Learning Enabled Optimization: Towards a Fusion of Statistical Learning and Stochastic Programming

Suvrajeet Sen, Yunxiao Deng
Daniel J. Epstein Department of Industrial and Systems Engineering
University of Southern California, Los Angeles, 90089
s.sen@usc.edu, yunxiaod@usc.edu

Several emerging applications call for a fusion of statistical learning (SL) and stochastic programming (SP). The Learning Enabled Optimization paradigm fuses concepts from these disciplines in a manner which not only enriches both SL and SP, but also provides a framework which supports rapid model updates and optimization, together with a methodology for rapid model-validation, assessment, and selection. Moreover, in many "big data/big decisions" applications, these steps are repetitive, and realizable only through a continuous cycle involving data analysis, optimization, and validation. This paper sets forth the foundation for such a framework by introducing several novel concepts such as *statistical optimality, a convex extension of stochastic decomposition, hypothesis tests for model-fidelity, generalization error of stochastic programming,* and finally, a *non-parametric methodology for model selection.* These new concepts provide a formal framework for modeling, solving, validating, and reporting solutions for Learning Enabled Optimization (LEO). We illustrate the LEO framework by applying it to an inventory control model in which we use demand data available for ARIMA modeling in the statistical package "R". In addition, we also study a production-marketing coordination model based on combining a pedagogical production planning model with an advertising data set intended for sales prediction.

Key words: Stochastic Linear Programming, Statistical Learning, Model Assessment

---

2/1/2018

1. Introduction

In recent years, optimization algorithms have become the work-horse of statistical (or machine) learning. Whether studying classification using linear/quadratic programming for support vector machines (SVM) or logistic regression using a specialized version of Newton's methods, deterministic optimization algorithms have provided a strong foundation for statistical learning. Indeed, statistical learning could be labeled as "optimization enabled learning". The class of models studied in this paper, entitled Learning Enabled Optimization (LEO), is intended to leverage advances in statistical learning to support the work-flow associated with stochastic programming.

In terms of scientific genealogy, one can trace the introduction of learning into optimization from the work on approximate dynamic programming (ADP, Bertsekas (2012), Powell (2011)) and approximate linear programming (ALP, e.g. De Farias and Van Roy (2004)). The canonical structure of these approaches pertains to DP, where one uses approximations of the DP value function by using basis functions. In this paper, the canonical setup derives from constrained optimization, although we will state our objectives in the context of approximate solutions. In this sense, one may refer to the technical content of our approach as "approximate stochastic programming."

An alternative data-driven approach to modeling uncertainty is via the ideas of Robust Optimization (RO) (Bertsimas and Sim (2004), Ben-Tal and Nemirovski (2001)), where performance is measured in terms of the ability to withstand extreme or near-extreme events. There are many applications (e.g., engineering design) where the ability to survive unforeseen circumstances is important. Slightly less demanding criteria come about via risk-averse optimization where the decision making model attempts to strike a balance between "risk" and "return"(e.g., Miller and Ruszczyński (2011)). The line of work pursued in this paper pertains to Stochastic Programming (SP, Birge and Louveaux (2011)). The above classes of models serve alternative types of applications of decisions under uncertainty. Each of these approaches haveg their "sweet spot" in terms of applications and ability to cope with constrained decisions under uncertainty.

We expect the LEO approach to be particularly powerful for environments with many data sources (volume), rapid information flow (velocity), and require adaptation to uncertain shifts in future data (volatility). In these situations, the modeling framework should also accommodate rapid updates and repeated execution of computations involving data, while the performance index is often the risk neutral (expectation) objective. However there are several hurdles to overcome. To elaborate, note that for most SP formulations (not all), one captures uncertainty via scenarios, and this calls for a sequence of steps involving preliminary scenario generation, followed by scenario aggregation/reduction, and one then obtains a solution using finite dimensional optimization. Unfortunately, many SP exercises stop at this point of the work-flow, although there is growing recognition of the need to assess solution quality which is often undertaken as a "post-optimality" exercise. While there are some variations to this sequence of steps, model verification and validation are usually not emphasized. Nevertheless, true decision-support using SP requires a far more care than has been forthcoming in the literature. In this paper, we recommend that the entire work-flow be undertaken via a fusion of Statistical Learning[1] and SP.

For the LEO paradigm, our goal is to import SL models as an integral part of an SP decision model, and in turn, this requires an overhaul of the current decision-support methodology supporting SP. Given the variety of SL models that are available, this approach can vastly expand the scope and influence of SP. To the best of our knowledge, this paper is the first to provide a formal basis for the flow of activities involving data analysis, uncertainty modeling, optimization algorithms, output analysis, as well as model assessment, and selection for decision-support. The specific capabilities required to achieve these goals are described in various sections of this paper.

Connections to the Literature and Contributions

In keeping with our goals to accomplish more with data in OR/MS applications, there have been some attempts to have optimization methods guide information gathering for predictive analytics in the work of Frazier (2012) and Ryzhov et al. (2012) which are intended to help an experimentalist improve the effectiveness of predictive models by using sensitivity of the response (using a concept known as knowledge gradient) to design experiments. This line of work uses algorithmic notions of optimization for experimental design (including simulation experiments). A more decision-oriented framework is proposed in Kao et al. (2009) where the regression coefficients are chosen based on a combined objective consisting of a loss function associated with regression, as well as that of optimization. The assumptions of Kao et al. (2009), namely, unconstrained quadratic optimization of both problems renders their simultaneous optimization manageable. However, if the optimization model were inequality constrained (as in many applications), such simultaneous optimization would lead to bilevel stochastic programs, which are much harder than the SP setting of the LEO model. Another viewpoint at the interface between predictive and prescriptive analytics is presented in the recent paper by Bertsimas and Kallus (2014). Their work demonstrates that in the presence of dependencies among random variables, using predictive models which capture dependencies (e.g., k-nearest neighbors) lead to more cost-effective decisions than using SAA without exploring dependency. It should be noted that the setting of the above paper is such that the random variables and the decision variables

[1] We will use the term statistical learning when the discussion is application-agnostic. For specific applications we use the term machine learning.

assume values in disjoint spaces (to be formalized in the next section as LEO models with disjoint spaces). The LEO models presented in this paper not only accommodate the above case, but they are also able to accommodate situations in which both SL and SP models share the same spaces. In this way, the LEO model can accept a statistical model whose parameters may be continuous random variables as in many types of regression. This may lead to an infinite dimensional optimization problem for which we propose an approximate solution through a new concept of statistical optimality.

As for the Operations Management (OM) literature, the primary focus of statistical learning has been on specific classes of problems (e.g. newsvendor models). For instance, Liyanage and Shanthikumar (2005) and more recently Rudin and Vahn (2014) have studied the integration of optimization and learning to identify optimal inventory ordering policies. Both papers use the specialized form of the newsvendor model to illustrate the potential for learning and optimization. Another avenue of application-specific use of learning within optimization arises from the introduction of specialized regularization in the context of portfolio optimization in Ban et al. (2017). More generally, the types of questions which motivate our work may be summarized as follows.

- Given that data has become such an widely available commodity, and SL models have begun to play a major role in most operations, are there OR/MS questions which might benefit from leveraging tools of SL and SP within a unified framework?
- If SL models are introduced into an SP framework, what is the appropriate way to incorporate them, even though SL models may be governed by continuous random variables?
- How should we compute optimal decisions in cases involving infinite dimensional models (i.e., extremely large/big data sets)?
- How should we assess the effectiveness of these models so that we can support decisions based on validation procedures based on statistical principles?

As the reader will recognize from this paper, the LEO approach provides a relatively general framework for decisions via SP, and uncertainty modeling via SL. This not only allows a more streamlined approach to building a statistical model to support decision-making, but also recognizes that as in SL, the world of SP could benefit from providing decision-makers with predictions of future costs, as well as estimates of validity of these predictions. In addition, the LEO approach is based on choosing the most promising model from among a collection of alternatives which seem relevant in a learning process. For each model-type, we will carry out a collection of tests both before and after optimization to help guide model-choice. In this context, we suggest statistical estimates, and tests which support this choice.

This paper is organized as follows. In section 2, we present two fundamental structures, which we refer to as "LEO Models with Disjoint Spaces" and "LEO Models with Shared Spaces". We illustrate the first of these structures with an inventory control problem, and the second one is illustrated using a production-marketing coordination model. Because LEO models will allow both continuous and discrete random variables (rvs), the statement of optimization will be relaxed to solutions with high probability (greater than 95%, say) of optimality. This concept, which is set forth in section 3, will be referred to as "statistical optimality" for sequential sampling algorithms (such as Stochastic Decomposition (SD)). In section 4 we study hypothesis tests for model validation. Such tests identify the contenders (models) which may be most promising. In addition, we also define a concept of generalization error which is motivated by an analogous concept in statistical learning. For LEO models, this measure aims to quantify the degree of flexibility expected in the decision model. This entire protocol is illustrated in section 5 via computations for the examples introduced in section 2. Finally, section 6 presents our conclusions and a possible path forward for this new genre of models.

Because many of the ideas in the paper span SL and SP, while the audience of may feel comfortable with only one of these topics, we have included more supplementary material than is common. Specifically, we associate one Appendix for each of the sections from section 2 to section 5, pairing Appendices I, II, III and IV with sections 2, 3, 4 and 5. There is also a fifth appendix (Appendix V) which includes the proofs. Readers unfamiliar certain aspects of a particular section might prefer reading the corresponding Appendix in conjunction with that section.

2. Learning Enabled Optimization

Statistical Learning provides support for predictive analytics, whereas, optimization forms the basis for prescriptive analytics, and the methodologies for these are built independently of each other. The process recommended for SL is summarized in Figure 1a in which the entire data set is divided into two parts (Training and Validation), with the former being used to learn model parameters, and the latter data set used for model assessment and selection. Once a model is selected, it can be finally tested via either simulation or using an additional "test data set" for trials before adoption. This last phase is not depicted in Figure 1 because the concepts for that phase can mimic those from the model validation phase. By inserting the prescriptive model into Figure 1b the LEO framework greatly enhances the potential for fusion.

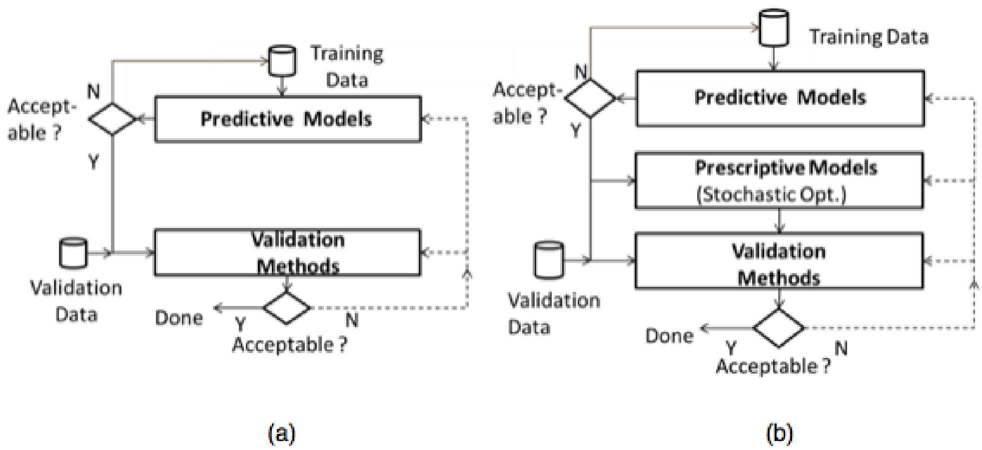

Figure 1 **Statistical Learning and Learning Enabled Optimization**

2.1. Model Instantiation

This section presents our aspirations for LEO models. As one might expect, this framework consists of two major pieces: the SL piece and the SP piece. We begin by stating a regression model in its relatively standard form. Towards this end, let m denote an arbitrary regression model for a training dataset $\{W_i, Z_i\}$, indexed by $i \in T$. Here W_i denotes the observation and Z_i the predictor. For notational simplicity we assume that $W_i \in \mathbb{R}$, whereas $Z_i \in \mathbb{R}^p$. Given the training data, a class of deterministic models $\hat{\mathcal{M}}$, and a loss function ℓ, the regression is represented as follows:

$$\hat{m} \in \operatorname{argmin}\Big\{ \frac{1}{|T|} \sum_{i \in T} \ell(m) \mid m \in \hat{\mathcal{M}} \Big\}. \tag{1}$$

We wish to emphasize that in many circumstances (e.g., modeling the impact of natural phenomena such as earthquakes, hurricanes etc.), model fidelity may be enhanced by building the statistical model of the phenomena independently of decision models. For such applications, one may prefer to treat the development of the SL piece prior to instantiating the SP piece. Other phenomena requiring simultaneous treatment of SL and SP will be treated in a future papers.

Alternative Error Models

Empirical additive errors: Suppose that an error random variable has outcomes defined by $\xi_i := W_i - \hat{m}(Z_i)$. By associating equal weights to these outcomes, one obtains a discrete random variable with an empirical distribution. For the case of multiple linear regression (MLR), $\hat{m}(z) = \sum_\tau \beta_\tau z_\tau$, where τ ranges over the index set of predictors. In that case, the empirical additive errors are given by substituting the deterministic affine function in place of $\hat{m}$. For $\tau = 0$, it is customary to use $z_0 = 1$. When the distribution does not depend on any specific choice of $Z = z$, the random variable is said to be homoscedasticity (as is commonly assumed for MLR). For a more general setup (known as projection pursuit (Friedman and Stuetzle (1981))), one may define $\hat{m}(z) = \sum_{\tau \in \mathcal{T}} \phi_\tau(\beta_\tau^\top z)$, where $\mathcal{T}$ is a finite index set. Again, the errors are assumed to have the same homoscedasticity properties. This projection pursuit approach leads to a rather general setting due to a result of Diaconis and Shahshahani (1984), which suggests that nonlinear functions of linear combinations of general directions (as opposed to coordinate directions) can produce arbitrarily close approximations of smooth nonlinear functions. We should note that the notion of homoscedasiticty translates to stationary error distributions in the context of time series models, and there are standard (and partial) autocorrelation tests which can discern stationarity. While on the context of error distributions, we should mention the work of Rios et al. (2015) which uses splines and epi-splines to model additive error distributions (see also Royset and Wets (2014)). However, the corresponding error distributions may not necessarily satisfy homoscedasticity. Consequently, extensions to non-convex optimization would be required to accommodate such generality.

