AFRL-AFOSR-VA-TR-2019-0219

Statistical Optimality, Algorithms and Resilience in Time-staged Stochastic Systems

Suvrajeet Sen
UNIVERSITY OF SOUTHERN CALIFORNIA

07/11/2019
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

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
6/30/2019	Final Report	2015-2019

4. TITLE AND SUBTITLE
Statistical Optimality, Algorithms and Resilience

5a. CONTRACT NUMBER
FA9550-15-1-0267

5b. GRANT NUMBER

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, Los Angeles, CA 90089

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
Circulate as necessary

13. SUPPLEMENTARY NOTES
None

14. ABSTRACT

This project focused on mathematical decision models which capture data uncertainty. In such situations, it is almost impossible to make choices which are deterministically optimum. However, by using statistical approaches, one can make decisions which are good up to a statistically verifiable guarantee. Algorithms which provide such decisions are said to achieve some level of statistical optimality. However, because there are no absolute certainties in such a setting, it is also important that the decisions are resilient to non-optimality. In other words, the decisions should be such that the downside of facing a bad scenario is not devastating to the decision-maker. Such decisions will be referred to as "resilient decisions". Our approaches were devoted to studying continuous optimization models which provide computational tools for resilient decision-making in two-stage (e.g., today and tomorrow) as well as multi-stage (sequential) decision models. Our approaches have been tested computationally, and the computational results speak to the effectiveness of these approaches. In all cases we have applied the new methods to decision problems arising in real-world settings such as network planning and system operations (e.g., power).

15. SUBJECT TERMS
Optimization Algorithms and Learning Algorithms

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

Reset

Statistical Optimality, Algorithms and Resilience

AFOSR Grant FA9550-15-1-0267

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

June 2018

Abstract

This project focused on mathematical decision models which capture data uncertainty. In such situations, it is almost impossible to make choices which are deterministically optimum. However, by using statistical approaches, one can make decisions which are good up to a statistically verifiable guarantee. Algorithms which provide such decisions are said to achieve some level of statistical optimality. However, because there are no absolute certainties in such a setting, it is also important that the decisions are resilient to non-optimality. In other words, the decisions should be such that the downside of facing a bad scenario is not devastating to the decision-maker. Such decisions will be referred to as "resilient decisions". Our approaches were devoted to studying continuous optimization models which provide computational tools for resilient decision-making in two-stage (e.g., today and tomorrow) as well as multi-stage (sequential) decision models. Our approaches have been tested computationally, and the computational results speak to the effectiveness of these approaches. In all cases we have applied the new methods to decision problems arising in real-world settings such as network planning and system operations (e.g., power).

Overview

This report organized by the publications which were produced as a result of AFOSR funding. The first group of publications presented here are dedicated to two-stage models, and following that we provide our work on multi-stage problems. Some applications-oriented papers are also listed in this report[1].

Two Stage Stochastic Linear Programming (2-SLP)

This section is sub-divided into three subsections, all of which are related 2-SLP

Variance Reduction

[1] Papers/Reports which are cited within the text will not appear in the Reference list. Other papers which are not cited within the report are listed in the References.

S. Sen and Y. Liu, "Mitigating Uncertainty via Compromise Decisions in Stochastic Linear Programming: Variance Reduction" *Operations Research,* 64(6):1422-1437, August 2016.

Stochastic Programming (SP) has long been considered a well-justified yet computationally challenging paradigm for practical applications. Computational studies in the literature often involve approximating a large number of scenarios by using a small number of scenarios to be processed via deterministic solvers, or running Sample Average Approximation on some genre of high performance machines so that statistically acceptable bounds can be obtained. In this paper we show that for a class of stochastic linear programming problems, an alternative approach known as Stochastic Decomposition (SD) can provide solutions of similar quality in far less computational time using ordinary desktop or laptop machines of today. In addition to these compelling computational results, we provide a stronger convergence result for SD, and introduce a new solution concept that we call the compromise decision. This new concept is attractive for algorithms that call for multiple replications in sampling-based convex optimization algorithms. For such replicated optimization, we show that the difference between an average solution and a compromise decision provides a natural stopping rule. We discuss three stopping criteria that enhance the reliability of the compromise decision, reducing bias and variance associated with the result. Finally our computational results cover a variety of instances from the literature, including a detailed study of SONET Switched Network (SSN), a network planning instance known to be more formidable test instances in the literature. This instance was also cited by the Defense Science Board as one of the challenges in DoD research (Defense Science Board Report, 2011).

