 :=$ Locations of aircrafts in the fleet	x_0 is given	$u_0 :=$ Target, Pay-load combination. (Discrete Choices)
1	$x_n :=$ Locations and data (conditions) prior to adjustments	$x_n = a_n + A_n x_0 + B_n u_0$	$u_n :=$ Re-assessment/re-assignment based on conditions. (Discrete Choices)
2	$x_{n+} :=$ Locations and data immediately after the attack	$x_{n+} = a_{n+} + A_{n+} x_n + B_{n+} u_n$	$u_{n+} :=$ Return path to base (Continuous Choices)

It is also important to note that the stagewise approach of ABC is also consistent with many multi-stage SLP algorithms such as Nested Benders Decomposition, Stochastic Dual Dynamic Programming, and others. Accordingly, this application is very well suited for a combination of SMIP and SLP models.

Computational Results

Computational results with the ABC algorithm has demonstrated that deterministic MIP solvers, even the most sophisticated ones (e.g. CPLEX) are not reliable enough to solve SMIP models. The paper by Qi and Sen (2015) presents many examples where CPLEX fails after an hour computing on a desktop machine. However, the same machine is able to compute provably optimal solutions using the ABC algorithm. Some of the larger examples reported by Qi and Sen (2015) are summarized in Table 2. Here an instance of the type (*m*x5), (*n*x5) pertains to a problem with *m* potential server locations, each of which can accommodate 5 different types of servers, which will serve *n* potential locations, again with at most 5 different types of clients. As one can observe, CPLEX 12.6, which is one of the most sophisticated deterministic MIP solvers, and has been under development for the past 25 years, struggles to solve instances with a large number of scenarios. Interestingly, when CPLEX 12.6 does find an optimal solution, it does so within fewer CPU secs. (s) as shown for the examples in Table 2. However, as the number of scenarios increases, the deterministic solver is unable to scale up, and failed on 6 out of 9 instances shown in Table 2. These results demonstrate the practical value behind the approaches produced as a result of this project.

Table 2. Computations Comparing ABC with CPLEX 12.6 with SMIP Instances

Instance	**$\lvert\mathcal{N}\rvert$ Size**	**Optimal Value**	**Variables**	**Constraints**	**ABC (s)**	**CPLEX 12.6 (s)**
(2x5),(15x5)	50	-457.24	1602	1601	4.26	> 3600
(2x5),(15x5)	100	-420.35	3202	3201	1.64	0.78
(2x5), (15x5)	500	-399.47	16002	16001	51.28	> 3600
(3x5),(15x5)	50	-436.42	2403	1651	191.92	9.73
(3x5),(15x5)	100	-371.24	4803	3301	19.56	> 3600
(3x5),(15x5)	500	-426.54	24003	16501	1069.92	> 3600
(4x5), (10x5)	50	-308.69	2204	1201	3.6	0.98
(4x5),(10x5)	100	-350.08	4404	2401	261.89	> 3600
(4x5),(10x5)	500	-311.3	22004	12001	745.62	> 3600

Large-scale Stochastic Linear Programming

As mentioned in the previous section, both 2-stage and multi-stage SMIP models can be embedded within the setting of algorithms which solve 2-stage and multi-stage SLP problems. As a result, we are also interested in solving large-scale SLP models which can end up as a platform for Stochastic-Dynamic MIPs. This section is devoted to two-stage and multi-stage SLPs.

Statistical optimality for two stage algorithms

The two stage SP model with recourse may be stated as the following optimization problem.

$$f^* = \underset{u_0 \in U}{\text{Min}}\, f(u_0) := d(u_0) + E[h(u_0, \tilde{\omega})]. \tag{12a}$$

The function h is referred to as the recourse function and is defined as

$$h(u_0, \omega) := \text{Min}\{d_1^\top u_1 \mid Du_1 = b(\omega) - C(\omega)u_0, u_1 \geq 0\}, \tag{12b}$$

where ω denotes an outcome of the random variable $\tilde{\omega}$ defined on an appropriate probability space. When the space of outcomes is very large (or we have continuous random variables), sampling-based algorithms provide a tractable approach to two stage SP. For such algorithms, optimality can only be verified in a statistical sense. There are two classes of statistical optimality tests: a) In-Sample stopping rules (e.g., Higle and Sen

1999) and b) Out-of-Sample stopping rules (e.g., Higle and Sen 1996, Mak et 1999 and Bayraksan and Morton 2011). The former apply to algorithms which are classified as "internal-sampling" algorithms, while the latter apply to both "internal" as well as "external" sampling algorithms. The current project has designed a new concept called the Compromise Solution for out-of-sample stopping (Sen and Liu 2015).