Multi-dimensional errors: Define a stochastic function $m(z, \xi) = \sum_\tau \tilde{\beta}_\tau \phi_\tau(z)$, where $\phi_0 = 1$, and $\phi_\tau(\cdot)$, are deterministic functions, but the parameters $\tilde{\beta}_\tau$ are elements of a multivariate random variable. Let $\hat{m}(z) = \sum_\tau \bar{\beta}_\tau \phi_\tau(z)$, where $\bar{\beta}_\tau = \mathbb{E}(\beta_\tau)$. In this case, a vector of errors associated with an outcome of random coefficients $\{\tilde{\beta}_\tau\}$ is given by the difference $\tilde{\xi}_\tau = \tilde{\beta}_\tau - \mathbb{E}(\beta_\tau)$. The coefficients $\{\tilde{\beta}_\tau\}$ may be correlated, but the multivariate distribution of these coefficients $\{\tilde{\beta}_\tau\}$ are assumed to not depend on any specific choice of $Z = z$ (homoscedasticity).

Remark 1. (a) In SL it is common to formulate a stochastic model $m(z,\xi)$, although predictions are deterministic and made with $\hat{m}(Z=z)$. In contrast, LEO will use an optimization model which will recognize inherent randomness of $m(x,\xi)$. For this reason, the deterministic model $\hat{m}(z)$ is not as relevant (for LEO) as the stochastic model $m(z,\xi)$. Such stochasticity will provide the optimization model a sense of uncertainty faced by the decision-maker, although the decisions produced by LEO must also be deterministic (at least for the first period, i.e., here-and-now). To draw another distinction between Directed Regression and LEO, we observe that the former uses the deterministic model $\hat{m}(z)$ rather than the stochastic model $m(z,\xi)$. (b) Note that alternative stochastic models m may lead to the same deterministic model $\hat{m}$. This is certainly the case of MLR. For this reason, the LEO setting will consider a finite list of plausible alternative stochastic models which will be indexed by the letter q. We suggest representing the class of models by $(\ell_q, \hat{\mathcal{M}}_q)$, and the specific deterministic model obtained in (1) is denoted $\hat{m}_q$. Whenever the specific index of a model is not important, we will refer to deterministic model as $\hat{m}$, and its stochastic counterpart by m. ■

Assumption 1 (A1). $\{W_i, Z_i\}$ are assumed to be i.i.d. observations of the data process $\{W, Z\}$. Moreover we assume that the errors are homoscedastic.

Assumption 2 (A2). We will assume that decisions in the SP model, denoted x, have no impact on the continuing data process $\{(W,Z)\}$ to be observed in the future.

To put this assumption in the context of some applications, note that in the advertising/financial market, it may be assumed that an individual advertiser/investor may not be large enough to change future market conditions.

For the remainder of this section, we present two alternative structures for the SP part of a LEO model.

2.1.1. LEO Models with Disjoint Spaces. This is the simplest version of a LEO model in which the statistical inputs (denoted Z) do not assume values in the space of optimization variables (x). Let $x \in \mathbf{X} \subseteq \mathbb{R}^{n_1}$ denote the optimization variables, and suppose that the predictors Z have p elements indexed by the set $\mathcal{J} = \{1, ..., p\}$. As suggested in Remark 1(b), SL models m_q are now assumed to be given. For the case of MLR, the parameters of the regression are Gaussian random variables with a corresponding distribution $\mathcal{P}_q$. Then we have an objective function of the following form.

$$f_q(x) := c(x) + \mathbb{E}_{\xi_q}[H(x, \tilde{\xi}_q | Z = z)] \tag{2}$$

where, $c: \mathbb{R}^{n_1} \to \mathbb{R}$ is a convex function, the expectation in (2) is calculated with respect to the rv $\tilde{\xi}_q|Z$, and this rv induces a random variable H whose outcomes h are given as follows.

$$h(x, \xi_q | Z = z) = \min\{d(y) | g(x, y) - m_q(z, \xi_q) \leq 0, y \in \mathbf{Y} \subseteq \mathbb{R}^{n_2}\} \tag{3}$$

Here $g: \mathbb{R}^{n_1+n_2} \to \mathbb{R}$. Clearly these constraints could be multi-dimensional, but some of the same conceptual challenges would persist. Nevertheless, there is an important algorithmic advantage resulting from the disjoint structure: because z and x belong to disjoint spaces, optimization with respect to x is not restricted by z. Hence, (2) can optimized by simply passing predicted values $m_q(z, \xi_q)$. As a result, the complexity of m_q does not impact the optimization part of the LEO model. As a result, very general regressions (e.g., Kernel-regression and others) can be included for the prediction process in (2).

Assumption 3 (A3-a). We assume that c, d, g are convex functions, the sets $\mathbf{X}, \mathbf{Y}$ are convex sets, and $h(x, \cdot)$ is a convex function in x as well.

The value function h defined in (3) goes by several alternative names such "recourse" functions in SP or "cost-to-go" function in DP. While the underpinnings of LEO models are closer to SP than DP, we adopt the "cost-to-go" terminology because we plan to extend LEO models to allow other types of "cost" predictions in future papers (e.g., forecasts of computational times, and other elements of an uncertain future). Note that H is a random cost-to-go function, and h are its outcomes, and $\mathbb{E}_\xi[H]$ is the expected cost-to-go function.

2.1.2. LEO Models with Shared Spaces. We continue with assumptions A1- A3, and will include another assumption for this class of models. Consider an SP model whose decisions x, have a subset of variables $(x_j, j \in J \subset \mathcal{J})$ which take values in the same space as the predictor data. Hence these models will be referred to as "models with Shared Spaces", and the objective function we formulate is as follows.

$$f_q(x) := c(x) + \mathbb{E}_{\xi_q}[H(x, \tilde{\xi}_q | Z = z, z_r = x_r, r \in J)]\}, \tag{4}$$

where, H is a convex function over the feasible set $\mathbf{X}$, and the outcomes h are defined as follows.

$$h(x, \xi_q | Z = z, z_j = x_j, j \in J) := \min\{d(y) | g(x, y) - m_q(z, \xi_q) \leq 0, y \in \mathbf{Y} \subseteq \mathbb{R}^{n_2}\} \tag{5}$$

As in (3) we assume that $h(x, \cdot)$ defined in (5) is a convex function in x. Note that in this form of the model, the decision maker is called upon to make a "bet" $(x_j, j \in J)$, and the response is a rv $H(x, \tilde{\xi}_q | Z = z, z_j = x_j, j \in J)$. While, the objective function of both Disjoint and Shared Spaces models have a similar forms, the interplay betwen decisions and random variables are different. To accommodate this, we state the following assumptions.

Assumption 3 (A3-b). In addition to Assumption (A3-a), (5) imposes the assumption that $m_q(z, \xi_q)$ is concave in $z_j, j \in J$ for all ξ except on a set of $\mathcal{P}_q$-measure zero.

Assumption 4 (A4). When decisions x are allowed to assume values in a subspace of observations of the rv Z, we assume that $\mathbf{X}$ is a subset of $\Pi_J\left(conv\ \{Z_i\}_{i \in T}\right)$, where the notation $\Pi_J(\cdot)$ denotes the projection on to the subspace of variables indexed by J. If number of decision variables are the same as those in Z, then set of feasible solutions ($\mathbf{X}$) is compact as well.

Due to the structural differences between (2) and (4) we will use a time-series model (ARIMA) to illustrate models with a disjoint structure, whereas, the illustration for the latter will be restricted to MLR because this form is much easier to manage in case of the latter.

2.1.3. Mathematical Formulation of a LEO Model. As mentioned in Remark 1, the decision model in LEO is a SP for which $m(z, \xi)$ is more critical that $\hat{m}(z)$ which is a deterministic forecast. The mathematical statement of a LEO model will be based on the recognition that importing a statistical model into an optimization problem can be demanding, but the payoff (via increased flexibility, and greater optimization) may be significant. The illustrative examples included in this paper demonstrate clear advantages of using Due to the possibility that the parameters of a statistical model (e.g. MLR) can have several coefficients which are continuous random variables (e.g, Gaussian), evaluating $f(x)$ may require some sampling-based SP algorithm. One of the more popular approaches for such SP is the Sample Average Approximation (SAA) which is summarized in Appendix II. However, it has been observed by many authors (e.g. Homem-de Mello and Bayraksan (2014)) that the sample size recommended by SAA theory can be extremely conservative (see also comments regarding sample sizes in the concluding section). For this reason, the algorithmic use of SAA consists of solving a sequence of sampled problems, each using a larger sample size (e.g., Linderoth et al. (2006)). Since most deterministic algorithms used for solving the SAA problem are not designed for changes in sample size, the algorithmic exercise becomes computationally burdensome due of the lack of coordination between solution algorithms and stopping criteria.

In order to bring the aspirations of a modeler (in search of decisions with performance guarantees), we state the LEO model in terms of a probabilistic guarantee. Let $\mathcal{P}_q$ denote the probability distribution of a random variable ξ_q. For LEO models, we will seek a pair (x_q, γ_q) such that for a pre-specified accuracy tolerance $\delta > 0$, we have

$$\gamma_q := \mathcal{P}_q \ (x_q \in \delta - \arg\min\{f_q(x) | x \in \mathbf{X}\}), \tag{6}$$

with $\gamma_q \geq \underline{\gamma}$ (= 0.95, say). As the reader might notice, (6) states our aspirations for a LEO model in such a manner that we can report the following critical quantities: δ, γ_q, x_q for a model indexed by q. The manner in which we verify these conditions will be discussed in the section on Statistical Optimality.

2.2. Examples of LEO Models

Example 1. Inventory Control: A LEO model with Disjoint Spaces -`LEO-ELECEQUIP`

The `ELECEQUIP` data-set in R provides 10 years of demand data for electrical equipment. We present an inventory control model with this data-set. Consider making equipment ordering choices in period t based on demand data from previous periods (i.e., periods $j < t$). Since the optimization model chooses decisions for periods $j \geq t$, we treat the optimization variables and that of the statistical model as disjoint. Clearly, this property holds for rolling horizon models as well. In essence disjoint spaces allow the statistical and optimization models to operate by simply passing values assumed by rvs. Because of the disjoint spaces, such a LEO model can entertain reasonably complex descriptions of data (e.g. time series, nonlinear and the so-called link functions). Our preliminary results provide an example using an ARIMA model, with (p,d,q), (P,D,Q) = (0,0,0), (1,1,0). This notation (p,d,q) is standard for ARIMA, and represents the number of periods used in predicting three parts of the stationary series (i.e., after removal of trends) representing autoregressive terms (p), integration terms (d) and moving average errors (q). The quantities P,D,Q refer to the "seasonal" component, which in this case was annual. In choosing these parameters, it is customary to test for stationarity using autocorrelation and partial autocorrelation functions (ACF and PACF) (Box et al. (2015)). Note that because P=1, D=1, the ARIMA model implies that two previous periods (from a year ago) are necessary for forecasting the demand for period $t = 1$. In order to acknowledge the rolling-horizon nature of the decision models, we also include an additional period look-ahead after period 1. Thus we have a three period model with $t = 0, 1, 2$. The detailed model formulation is provided in Appendix I. ■

Example 2. Production-Marketing Coordination: A LEO model with Shared Spaces - LEO-Wyndor

An important piece of data for production planning is predicted sales, which in turn depends on how much advertising is carried out. Suppose that a company buys advertising slots (units of time) on several media channels, then, this decision has an impact on future sales figures. We present an example which we refer to as the LEO-Wyndor data in which the decision vector x represents the allocation of the advertising time slots to each type of media (TV and radio). The name Wyndor and the production part of this problem is borrowed from a very popular OR textbook (Hillier and Lieberman (2012)). Our example extends the original Wyndor model to one in which the production plan is to be made while bearing in mind that the allocation of the advertising time slots affects sales of Wyndor products (two types of doors), and the final production plan will be guided by firm orders (sales) in the future. For this example, a statistical model predicting future sales is borrowed from the advertising data set of James et al. (2013) which also presents an MLR model relating sales (W) with advertising slots (Z) on TV and radio. In this example, the advertising decisions constitute a "bet" on the first stage (advertising) decisions x, and the second stage decisions are the production planning choices, given "firm orders" (sales). A specific numerical instance of this model is given in Appendix I. ■

The models presented in Appendix II have the same structure as a stochastic linear program. In case of the `LEO-ELECEQUIP` example, the cost-to-go function h has the form $h(x, \xi) = \min\{d^\top y \mid Dy = \xi - Cx, y \geq 0\}$, leading to a SP with deterministic matrices C, D, while randomness only appears as a vector ξ on the right hand side of the second-stage model. This is also the structure studied in Bertsimas and Kallus (2014).

In case of the `LEO-Wyndor` example, several alternative LEO models are plausible based on the SL model used. When we use empirical additive errors (EAE), the MLR coefficients ($\beta_j, j = 0, 1, 2$) are fixed scalars, and they become the deterministic entries of the matrix C, while the empirical additive error ξ is the random right hand side. On the other hand, when the MLR model with normally distributed coefficients are used, then $\{\tilde{\beta}_j\}_{\{j=0,1,2\}}$, then entries of C are also random. If the cross-correlations among these coefficients are ignored, then the matrix C is diagonal (with normally distributed uncorrelated random variables - denoted NDU). On the other hand, if the cross-correlations of $\{\tilde{\beta}_j\}_{\{j=0,1,2\}}$ are included, then the resulting LEO model has error terms which are normally distributed and correlated (denoted NDC). While these alternative statistical models are plausible, statistical theory recommends that we choose the model with the least generalization error (in a statistical sense). Since the goal of LEO models is to recommend optimal decisions e.g. order quantities or advertising allocations, we will choose that x_q for which the validated objective value $\hat{f}$ turns out to be the lowest, where,

$$\hat{f}(x_q) := c(x_q) + (1/|V|) \sum_{j \in V} h(x_q, W_j | z = Z_j) \tag{7}$$

Note that this validation process is based on observations set aside for the validation set V, and is entirely data-driven. Hence the index q only appears for the solutions x_q, but this index does not appear in the definition of $\hat{f}$. Finally, there are some obvious amendments to the definition of $\hat{f}$ when we use cross-validation (see section 4).

3. Statistical Optimality

In this section, we assume that the index q is fixed, and hence it is suppressed below. The algebraic statements of most optimization models lead to algebraic optimality conditions, and in turn, those engender deterministic algorithms. The introduction of a "reliability" requirement in (6) directs us towards computational statistical optimality so that the reliability requirement can be estimated, as well as new statistically motivated algorithms.