Formally speaking, SSN is a two-stage stochastic linear programming model. The basic "operations"-issue in the SSN model is to recommend link sizes of a given network so that the network will experience the least number of "lost calls" (in expectation), while operating under a given budget constraint. In the SP literature, such models are often classified as "here-and-now" because the link capacities must be decided before actual demands are known. Models of this type, which are based on introducing randomness to linear programming models, must contend with multidimensional random vectors, which, in the SSN model represent point-to-point demand uncertainty. In this example, there are 86 point-to-point pairs, which, by standards of LP models, is modest. As is common today, these demands are available through forecasting systems, and errors in forecasts may be treated as independent random variables. For the model presented in Sen et al. (1994), each marginal error random variable was deemed to be sufficiently approximated by a discretization using about 5–9 outcomes per demand pair. Clearly the total number of scenarios involves an astronomical number of parametric LPs (approximately of magnitude 10^{71}). Even if one had access to an exascale (10^{18} flops) computing platform, it would be pointless to seek a solution whose optimality could be verified in a deterministic sense. It is therefore pragmatic to seek approximate solutions that are near-optimum in a statistical sense.

The most widely cited experiment for the SSN problem is the 2006 study of Linderoth et al which reported lower and upper bounds on the optimal solution of the instance to

belong to the interval [L,U] = [9.84, 9.914]. This was accomplished using grid computing with hundreds of PCs[2] running for about 30-45 minutes per replication. that these hundreds of desktop PCs and the MacBook Air used in [99] had very similar processing speeds (about 2 GHz clock speed). Thanks to research support from this AFOSR grant, our team was able to produce the following results using only a conventional laptop[3].

Relative Tolerance	Sample Size (Std. Dev.)	Lower Bound (L) Conf. Int.	Upper Bound (U) Conf. Int.	U - L	CPU secs (Std. Dev)
Tight (0.0001)	3137 (605.17)	9.876 (+/- 0.107)	9.925 (+/- 0.05)	0.049	189.79 (74.57)

Table 1: Results using Stochastic Decomposition (SD) for SSN on a MacBook Air

The quantity U is an upper bound on the minimum (in expectation) and L is a lower bound on the minimum. The difference shown in the column U – L reports that the optimum solution is 0.5% (on average). However, the processing time reported by our experiments is only 3 minutes per replication. Because the processor speeds of the two studies were similar, one concludes that the "speed-up" can be attributed to algorithmic efficiency. From a detailed discussion reported on page 1427 of our *Operations Research* (2016) paper, we were able to conclude that our approach, which produced Table 1, provides an algorithmic speed-up similar to the renowned Moore's Law for computer chips (i.e., doubling the speed every 1.5 years)!

Convergence Rate

J.Liu and S. Sen, "Asymptotic Results on Two-stage Stochastic Quadratic Programming" submitted last minor revision to *SIAM J. on Optimization*

While the computational results reported in the previous section was remarkable, and the asymptotic convergence of SD has been proved over 25 years ago (Higle and Sen 1994) no theoretically proven convergence *rates* were available for this method. In this sense, the mathematical properties of stochastic bundle methods were not fully understood. Thanks to the current AFOSR funded project, this issue has been resolved: we have obtained a *sublinear convergence rate* for the SD algorithm, under some assumptions. Table 2 compares the convergence rate that we have obtained for SD with those available in the literature for other algorithms which might be used for large scale Linear/Quadratic Programming problems. The notation in Table 2 uses x^N to denote the solution obtained when an algorithm stops after N iterations, and the notation $f(x^N)$ denote the "true value" of the decision x^N, while f^* and x^* denote the "true optimal value" and the "true optimal solution" of the SP model. One should recognize that since most optimization

[2]An average Pentium IV PC in 2004/2005 had clock speed of about 2 GHz, which is similar to the laptop
[3]MacBook Air using (Core i5 processor running at 1.8 GHz)

algorithms are used for the purposes of decision-making, what matters is the quality of x^N, and although $f(x^N)$ is an important deliverable (for optimization), many Air Force and DoD applications seek decisions or actions, and therefore, having good quality solutions is or primary importance.

It is important to note that SAA is more-or-less a mathematical tool, rather than a numerical algorithm, because the latter is an essential choice in implementing SAA. Hence the convergence rate listed for SAA does not tell the whole story. Of the remaining algorithms which can be applied for Stochastic Programing with Linear or Quadratic Programs defining their value functions, we show that SD provides a faster rate of convergence than of the algorithms which can be used. This theoretical result actually supports the computational results which were observed on page 2 of this report.