.

In order to give the reader a concrete sense of progress, we present recent computational results with the Sonet Switched Network (SSN) cited in the Defense Science Board Report (2011) for the potential power of SP. This is a two-stage SP, and in 1994, we had reported major improvements in network robustness by using solutions which were obtained on a Sun Sparc II machine after about an hour of computing using SD (Sen et al 1994). Because the SSN model involves a sample space of O(10^{70}) possible scenarios, we had no hope of ascertaining whether the design we obtained was close to optimum. From a deterministic point of view, this is true even today. Even if one had access to exascale computing (O(10^{18}) flops), there is no hope of predicting the expected number of lost packets deterministically. If one accepts statistical assessments of optimality, then there is hope.

SAA Computations for SSN

The methods discussed below (i.e. SAA and SD) use the above rules in a manner suited within the context of their specific algorithmic procedures. SAA is an acronym for Sample Average Approximation (Kleywegt et al 2002), and the strategy here is to replace the expectation $E[h(u,\tilde{\omega})]$ in the objective function of (12a) by a sample average approximation created by sampling a *batch* of N outcomes $\{\omega^t\}_{t=1}^N$. Thus, the expectation is replaced by

$$\widehat{H}(u) = \frac{1}{N}\sum_{t=1}^{N} h(u,\omega^t). \tag{13}$$

In the machine learning literature, this function is referred to as "empirical risk". Using concentration inequalities and large deviations theory, suggests a sample size which ensures an ϵ-optimum with high probability. Nevertheless, as the authors of SAA suggest that such sample sizes are much too large to be computationally practical (Kleywegt et al 2002). Instead a sequential application of SAA is recommended (Linderoth et al 2006). For each sample size N, the sequential approach creates $m = 1,,\dots,M$ replications denoted $\{\widehat{H}^m\}$ of the form (13). After optimizing these M replications, one calculates a lower bound confidence interval, and an upper bound confidence interval as suggested in Mak et al (1999). If the pessimistic gap[2] is large, then

[2] The pessimistic gap is the difference between the upper end-point of the upper bound confidence interval and the lower end-point of the lower bound confidence interval.

the sample size N is increased; otherwise, the method stops and declares the solution to be close to optimum.

Table 3: Results for SSN using SAA and LHS Sampling on a Computing Grid

Sample Size (N)	Lower Bound (L) Confidence Interval	Upper Bound (U) Confidence Interval	Pessimistic Gap Max $(U - L)$
50	10.10 (+/-0.81)	11.380 (+/-0.023)	2.113
100	8.90(+/-0.36)	10.542(+/-0.021)	2.023
500	9.87(+/-0.22)	10.069(+/-0.026)	0.445
1000	9.83(+/-0.29)	9.996(+/-0.025)	0.445
5000[3]	9.84(+/-0.10)	9.913(+/-0.022)	0.195

The results for SSN in Linderoth et al (2006) are reproduced in Table 3. For $N = 5000$ the pessimistic gap is less than 0.5%. Thus with very high probability the optimal value of SSN is in the suggested confidence intervals. The results in Table 2 were obtained using a computational grid with several hundred desktop PCs, although no more than one hundred machines were in operation at any given time. Even so, each SAA instance of SSN in the final row of Table 3 (with $N = 5000$) required about 30-45 (wall clock) minutes. To the best of our knowledge, the authors use $M = 6$ for SSN.

Two-stage SD computations for SSN

The statistical tests used in SD include both in-sample as well as out-of samples rules. The in-sample rule verifies whether a large proportion of resampled instances provide duality gap estimates below a given tolerance (loose, nominal, or tight). In addition, for out-of-sample tests SD chooses the number of replications $m = 1, \dots, M$. Because the specifics of the algorithms (SAA and SD) are different, the tables present slightly different information, although the bounds and pessimistic gap appear in both.