3.1. Parallel Replications and Statistical Optimality

Statistical optimality bounds have been studied in the literature for a while (e.g., Higle and Sen (1991), Higle and Sen (1996b), Mak et al. (1999), Kleywegt et al. (2002), Bayraksan and Morton (2011), Glynn and Infanger (2013)). A complete mathematical treatment of these concepts appears under the banner of "Statistical Validation" in Shapiro et al. (2009), and a detailed tutorial for SAA appears in Homem-de Mello and Bayraksan (2014). In addition, there have been theoretical investigations on how the computational budget ought to be allocated so that increases in sample size can be determined in an online manner (e.g., Bayraksan and Pierre-Louis (2012), Royset and Szechtman (2013)). Despite this relatively long history, their use in identifying near-optimal *decisions* for realistic instances has been limited. There are at least two hurdles to overcome here: 1) as mentioned earlier, the sample size requirements predicted by the current theory leads to relatively large approximation problems, and 2) while replications of sampled algorithms are important for lower variance estimates of the objective function f, the unfortunate reality of sampling-based optimization is that replications may introduce significant variability in decisions (see experience with SSN reported in Freimer et al. (2012)). One remedy to overcome variability is to use compromise decisions as described in Appendix II. The current section extends the result of Appendix II to a situation in which statistical optimality of a compromise decision can be verified up to an accuracy of δ as in (6). To accomplish this goal, we combine the algorithmic framework of Sen and Liu (2016) and the convergence rate results of SAA (Chapter 5, Shapiro et al. (2009)) to obtain a distribution-free estimate of the probability of optimality of the proposed decision. Thus our approach combines concepts from external sampling (SAA), as well as internal sampling (SD) within one common framework. In the interest of preparing a self-contained presentation, we provide brief summaries of SAA and SD in the Appendix II.

In the following, we impose the assumptions required for the asymptotic convergence of SD. Let $\nu = 1, ..., M$ to denote the index of replications, and for each ν, the SD algorithm is assumed to run for $K_\nu(\varepsilon)$ samples, to produce a terminal solution $\mathbf{x}^\nu(\varepsilon)$, and a terminal value f^ν_ε, where ε is the stopping tolerance used for each replication. From Appendix II, note that the grand-mean approximation $\bar{F}_M(x) := (1/M)\sum_{\nu=1}^M f^\nu(x)$, where $\{f^\nu\}_{\nu=1}^M$ denotes terminal value function approximations for each replication m. In addition, $\bar{\mathbf{x}} = (1/M)\sum_\nu x^\nu$, and the compromise decision $\mathbf{x}^c$ is defined by $\mathbf{x}^c \in \arg\min\{\bar{F}_M(x) + \frac{\bar{\rho}}{2}||x - \bar{\mathbf{x}}||^2 : x \in \mathbf{X}\}$, where $\bar{\rho}$ is the sample average of $\{\rho^\nu\}$, which denote the terminal proximal parameter for the ν^{th} replication.

Theorem 1. Assume $\mathbf{X}$ is non-empty, closed and convex, and the approximations f^ν are proper convex functions over $\mathbf{X}$. For $\delta = \bar{\rho}||\mathbf{x}^c - \bar{\mathbf{x}}||^2$, we have,

$$\frac{1}{M}\sum_{\nu=1}^{M} f^\nu_\varepsilon + \delta \geq \bar{F}_M(\mathbf{x}^c). \tag{8}$$

which implies $\mathbf{x}^c$ is δ-argmin to $\frac{1}{M}\sum_{\nu=1}^M f^\nu(\cdot)$, and the tolerance level satisfies $\delta = \bar{\rho}||\mathbf{x}^c - \bar{\mathbf{x}}||^2$.

Proof: See Appendix V.

If we define $\hat{S}_M(\delta) = \{x \in \mathbf{X} \mid \bar{F}_M(x) \leq \frac{1}{M}\sum_\nu f^\nu(\mathbf{x}^\nu) + \delta\}$, f^* the optimal value, and $S(\delta_u) = \{x \in \mathbf{X} \mid \bar{F}_M(x) \leq f^* + \delta_u\}$, then Theorem 1 has proved that $\mathbf{x}^c \in \hat{S}_M(\delta)$. Note that

$S(\delta_u)$ defines the solution set which is δ_u-optimal to the true optimal solution, and as a result, we should also analyze the relationship between $\mathbf{x}^c$ and $S(\delta_u)$.

Unless one restricts the model to using only Empirical Additive Errors (EAE), it is difficult for a user to prescribe a sample size for a stochastic programming model for reasons discussed earlier (also see computational experience reported in Sen and Liu (2016)). Hence we do not recommend this approach for cases where the distribution used is continuous or discrete with countably infinite number of outcomes. Instead, we use SD to suggest sample sizes, and discover the probability that a recommendation $\mathbf{x}^c \in S(\delta_u)$. The following theorem gives the probability bound of $\mathbf{x}^c \in S(\delta_u)$.

Theorem 2. Let $F(x,\tilde{\xi}) := c(x) + H(x,\tilde{\xi})$ denote the objective rv in (2) and (4). Suppose for each outcome ξ, $\kappa(\xi)$ satisfies $|F(x',\xi) - F(x,\xi)| \le \kappa(\xi)||x' - x||$. We define the Lipschitz constant of $\mathbb{E}_\xi[F(x,\tilde{\xi})]$ as $L = \mathbb{E}_\xi[\kappa(\tilde{\xi})]$. Suppose $\mathbf{X} \subseteq \mathbb{R}^n$ has a finite diameter D, M stands for the number of replications in solving the SP, N denotes the minimum of sample size of all the replications, and let the tolerance level $\delta_u > \delta$, with δ defined in Theorem 1. Then we have the following inequality:

$$Prob(\hat{S}_M(\delta) \subset S(\delta_u)) \ge 1 - \exp\left(- \frac{NM(\delta_u - \delta)^2}{32L^2D^2} + n\ln\left(\frac{8LD}{\delta_u - \delta}\right)\right). \tag{9}$$

Proof: See Appendix V.

Remark 2. To the best of our knowledge, the sample size formulas for SAA (Chapter 5, Shapiro et al. (2009)) are not intended for use to set up computational instances for solution. Instead, their primary role has been in showing that the growth of sample size for SAA depends logarithmically on the size of the feasible set and the reliability level $(1-\alpha)$. Our approach, seeking probabilistic bounds, allows us to estimate the reliability of a solution for a sampling-based algorithm. ■

We make two other observations in connection with the vision of statistical optimality: (i) due to replications, there is a natural affinity towards parallel algorithms, and (ii) it promotes the use of adaptive solution algorithms which can increase the sample size of any replication without having to restart the algorithmic process from scratch. The SD algorithm for SLP problems, and the convex SD algorithm introduced in the next subsection illustrate this strategy.

3.2. An Algorithm: Convex Stochastic Decomposition

Because the LEO approach may be required to solve models which use continuous distributions (or address very large data sets), the standard formulation of finite scenario SP can be very cumbersome. For instance, the SAA sample size formula is known to provide sample sizes which are far too conservative for algorithmic purposes. As a result we turn to a so-called internal sampling algorithm: the stochastic cutting plane (SCP) method. This was a precursor to SD and its presentation in (Higle and Sen (1996b)) shows its asymptotic convergence.

In the following we present an extension of SCP which includes a proximal mapping. As shown in Liu and Sen (2017), the proximal mapping in SD allows us to study convergence rates of such algorithms, and hence advantageous. However, our goal in this subsection is limited to showing that statistical optimality leads to finite convergence of cutting plane algorithms including the proximal point version, which we refer to as the convex SD algorithm. As in the standard SD method (Sen and Liu (2016) the piecewise linear approximations we maintain requires only finitely many cutting planes.

To simplify the notation for algorithmic purposes, we put $h_i(x) \equiv h(x, \xi_i)$ and assume that these recourse functions are non-negative. If non-negativity condition is not satisfied, then we require a lower bound $w < 0$ such that $h_i(x) \geq w$ for all $x \in \mathbf{X}$. In this case, we modify the optimization objective by $h_i(x) \leftarrow \{h_i(x) - w\}$, thus ensuring that the revised function h_i are all non-negative. Any way, we continue with the assumption that $w = 0$. As in the original SD method (for SLPs), the algorithm will work in an "online" manner, augmenting the sample by a small number of additional samples (say 1) at each iteration. Any iteration of the algorithm, denoted k, starts with an incumbent solution denoted $\hat{x}^k$. The method is as follows.

Initialization. Step 0. $k = 1$ and choose a starting incumbent $\hat{x}^1 \in X$, and set a value for $\rho_1 > 1$, and set $f_0(x) := c(x)$.

Preview of Step 1. For $k > 1$, assume $\hat{x}^{k-1}$ which is available from the previous iteration. In addition, assume that the "max" function shown below is also available.

$$f_{k-1}(x) := c(x) + \max_{j \in J_k} \frac{t_j}{k} \{\bar{\alpha}_j^{t_j} + (\bar{\beta}_j^{t_j})^\top x\}, \tag{10}$$

where J_k is the collection of subgradients in iteration $k > 1$, $t_j \leq k$ denotes the number of samples that were used to create the j^{th} subgradient, and $\bar{\beta}_j^{t_j} \in \partial[\frac{1}{t_j}(\sum_{i \leq t_j} h_i(x))]$ is a subgradient evaluated at points discovered in previous iterations. Each affine function will be referred to as a Sample Average Subgradient Approximation (SASA), and the notation $(\bar{\alpha}, \bar{\beta})$ reflects their sample averaging property. We use a factor $\frac{t_j}{k}$ in (10) to ensure that all previously generated affine functions are minorants for $\frac{1}{k}\sum_i^k h_i(x)$ for all x.

Step 1. Solve the proximal problem. Find a candidate solution which will challenge the incumbent by solving the following.

$$x^k = \arg\min\{f_{k-1}(x) + \rho_k ||x - \hat{x}^{k-1}||^2 \mid x \in \mathbf{X}\}, \tag{11}$$

Preview of Step 2. Once the candidate solution has been obtained, we will update the approximation of f_{k-1} by appending two new SASA functions. These affine approximations are computed (in Step 2 below) using the incumbent $\hat{x}^{k-1}$, and at a candidate x^k.

Step 2. Calculate two new SASA functions. For $u \in \{\hat{x}^{k-1}, x^k\}$, define $\beta_i(u) \in \partial h_i(u)$, and a SASA function with k data points is given by $\frac{1}{k}\sum_{i=1}^k h_i(u) + (\beta_i(u))^\top(x-u)$. Equivalently, put $\alpha_i(u) = h_i(u) - \beta_i(u)^\top u$, and define the new SASA functions as:

$$\bar{\alpha}(u) = \frac{1}{k}\sum_i \alpha_i(u), \quad \bar{\beta}(u) = \frac{1}{k}\sum_i \beta_i(u), \quad u \in \{\hat{x}^{k-1}, x^k\}. \tag{12}$$

Preview of Step 3. If there are $n_1 + 3$ constraints in f_{k-1}, then Caratheodory's theorem suggests that there are at least two SASA functions which need not be used to create the same search direction as obtained in step 1. In place of the deleted functions, we include the SASA functions corresponding to $u \in \{\hat{x}^{k-1}, x^k\}$.

Step 3. Determine an incumbent for the next iteration. Delete those SASA functions with zero Lagrange (weight) multipliers. Append the new SASA functions from step 2 for $u \in \{\hat{x}^{k-1}, x^k\}$ to obtain the new approximation denoted f_k. Choose the next incumbent by the following rule:

$$\hat{x}^k = \arg\min\{f_k(x^k), f_k(\hat{x}^{k-1})\}. \tag{13}$$

Update the parameter ρ_k by using either a "trust region"-type rule (Nocedal and Wright (1999)) (or an acceleration/decceleration rule) and return to step 1.

For the case in which the cost-to-go function h is defined by a linear program, the convex SD algorithm reduces to a standard SD algorithm for iterations beyond which no new dual extreme points are discovered (for the second stage LP). In this sense, asymptotic convergence of this algorithm can be proved by adhering to the assumptions imposed for the standard SD algorithm (see Appendix II, assumptions SD-a,b,d or see Higle and Sen (1994) and Sen and Liu (2016)). However the compactness of the set of outcomes will no longer be necessary because the reliability level required by (6) does not call for almost sure convergence. On the other hand, the boundedness of subgradients, which is implicit in the SLP setting (via the "relatively complete recourse" assumption), must be imposed explicitly.

Theorem 3. Let the assumptions SAA-a-d and SD-b-d as stated in Appendix II hold. In place of assumption SD-a, assume that $g(x,y)$ is convex in both variables. In case of LEO models with disjoint spaces (2)-(3), no additional assumptions for $m(z,\xi)$ are necessary (beyond those of section 2). In case of LEO models with shared spaces, i.e., (4)-(5), assume that $m(z,\xi)$ is a proper concave integrand. Finally, we assume that the set of all subgradients of the expected cost-to-go function are bounded (i.e., $\exists B > 0 | \bar{\beta} \in \partial \mathbb{E}[H(x,\xi)] \implies ||\bar{\beta}||^2 \leq B^2, \forall x \in X$). Then, the convex SD algorithm produces a γ optimal solution which satisfies (6) in finitely many iterations.

Proof: See Appendix V.

4. Model Validation, Assessment and Selection

The field of statistics, and more recently, Statistical Learning have developed notions of model selection on the basis of estimated errors for models which use empirical distributions. Because of their data driven emphasis, concepts such as model assessment and selection are important for LEO as well. The stochastic programming (SP) literature has some foundational results for assessing solution quality as proposed in Mak et al. (1999). Shapiro and Homem-de Mello (1998) and Higle and Sen (1996a). However, these tests are not proposed within the larger context of model validation and assessment. Because the LEO setup includes both statistical modeling as well as optimization, we have the potential for both model validation, assessment and selection. Note that any model validation schema should depend on measuring the specific response function we wish to optimize. In this paper, our optimization objective reflects a sample average criterion. In cases where the optimization objective is not the expectation (such as robust optimization, mean-risk optimization), the validation methods proposed in this paper may not be appropriate.

The protocol we adopt is one based on Figure 1b where validation is critical part of the modeling process. In section 4.1 we discuss metrics for any LEO model, and comparisons between alternative LEO models will be presented in section 4.2. These tests correspond to the hypothesis tests used in the lower diamond-shaped block of Figure 1b, and require a decision as an input.

4.1. Metrics and Model Validation

The following tests will be included for each alternative LEO model (indicated by an index q are mentioned in section 2).

- χ^2 test for error terms and cost-to-go objectives
- T-test for the mean of cost-to-go function
- F-test for the variance of cost-to-go function
- Tests to identify outliers.
- Prediction and confidence intervals such as those motivated by SL.