Method	**Structural Assumptions**	**Convergence Type**	**Convergence Rate**
SAA for Nonconvex Problems	Non-convex	$P\{f(x^N) - f^* > \epsilon\}$	$O(C_\epsilon e^{-\beta_\epsilon N})$
SAA for Convex Problems	convex	$P(x^N \notin S^*)$	$O(C_0 e^{-\beta_0 N})$
Robust Stochastic Approximation (RSA)	convex	$E\|f(x^N) - f^*\|$	$O(N^{-\frac{1}{2}})$
Stochastic Approximation (SA)	Strongly convex	$E\|\|x^N - x^*\|\|$	$O(N^{-\frac{1}{2}})$
Stochastic Approximation (SA)	Lipschitz Gradient	$E\|f(x^N) - f^*\|$	$O(N^{-1})$
Stochastic Decomposition (SD) (**This Grant**)	Positive Definite Quadratic Forms	$E\|\|x^N - x^*\|\|$	$O(N^{-1})$

Table 2. Comparison of Convergence Rates for Alternative Algorithms for SP.

Random Cost Models

H. Gangammanavar, Y. Liu and S. Sen, "Stochastic Decomposition for Two-Stage Stochastic Linear Programs with Random Cost Coefficients," submitted last minor revision to *INFORMS Journal on Computing*

The key to the SD algorithm mentioned above is the incremental sampling approach which is designed to discover an appropriate sample size for a given SP instance, thus precluding the need for either scenario reduction or arbitrary sample sizes to create

sample average approximations (SAA). As discussed above, SD provides solutions of similar quality in far less computational time using ordinarily available computational resources. However, previous versions of SD were not applicable to problems with randomness in second-stage cost coefficients. In this paper, we have extended its capabilities by relaxing this assumption on cost coefficients in the second-stage. In addition to the algorithmic enhancements necessary to achieve this, our paper also presents the details of implementing these extensions which preserve the computational edge of SD. The computational results reported in the paper illustrate the continued success of the SD methodology. To help the reader get a sense of the computational advantages of SD we compare our computational results with those obtained from the regularized Benders decomposition method applied an extension of the SSN instance. Both methods were run using 30 replications so that we could get a sense of the variability in performance. These results are summarized in Table 2.

Algorithm	Sample Size (std. dev.)	Lower Bound (95% CI)	Upper Bound (95% CI)	Pessimistic Gap	Avg.Time (s) (std.dev.)
Regularized Benders	50	4.46 ($\pm$0.36)	13.63 ($\pm$0.36)	10.32 (231.24%)	12.54 (4.41)
	500	9.06 ($\pm$0.39)	10.43 ($\pm$07)	1.84 (20.30%)	470.34 (208.88)
	5000	9.85 ($\pm$0.1)	9.95 ($\pm$0.01)	0.22 (2.2%)	30,800.76 (15,187.34)
SD-Loose	1567 ($\pm$286)	9.59 ($\pm$0.22)	10.21 (0.05)	0.89 (9.27%)	38.91 (17.60)
SD-Nom	2315 ($\pm$251)	9.72 ($\pm$0.13))	10.14 (0.04)	0.59 (6.03%)	103.86 (30.48)
SD-Tight	3318 ($\pm$670)	9.88 ($\pm$0.11)	10.12 (0.04)	0.39 (3.98%)	299.02 (177.00)

Table 3. Computational Comparison of Regularized Benders and SD for SSN with *Random Costs in the Second Stage*

Based on the upper and lower bounds reported in Table 3 (see also the column labeled pessimistic gap), it is clear that SD method provides decisions which are far closer to optimality in far less time (see the times reported in the last column) than the Benders' Decomposition method which is standard for Large Scale Stochastic Programming problems.

Multi-stage Stochastic Linear Programming with a Simulator (Multi-stage SLP)

H. Gangammanavar, and S. Sen, "The Stochastic Dynamic Linear Programming Algorithm" under revision 2019.
H. Gangammanavar, and S. Sen, "Two-scale Stochastic Optimization Framework for Controlling Distributed Storage Devices" *IEEE Transactions on Smart Grid*, vol. 9, pp. 2691-2702, 2018.

