SANDIA REPORT

SAND 2010-8811
Unlimited Release
Printed December 20, 2010

Computing Confidence Intervals on Solution Costs for Stochastic Grid Generation Expansion Problems

Jean-Paul Watson and David L. Woodruff

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND 2010-8811
Unlimited Release
Printed December 20, 2010

Computing Confidence Intervals on Solution Costs for Stochastic Grid Generation Expansion Problems

Jean-Paul Watson and David L. Woodruff

Abstract

A range of core operations and planning problems for the national electrical grid are naturally formulated and solved as stochastic programming problems, which minimize expected costs subject to a range of uncertain outcomes relating to, for example, uncertain demands or generator output. A critical decision issue relating to such stochastic programs is: How many scenarios are required to ensure a specific error bound on the solution cost? Scenarios are the key mechanism used to sample from the uncertainty space, and the number of scenarios drives computational difficultly. We explore this question in the context of a long-term grid generation expansion problem, using a bounding procedure introduced by Mak, Morton, and Wood. We discuss experimental results using problem formulations independently minimizing expected cost and down-side risk. Our results indicate that we can use a surprisingly small number of scenarios to yield tight error bounds in the case of expected cost minimization, which has key practical implications. In contrast, error bounds in the case of risk minimization are significantly larger, suggesting more research is required in this area in order to achieve rigorous solutions for decision makers.

Acknowledgements

We gratefully acknowledge the assistance of Shan Jin and Prof. Sarah Ryan at Iowa State University for their model and data, and for providing a rich environment in which to explore computational aspects of a key grid planning problem.

Contents

Appendix

List of Tables

Introduction

Formal optimization models play a central role in a number of foundational problems in electrical grid operations and planning. For example, unit commitment – the problem of scheduling thermal generator on/off states and corresponding output levels – is now widely expressed and solved as a deterministic mixed-integer program, using commercial off-the-shelf solvers such as CPLEX [2]. Similarly, generation expansion – the problem of determining the type, count, and location of generators that must be built over the long term to meet anticipated demand, is naturally formulated as a deterministic mixed-integer program – a traditional formalism for expressing such classic resource allocation problems. While these optimization problems are in theory computationally intractable, in practice high-quality and often optimal solutions are obtainable using commercial solvers on commodity hardware (workstations) in minutes to hours of run-time.

Historically, grid optimization models have been formulated and solved as deterministic mixed-integer programs, i.e., where the values of all parameters are known in advance. Some parameters, including short-term (hourly) demand, can be predicted with high accuracy based on historical data. However, projected demands over yearly or decade-long time horizons, or generator outputs in the context of renewable sources such as solar and wind, are very difficult to predict, yielding a large uncertainty envelope. One approach to deal with such uncertainty in the context of mixed-integer programming is to approximate the distribution of a parameter value with its mean. However, this approach commonly leads to optimistic – translating to higher-cost and higher-risk – solutions when they are evaluated in the true stochastic context [1].

To properly deal with uncertainty in the context of mathematical optimization, the formalism of stochastic mixed-integer programming is a widely studied and powerful paradigm [8, 9]. Informally, stochastic programming extends deterministic mathematical programming through three key mechanisms. First, the notion of a scenario tree is introduced to represent the evolution of parameter uncertainty through time. Second, the optimization objective is modified to minimize expected cost across all scenarios in the tree. Third, non-anticipativity constraints are enforced to ensure that no solution can take advantage of knowledge of the future, as encoded in the scenario tree.

Stochastic programming is a powerful formalism, and is starting to become more widely used in practice. One impediment to widespread adoption is the increase in computational difficulty relative to the deterministic case. While still NP-hard, the typically large number of scenarios employed dramatically inflates the computational difficulty of such problems in practice. Less explored, particularly in the case of practical, real-world optimization problems, is the following question: How do we know we are using a sufficient number of scenarios? The scenario tree typically represents a discretized approximation of an underlying continuous stochastic process. In the process of discretization, error is necessarily introduced, such that solutions to a discretized scenario tree will have a different cost relative to the cost of the "true" (if it could be analytically computed), infinite-sample scenario tree.