The first three tests are relatively standard in the statistical literature, but adapted to the LEO setup. In keeping with our goals of a self-contained presentation, we relegate the standard tests to Appendix III. Hence this section begins with the fourth bullet item.

Identifying Outliers. In stating the LEO model, the class of regressions $\mathcal{M}$ can be quite general. However, a model with Shared Spaces may call for a constrained regression where $\mathcal{M}$ may include bounds on predictions. For instance, in the LEO-Wyndor example, an unconstrained regression may lead to predictions which violate the bounds of the data-set. Unlike robust optimization where outliers may be critical to a decision model, our setting is more in line with regression model of statistics where the outliers can have a detrimental impact on the estimated conditional expectation. As in regression, where the focus is approximating a sample average prediction (using empirical residual minimization), data that are considered to be outliers should be removed. Similar considerations also hold for clustering algorithms (see Bertsimas and Shioda (2007)).

To identify outliers, it is important to choose the type of errors (additive scalar or multi-dimensional) to be used during cross-validation. Outliers from additive errors can be identified via Q-Q plots, whereas the case of multi-dimensional errors require greater computational effort. In cases such as MLR, the regression coefficients are allowed to be random variables, and hence lead to multi-dimensional error terms.

Outliers for an additive scalar model. Let $W_L = \min_i\{W_i : i \in T\}$ and $W_U = \max_i\{W_i : i \in T\}$. Once these bounds W_L, W_U have been computed, we identify those $i \in V$ as outliers by checking whether $m(Z_i, \xi) \in [W_L, W_U]$. Hence, data points with predictors outside the bounds ($[W_L, W_U]$) are considered to be outliers, and should be removed. Figure 4 shows the q-q plots for the error terms of the Training and Validation data sets of the LEO-Wyndor example before and after data preprocessing. We also compared the χ^2 test result of error sets before and after preprocessing. The detailed results of χ^2 test are included in section 5, where all computational results are presented.

An alternative way to identify outliers for the additive scalar SL model is to identify a $1-\alpha$ percent range where the parameter β_0 should belong. Choosing $\alpha = 0.05$, the acceptable range of β_0 is

$$\mathcal{E}_{M_0} = \left\{ \beta_0 \;\middle|\; \frac{(\beta_0 - \bar{\beta}_0)^2}{s^2_{\beta_0}} \leq \chi^2(\alpha) \right\},$$

where $\bar{\beta}_0$ denotes the mean value of the constant coefficient from MLR, and s_{β_0} is the standard deviation of the coefficient. We declare a data point (W_i, Z_i) to be an outlier if the following set is empty.

$$\begin{cases} (\bar{\beta}_0 + \xi_{i0}) + \bar{\beta}^\top Z_i = W_i \\ \bar{\beta}_0 + \xi_{i0} \in \mathcal{E}_{M_0} \end{cases}$$

The above test for uni-dimensional outliers is generalized to the multi-dimensional case below.

Outliers for a multi-dimensional model. Here we consider the case of statistical models in which the parameters are considered to be rvs. Given our current emphasis on MLR, we study the case of parameters which have multivariate normal distributions. For such statistical models, the Mahalanobis distance is known to provide a justifiable test for identifying outliers (Kulis et al. (2013)). The essence of our test revolves around identifying parameters β_i, which are expected to belong to a multi-dimensional ellipsoid

$$\mathcal{E}_M = \{\beta \mid (\beta - \bar{\beta})^\top \Sigma_\beta^{-1} (\beta - \bar{\beta}) \leq \chi^2_p(\alpha)\},$$

where $\bar{\beta}$ denotes the mean value of coefficients reported for MLR, Σ_β is the variance-covariance matrix associated with the coefficients, p denotes the degrees of freedom for the χ^2 distribution and $1-\alpha$ is the probability. If for a given data point (W_i, Z_i), the following system is infeasible, then we declare such a data point as an outlier.

$$\begin{cases} (\bar{\beta}_0 + \xi_{i0}) + (\bar{\beta} + \xi_i)^\top Z_i = W_i \\ (\bar{\beta}_0 + \xi_{i0}, \bar{\beta} + \xi_i) \in \mathcal{E}_M \end{cases}$$

These multi-dimensional feasibility problems are best solved in parallel using a convex optimization solver for each i.

Prediction Intervals. Under uncertainty, decision-makers (DM) not only seek a recommendation for a well-hedged decision, but they are interested in predictions of a variety of quantities associated with the decision. Thus, DMs may be interested in estimated expected costs, its confidence interval, and finally, as in statistical analysis, prediction interval of future costs. The prediction interval is a population-wide property which is intended to provide the DM a 95% range in which future costs might belong. Such estimates provide a clearer picture of the uncertain future that lies ahead. As in regression analysis, we recommend prediction intervals which represent a non-parametric interval estimate of the population value of the cost-to-go-function at a 95% level. Unlike confidence intervals (even prediction intervals in regression), non-parametric prediction intervals may not be symmetric (See Frey (2013)). As for the prediction interval, we use the shortest interval which covers 95% of the validated cost-to-go population. While this problem can be stated as an extension of a knapsack problem, we propose a somewhat tighter formulation (with additional valid inequalities) below.

Let $\{h_j\}_{j=1}^{|V|}$ denote the validated cost-to-go data points, where $|V|$ represents the sample size of validation dataset. We recommend that the dataset be sorted increasing order (i.e., $h_j \leq h_{j+1},\ \ j=1,...,|V|-1$). Let z_j denote a binary variable which assumes a value 1 if h_j is included in the prediction interval, and 0 otherwise. In addition, define a binary variable u_j (v_j) which assumes a value 1 if h_j is the smallest (largest) index to be included in the prediction interval, and 0 otherwise. The problem is formulated as follows.

$$
\begin{aligned}
\min\ & \sum_{j=1}^{|V|} h_j v_j - \sum_{j=1}^{|V|} h_j u_j \\
\text{s.t.}\ & \sum_{j=1}^{|V|} v_j = 1, \quad \sum_{j=1}^{|V|} u_j = 1, \quad \sum_{j=1}^{V} z_j \geq (1-\alpha)|V| \\
& v_j \geq z_{j-1} - z_j, \quad u_j \geq z_j - z_{j-1},\ j=1,...,|V| \\
& u_j \leq 1 - v_{j-t}, \quad t=0,...,j-1,\ j=1,...,|V| \\
& \sum_{t \leq j-1} u_t \geq v_j, \quad j=2,...,|V| \\
& z_t \leq 1-u_j, \quad z_\tau \leq 1 - v_j,\ t=0,...,j-1,\ \tau=j+1,...,|V|,\ j=1,...,|V|.
\end{aligned}
$$

The $1-\alpha$ prediction interval is $\left[\sum_{j=1}^{|V|} h_j u_j, \sum_{j=1}^{|V|} h_j v_j\right]$. In the interest of brevity, we leave the interpretation of the formulation to the reader.

4.2. Comparison across LEO Models

In this subsection, we discuss how alternative LEO models are assessed and which of these should be recommended as the most appropriate. In order to do so, we first estimate generalization error and optimization error. Finally, we include the Kruskal-Wallis test, which provides a sense of reliability of the estimates.

Generalization Error. This quantity is a prediction of out-of-sample cost-to-go error which may be observed when the system is implemented in practice. Let the in-sample cost-to-go error be approximated as

$$\mathrm{Err}_{in} \approx \frac{1}{|T|} \sum_{i=1}^{|T|} \mathbb{E}_{h^+}(h_i^+ - \hat{h}_i)^2, \tag{14}$$

where h_i^+ represents a new observation of the cost-to-go function, and $\hat{h}_i$ denotes a cost-to-go function value in the training dataset of sample size $|T|$. The approximate equality ($\approx$) in (14) is intended to convey that the right hand side is asymptotically equal to the left hand side as $|T|$ approaches infinity. In any event, the in-sample cost-to-go error estimates an average error between a new cost-to-go response and the training set cost-to-go.

Let h_i represent the validation cost-to-go objective, and the cost-to-go training error (err) is defined as

$$\mathrm{err} = \frac{1}{|T|} \sum_{i=1}^{|T|} (h_i - \hat{h}_i)^2. \tag{15}$$

Given (14) and (15), the generalization error is estimated by $\mathbb{E}_h(\mathrm{Err}_{in} - \mathrm{err})$. The following theorem suggests a mechanism to estimate generalization error.

Theorem 4. Assume that the expected value of new observations of the cost-to-go function ($\mathbb{E}_{h^+} h_i^+$) is equal to the expectation of the validated cost-to-go function ($\mathbb{E}_h h_i$), and let A1 and A2 hold. Then the generalization error is estimated by

$$\mathbb{E}_h(\mathrm{Err}_{in}) - \mathbb{E}_h(\mathrm{err}) \approx \frac{2}{|T|} \sum_{i=1}^{|T|} \mathrm{Cov}(h_i, \hat{h}_i) \tag{16}$$

Proof: See Appendix V.

Therefore, the covariance above is an estimate of the generalization error. Among alternative models, if we observe one with a larger covariance than another, then we may conclude that the one with a lower covariance has a lower generalization error.

As is common in statistical learning, one obtains better estimation of generalization error by using cross-validation (Hastie et al. (2011)). In one run of cross-validation, the data is partitioned randomly into two complementary subsets. To analyze the generalization error for a given decision, we calculate the covariance of cost-to-go objectives from these independent subsets. Multiple runs of cross-validation will be performed to sample a generalization error, and finally, we report the estimate of the generalization error as the average value over k runs.

Optimization Error. To choose an optimum from all decisions, we need to find a proper metric to compare the estimated objectives among different models. In this case, we propose to undertake the Kruskal-Wallis test (Kruskal and Wallis (1952)), which does not assume normality as a condition for the test. The null hypothesis of the Kruskal-Wallis test is that the ranked medians of bins (of samples from two competing models) are the same. When the hypothesis is rejected, the cost-to-go values of one method stochastically dominates the cost-to-go of the other method. Note that the chance of committing a type I error increases when comparing many pairs of models. To prevent the inflation of type I error rates, it is also possible to use multiple cost-to-go data sets in a manner which only test the hypothesis that all cost-to-go data sets belong to the same distribution. Such comparisons are used to determine whether there is a difference in the median among all the groups of estimated objectives.

Now, suppose we wish to identify the best model and its estimated objective value $\hat{f}^*$. This value can be obtained by choosing the best model identified via the Kruskal-Wallis test, which will be performed by the pairwise comparisons. Let Q denote the index set of alternative LEO models being compared. Then identifying the best model is accomplished by carrying out $\frac{|Q|(|Q|-1)}{2}$ hypothesis tests. Once $\hat{f}^*$ is identified by these tests, we calculate the optimization error by the difference $|\hat{f}_q - \hat{f}^*|$, where $\hat{f}_q$ denotes the estimated cost provided by model q.

5. Illustrative Computations

We now turn to the LEO-`ELECEQUIP` and the LEO-Wyndor illustrations. All computations reported below are carried out by the original SD algorithm because the models are SLPs.

5.1. LEO-`ELECEQUIP`

The specific model we solve is given in Appendix I. In this example, we use $c_u = 1, c_v = 3$ and $U_t = R_t = \infty$.

(a) Deterministic ARIMA Forcasting (DAF). Since U_t and R_t are infinity, we can use the predicted demand to define the order quantity as: $\Delta_t = \text{Max}\{0, \hat{D}_t - u_t\}$, where $\hat{D}_t$ is the expected value of the ARIMA model. This is a case of using $\hat{m}$ in section 2.

(b) Stochastic Linear Programming (SLP), which gives the decision by solving the instance in Appendix I (equation (17)). Note that our rolling horizon approach solves three period problems (0,1,2), and we use the solution of period 0 as our current decision, and drop the other decisions. We then use the demand of the following period, update the inventory status, and move the clock forward to the next period. This is a case of $m(z,\xi) = \hat{m}(z) + \xi$, where ξ is an outcome of $\tilde{\xi}$, the normal error from ARIMA.

5.1.1. Month by Month Validation Results for 2001-2002. The ARIMA model was trained on data from 1996-2000, and the performance of the models were validated during the two year period 2001-2002. Table 1 presents costs for the year 2001 and 2002 (24 months) for each of the two inventory policies DAF and SLP specified in (a) and (b) above. Note that of the 24 runs (simulating two years of inventory), the LEO approach cost higher only for month 1. Thereafter, it cost less in each subsequent month, with some (months) reducing costs by over 66%. The average inventory cost reduction over the deterministic ARIMA forecast is approximately 34% over the 2 year run.

5.1.2. Snapshot Statistical Comparisons. For practical decision-making the back-testing exercise performed above is more convincing, than any snapshot study of a dynamic process. Nevertheless, we present a snapshot of this inventory model in the interest of parsimony of presentation for both examples of this paper. However, we relegate these computational results to Appendix IV.

5.2. LEO-Wyndor

We now present the LEO-Wyndor problem under alternative models. DF/LP represents learning enabled optimization using deterministic forecasts, in which we use the expected value of the linear regression as the demand model. This results in a deterministic LP. In addition, we also study other models where linear regression suggests alternative parameters: a) the additive scalar error model, using the empirical additive errors (EAE) and deterministic model coefficients $\beta_0, \beta_1, \beta_2$ where the first is the constant term, the second is the coefficient for TV expenditures, and the third is the coefficient for radio expenditures; b) a linear regression whose coefficients are random variables $\{\tilde{\beta}_j\}$, which are normally

Month	1	2	3	4	5	6
DAF Cost	12.33	14.41	39.02	14.54	26.44	28.86
SLP Cost	16.53	3.06	12.28	9.49	20.63	17.77
Month	7	8	9	10	11	12
DAF Cost	7.65	37.99	25.26	38.46	16.92	30.34
SLP Cost	7.38	31.27	14.82	28.66	13.23	21.92
Month	13	14	15	16	17	18
DAF Cost	11.35	3.05	15.11	26.74	15.67	38.98
SLP Cost	6.04	1.11	11.06	15.78	14.22	24.56
Month	19	20	21	22	23	24
DAF Cost	33.23	23.81	17.90	16.62	15.31	29.72
SLP Cost	11.90	19.88	5.13	8.66	9.05	23.20

Table 1 LEO-ELECQUIP: Monthly Back-Testing Costs

distributed and uncorrelated (NDU); c) a linear regression whose coefficients are random variables $\{\tilde{\beta}_j\}$ which are normally distributed and correlated (NDC). We reiterate that all three models EAE, NDU, NDC correspond to specific types of errors (which are indexed by q in the presentation in section 2). Note that for models NDU and NDC, we have continuous rvs, and as a result we adopted SD as the solution methodology because it manages discrete and continuous random variables with equal ease. We refer to these results by NDU/SD and NDC/SD. Also note that for the case of EAE, the dataset is finite and reasonably manageable. Hence we will use both SAA and SD for this model, and refer to them by EAE/SAA and EAE/SD.