Among the most popular methods for Multi-stage Stochastic Linear Programming is the algorithm of Pereira and Pinto (1991). That method, which goes by the acronym SDDP (for Stochastic Dual Dynamic Programming) has become among the most popular multi-stage SP algorithms, primarily based on its ability to address large scale power system applications. SDDP is based on a forward sampling/simulation scheme in which only a small subset of scenarios of the multi-stage SP model are sampled, and one creates piecewise linear approximations for each node in the sample generated during the forward sampling pass. By restricting the number paths generated during each forward pass, SDDP restricts the number of nodes (of the scenario tree) that one must visit during any iteration. While this does reduce the computational burden of associated with any iteration of SDDP, as compared with nested Benders decomposition, there are several bottlenecks which remain for large scale implementations. First and foremost, the model requires a full description of the scenario tree, including the probability distribution associated with each node of the scenario tree. This severely restricts the ability to accommodate very large scale SP models where distributions associated large scale trees may be too cumbersome to specify. In such applications, one often works via simulators which can work as an oracle that the next state of the stochastic process, and essentially, leaves it to an algorithm to infer the probability of visiting nodes of the scenario tree. For such models, our group has presented the multi-stage Stochastic Decomposition (MSD) algorithm in Sen and Zhou (2014). As part of this project, we have developed the so-called Stochastic Dynamic Linear Programming (SDLP) algorithm (Gangammanavar and Sen (2018, 2019)). These two are companion papers, with the former providing computational results for a smart electricity grid, and the latter providing the basic theory. Unlike SDDP whose convergence proof depends on discovering all (finitely-many) dual-extreme points (of a nodal dual problem), the proof of convergence of SDLP relies on the convergence of values of nodal approximations generated during the course of the forward/backward updates of MSD. Other than such mathematical differences, there is an important modeling consequence of SDLP: it allows the recursive use of simulators within the algorithm setup, and as a result provides greater fidelity than a simple probability distribution (as in SDDP).

The computational work reported in Gangammanvar and Sen (2018) is transcribed in Table 4. These computational instances were based on an IEEE Data Sets (IEEE 14, 30, and 57) with parameters n, z, p in the data, some with decreasing renewable energy (n), others with zero change in renewables (z) and still others with positive change (p) in renewable energy. It is important to note that as the size of the instance grows (from 14 nodes to 57 nodes), the time per iteration for SDLP increases at a far lower rate than for SDLP. This is particularly important in large scale applications. Many more computational experiments are reported in Gangammanavar and Sen (2018) and the ability to simulate the operations of batteries and other storage devices (with losses) is critical for optimization, and this aspect is highlighted by the features provided in SDLP.

Instances	Time per iter.	SDDP Prediction value (v_1)	Sample Mean	U.B. Estimation 95 % C.I. [CI_ℓ, CI_u]	Time per iter.	SDLP Prediction value(v_2)	Difference $\lvert v_2 - v_1 \rvert / v_1$
ieee14n	0.001	6534.47	6535.76	[6530.67,6540.86]	0.020	6537.89	5e-4
ieee14z	0.001	6353.72	6351.59	[6346.25,6356.93]	0.021	6353.72	3e-4
ieee14p	0.001	5991.30	5992.64	[5987.30,5997.98]	0.023	5994.77	6e-4
ieee30n	0.070	8489.47	8487.38	[8470.19,8504.58]	0.052	8482.08	8e-4
ieee30z	0.025	7804.19	7801.54	[7789.26,7813.82]	0.042	7786.69	2e-3
ieee30p	0.021	7334.09	7340.13	[7326.80,7353.46]	0.038	7309.49	3e-3
ieee57n	1.023	24684.45	24695.83	[24686.51, 24705.14]	0.070	24639.42	2e-3
ieee57z	0.793	24279.30	24287.18	[24277.20, 24297.16]	0.076	24229.24	2e-3
ieee57p	0.641	22850.14	22868.82	[22849.84, 22867.81]	0.068	22854.57	2e-4

Table 4. Comparison of SDDP and SDLP

Two-stage Stochastic Mixed-Integer Programming (2-SMIP)

Some of the most challenging models in optimization arise from a combination of discrete optimization and decisions under uncertainty. One of the lessons from deterministic discrete optimization has been that accommodating discrete variables in optimization is best accomplished by exploiting special structures. In a previous AFOSR project, we had obtained results for multiple cases of SMIP models using binary variables. This class of models possesses the so-called facial structure which means that all solutions to these types of models belong to some face of the polyhedron representing the feasible set. Thanks to AFOSR support, our team was awarded the prestigious INFORMS Computer Society award in 2015 for seminal work in Stochastic Mixed-Integer Programming.

Concurrently with the above award, we undertook a study of one of the more general structures in discrete optimization, namely, the two-stage SMIP with mixed-integer variables in both stages. The resulting paper is listed below.

Y. Qi and S. Sen, "Ancestral Benders' Cuts and Multi-term Disjunctions for Mixed-Integer Recourse Decisions in Stochastic Programming," *Mathematical Programming* January 2017, Volume 161, pp 193–235.

The above paper had already been started during the previous AFOSR grant, but the full extent of its power was only realized during the course of this particular grant. As we explored the implications of the above paper during the current grant, several individuals with background in Homeland Security contacted us explaining that our setup would lend itself to modeling situations in which the goal is to deploy alternative technologies (cameras, software etc.) in such a manner as to thwart several categories of perpetrators (e.g. smugglers, terrorists etc.) within the available resource constraints. Given the major interested in such problems, we were invited to present a tutorial paper at the INFORMS conference in Nov. 2017. Based on that invitation, the following paper was presented in Houston.