[3] Runs with $N = 5000$ took 30-45 mins. of wall clock time per replication with 100 PCs on a computing grid

Table 4: SD with Common Random Numbers (on a MacBook Air with CPLEX12.4)

Stopping Tolerance (Relative)	Sample Size (Standard Dev)	Lower Bound (L) Confidence Interval	Upper Bound (U) Confidence Interval	Pessimistic Gap Max ($U - L$)	CPU sec. per replication
Loose (0.01)	1023.33 (167.62)	9.366 (+/-0.244)	9.953 (+/-0.050)	0.881	32.73(6.97)
Nominal (0.001)	2353.43 (343.33)	9.764 (+/-0.120)	9.928(+/- 0.050)	0.334	109.96(26.31)
Tight (0.0001)	3137.50 (605.17)	9.876 (+/-0.107)	9.925 (+/-0.050)	0.206	189.79(74.57)

The main observations from the above comparison (Tables 3 and 4) are provided below:

1. (*Choosing decisions)*. As reported by Linderoth et al (2006), the 6 solutions for SSN found using SAA were quite disparate even though these experiments were done using Latin-Hypercube Sampling, a variance reduction tool commonly used for discrete-event simulation. This was attributed to the inherent ill-conditioning of SSN. Nevertheless, solution variability (due to sampling) does appear in many practical instances, and decision-makers need guidance on which choice to adopt.
2. *(Computing Platforms v Algorithms)*. The MacBook Air has a processor speed of about 1.8 GHz, and the average Pentium IV processor from 2004/2005 ran at about 2 – 2.2 GHz. Based on the similarity in processing speeds, and following the arguments in a PCAST report of (Holdren et al 2011), Sen and Liu (2015) argue that the speed-up reported in Table 4 is on par with Moore's law, as long as the grid computing platform spent less than 87.2% of its time on either idling or communicating. Similar conclusions have been drawn while comparing Batch and Online Stochastic Learning (Bottou 2003), where "Batch" refers to empirical risk minimization as in SAA, and "Online" is similar to SD. Similarly computations reported by Nemirovski et al (2009) also support this conclusion.

Out-of-sample Test: Compromise Decision.

is that the lines on the left represent estimated objective function values at the solution to a *Compromise Problem* (14) which uses the replications to recommend a nearly optimal decision.

$$\underset{u\in U}{\text{Min}}\left\{d(u)+\frac{1}{M}\sum_{m}h^m(u)+\frac{\bar{\sigma}}{2}\|u-u^m\ \|^2)\right\}. \tag{14}$$

Consider models which have the structure shown in (12), and assume that i) (U,Ω) are compact sets, U is a convex set, and $d(u)$ is a convex function, ii) the second stage satisfies the relatively complete recourse property, and iii) the random process affects either activities (columns) of the first or resources (right hand side) of the second stage. Let an interior sampling algorithm (such as SD) run M times with each seed creating i.i.d. sequences of outcomes $\{\omega_k^m\}$, for $m=1,\dots,M$, where M is large enough and N denotes the smallest sample size discovered by SD among all M runs.. Suppose each run produces a solution u^m as well as an approximate value function $h^m(u)$ such that

$$\begin{aligned}&u^m\in \text{argmin}\{f^m(u) \coloneqq\ d(u)+h^m(u) : u\in U\}\\ &\text{and } |E[h(u^m,\tilde{\omega})]-h^m(u^m)|=O_p\left(\frac{1}{\sqrt{N}}\right).\end{aligned} \tag{15}$$

where the right-hand side[4] in (15) is due to the Central Limit Theorem. Let $\bar{F}_M(u)$ denote the grand mean value function approximation $\frac{1}{M}\sum_m f^m(u)$ used in (11). Suppose that $u^c\in$ argmin$\{\ \bar{F}_M(u)|u\in U\ \}$. We say that the statistical optimality principle is satisfied if

$$\bar{F}_M(u^c)-\eta\le f^*\le\ \bar{F}_M(u^c)+\eta, \tag{16}$$

where f^* denotes the optimal value in (12), and $\eta=O_p\left(\frac{1}{\sqrt{NM}}\right)$. This is the principle that leads us to the kind of computational performance that is reported in Table 4. Further details for SLP problems are provided in the Sen and Liu (2015).