5.2.1. Results for Error Terms. The calculations begin with the first test as the top diamond block in Figure 1b. Table 2 shows p-values and test results of χ^2 test for NDU/SD, NDC/SD and EAE. From values reported in Table 2, the fit appears to improve when a few of the data points near the boundary are eliminated. (see Figure 4).

5.2.2. Results for Decisions and Optimal Value Estimates. The decisions and various metrics discussed earlier are shown in Table 3. Although the prediction interval is listed in a symmetric way, the actual data points are asymmetric with respect to the estimated mean. The last two rows report the probability γ and the corresponding tolerance level

	NDU/SD	NDC/SD	EAE
Before Data Preprocessing	0.44, not rejected	0.42, not rejected	0.45, not rejected
After Data Preprocessing	0.59, not rejected	0.57, not rejected	0.78, not rejected

Table 2 LEO-Wyndor: Comparison of Chi-square test

δ, which are provided by SD algorithm based on the theorems in section 3. We choose 1% of the mean value of validated objective to be δ_u in Theorem 2. Once again, notice that for both DF/LP and EAE/SAA, we do not report any probability because we use a deterministic solver as in (4).

The hypothesis test results for the cost-to-go objectives (the lowest diamond in Figure 1b) for each model are reported in Table 4. The T-test rejects the DF/LP model, where the others were not. The next two rows give the test results of variance based on F-statistic, and we conclude that none of the models can be rejected. We also performed a χ^2 test for the cost-to-go objectives using the training and validation sets. Again, the DF/LP model was rejected where as the others were not.

Remark 3. The concept of cross-validation (k-fold) is uncommon in the stochastic programming literature. With $k > 1$, this tool is a computational embodiment of (14), and provides a prediction of the error. Without such cross-validation, it is often likely that model assessment can go awry. For instance, in this example we have observed that if we use $k = 1$, then the EAE/SAA model can get rejected although using $k = 5$, the EAE/SAA is no longer rejected. This can be attributed to the fact that variance reduction due to $k = 5$-fold cross-validation reduces Type I error (when compared with $k = 1$). ■

Table 5 reports the optimization error, as well as the generalization error for all models. DF/LP shows the largest optimization error, which indicates that it is not an appropriate model to recommend for this application. On the other hand, NDU/SD and NDC/SD have comparable and relatively small generalization errors. However the optimization errors appear to be significant, therefore NDU/SD and NDC/SD do not yield the most profitable decisions. Had the criterion been simply the generalization error (as in SL), we might have chosen non-profit-maximizing decisions.

In Table 6 we present the pairwise comparison of Kruskal-Wallis test. For the tests of DF/LP with other methodologies, the p-values are all smaller than 0.01, which implies that there are significant differences between the median ranks of DF/LP and each of the other four approaches. The stepped curve in Figure 2 illustrates the ordering discovered by the Kruskal-Wallis test. Note that DF/LP shows significant difference from the other approaches. Moreover, the curves for NDU/SD and NDC/SD are relatively close, whereas EAE/SAA and EAE/SD are indistinguishable. These similarities were quantified in Table 6 by the fact that the p-values for these comparisons are greater than 0.01. Finally, EAE/SAA and EAE/SD give the largest objective value, which is also reported in Table 3. LEO-Wyndor example is a profit maximization problem, therefore EAE/SAA and EAE/SD lead to more profitable decisions since they stochastically dominate the others. The Kruskal-Wallis test suggests that the difference of EAE/SAA and EAE/SD is not significant, therefore both EAE/SAA and EAE/SD provide the most profitable decision (see Table 6 for the validated objective value estimated for EAE/SAA and EAE/SD).

6. Future Directions and Conclusions

In this paper, we introduced a fusion of concepts from SL and SP which is made possible by the manner in which we allow SL models to be imported as a model of uncertainty for decisions using SP. The interplay between SL and SP models in the latter setting may be used in a variety of ways, although we have only explored a couple of possibilities in this paper. One can easily envision other ways to integrate these types of models, for example, by treating SL and SP as agents in a game. Another possibility is to introduce interactions between SL and SP by incorporating search directions from optimization into directions for use within projection pursuit. There are many other research questions which are completely open, such as, coupled dynamic SL and SP models, including various combinations of filtering and optimization.

To conclude the paper, let us revisit the four questions posed in the introduction.

- We discussed two classes of LEO models, and note that the one with disjoint spaces can allow very general SL models (e.g., ARIMA), while those with shared spaces call for more nuanced usage. Our illustrations also demonstrate the power of using stochastic functions for forecasting in SP over deterministic forecasts in LP.

Models	DF/LP	NDU/SD	NDC/SD	EAE/SAA	EAE/SD
x_1	173.48	181.70	181.40	191.27	191.40
x_2	26.52	18.30	18.60	8.73	8.60
Estimated Obj.	\$41,391	\$41,580	\$41,492	\$42,009	\$42,045
Validated Obj. 95% C.I.	\$39,869(±692)	\$41,903 (±335)	\$41,865 (±302)	\$42,269 (±522)	\$42,274 (±493)
Validated Obj. 95% P.I.	\$39,856(±1,302)	\$41,911 (±632)	\$41,841 (±639)	\$42,258 (±1,012)	\$42,279 (±973)
Probability (γ)		0.9633	0.9698		0.9872
Tolerance (δ)		760	694		842

Table 3 LEO-Wyndor: Comparison of Solutions for Alternative Models

Models	DF/LP	NDU/SD	NDC/SD	EAE/SAA	EAE/SD
T-statistics($t < 1.96$)	$t = 2.18$	$t = 0.72$	$t = 0.84$	$t = 0.62$	$t = 0.49$
Cost-to-go Test(Mean)	rejected	not rejected	not rejected	not rejected	not rejected
F-statistics($0.67 < f < 1.49$)	$f = 1.23$	$f = 1.43$	$f = 1.29$	$f = 0.79$	$f = 1.16$
Cost-to-go Test(Variance)	not rejected	not rejected	not rejected	not rejected	not rejected
χ^2 Test p-value ($p > 0.05$)	$p = 0.038$	$p = 0.34$	$p = 0.32$	$p = 0.42$	$p = 0.42$
Cost-to-go Test(Distribution)	rejected	not rejected	not rejected	not rejected	not rejected

Table 4 LEO-Wyndor: Hypothesis Test Results for Alternative Models

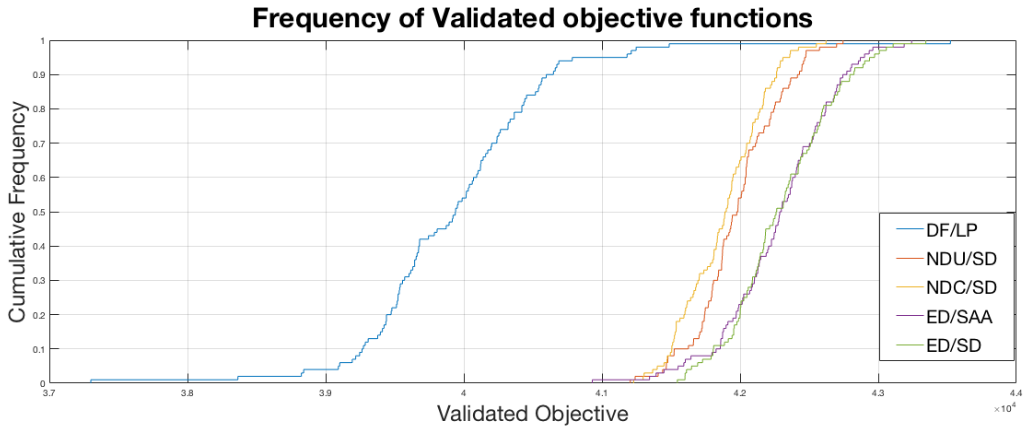

Figure 2 LEO-Wyndor: Stochastic Dominance of Validated Objectives under Alternative Models

Models	DF/LP	NDU/SD	NDC/SD	EAE/SAA	EAE/SD
Optimization Error	2405	371	409	5	
Generalization Error	29.751	19.406	19.554	21.889	21.326

Table 5 LEO-Wyndor: Errors for Alternative Models

Models	EAE/SD	EAE/SAA	NDC/SD	NDU/SD
DF/LP	2.76×10^{-8}	1.34×10^{-7}	1.12×10^{-7}	5.60×10^{-7}
NDU/SD	8.46×10^{-7}	6.2×10^{-3}	0.37	
NDC/SD	2.05×10^{-7}	1.72×10^{-3}		
EAE/SAA	5.87×10^{-2}			

Table 6 LEO-Wyndor: Kruskal-Wallis Test Results ($p > 0.01$)

- We offered a new notion of statistical optimality which seeks decisions with high probability of being within a given tolerance of the optimal value. The new optimality criterion also motivated a new convex SD algorithm. To the best of our knowledge, this is the first effort to report such statistically estimated decisions, with an acceptable probability of ($\delta-$optimality), and we are able to report those probabilities.
- To give the reader a practical sense of the use of samples in our approach versus those of SAA, we estimated the sample sizes which would be necessary according to (21). For the NDC model in the LEO-Wyndor example, the sample size for SAA is $K = 20,182$.

On the other hand, using a compromise decision of SD requires a total of 15,990 samples over 30 runs in parallel. In case of such embarrasingly parallel runs, it is safe to say that the estimated completion time of SD is governed by the run with the largest sample size, which in this instance was 793. If computation time is roughly proportional to the number of samples, then the *reduction* due to SD and parallelization *is 96% percent* of the computational time of SAA with a sample size of 20,182.

- We also introduced a model validation protocol for SP based on concepts of cross-validation in SL. In order to make this protocol realizable, we carried out statistical tests such as the Kruskal-Wallis test to identify which of the models provided the most desirable decision.

The novelty of these contributions is self-evident, not only from a conceptual (or theoretical) point of view, but also from modeling and computational perspectives. Using examples from the OR/MS discipline, we have shown how these ideas provide decision support which combines both statistical learning as well as stochastic programming. While our examples are drawn from linear models[2], the contributions summarized above have the potential to change the future of OR/MS modeling, especially for data-intensive and time-critical decision models. Such models are likely to include streaming and/or spatio-temporal data, and a variety of different classes of decision models (e.g., nonlinear, mixed-integer, dynamic and others). These will be particularly important to accommodate more general error distributions which do not satisfy homoscedasticity. We are hopeful that concepts such as statistical optimality, model validation, assessment and selection will be an integral part of the next generation of stochastic programming models and work-flow.

[2] combining multiple linear regression and stochastic linear programming

Appendix I: Details of Examples of Some LEO Models

The instances discussed below are developed using existing data-sets and existing optimization models. As with the rest of the paper, the novelty here is in the fusion of learning data-sets and optimization models. We include one example for each type of a LEO structure: Disjoint Spaces and Shared Spaces. Since the data-sets are not new, we append the acronym LEO to the names of the existing data-sets. Each model requires two aspects: one is the SL aspect, and the other is the decision/optimization aspect. In case of the SL part of the model, we assume that measures that are necessary to create acceptable SL models are undertaken. For the models given in this appendix, the time series setting (LEO-`ELECEQUIP` Inventory model) calls for stationarity tests, whereas in case of cross-sectional data (LEO-Wyndor model), outliers in the data should be identified and removed.

I.1. A Model with Disjoint Spaces: LEO-`ELECEQUIP` (Time-Series Data)

This model is based on a "rolling-horizon" decision model commonly used in inventory control. Before starting a sequence of decisions, one typically analyzes historical demand. In this particular example, we use a commonly available data set referred to as `ELECEQUIP` which provides demand of electrical equipment over a ten-year period. We will use the first five years to discover the time series pattern of demand, and then, use it within a rolling horizon inventory model. We conduct the validation exercise for two years, 2001-2002.

When fitting an ARIMA model to the training data, one identifies (p,d,q) and $(P,D,Q)_\tau$ where τ represents the seasonal backshift, and (p,d,q) specify the AR(p), I(d) and MA(q) components of ARIMA. In order to ascertain whether a series is stationary, it is customary to create an autocorrelation function (ACF) plot, and then one examines whether such a plot drops off to zero relatively fast. If so, we venture to guess that the data are stationary. While checking for correlation is not a substitute for independence, the lack of correlation can be considered as a surrogate.

<u>Model Details</u>: Without loss of generality, we can view the decision epoch as $j=0$, and the most recent demand will be denoted $d_0 = D_{j+1}$ (from the validation data set). The beginning inventory y_0 and and ending inventory u_0 are also available. The inventory model will look ahead into periods $t=0,\ldots,T,$, although as in rolling horizon schemes, only Δ_0 will be implemented. The model will be a two-stage multi-period stochastic program with the first stage making ordering decisions $x=(\Delta_0,\ldots,\Delta_{T-1})$, and the second stage predicting the potential cost of the choice Δ_0. As the decision clock moves forward in time,

the total cost of inventory management becomes estimated by this process. The various relationships in the inventory model are summarized below.