S. Küçükyavuz and S. Sen, "Introduction to Two-stage Stochastic Mixed Integer Programming" in *INFORMS TutORials*, pp. 1-27, 2017.

Finally, we should mention that recently, we have made progress on using sequential sampling (Stochastic Decomposition) for the case of 2-SMIP models. As noted earlier, our SD approach allows seamless integration with simulators, allowing SP algorithms to work directly with simulation software, as well as discrete optimization will open a large class of unsolved problems. This will allow us to adopt a distribution-free approach, although no such methods exist at this point. This will be particularly powerful for models in which the second stage is unimodular, although the first-stage may be very general. We are in the process of exploring such models in the near future. While this class of models appears to rely entirely of unimodularity theory, that is not exactly the case because the entire model is not unimodular: only the second stage. It turns out that

many practical problems of logistics and distribution invoke such properties, and the challenge

Coalescing Data and Decision Sciences

One of the deficiencies of Stochastic Programming (SP) is that it does not treat data in the form of predictors and responses. Nevertheless, most of statistical and machine learning (especially supervised learning) approaches are based on data being available in this form. From a decision-making point of view, this form of data-driven decision model allows decisions to be far more agile because the decisions respond to observed data. Our project is working towards the fusion of statistical learning and stochastic programming, which we refer to as Predictive Stochastic Programming (PSP). Two papers in this realm were produced in the current project.

Y. Deng and S. Sen, "Learning Enabled Optimization: Towards a Fusion of Statistical Learning and Stochastic Programming" *Optimization Online, 2018.*

The above paper is still in the revision process, but based on the work of that paper, my students and I were invited to present a tutorial which appeared in the following paper. The appears to be a lot of interest in this class of decision models because they bridge the areas of Stochastic Programming and Statistical Learning. These concepts form the basis for the following paper.

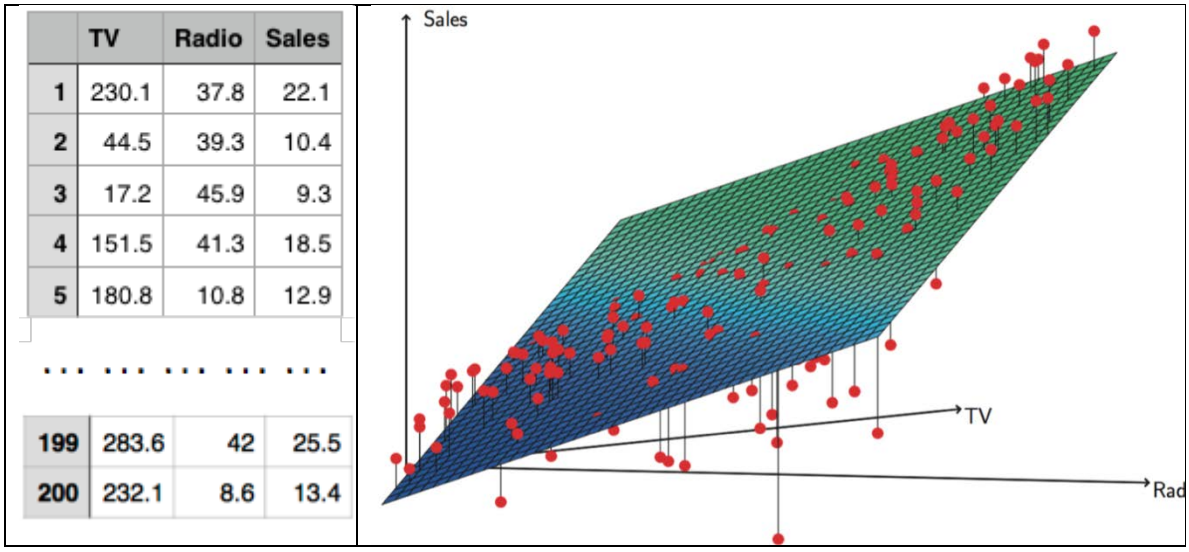
Y. Deng, J.Liu and S. Sen, "Coalescing Data and Decision Sciences for Analytics" *INFORMS TutORials pp. 20-49, 2018.*

In order to provide the reader a sense of a specific drawback of SP, we present a pedagogical example in which we wish to study the coordination between advertising and production shown in Figure 1. Here predictors $\{Z_i\}$ represent minutes of TV and Radio advertising, and response $\{W_i\}$ denotes sales. In this case, management seeks to use this data for *prospective* decision-making where it wishes to know how much of its 200,000 advertising minutes should it commit to TV, and radio so that the total expected profit (accounting for production costs of the "Wyndor Production" model) can be maximized. Thus the set $\boldsymbol{X}$ represents the 200,000 min. advertising limit, and certain policies (e.g. non-negative expenditures). This is an application in which the management must make a bet as to how well the investments in advertising will translate into future sales W, which in turn affects profits from production in the future. The profit (which is a random variable) is estimated using a resource allocation LP whose variables are production quantities $(y_A(\xi), y_B(\xi))$ which are variables in the resource allocation LP, and these

choices depend both on the regression model $m(z,\xi)$, which includes the predictor as well as the error random variable ξ (which are in fact latent).