Multi-stage Stochastic Linear Programming

This part of the project deals with successive approximation schemes within the context of multi-stage stochastic programming, and should be interpreted as a bridge between stochastic and dynamic programming. The method presented in Sen and Zhou (2014) is the only MSLP algorithm which discovers the stochastic process "on the fly", and in this sense it is the multis-stage analog of two-stage Stochastic Decomposition (SD). Among sampling-based approaches for MSLP, the most popular ones trace back to the work on stochastic dual dynamic programming (SDDP) of Pereira and Pinto (1991). Algorithmic

[4] Let $O_p(1)$ denote stochastic boundedness. Any sequence of random variables $\{R_n\}$ is stochastically bounded if for every $\epsilon>0$ there exists $C>0$ such that $\Pr[|R_n|>C]<\epsilon$.

enhancements for this class of methods have appeared in Infanger and Morton (1996) and more recently as abridged Benders' decomposition in Donahue and Birge (2006). Asymptotic results by Philpott and Guan (2008) highlight the need for certain analytical safeguards for asymptotic convergence.

One of our main goals was to present an approach that generalizes the two-stage SD algorithm to the multi-stage setting. More specifically, the MSD algorithm provides a dynamic extension of regularized multi-stage SLP. As with the two stage case, the multistage extension provides an asymptotically optimal solution (wp1), even without requiring the scenario probabilities as input into the algorithm. While the approach for sampled approximations share some of the recursive features of ADP, some approximation tools (e.g. duality) and asymptotic analysis draw upon concepts that are central to SP (e.g. epigraphical nesting). In this sense, MSD provides a bridge between SP and ADP.

In order to appreciate the above goals, it is important to place the two-stage SD algorithm in the context of other sample-based methods for two-stage SP. Among the earliest approaches for sample-based algorithms we have the Stochastic Approximation method (SA, Robbins and Monro 1951). The work of Ermoliev and Shor (1968) appears to be the first application of SA to SP, and as demonstrated by the work of Nemirovski et al (2009), there is continuing interest in using SA methods for SP problems. This genre of methods creates a sequence of sampled subgradients which are used with certain step size rules to ensure convergence (wp1) of the random sequence of solutions generated by SA. Thus one might consider an SA algorithm to be a sampled version of the deterministic subgradient method, just as two-stage SD is a sampled version of Benders' decomposition. The advantage of these sample-based schemes is that they are able to work with statistical estimates, and for large scale models they have the advantage of not having to calculate a subgradient using every potential outcome of the random variable. The main point here is that SA and SD are both randomized versions of their deterministic counterparts.

In order to extend algorithms that use Benders' cuts (e.g. Nested Benders, SDDP, etc.) to cases where a simulation of the randomness is available, but not exact distributional information, a variety of issues were addressed.

- (*Sampling and Convergence*) In current sampling-based methods motivated by Benders' decomposition the essential role of sampling is to reduce the number of nodes traversed by the algorithm in any iteration. However, the approximations themselves are formed by using deterministic Benders' cuts. This requires a probabilistic description of uncertainty to be characterized before an SP algorithm can be used. Note that there are many emerging applications in which stagewise independence may not be justified. For instance, in Philpott and Matos (2012) the authors try to include a Markov Chain within an SDDP framework (see also Higle and Kempf 2011). However, the MSD framework provides a more natural setting for such applications because the setup is based on a dynamic systems framework which admits Markov chains seamlessly. The question then arises whether asymptotically convergent algorithms can be designed for such applications. For MSLP, this open question will be resolved, in a mathematical sense, by the MSD algorithm. This paper will establish asymptotic results, although computational realizations will require further attention to data structures, stopping rules, and other advances.

- (*Uniqueness*) Under what circumstances will the algorithm produce a unique solution?
- (*Specializations for stochastic structure*) Are there specializations (e.g. stagewise independence or autoregressive structures) of the general MSLP structure that will allow a streamlined algorithm compared with the case of correlated random variables?
- (*Specializations for constraint structure*) Suppose that some of the matrices have specialized structures (e.g. networks). Are there ways to specialize the MSD algorithm in a way that can exploit such special structures? Finally, does the MSD algorithm reduce to two-stage SD when applied to two-stage SLP?

By developing the MSD algorithm, our project has addressed all of the questions posed above. The MSD algorithm is a dynamic version of Higle and Sen (1994) which is a regularized approach to the original version of SD. In the SP literature, such regularization was already an effective tool within the deterministic approach proposed in Ruszczyński (1986), as well as non-smooth optimization methods, as in Kiwiel (1990). Finally, these ideas were also studied more recently in Oliviera et al (2012). However, there has been very little attention paid to multistage extensions using regularized algorithms (e.g. see comments by Shapiro 2011). The MSD approach is the first regularized algorithm for MSLP. Details of this approach are available in Sen and Zhou (2014).