- Because of the delivery capacity U_t, we must have $0 \leq \Delta_t \leq U_t$.
- We will sample demand realizations in period t using a forecast $D_t(\xi)$ (using the time series model created in the training phase). Here the notation ξ denotes one sample path of demands over the periods $t = 1, \ldots, T-1$. The notation d_0 (in lower case) denotes a deterministic quantity, whereas, the notation $D_t(\xi)$ denotes the outcome ξ of the demand (stochastic process) observed in period t. The planning horizon used in the linear program will be $T = 3$, which requires us to simulate two periods of demand.
- We assume that y_0 and d_0 are given. Let $u_t(\xi)$ denote the ending inventory in period t, and $y_{t+1}(\xi)$ denote the beginning inventory in period $t+1$. We have $y_{t+1}(\xi) = u_t(\xi) + \Delta_t$, and a storage (refrigerator) capacity constraint requires that $y_{t+1}(\xi) \leq R_{t+1}$, where the latter quantity is given. Then the ending inventory of period t, denoted $u_t(\xi)$, must obey the relationship $u_t(\xi) = \text{Max}\{0, y_t(\xi) - D_t(\xi)\}$. The unit cost of holding inventory is c_u, where $c_u \geq 0$. The total inventory holding cost for period t is then $c_u u_t(\xi)$.
- Let $v_t(\xi)$ denote the lost sales in period t, so that $v_t(\xi) = \text{Max}\{0, D_t(\xi) - y_t(\xi)\}$. Suppose that the per unit cost of lost sales in period t is c_v, where $c_v \geq 0$. Then the total cost of lost sales for period t is $c_v v_t(\xi)$, and the first stage cost is zero. Recalling that $x = (\Delta_0, \ldots, \Delta_{T-1})$, the cost-to-go function is defined as follows.

$$\text{Min}_{0 \leq \Delta_t \leq U_t}\ f(x) = \mathbb{E}_{\tilde{\xi}}\Big[h(x,\xi) = \sum_{t=0}^{T-1} c_u u_t(\tilde{\xi}) + c_v v_t(\tilde{\xi})\Big] \tag{17a}$$

$$\begin{aligned}
\text{s.t.} \quad y_{t+1}(\xi) - u_t(\xi) &= \Delta_t \quad \text{for almost all } \xi && \text{(17b)}\\
y_{t+1}(\xi) &\leq R_{t+1} \quad \text{for almost all } \xi && \text{(17c)}\\
-y_t(\xi) + u_t(\xi) &\geq -D_t(\xi) \quad \text{for almost all } \xi && \text{(17d)}\\
y_t(\xi) + v_t(\xi) &\geq D_t(\xi) \quad \text{for almost all } \xi && \text{(17e)}\\
u_t(\xi), v_t(\xi) &\geq \mathbf{0} && \text{(17f)}
\end{aligned}$$

Note that constraints in (17) should be imposed for all possible errors in the training set. However, not all error combinations are sampled, and as result, we say that the constraints must hold for a large enough sample size (which is what we mean by the phrase "almost all" ξ). It suffices to say that the sample size used in optimization is decided during the

Stochastic Decomposition (SD) algorithmic process. The computational results for T=2 are presented in the body of this paper.

I.2. A Model with Shared Spaces: LEO-Wyndor (Cross-Sectional Data for Production - Marketing Coordination)

We study a "textbook"-ish example which has been amalgamated from two textbooks: one on Operations Research (Hillier and Lieberman (2012)) and another on Statistical Learning (James et al. (2013)). Consider a well known pedagogical product-mix model under the banner of "The Wyndor Glass Co." In this example, Hillier and Lieberman (2012) address resource utilization questions arising in the production of high quality glass doors: some with aluminum frames (A), and others with wood frames (B). These doors are produced by using resources available in three plants, named 1, 2, and 3. The data associated with this problem is shown in Table 7. The product mix will not only be decided

Plant	Prod. time for A (Hours/Batch)	Prod. time for B (Hours/Batch)	Total Hours Available
1	1	0	4
2	0	2	12
3	3	2	18
Profit per Batch	\$3,000	\$5,000	

Table 7 Data for the Wyndor Glass Problem (Hillier and Lieberman (2012))

by production capacity, but also the potential of future sales. Sales information, however, is uncertain and depends on the marketing strategy to be adopted. Given 200 advertising time slots, the marketing strategy involves choosing a mix of advertising outlets through which to reach consumers. Exercising some "artistic license" here, we suggest that the advertising data set in James et al. (2013) reflects sales resulting from an advertising campaign undertaken by Wyndor Glass. That is, the company advertises both types of doors through one campaign which uses two different media, namely, TV and radio[3]. Note that in the original data set advertising strategy is represented as budgeted dollars, whereas we have revised it to represent advertising time slots. Thus in our statistical model,

[3] The actual data set discussed in James et al. (2013) also includes newspapers. However we have dropped it here to keep the example very simple.

sales predictions are based on number of TV and radio advertising time slots[4]. In our interpretation, product-sales reflect total number of doors sold ($\{W_i\}$) when advertising time slots for TV is $Z_{i,1}$ and that for radio is $Z_{i,2}$, again as number of advertising time slots. (This data set has 200 data points, that is, $i = 1, \ldots, 200$). For the SP side, x_1 denotes the number of TV advertising time slots, and x_2 denotes the number of radio advertising time slots.

	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

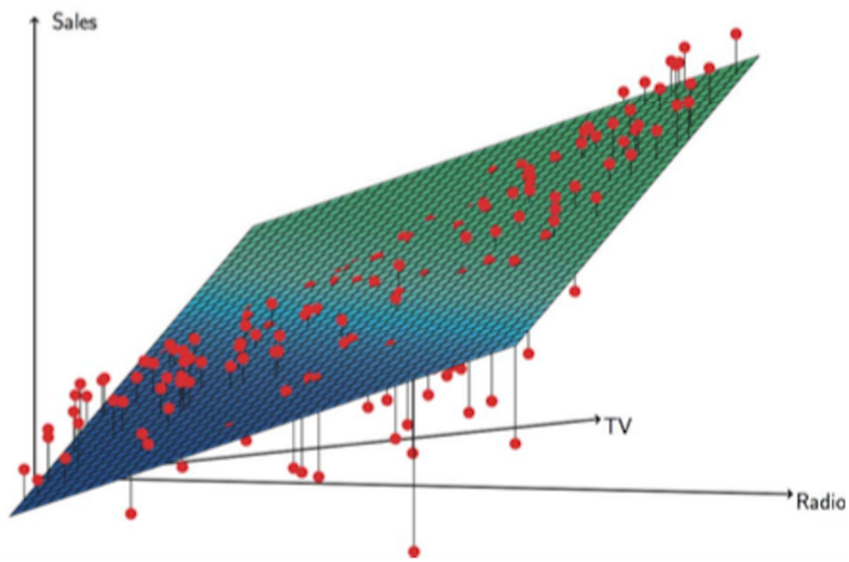

Figure 3 **The Advertising Data Set (Source: James et al 2011).**

The linear regression model for sales is shown in Figure 3, and will be represented by $\hat{m}(x)$. We consider the following statistical models reported in Section 5.

Data Preprocessing

1. (DF) For deterministic forecasts (DF) we simply use the sales given by $\hat{m}_1(z) = \bar{\beta}_0 + \bar{\beta}_1 z_1 + \bar{\beta}_2 z_2$. Thus, for deterministic predictions, we do not account for errors in forecast, whereas the remaining cases below use some form of error distributions.

2. (NDU) One approximation of the sales forecast is one where the correlation between the coefficients are ignored, and the resulting model takes the form $m_2(z, \xi) = (\bar{\beta}_0 + \xi_0) + (\bar{\beta}_1 + \xi_1) z_1 + (\bar{\beta}_2 + \xi_2) z_2$, where (ξ_0, ξ_1, ξ_2) are normally distributed and uncorrelated with mean zero, and the standard deviations are computed within MLR.

[4] The numbers used are the same as in James et al. (2013)

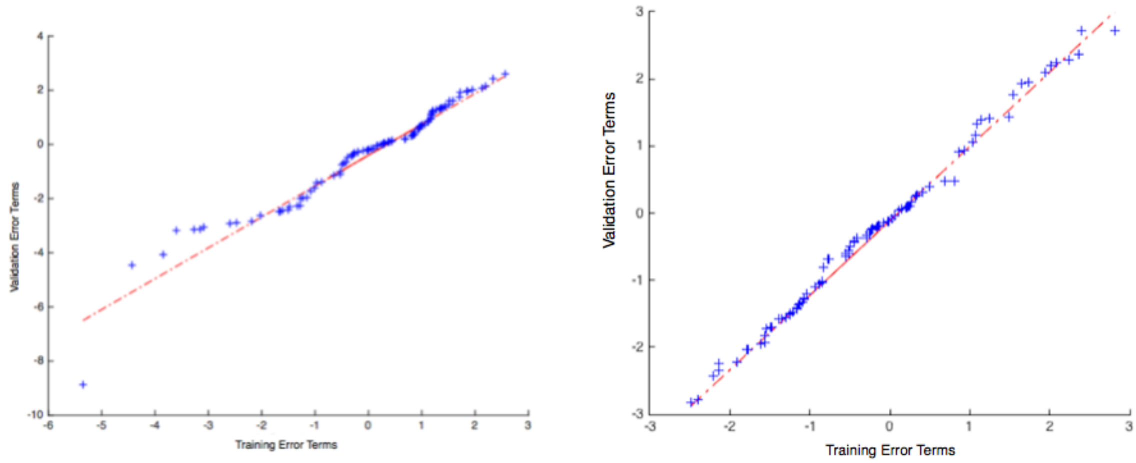

Figure 4 **q-q plot before and after data preprocessing**

3. (NDC) Another approximation of the sales forecast is $m_3(z,\xi) = (\bar{\beta}_0 + \xi_0) + (\bar{\beta}_1 + \xi_1)z_1 + (\bar{\beta}_2 + \xi_2)z_2$, where (ξ_0, ξ_1, ξ_2) are normally distributed and correlated according to the variance-covariance matrix reported by MLR.

4. (EAE) This is the empirical additive error model, where $m_4(z,\xi) = \bar{\beta}_0 + \bar{\beta}_1 z_1 + \bar{\beta}_2 z_2 + \xi_0$, and ξ_0 denotes a rvs whose outcomes are $W_i - \hat{m}_4(Z_i)$. We refer to this model as empirical additive errors.

As in the set up for (4), the index q refers to the alternative error models (DF, NDU, NDC and EAE). The corresponding first stage is given as follows:

The formulation presented below mimics (4), and since all decisions variables x share the same space as the rv Z, we explicitly remind the reader that $Z = z = x$.

Index Sets and Variables

$i \equiv$ index of product, $i \in \{A, B\}$.

$y_i \equiv$ number of batches of product i produced.

$$f(x) = -0.1x_1 - 0.5x_2 + \mathbb{E}[h(x, \tilde{\xi}_q \mid Z = z = x)] \tag{18a}$$

$$\text{s.t.} \quad x_1 + x_2 \leq 200 \tag{18b}$$

$$x_1 - 0.5x_2 \geq 0 \tag{18c}$$

$$L_1 \leq x_1 \leq U_1, L_2 \leq x_2 \leq U_2 \tag{18d}$$

$$h(x, \xi_q \mid Z = z = x) \quad = \quad \text{Max} \quad 3y_A + 5y_B \tag{19a}$$

$$\text{s.t.} \quad y_A \leq 4 \tag{19b}$$

$$2y_B \leq 12 \tag{19c}$$

$$3y_A + 2y_B \leq 18 \tag{19d}$$

$$y_A + y_B \leq m_q(z, \xi_q) \tag{19e}$$

$$y_A, y_B \geq 0 \tag{19f}$$

Note that the choice of ranges $[L_1, U_1]$ and $[L_2, U_2]$ are chosen so that assumption A2 is satisfied. Note that this instance is stated as a "maximization" model, whereas, our previous discussions were set in the context of "minimization". When interpreting the results, it helps to keep this distinction in mind. The LEO models presented above are relatively general, allowing very general regression models such as kernel-based methods, projection pursuit, and others. However, our current computational infrastructure is limited to stochastic linear programming (SLP) and as a result the regression used for models with Shared Spaces will be restricted to MLR.

Appendix II: Stochastic Programming Background - Sample Average Approximation (SAA) and Stochastic Decomposition (SD)

Sample Average Approximation(SAA)

Sample Average Approximation is a standard sampling-based SP methodology, which involves replacing the expectation in the objective function by a sample average function of a finite number of data points. Suppose we have sample size of K, an SAA example is as follows:

$$\min_{x\in X} F_K(x) = c^\top x + \frac{1}{K}\sum_{i=1}^{K} h(x,\xi^i). \tag{20}$$

As an overview, the SAA process may be summarized as follows.

1. Choose a sample size K, and sample K outcomes from the training data-set.
2. (Optimization Step). Create the approximation function $F_K(x)$, and solve an SAA instance (20).
3. (Validation Step). Take the decision from $F_K(x)$, follow the steps in section 4, estimate the validated confidence interval, generalization error and optimization error.
4. If the estimated objective does not agree with validated confidence interval, or generalization error and optimization error are not acceptable, increase the sample size K and repeat from step 1.

Assumption SAA-a. The expectation function $f(x)$ remains finite and well defined for all $x \in \mathbf{X}$. For $\delta > 0$ we denote by

$$S(\delta) := \{x \in \mathbf{X} : f(x) \le f^* + \delta\} \quad \textit{and} \quad \hat{S}_K(\delta) := \{x \in \mathbf{X} : \hat{f}_K(x) \le \hat{f}^*_K + \delta\},$$

where f^* denotes the true optimal objective, and $\hat{f}^*_K$ denotes the optimal objective to the SAA problem with sample size K.

Assumption SAA-b. There exists a function $\kappa : \Xi \to \mathbb{R}_+$ such that its moment-generating function $M_\kappa(t)$ is finite valued for all t in a neighborhood of zero and

$$|F(x',\xi) - F(x,\xi)| \le \kappa(\xi)||x' - x||$$

for a.e. $\xi \in \Xi$ and all $x', x \in \mathbf{X}$.

Assumption SAA-c. There exists constant $\lambda > 0$ such that for any $x', x \in \mathbf{X}$ the moment-generating function $M_{x',x}(t)$ of rv $[F(x',\xi) - f(x')] - [F(x,\xi) - f(x)]$, satisfies

$$M_{x',x}(t) \le exp(\lambda^2||x' - x||^2 t^2/2), \forall t \in \mathbb{R}.$$

From assumption SAA-b, $\Big|[F(x',\xi) - f(x')] - [F(x,\xi) - f(x)]\Big| \leq 2L||x' - x||$ *w.p. 1, and* $\lambda = 2L$.

PROPOSITION 1. Suppose that assumptions SAA(a-c) hold, the feasible set $\mathbf{X}$ has a finite diameter D, and let $\delta_u > 0, \delta \in [0, \delta_u), \varepsilon \in (0,1)$, and $L = \mathbb{E}[\kappa(\xi)]$. Then for the sample size K satisfying

$$K(\varepsilon,\delta) \geq \frac{8\lambda^2 D^2}{(\delta_u - \delta)^2}\left[n \ln\left(\frac{8LD}{\delta_u - \delta}\right) + \ln\left(\frac{1}{1-\varepsilon}\right)\right], \tag{21}$$

we have

$$Prob(\hat{S}_K(\delta) \subset S(\delta_u)) \geq \varepsilon.$$

Proof: This is Corollary 5.19 of Shapiro et al. (2009) with the assumption that the sample size K is larger than that required by large deviations theory (see 5.122 of Shapiro et al. (2009)). ■

Stochastic Decomposition (SD)

Unlike SAA which separates sampling from optimization, SD is based on sequential sampling and the method discovers sample sizes "on-the-fly" (Higle and Sen (1991),Higle and Sen (1994)). Because of sampling, any stochastic algorithm must contend with both variance reduction in objective values as well as solutions. SD uses M independent replications of value functions denoted f^ν, $\nu = 1, \dots, M$. Each of these functions is a max-function whose affine pieces represent some Sample Average Subgradient Approximations (SASA). Because SD uses proximal point iterations, Higle and Sen (1994) shows that the maximum number of affine pieces is $n_1 + 3$, where n_1 is the number of first stage decisions, and these pieces are automatically identified using positive Lagrange multiplier estimates during the iterations. In theory, one can control the number of affine pieces to be smaller, but that can also be chosen "on-the-fly" depending on the size of the first stage decision variables n_1. When the number of affine functions reduces to only 1, the SD method reduces to a proximal stochastic variance reduction method (prox-SVRG) (Xiao and Zhang (2014)). Among other strengths such as parallelizability and variance reduction, one of the main usability issues which SD overcomes is that when a model has a large number of random variables (as in the case of multi-dimensional random variables) it does not require users to choose a sample size because the sequential process automatically discovers an appropriate sample. Together with the notion of Statistical Optimality as set forth in section 3, SD provides a very convenient optimization tool for LEO models.