As part of our pedagogical example, we assess three different versions of the PSP model:

a) Use a deterministic approximation $\hat{m} \coloneqq m(x,0)$; this approximation leads to a linear program. (This appears to be a common among many consulting outfits).
b) All errors (ξ_0, ξ_1, ξ_2) are 0-mean, correlated Gaussian random variables (obtained after linear regression). Here ξ_0 denotes the error for the constant term, while ξ_1, ξ_2 denote the errors of the coefficients of the regression.
c) There is only additive error and hence outcomes of ξ_0 are $\xi_{0,i} = W_i - m(Z_i, 0)$.



	TV	Radio	Sales
1	230.1	37.8	22.1
2	44.5	39.3	10.4
3	17.2	45.9	9.3
4	151.5	41.3	18.5
5	180.8	10.8	12.9
...	...	...	...
199	283.6	42	25.5
200	232.1	8.6	13.4

Figure 1. The Advertising Data Set (Source: James et al 2011).

Given the above regression, the decision model involves choosing the advertising expenditures in a manner that maximizes expected (profits – cost of advertising) as on the left hand panel below.

$\text{Max} \quad -0.1x_1 - 0.5x_2 + E_{\xi}[h(x=z,m)]$ $x_1 + x_2 \le 200$ $x_1 - 0.5\,x_2 \ge 0$ $L_1 \le x_1 \le U_1,\ L_2 \le x_2 \le U_2$	where, $h(x=z,m) = \text{Max}\ 3\,y_A + 5\,y_B$ s. t. $y_A \le 8$ $2\,y_B \le 24$ $3y_A + 2y_B \le 36$ $y_A + y_B \le m(z,\xi)$ $y_A,\ y_B \ge 0$

The right hand side of the above panel represents the profit under resource constraints (as in Hillier and Lieberman 2011) with a total sales constraint represented as $y_A + y_B \le m(z,\xi)$.

It is important to notice that the prediction introduced above is a random variable derived from the regression in Figure 3. Using the randomness of errors in $\tilde{\xi}$ *(which includes the randomness of coefficients* B*) leads to a fully data-driven and automated setup which is in the spirit of SL/ML and is sometimes referred to as Learning Enabled Optimization (LEO, Deng et al 2018), but more generally as PSP.*

In order to illustrate the range of models , which arise via the combination of learning and optimization, we present three alternative plausible models. The first model a) is based on a deterministic forecast for the case of $\xi = 0$. On the other hand, models b) and c) involve estimated Gaussian random variables (sales). In any event, a) is solved using linear programming, while b) and c) is an SP model, solved using Stochastic Decomposition (SD) which is a specialized SP algorithm combining *stochastic* approximation and Benders *decomposition* (Higle and Sen, 1994, and Sen and Liu 2016). The summary of results for each of these models is shown in Table 5. It turns out that Model a) is rejected by the hypothesis test because the objective function estimated using the training data has a predicted 95% confidence interval (CI) which has no overlap with the 95% validation CI shown in Table 5. This is essentially because of the overfitting problem of linear programming (using a deterministic prediction $m(x, 0)$). In the same Table we also report estimated optimization and generalization errors. The former is obtained using a non-parametric analysis of variance, via the Kruskal-Wallis test, while the generalization error is based on results in Deng and Sen (2018). Since the optimization error is so dominant, compared with the generalization error for models b) and c), one recommends c) as the model to choose.