1. Dissemination

Stochastic Programming Software (Freely Distributed)

- Available at www.neos-server.org/neos/solvers/slp:sd/SMPS.html
- Available through Github at: https://github.com/imliuyifan/sd
- Have created interfaces for PySP (Open Source Software for Stochastic Programming)

Presentations

(Featured Presentation) S. Sen, "Stochastic Mixed-Integer Programming," INFORMS Optimization Conference, Miami, FL, February 2012.

S. Sen, "Multi-stage Stochastic Decomposition," Symposium of the Mathematical Optimization Society, Berlin, August 2012.

S. Sen and Y. Liu, "Optimization Statistics for Stochastic Linear Programming" INFORMS Computing Society Meeting, Santa Fe, NM, January 2013.

S. Sen and Y. Qi, "The ABC algorithm for two-stage Stochastic Mixed-Integer Programming," International Conference on Stochastic Programming, Bergamo, Italy, 2013 (also presented at the INFORMS Annual Conference in Minneapolis)

S. Sen "Specializations of Multi-stage Stochastic Decomposition," Applied Probability Society Conference, Costa Rica, July 2013

S. Sen (Plenary/Advance Tutorial) "Conceptual v. Computational Stochastic Programming: Perspectives as SP turns 60" INFORMS Computing Society Conference, Richmond, VA, January 2015.

Lectures at Universities

90 min. Lecture at Winter School at Tignes, France (2013)
Lecture at Ohio State University (2013)
Lecture at UC – Davis (2014)
Offered ISE 638 to Ph.D. students at USC (9 students enrolled)

Students Supported

Y. Liu, S. Atakan and A. Ozkan. Their dissertations are on-going at this point.

Publications

[1] B. Chen, S. Küçükyavuz, and S. Sen, “A Computational Study of the Cutting Plane Tree Algorithm for General Mixed-Integer Linear Programs,” *Operations Research Letters,* Vol. 40, pp. 15-19, 2012.

[2] D. Gade, S. Küçükyavuz, and S. Sen, “Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs,” *Mathematical Programming*, vol. 144, pp. 39-64, 2012.

[3] K. Huang, S. Sen, and F. Szidarovszky, “Connections among Decision Field Theory models of Cognition,” *Journal of Mathematical Psychology*, Vol. 56, pp. 287 – 296, 2012.

[4] Y. Qi and S. Sen, “Ancestral Benders' Cuts and Multi-term Disjunctions for Mixed-Integer Recourse Decisions in Stochastic Programming,” revised and resubmitted to *Mathematical Programming.*

[5] S. Sen, “Stochastic Programming," *Encyclopedia of Operations Research and Management Science* (S. Gass and M. Fu, eds.), Springer, pp. 1486 – 1497, 2013.

[6] S. Sen and Y. Liu, “Mitigating Uncertainty via Compromise Decisions in Stochastic Linear Programming” revised and resubmitted to *Operations Research*

[7] S. Sen and Z. Zhou, “Multi-stage Stochastic Decomposition” *SIAM Journal on Optimization*, Vol. 24, pp. 127-153, 2014.

[8] H. Gangammanavar, S. Sen and V. Zavala, “Simulation and Stochastic Optimization for Sub-hourly Economic Dispatch of Wind Energy” accepted for publication in *IEEE Transactions on Power Systems*.

References

[1] Balas, E. (1979). Disjunctive programming. Annals of Discrete Mathematics, 5:3-51.

[2] Birge, J. R. 1985, “Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs,” *Operations Research*, 33:989–1007.

[3] Bottou, L.(2003), “Statistical Learning,” in Advanced Lectures on Machine Learning (Bosquet, Luxburg, Ratsch eds.), pp. 146 – 168.

[4] Caroe, C. and J. Tind, (1998), “ L-shaped decomposition of two-stage stochastic programs with integer recourse.,” *Mathematical Programming*, 83:451-464.

[5] B. Chen, S. Küçükyavuz, and S. Sen (2012), “A Computational Study of the Cutting Plane Tree Algorithm for General Mixed-Integer Linear Programs,” *Operations Research Letters*, Vol. 40, pp. 15-19.

[6] Defense Science Board Report (2011). "Enhancing Adaptability of U.S. Military Forces," p.31

[7] Donohue, C.J. and J. R. Birge (2006) “The Abridged Nested Decomposition Method for Multistage Stochastic Linear Programs with Relatively Complete Recourse,” *Algorithmic Operations Research*, 1:20-30.