For SLP models, Sen and Liu (2016) have already presented significant computational evidence of the advantage of SD over plain SAA. The reduced computational effort also facilitates replications for variance reduction (VR). VR in SD is achieved by creating the so-called compromise decision, denoted $\mathbf{x}^c$, which minimizes a grand-mean approximation $\bar{F}_M(x) := \frac{1}{M}\sum_{\nu=1}^{M} f^\nu(x)$, where $\{f^\nu\}_{\nu=1}^{M}$ denotes a terminal value function approximation for each replication m. Suppose that solutions $x^\nu(\varepsilon) \in (\varepsilon - \arg\min \ \{f^\nu(x) \mid x \in \mathbf{X}\})$ and $\mathbf{x}^c(\delta) \in (\delta - \arg\min\{\bar{F}_M(x) \mid x \in \mathbf{X}\})$. Then, Sen and Liu (2016) has proved consistency in the sense that $\lim_{\delta\to 0} \Pr(\bar{F}_M(\mathbf{x}^c(\delta)) - f^*) \to 0$. Here are the critical *assumptions* of SD (Higle and Sen (1996b)).

Assumption SD-a. The objective functions in the first and second stage models are either linear or quadratic, and all constraints are linear. Moreover, the set of first stage solutions is compact.

Assumption SD-b. The second stage optimization problem is feasible, and possesses a finite optimal value for all $x \in X$, and outcomes ξ (i.e., the relatively complete recourse assumption holds).

Assumption SD-c. The second stage constraint functions are deterministic (i.e., fixed recourse), although the right hand side can be governed by random variables. The set of outcomes of the random variables is compact.

Assumption SD-d. The recourse function h is non-negative. So long as a lower bound on the optimal value is known, we can relax this assumption. (Higle and Sen (1996b))

A high-level structure of SD algorithm can be summarized as follows (Sen and Liu (2016)). For greater specifics regarding two-stage SD, we refer to Higle and Sen (1991), Higle and Sen (1994), and Higle and Sen (1996b), and for the multi-stage version we refer to Sen and Zhou (2014).

1. (Initialize). Let ν represent the number of completed replications, and set $\nu = 0$.
2. (Out-of-Sample loop). Set the number of completed replications $\nu = 0$. Increment ν at each time and start the next replication.
3. (In-Sample loop). Add one sampled outcome to the available samples and update the empirical frequencies.

4. (Updated Value Function Approximation). Using the new outcome from step 3 and previously generated approximations, update the new value function approximation $f_k^\nu(x)$.

5. (Optimization Step). Solve the regularization of $f_k^\nu(x)$ in step 4, and update an incumbent solution for the first stage.

6. (In-Sample Stopping Rule). If an In-Sample stopping rule is satisfied, output the incumbent solution $\mathbf{x}^\nu$ and continue to step 7. Else repeat from step 3.

7. (Out-of-Sample Stopping Rule). If the number of replications is greater than or equal to M, calculate a compromise decision $\mathbf{x}^c$ using a set of $\{\mathbf{x}^\nu\}_{\nu=1}^M$. Else, repeat from step 2.

The value function approximation for replication ν is denoted f^ν and the terminal solution for that replication is x^ν. Note that we generate sample average subgradient approximations (SASA) using $K_\nu(\varepsilon)$ observations. Since these observations are i.i.d, the in-sample stopping rule ensures an unbiased estimate of the second stage objective is used for the objective function estimate at x^ν. Hence, the Central Limit Theorem (CLT) implies that $[K_\nu(\varepsilon)]^{\frac{1}{2}}[f(x^\nu) - f^\nu(x^\nu)]$ is asymptotically normal $\mathbf{N}(0, \sigma_\nu^2)$, where $\sigma_\nu^2 < \infty$ denotes the variance of $f^\nu(x^\nu)$. Since

$$N = \min_\nu K_\nu(\varepsilon), \tag{22}$$

it follows that the error $[f(x^\nu) - f^\nu(x^\nu)]$ is no greater than $O_p(N^{-\frac{1}{2}})$. The following result provides the basis for compromise solutions $\mathbf{x}^c$ as proved in Sen and Liu (2016).

Proposition 2. Suppose that assumptions SD(a-d) stated in the Appendix II hold. Suppose $\bar{\mathbf{x}}$ is defined as in Theorem 1, and $\mathbf{x}^c = \bar{\mathbf{x}}$. Then,

a) $\mathbf{x}^c$ solves

$$\operatorname*{Min}_{x \in X} \quad \bar{F}_M(x) := \frac{1}{M} \sum_{\nu=1}^M f^\nu(x), \tag{23}$$

b)

$$f(\mathbf{x}^c) \leq \bar{F}_M(\mathbf{x}^c) + O_p((NM)^{-\frac{1}{2}}), \tag{24}$$

c) $\mathbf{x}^c(\delta)$ denote an δ-optimal solution to (23). Let f^* denote the optimal value of the problem,

$$\lim_{\delta \to 0} ||\bar{\mathbf{x}}(\delta) - \mathbf{x}^c(\delta)|| \ \rightarrow \ 0 \text{ (wp1)}, \tag{25}$$

d)

$$\lim_{\delta \to 0} P(|\bar{F}_{\delta,N}(\bar{\mathbf{x}}(\delta)) - f^*| \geq t) \;\to\; 0 \;\text{ for all } t \geq 0. \tag{26}$$

Proof: See Sen and Liu (2016).

Appendix III: Statistical Tests

The three standard statistical tests summarized below include χ^2, T and F tests. For any of these tests, if the null hypothesis is rejected in either case, then the corresponding LEO model is rejected. Otherwise, it is NOT rejected (i.e., models will be either rejected or not). In addition, we include the calculation of confidence intervals of the estimated expected cost-to-go function for a given decision.

The T-test and F-test are based on asymptotic normality of the optimal value and the optimal solutions of the stochastic programming problem. The property of asymptotic normality is obtained via uniform convergence of SAA (Homem-de Mello and Bayraksan (2014)) whereas the asymptotic normality of solutions follows from the uniqueness of limits as shown in Sen and Liu (2016). The latter property does not necessarily hold for the general SAA setting which does not impose any algorithmic conditions.

<u>χ^2 Test for Error Terms and Cost-to-go Objectives.</u> We perform χ^2 tests for error terms and the cost-to-go objective functions. The data-set is expected to have two parts, and we test the null hypothesis that both parts of the data-set share a common distribution. Given a data-set we allocate the data into B bins, for the i^{th} bin, denote E_{1i} as the observed frequency for bin i from one sample, and E_{2i} as the observed frequency for bin i from validation the other sample. Then the χ^2 statistic for this data is estimated as:

$$\hat{\chi}^2 = \sum_{i=1}^{B} \frac{(E_{1i} - E_{2i})^2}{E_{1i} + E_{2i}} \tag{27}$$

We check the χ^2 distribution with B degrees of freedom which provides the standard value $\chi^2(B)$, and the probability $p = Prob(\chi^2(B) > \hat{\chi}^2)$. Given a significance level α, we reject the null hypothesis if $p \leq \alpha$; otherwise, we do not reject the null hypothesis.

<u>Cost-to-go Hypothesis test for the Mean value using T-statistic.</u>

For cost-to-go objectives, we introduce the null hypothesis test for the mean value. Suppose we have two independent samples which reflect the training and validation sets, the mean values of them are h_1, h_2, and the standard deviations are s_1, s_2. To determine whether two sample means are significantly different with $\alpha = 0.05$, then T-statistic of two group means is $t = \frac{|h_1 - h_2|}{\sqrt{s_1^2 + s_2^2}}$. Compare the calculated T-value with $t_0 = 1.96$, if $t > t_0$, we will reject the null hypothesis, which indicates that the mean values of two samples are significantly different. If this hypothesis is rejected, the objectives of training and validation set are considered to be different on the basis of the first moment.

Cost-to-go Hypothesis test for the Variance value using F-statistic.

Besides the hypothesis test on the first moment level, we also perform a test for the variance value based on the F distribution. The F-test is often used to test if the variances of two samples are consistent. The null hypothesis H_0 is defined as: the variances of two samples are equal. Suppose we denote the observed variances of two samples as s_1^2 and s_2^2, then the F statistic is the following $f = s_1^2/s_2^2$. Suppose we choose the significance level α, sample 1 has sample size N_1, and sample 2 has sample size N_2, then the critical region is decided by two values from F-distribution: $F_{\alpha/2,N_1-1,N_2-1}, F_{1-\alpha/2,N_1-1,N_2-1}$. If $F_{\alpha/2,N_1-1,N_2-1} \leq F \leq F_{1-\alpha/2,N_1-1,N_2-1}$, we do not reject the null hypothesis.

Confidence Intervals of the estimated expected cost-to-go functions.

Next we present the calculation of confidence interval (of the mean). Suppose we let $\bar{h}$ denote the mean of validated cost-to-go values, and d denote the approximate standard deviation of validated cost-to-go mean values among all the replications. The $1-\alpha = 95\%$ confidence interval is given by $[\bar{h} - \frac{t_{\alpha/2,k-1}}{\sqrt{k-1}}d, \bar{h} + \frac{t_{\alpha/2,k-1}}{\sqrt{k-1}}d]$, in which $k \in [5, 10]$ represents the number of folds in cross validation and $t_{\alpha/2,k-1}$ denotes the t distribution value with probability $1-\alpha$ and $k-1$ degrees of freedom.

Appendix IV: Snapshot Computational Results for `ELECEQUIP`

A “snapshot” study for the inventory model provided in this Appendix. We select the end of the first year as the point in time when statistical comparisons are made. For this snapshot study, such a choice, allowing the model to run for a year, helps to avoid initialization bias.

Table 8 provides the estimated objective, validated objective (95% confidence and prediction intervals), and the solution quality. The probability of optimality reported in Table 8 is a result of the computations suggested in Theorem 2, where δ_u is chosen to be 1% of the total cost. Notice that we do not report a probability for the DAF model because it is simply a result of the ARIMA forecast. On the other hand, we include the probability for the SLP model, and this is consistent statistical optimality of section 3.

Table 9 summarizes results for three hypothesis tests for both DAF and SLP cases. A hypothesis test rejects the null hypothesis at the 95% level when the statistic lies outside the range provided in the table. Upon examining the entries for the T-test, the null hypothesizes for both DAF and SLP are not rejected. We also perform the F-test for DAF and SLP, and the F-test rejects the hypothesis that the variance of training and validation are the same at the 95% level. The results of the χ^2 test are presented in the last two rows, which analyzes the consistency of two data sets. Note that both DAF and SLP are not rejected at level $\alpha = 0.05$, but SLP shows a higher p-value. From these test results, we conclude that the SLP approach performs better in terms of consistency between training and validation sets.

The comparison across models is provided in Table 10. The cost of SLP in validation is smaller than DAF by 9.42, and shows smaller generalization error as well. We include the $p-$value of Kruskal-Wallis test between DAF and SLP approaches, and the result shows that objectives of DAF and SLP methodologies have significantly different ranked medians. Since LEO-`ELECEQUIP` is a minimization problem, better solutions result in costs that are at the lower end of the horizontal (cost) axis. In this case, the better decision results from SLP, and Figure 5 gives evidence of this conclusion because for all cost levels C, the $Prob\ (Cost \leq C)$ is higher for SLP than it is for DAF.

Models	DAF	SLP
Estimated Obj.	25.52	22.75
95% Validated Confidence Interval	30.34($\pm$3.84)	21.92($\pm$3.38)
95% Validated Prediction Interval	30.87($\pm$9.03)	21.75($\pm$8.32)
Probability (γ)		0.9934
Tolerance (δ)		0.092

Table 8 LEO-ELECQUIP: Comparison of Solutions under Alternative Models

Models	DAF	SLP
T-statistic ($t < 1.96$)	$t = 1.21$	$t = 0.20$
Cost-to-go Test (Mean)	not rejected	not rejected
F-statistic ($0.62 < f < 1.62$)	$f = 2.53$	$f = 1.23$
Cost-to-go Test (Variance)	rejected	not rejected
χ^2 Test p-value ($p > 0.05$)	$p = 0.13$	$p = 0.37$
Cost-to-go Test (Distribution)	not rejected	not rejected

Table 9 LEO-ELECQUIP: Hypothesis Test Results under Alternative Models

Models	DAF	SLP
Generalization Error	1.45	0.96
Kruskal-Wallis Test ($p-$value)	1.24×10^{-6}	
Optimization Error	9.42	

Table 10 LEO-ELECQUIP: Errors under Alternative Models

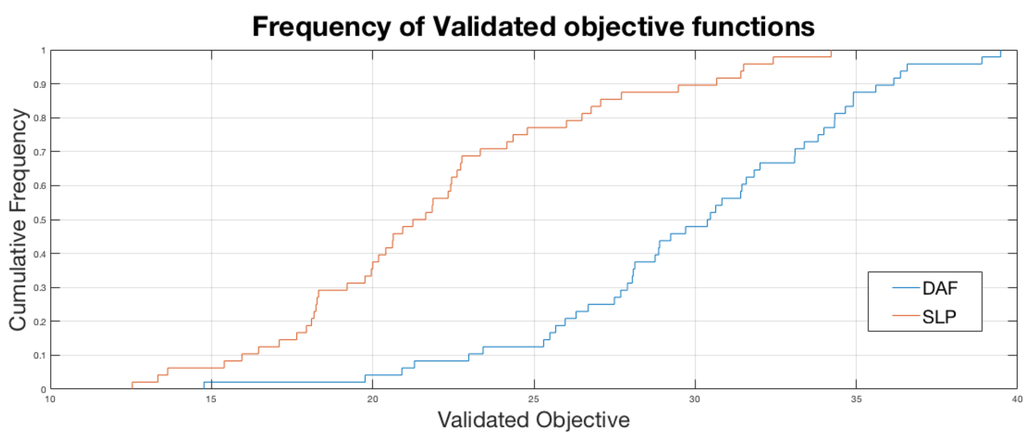

Figure 5 LEO-ELECQUIP: Stochastic Dominance of SLP Validated objectives over DAF

Appendix V: Proofs

Proof of Theorem 1

Since $\mathbf{x}^c \in \arg\min\{\bar{F}_M(x) + \frac{\bar{\rho}}{2}||x - \bar{\mathbf{x}}||^2 : x \in \mathbf{X}\}$, we have,

$$0 \in \partial \bar{F}_M(\mathbf{x}^c) + \mathcal{N}_X(\mathbf{x}^c) + \bar{\rho}(\mathbf{x^c} - \bar{\mathbf{x}}).$$