Model	Model a)	Model b)	Model c)
x_1	172.48	181.40	191.40
x_2	26.52	18.60	8.6
Predicted 95% CI (Training)	\$41,391 (± 601)	\$41,492(± 272)	\$42,045 (± 465)
Validated 95% CI	\$39,869 (± 692)	\$41,865 (± 302)	\$42, 274 (± 493)
Optimization Error	\$2405	\$409	$f^* = 42{,}274$
Generalization Error	N/A	19.554	21.326

Table 5. Model Comparisons for LEO-Wyndor Example (Deng et al 2018)

Statistical Optimality:

One of the challenges with PSP models is that it is specified by using statistical inference on the coefficients of a regression involving $\{Z, W\}$, leading to random variables which are continuous (Gaussian). For the LEO-Wyndor example presented in Figure 1 and Table 2, we have $p = n_z = 2$, and $n_w = 1$. As a result, the PSP model has 3 Gaussian random variables, one for each regression coefficient and one for the bias term ($(n_z + 1) \times n_w = 3$). Because the number of random variables in PSP grows as O($n_z \times n_w$), these models grow in complexity more rapidly than a standard SP. Of course, the latter does not use predictive capabilities because $n_z = 0$). In other words, there is a computational cost to pay to include predictability: the increased number of random

variables is akin to lifting the problem into higher dimensional, albeit Gaussian, random variables. One way to reduce the increased complexity of the new class of models is to seek approximately optimal solutions. Moreover, our work is also motivated by the potential for automated scenario generation, which is typically not included as part of *the complexity of standard SP*. Because there is no free lunch, we expect an increase in overall complexity, and as a result, we propose to lower the burden of seeking decisions via the concept of "statistically" optimal solutions. The idea here is to reflect the aspirations of a modeler who is willing to accept decisions which are close to optimum (i.e., δ-optimum, $\delta > 0$) with a high degree of confidence (say 95%). Let $\mathcal{P}_q$ denote the Gaussian distribution of the random variables ($\tilde{\xi}$) for model q. Then we will seek a pair (x_q, γ_q) such that for a pre-specified accuracy tolerance $\delta > 0$, we have

$$\gamma_q \coloneqq \mathcal{P}_q(\boldsymbol{x}_q \in \delta - \operatorname{argmin}\{\hat{f}_q(\boldsymbol{x}) \mid \boldsymbol{x} \in X\}) \tag{5}$$

with $\gamma_q \geq \underline{\gamma}$ = (0.95, say). We refer to the requirement in (5) as Statistical Optimality. As the reader might notice, this condition states our aspirations for a PSP model in such a manner that we expect to report the following critical quantities: δ, γ_q, and $\boldsymbol{x}_q$. The precise manner of estimating such a probability consists of combining the notion of a compromise decision (Sen and Liu 2016) with (5), and the mathematical details appear in Deng and Sen (2018). When (4) is a convex optimization model (e.g. when the regression has a multiple linear regression structure), Deng and Sen (2018) shows that one can use sampling algorithms, with replications (perhaps in parallel), to identify solutions which satisfy (5) under certain assumptions (e.g., convexity, Lipschitz continuity, and knowledge of some parameters) identified in the Theorem below. In the following theorem we use F as in (1) and $\omega \equiv \xi$.

Theorem. Let L denote the expectation of the family of random Lipschitz functions $F(x, \xi)$, and D the diameter of the ball containing the set X. Assume that an instance of the type stated in (4) is solved by i.i.d. sampling of K_ν points from the Gaussian distributions provided by the regression. Assume $N = \min_{\nu} K_\nu$ and that sampling-based algorithms are independently replicated M times, using a proximal-point (regularized) algorithm using proximal parameters $\{\rho^\nu\}$. Assume that for each replication, the solution algorithm produces an approximation $\hat{f}^\nu$ such that $\hat{f}^\nu(\cdot) \leq (K_\nu)^{-1}(\sum_t F(\cdot, \xi^t))$, finds an ε-optimal solution for each proximal replication ν, denoted x^ν, where $\varepsilon < \delta_u$ (assumed given). Let x^c denote $\delta - \operatorname{argmin}\{\bar{F}_M(x) \coloneqq \frac{1}{M}\sum_\nu f^\nu(x)\}$, where $\delta < \delta_u$. Moreover define $\bar{x} = \frac{1}{M}\sum_\nu x^\nu$. If $\delta = \bar{\rho}\|x^c - \bar{x}\|^2$, where $\bar{\rho}$ is the average among the terminal values of ρ^ν . Then,

$$\mathcal{P}\left(x^c \in S(\delta_u)\right) \geq 1 - \exp\left(-\left(\frac{NM(\delta_u - \delta)^2}{32L^2D^2}\right) + n \ln\left(\frac{8LD}{\delta_u - \delta}\right)\right)$$

where $S(\delta_u)$ is the set of solutions $\{x \mid f(x) \leq f^* + \delta_u\}$, where f^* is the optimal value of (4), under the assumptions of this theorem. ■

Thus although the PSP setting is more demanding than standard SP, computational algorithms such as stochastic proximal point methods (e.g., Stochastic Decomposition)

can be used to obtain solutions with about the same level of accuracy in case of convex PSP.