[8] Ermoliev Y.M. and N.Z. Shor (1968), “Random search method for a two-stage stochastic programming problem,” Kibernetika (translated into English as Cybernetics), 1:85-88.

[9] Higle, J.L. and K. Kempf (2011), “Production Planning under Supply and Demand. Uncertainty: A Stochastic Programming Approach,” *Stochastic Programming: The State of the Art – In Honor of George B. Dantzig* (G. Infanger, ed.), pp. 297-316, Springer.

[10] Higle, J.L. and S. Sen (1991), “Stochastic Decomposition: An Algorithm for Two Stage Linear Programs with Recourse,” *Mathematics of Operations Research*, vol. 16:650-669.

[11] Higle, J.L. and S. Sen (1994), “Finite master programs in regularized stochastic decomposition," *Mathematical Programming*, 67:143-168.

[12] Higle, J. L. and S. Sen (1996), *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*, Kluwer Academic Publishers.

[13] Holdren, J.P., Lander, E. and Varmus, H. (2010). “Report to the President and Congress – Designing a Digital Future: Federally Funded Research and Development in Networking and Information Technology.” President’s Council of Advisors on Science and Technology.

[14] Infanger, G. and D.P. Morton (1996),“Cut sharing for multistage stochastic linear programs with interstage dependency,” *Mathematical Programming* 75: 241-256.

[15] Kiwiel, K.C. (1990), “Proximity Control in Bundle Methods for Convex Non-differentiable Minimization,” *Mathematical Programming*, 46, 105-122.

[16] Kleywegt, A, Shapiro, A, and Homem-de-Mello, T, (2002) SIAM Journal on Optimization archive, vol. 12Journal SIAM Journal on Optimization archive, , 2002

[17] Linderoth, J., A. Shapiro, S. Wright. 2006. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research* 142:215-241.

[18] Nemirovski, A., A. Juditsky, G. Lan, G. and A. Shapiro (2009), "Robust Stochastic Approximation Approach to Stochastic Programming," *SIAM Journal of Optimization*, 19, 1574-1609.

[19] Oliviera, W., C. Sagaastizabal and S. Scheimberg (2011), "Inexact Bundle Methods for Two-Stage Stochastic Programming," *SIAM Journal of Optimization*, 21: 517–544.

[20] Pereira, M.V. and L. M. Pinto (1991), "Multi-stage Stochastic Optimization Applied to Energy Planning," *Mathematical Programming*, 52:359-375.

[21] Philpott, A.B. and Z. Guan (2008), "On the Convergence of Stochastic Dual Dynamic Programming and Related Methods," *Operations Research Letters*, 36: 450-455.

[22] Philpott, A.B. and V. L. de Matos (2012), "Dynamic Sampling Algorithms for Multi-stage Stochastic Programs with Risk Aversion. *European Journal of Operational Research* 218(2): 470-483.

[23] Powell, W. B. (2007), *Approximate Dynamic Programming: Solving the curses of dimensionality*, Wiley Series in Probability and Statistics.

[24] Y. Qi and S. Sen (2015), "Ancestral Benders' Cuts and Multi-term Disjunctions for Mixed-Integer Recourse Decisions in Stochastic Programming," revised and resubmitted to *Mathematical Programming* .

[25] Robbins, H. and S. Monro (1951), "A Stochastic Approximation Method," *Annals of Mathematical Statistics*, 22:400-407.

[26] Rockafellar, R.T. and R. J-B. Wets (1991), "Scenarios and Policy Aggregation in Optimization under Uncertainty," *Mathematics of Operations Research*, 16(1):119–147.

[27] Ruszczyński, A. (1986), "A Regularized Decomposition Method for Minimizing a Sum of Polyhedral Functions," *Mathematical Programming*, 35:309-333.

[28] Sen, S., R.D. Doverspike and S.C S. Cosares (1994), "Network Planning with Random Demand, *Telecommunication Systems*, 3, pp. 11-30.

[29] Sen, S. and J.L. Higle (2005), "The C3 Theorem and a D2 Algorithm for Large Scale Stochastic Integer Programming," *Mathematical Programming*, 104, pp. 1-20.

[30] S. Sen and Y. Liu (2015), "Mitigating Uncertainty via Compromise Decisions in Stochastic Linear Programming" revised and resubmitted to *Operations Research.*

[31] Sen, S. and H.D. Sherali (2006), "Decomposition with Branch-and- Cut Approaches for Two Stage Stochastic Integer Programming," *Mathematical Programming*, 106, pp. 203-223, 2006.