Hence, $-\bar{\rho}(\mathbf{x^c} - \bar{\mathbf{x}})$ can be used as a subgradient of the function $\bar{F}_M(x) + \mathcal{I}_X(x)$ at $x = \mathbf{x}^c$. Hence, for all $x \in \mathbf{X}$,

$$\bar{F}_M(x) + \mathcal{I}_X(x) \geq \bar{F}_M(\mathbf{x}^c) + \mathcal{I}_X(\mathbf{x}^c) - \bar{\rho}(\mathbf{x}^c - \bar{\mathbf{x}})^\top (x - \mathbf{x}^c)$$

Since $\bar{\mathbf{x}}, \mathbf{x^c} \in \mathbf{X}$, the indicator terms vanish, and therefore,

$$\bar{F}_M(\bar{\mathbf{x}}) + \bar{\rho}(\mathbf{x}^c - \bar{\mathbf{x}})^\top (\bar{\mathbf{x}} - \mathbf{x^c}) \geq \bar{F}_M(\mathbf{x}^c).$$

Since $\bar{\rho}(\mathbf{x^c} - \bar{\mathbf{x}})^\top (\bar{\mathbf{x}} - \mathbf{x^c}) \leq \bar{\rho}||\mathbf{x^c} - \bar{\mathbf{x}}||\ ||\bar{\mathbf{x}} - \mathbf{x^c}||$, we have

$$\bar{F}_M(\bar{\mathbf{x}}) + \bar{\rho}||\mathbf{x^c} - \bar{\mathbf{x}}||^2 \geq \bar{F}_M(\mathbf{x}^c). \tag{28}$$

Recall that $\bar{\mathbf{x}} = \frac{1}{M}\sum_\nu \mathbf{x}^\nu$, and $\bar{F}_M$ is convex, therefore, $F_M(\bar{\mathbf{x}}) \leq \frac{1}{M}\sum_\nu \bar{F}_M(\mathbf{x}^\nu)$. Because $f^j(\mathbf{x}^\nu) \leq f^\nu(\mathbf{x}^\nu)$ for all pairs (j, ν), and $f^\nu_\varepsilon = f^\nu(\mathbf{x}^\nu)$, we have

$$\bar{F}_M(\bar{\mathbf{x}}) \leq \frac{1}{M}\sum_\nu \bar{F}_M(\mathbf{x}^\nu) \leq \frac{1}{M}\sum_\nu f^\nu_\varepsilon. \tag{29}$$

Combining (28) and (29), we get

$$\frac{1}{M}\sum_\nu f^\nu_\varepsilon + \delta = \frac{1}{M}\sum_\nu f^\nu_\varepsilon + \bar{\rho}||\mathbf{x^c} - \bar{\mathbf{x}}||^2 \geq \bar{F}_M(\mathbf{x}^c). \quad \blacksquare$$

Proof of Theorem 2

If we solve for the probability from (21) in Proposition 1, the following inequality holds:

$$Prob(\hat{S}_M(\delta) \subset S(\delta_u)) \geq 1 - \exp\left(-\frac{K(\delta_u - \delta)^2}{8\lambda^2 D^2} + n\ln\left(\frac{8LD}{\delta_u - \delta}\right)\right). \tag{30}$$

From assumption SAA-c in Appendix II, $\lambda = 2L$. Also, recall from (22) in Appendix II, each replication uses a sample size of at least N. Therefore, in this case the total sample size K is at least NM. The conclusion holds by replacing λ and K in (30). $\blacksquare$

Proof of Theorem 3

Under assumptions for SAA, we may set a tolerance level $\delta/2$, and a reliability level $\varepsilon = (1+\gamma)/2 > \gamma$. Then Proposition 1 in Appendix II ensures that there exists a finite sample size $K(\varepsilon, \delta/2) < \infty$ such that $K(\gamma, \delta) < K(\varepsilon, \delta/2)$, and the SAA approximation provides a $\delta/2$-optimum solution with probability $\varepsilon > \gamma$. Because SP with the sample size $K(\varepsilon, \delta/2)$ has finitely many outcomes, its sample space is compact. Hence if SD is applied to this empirical problem, assumptions SD-a,b,c,d of traditional SD are satisfied. Therefore the convex SD algorithm provides a $\delta/2$-optimum solution with probability 1 for SAA with sample size $K(\varepsilon, \delta/2)$. Since $\gamma < \varepsilon < 1$, there exists a finite iteration $K(\gamma, \delta) \leq K(\varepsilon, \delta/2)$ such that the solution of convex SD is δ-optimum. ■

Proof of Theorem 4

Although this result in discussed in the SL context of Hastie et al. (2011), the proof is not provided there. Hence we have include it here for completeness. By the definitions of $\mathbb{E}_h(\text{Err}_{in})$ and $\mathbb{E}_h(\text{err})$, the following equations hold:

$$\begin{aligned}
\mathbb{E}_h(\text{Err}_{in}) - \mathbb{E}_h(\text{err}) &\approx \frac{1}{|T|}\sum_{i=1}^{|T|} \mathbb{E}_h\mathbb{E}_{h^+}(h_i^+ - \hat{h}_i)^2 - \frac{1}{|T|}\sum_{i=1}^{|T|} \mathbb{E}_h(h_i - \hat{h}_i)^2 \\
&= \frac{1}{|T|}\sum_{i=1}^{|T|}\left[\mathbb{E}_h\mathbb{E}_{h^+}(h_i^{+2} + \hat{h}_i^2 - 2h_i^+\hat{h}_i) - \mathbb{E}_h(h_i^2 + \hat{h}_i^2 - 2h_i\hat{h}_i)\right] \\
&= \frac{1}{|T|}\sum_{i=1}^{|T|}\left[\mathbb{E}_{h^+}h_i^{+2} + \mathbb{E}_h\hat{h}_i^2 - 2\mathbb{E}_h\mathbb{E}_{h^+}(h_i^+\hat{h}_i) - \mathbb{E}_h h_i^2 - \mathbb{E}_h\hat{h}_i^2 + 2\mathbb{E}_h(h_i\hat{h}_i)\right] \\
&= \frac{1}{|T|}\sum_{i=1}^{|T|}\left[\mathbb{E}_{h^+}h_i^{+2} - 2\mathbb{E}_{h^+}\mathbb{E}_h(h_i^+\hat{h}_i) - \mathbb{E}_h h_i^2 + 2\mathbb{E}_h(h_i\hat{h}_i)\right] \\
&= \frac{1}{|T|}\sum_{i=1}^{|T|}\left[2\mathbb{E}_h(h_i\hat{h}_i) - 2\mathbb{E}_h(h_i)\mathbb{E}_h(\hat{h}_i)\right] \\
&= \frac{2}{|T|}\sum_{i=1}^{|T|}\text{Cov}(h_i, \hat{h}_i),
\end{aligned}$$

where the first equation is a result of (14) and (15), the second and third are due to algebraic manipulcations, the fourth follows from assumption A2 that $\mathbb{E}_{h^+}h_i^{+2} = \mathbb{E}_h h_i^2$, and the fifth by definition. ■

Acknowledgments

This research was supported by AFOSR Grant FA9550-15-1-0267, NSF Grant ECCS 1548847 and NSF Grant CMMI 1538605. We also thank Gabe Hackebeil for extending PySP functionality to allow SMPS output which is necessary for our SD code (available from the authors as well as the Github repository).

References

Ban, Gah-Yi, Noureddine El Karoui, Andrew E. B. Lim. 2017. Machine learning and portfolio optimization. *Management Science* doi:10.1287/mnsc.2016.2644.

Bayraksan, Güzin, David P Morton. 2011. A sequential sampling procedure for stochastic programming. *Operations Research* **59**(4) 898–913.

Bayraksan, Guzin, Péguy Pierre-Louis. 2012. Fixed-width sequential stopping rules for a class of stochastic programs. *SIAM Journal on Optimization* **22**(4) 1518–1548.

Ben-Tal, Aharon, Arkadi Nemirovski. 2001. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM.

Bertsekas, D.P. 2012. *Dynamic Programming and Optimal Control*. No. v. 2 in Athena Scientific optimization and computation series, Athena Scientific.

Bertsimas, Dimitris, Nathan Kallus. 2014. From predictive to prescriptive analytics. *arXiv preprint arXiv:1402.5481* .

Bertsimas, Dimitris, Romy Shioda. 2007. Classification and regression via integer optimization. *Operations Research* **55**(2) 252–271.

Bertsimas, Dimitris, Melvyn Sim. 2004. The price of robustness. *Operations Research* **52**(1) 35–53.

Birge, John R, Francois Louveaux. 2011. *Introduction to Stochastic Programming*. Springer Science Business Media.

Box, George EP, Gwilym M Jenkins, Gregory C Reinsel, Greta M Ljung. 2015. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons.

De Farias, Daniela Pucci, Benjamin Van Roy. 2004. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research* **29**(3) 462–478.

Diaconis, Persi, Mehrdad Shahshahani. 1984. On nonlinear functions of linear combinations. *SIAM Journal on Scientific and Statistical Computing* **5**(1) 175–191.

Frazier, Peter. 2012. Optimization via simulation with bayesian statistics and dynamic programming. *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference, 7.

Freimer, Michael B, Jeffrey T Linderoth, Douglas J Thomas. 2012. The impact of sampling methods on bias and variance in stochastic linear programs. *Computational Optimization and Applications* **51**(1) 51–75.

Frey, Jesse. 2013. Data-driven nonparametric prediction intervals. *Journal of Statistical Planning and Inference* **143**(6) 1039–1048.

Friedman, Jerome H, Werner Stuetzle. 1981. Projection pursuit regression. *Journal of the American Statistical Association* **76**(376) 817–823.

Glynn, Peter W, Gerd Infanger. 2013. Simulation-based confidence bounds for two-stage stochastic programs. *Mathematical Programming* **138**(1-2) 15–42.

Hastie, Trevor J.., Robert John Tibshirani, Jerome H Friedman. 2011. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Higle, Julia L, Suvrajeet Sen. 1991. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research* **16**(3) 650–669.

Higle, Julia L, Suvrajeet Sen. 1994. Finite master programs in regularized stochastic decomposition. *Mathematical Programming* **67**(1-3) 143–168.

Higle, Julia L, Suvrajeet Sen. 1996a. Duality and statistical tests of optimality for two stage stochastic programs. *Mathematical Programming* **75**(2) 257–275.

Higle, Julia L, Suvrajeet Sen. 1996b. *Stochastic Decomposition*. Springer.

Hillier, Frederick S, G J Lieberman. 2012. *Introduction to operations research*. Tata McGraw-Hill Education.

Homem-de Mello, Tito, Güzin Bayraksan. 2014. Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science* **19**(1) 56–85.

James, Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani. 2013. *An Introduction to Statistical Learning*, vol. 6. Springer.

Kao, Y-H., Benjamin V. Roy, Xiang Yan. 2009. Directed regression. Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, A. Culotta, eds., *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc., 889–897. URL `http://papers.nips.cc/paper/3686-directed-regression.pdf`.

Kleywegt, Anton J, Alexander Shapiro, Tito Homem-de Mello. 2002. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* **12**(2) 479–502.

Kruskal, William H, W Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association* **47**(260) 583–621.

Kulis, Brian, et al. 2013. Metric learning: A survey. *Foundations and Trends® in Machine Learning* **5**(4) 287–364.

Linderoth, Jeff, Alexander Shapiro, Stephen Wright. 2006. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research* **142**(1) 215–241.

Liu, Junyi, Suvrajeet Sen. 2017. Asymptotic results for two-stage stochastic quadratic programming. *SIAM Journal on Optimization (submitted)* .

Liyanage, Liwan H, J George Shanthikumar. 2005. A practical inventory control policy using operational statistics. *Operations Research Letters* **33**(4) 341–348.

Mak, Wai-Kei, David P Morton, R Kevin Wood. 1999. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters* **24**(1) 47–56.

Miller, Naomi, Andrzej Ruszczyński. 2011. Risk-averse two-stage stochastic linear programming: Modeling and decomposition. *Operations Research* **59**(1) 125–132.

Nocedal, Jorge, Stephen J Wright. 1999. *Numerical Optimization*. Springer Science in Operations Research.

Powell, Warren B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, vol. 842. John Wiley & Sons.

Rios, Ignacio, Roger JB Wets, David L Woodruff. 2015. Multi-period forecasting and scenario generation with limited data. *Computational Management Science* **12**(2) 267–295.

Royset, Johannes O, Roberto Szechtman. 2013. Optimal budget allocation for sample average approximation. *Operations Research* **61**(3) 762–776.

Royset, Johannes O, Roger JB Wets. 2014. From data to assessments and decisions: Epi-spline technology. *Tutorials in Operations Research: Bridging Data and Decisions*. INFORMS, 27–53.

Rudin, Cynthia, Gah-Yi Vahn. 2014. The big data newsvendor: Practical insights from machine learning. *DSpace* .

Ryzhov, Ilya O, Warren B Powell, Peter I Frazier. 2012. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research* **60**(1) 180–195.

Sen, Suvrajeet, Yifan Liu. 2016. Mitigating uncertainty via compromise decisions in two-stage stochastic linear programming: Variance reduction. *Operations Research* **64**(6) 1422–1437.

Sen, Suvrajeet, Zhihong Zhou. 2014. Multistage stochastic decomposition: a bridge between stochastic programming and approximate dynamic programming. *SIAM Journal on Optimization* **24**(1) 127–153.

Shapiro, Alexander, Darinka Dentcheva, Andrzej Ruszczynski. 2009. *Lectures on Stochastic Programming*, vol. 10. SIAM, Philadelphia.

Shapiro, Alexander, Tito Homem-de Mello. 1998. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming* **81**(3) 301–325.

Xiao, Lin, Tong Zhang. 2014. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization* **24**(4) 2057–2075.