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/USC3DLAB
- A repository of experiments is available at https://core.isrd.isi.edu

Dissertations (Four Completed During the Course of this Project; Two Ongoing)

[1] Y. Liu (Ph.D) "Statistical Aspects of Stochastic Decomposition," At 85.41, a Data Science subsidiary of Kroger.
[2] S. Atakan (Ph.D) "Advances in Stochastic Mixed-Integer Programming with Applications in Power Systems Planning," May 2018. OR Analyst at Amazon
[3] Y. Deng (Ph.D.) "Learning Enabled Optimization and Applications", joining Google in December, 2018.
[4] J. Liu (Ph.D.) "Stochastic Programming for Prescriptive and Predictive Analytics" June 2019
[5] S. Diao (Current Ph.D. Student) "Non-parametric Learning Enabled Optimization and Applications",(ongoing)
[6] J. Xu (Current Ph.D. Student) "Deep Learning with Stochastic Optimization" (ongoing)

Major Presentations Only (2016/2017/2018/2019)

[1] (Plenary Presentation). "Big Data and Big Decisions" INFORMS Annual Conference, Nashville, TN, Nov. 2016.
[2] (Tutorial Presentation) "Tutorial on Stochastic Mixed-Integer Programming," INFORMS Annual Conference, Houston, TN, Nov. 2017.
[3] (Distinguished Lecture) "Learning Enabled Optimization," ISEN Department, Texas A&M University, Oct. 2017
[4] Lecture at German Mathematics Institute, Oberwolfach, August, 2018
[5] "Learning Enabled Optimization" Sabbatical Lectures at George Washington Univ. University of California-Berkeley, Johns Hopkins, Virginia Tech., Virginia Commonwealth University, George Mason University, Norwegian University of Economics, September-October 2018.
[6] (Tutorial Presentation) "Tutorial on Learning Enabled Optimization," INFORMS Annual Conference, Phoenix, TN, Nov. 2018.
[7] "Stochastic Decomposition Revival," Sabbatical Lectures Univ. of Arizona, Univ. of Maryland, Univ. of California-Davis , Sept. 2018, March 2019.
[8] (Inaugural Lecture, Operations Research Society Conference), "Learning Enabled Optimization," December, 2018

Students Supported

J. Liu, Y. Deng, J. Xu. The First two have recently graduated, and the third is continuing to work on his Ph.D.

Publications

[9] H. Gangammanavar, Y. Liu and S. Sen, "Stochastic Decomposition for Two-Stage Stochastic Linear Programs with Random Cost Coefficients," submitted last minor revision to *INFORMS Journal on Computing*

[8] J.Liu and S. Sen, "Asymptotic Results on Two-stage Stochastic Quadratic Programming" submitted last minor revision to *SIAM J. on Optimization*

[7] Y. Deng, J.Liu and S. Sen, "Coalescing Data and Decision Sciences for Analytics" *INFORMS TutORials pp. 20-49, 2018.*

[6] S. Atakan and S. Sen, "A Progressive Hedging Based Branch and Bound Algorithm for Stochastic Mixed Integer Programming" *Computational Management Science*, vol. 15, pp. 501-540, 2018.

[5] H. Gangammanavar, and S. Sen, "Two-scale Stochastic Optimization Framework for Controlling Distributed Storage Devices" *IEEE Transactions on Smart Grid*, vol. 9, pp. 2691-2702, 2018.

[4] A. Atakan, G. Lulli, and S. Sen, "A State-Transition MIP Formulation for the Unit Commitment Problem," *IEEE Transactions on Power Systems*, Vol. 33 (1), pp. 736-748, 2018.

[3] S. Küçükyavuz and S. Sen, "Introduction to Two-stage Stochastic Mixed Integer Programming" in *INFORMS TutORials*, pp. 1-27, 2017.

[2] Y. Qi and S. Sen, "Ancestral Benders' Cuts and Multi-term Disjunctions for Mixed-Integer Recourse Decisions in Stochastic Programming," *Mathematical Programming* January 2017, Volume 161, pp 193–235.

[1] S. Sen and Y. Liu, "Mitigating Uncertainty via Compromise Decisions in Stochastic Linear Programming: Variance Reduction" *Operations Research*, 64(6):1422-1437, August 2016.

References

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

[2] Gangammanavar, H. and S. Sen (2019) “The Stochastic Dynamic Linear Programming Algorithm,” currently under revision.

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

[4] Hillier, F. and G. Lieberman (2012). Introduction to Operations Research.

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

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

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

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

[9] S. Sen and Z. Zhou, “Multi-stage Stochastic Decomposition: A Bridge Between Stochastic Programming and Approximate Dynamic Programming” *SIAM Journal on Optimization* , Vol. 24, pp. 127-153, 2014.

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

[11] 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.

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

[13] 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.

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

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