[32] S. Sen and Z. Zhou (2014), "Multi-stage Stochastic Decomposition" *SIAM Journal on Optimization* , Vol. 24, pp. 127-153, 2014.

[33] Shapiro, A. (2011), “Analysis of Stochastic Dual Dynamic Programming Method,” *European J. of Operations Research*, vol. 209: 63-72.

AFOSR Deliverables Submission Survey

Response ID:4521 Data

1.

1. Report Type

Final Report

Primary Contact E-mail
Contact email if there is a problem with the report.

s.sen@usc.edu

Primary Contact Phone Number
Contact phone number if there is a problem with the report

213-740-6810

Organization / Institution name

University of Southern California

Grant/Contract Title
The full title of the funded effort.

Stochastic Dynamic Mixed-Integer Programming (SD-MIP)

Grant/Contract Number
AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".

FA9550-13-1-0015

Principal Investigator Name
The full name of the principal investigator on the grant or contract.

Suvrajeet Sen

Program Manager
The AFOSR Program Manager currently assigned to the award

Fariba Fahroo

Reporting Period Start Date

02/15/2012

Reporting Period End Date

12/15/2014

Abstract

Mixed-Integer Programming has traditionally been restricted to deterministic models. Recent research has opened the door to stochastic optimization models, which are typically dynamic in nature. This project lays the foundation for stochastic dynamic mixed-integer and linear programming (SD-MIP). This project has produced several new ideas in connection with a) convexification of two-stage mixed-integer sets and b) multi-stage (including two-stage) stochastic linear programming. Together a) and b) provide the foundations for SD-MIP problems. From new concepts and algorithms to applications and software, this project has made significant breakthroughs in all aspects.

This report provides a synopsis of both theoretical and computational results. As a preview, we mention that currently available deterministic MIP solvers, as powerful as they are known to be, are unable to solve SD-MIP models of modest size

within an hour of computing. In contrast, our decomposition approach provides provably optimal solutions within the hour time-limit.

Distribution Statement

This is block 12 on the SF298 form.

Distribution A - Approved for Public Release

Explanation for Distribution Statement

If this is not approved for public release, please provide a short explanation. E.g., contains proprietary information.

SF298 Form

Please attach your SF298 form. A blank SF298 can be found here. Please do not password protect or secure the PDF The maximum file size for an SF298 is 50MB.

SF298-2015.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.

FinalReport4USC-April 2015.pdf

Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.

Archival Publications (published) during reporting period:

[1] B. Chen, S. Küçükyavuz, and S. Sen, "A Computational Study of the Cutting Plane Tree Algorithm for General Mixed-Integer Linear Programs," Operations Research Letters, Vol. 40, pp. 15-19, 2012.

[2] D. Gade, S. Küçükyavuz, and S. Sen, "Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs," Mathematical Programming, vol. 144, pp. 39-64, 2012.

[3] K. Huang, S. Sen, and F. Szidarovszky, "Connections among Decision Field Theory models of Cognition," Journal of Mathematical Psychology, Vol. 56, pp. 287 – 296, 2012.

[4] Y. Qi and S. Sen, "Ancestral Benders' Cuts and Multi-term Disjunctions for Mixed-Integer Recourse Decisions in Stochastic Programming," revised and resubmitted to Mathematical Programming.

[5] S. Sen, "Stochastic Programming," Encyclopedia of Operations Research and Management Science (S. Gass and M. Fu, eds.), Springer, pp. 1486 – 1497, 2013.

[6] S. Sen and Y. Liu, "Mitigating Uncertainty via Compromise Decisions in Stochastic Linear Programming" revised and resubmitted to Operations Research

[7] S. Sen and Z. Zhou, "Multi-stage Stochastic Decomposition" SIAM Journal on Optimization, Vol. 24, pp. 127-153, 2014.

[8] H. Gangammanavar, S. Sen and V. Zavala, "Simulation and Stochastic Optimization for Sub-hourly Economic Dispatch of Wind Energy" accepted for publication in IEEE Transactions on Power Systems.

Changes in research objectives (if any):

None

Change in AFOSR Program Manager, if any:

Begun under Donald Hearn. New Program Officer: Fariba Fahroo

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

Apr 28, 2015 14:54:00 Success: Email Sent to: s.sen@usc.edu