Quantification of this error is critical, for two key reasons. First, due to computational difficulty, we want to use the minimal number of scenarios required to achieve a solution with a given error. A significant reduction is the number of scenarios can reduce a practically uncomputable problem to one for which solutions are attainable. Second, we are often forced due to external circumstances – e.g., when scenarios are obtained via expensive computer simulations – to deal with a fixed number of scenarios. In these situations, we must make maximal use of the scenarios we are given, identifying the best solution with the smallest possible error bound.

In this report, we focus on answers to this question in the context of a stochastic grid generation expansion problem, developed by external collaborators from Iowa State University's Industrial and Manufacturing Systems Engineering Department. To compute error bounds on the optimal investment costs, we use the Multiple Replication Procedure originally introduced by Mak, Morton, and Wood. Due to the computational challenge, this procedure has not been widely used in practical, large-scale stochastic programming problems. Our investigations represent a first in terms of large-scale, practical application. Further, our results yield practical impact in terms of answering questions relating to the scale of problem that should be addressed, and identify future research directions involving the quantification of risk-oriented optimization metrics.

The remainder of this report is organized as follows. We begin with an overview of stochastic programming. The grid generation expansion problem we consider is described next, followed by a discussion of the corresponding test case. We briefly survey the Mak, Morton, and Wood Multiple Replication Procedure, and follow with a discussion of our software implementation and associated effort. Experimental results concerning solution error bounds on the generation expansion problem are then presented. Finally, we discuss future research directions.

Stochastic Programming: An Overview

We now briefly introduce the concept of a stochastic program. More comprehensive introductions to both the theoretical foundations and the range of potential applications can be found in [1], [8], and [9].

We concern ourselves with stochastic optimization problems where uncertain parameters (data) can be represented by a set of scenarios $\mathcal{S}$, each of which specifies both (1) a full set of random variable realizations and (2) a corresponding probability of occurrence. The random variables in question specify the evolution of uncertain parameters over time. We index the scenario set by s and refer to the probability of occurrence of s (or, more accurately, a realization "near" scenario s) as $\Pr(s)$. Let the number of scenarios be given by $|S|$. The source of these scenarios does not concern us in this paper, although we observe that they are frequently obtained via simulation or formed from expert opinions. We assume that the decision process of interest consists of a sequence of discrete time stages, the set of which is denoted T. We index T by t, and denote the number of time stages by $|T|$.

We develop the notation primarily for the linear case in the interest of simplicity and practicality (little attention has been devoted to nonlinear stochastic programs, due to their extreme difficulty). For each scenario s and time stage t, $t \in \{1, \ldots, |\mathcal{T}|\}$, we are given a row vector $c(s,t)$ of length $n(t)$, a $m(t) \times n(t)$ matrix $A(s,t)$, and a column vector $b(s,t)$ of length $m(t)$. Let $N(t)$ be the index set $\{1, \ldots, n(t)\}$ and $M(t)$ be the index set $\{1, \ldots, m(t)\}$. For notational convenience, let $A(s)$ denote $(A(s,1), \ldots, A(s, |\mathcal{T}|))$ and let $b(s)$ denote $(b(s,1), \ldots, b(s, |\mathcal{T}|))$.

The decision variables in a stochastic program consist of a set of $n(t)$ vectors $x(t)$; one vector for each scenario $s \in \mathcal{S}$. Let $X(s)$ be $(x(s,1), \ldots, x(s, \mathcal{T}))$. We will use X as shorthand for the entire solution system of x vectors, i.e., $X = x(1,1), \ldots, x(|\mathcal{S}|, |\mathcal{T}|)$.

If we were prescient enough to know which scenario $s \in \mathcal{S}$ would be ultimately realized, our optimization objective would be to minimize

$$f_s(X(s)) \equiv \sum_{t \in T} \sum_{i \in N(t)} [c_i(s,t) x_i(s,t)] \qquad (\mathrm{P}_s)$$

subject to the constraint

$$X \in \Omega_s.$$

We use Ω_s as an abstract notation to express all constraints for scenario s, including requirements that some decision vector elements are discrete or more general requirements such as

$$A(s)X(s) \geq b(s).$$

The notation $A(s)X(s)$ is used to capture the usual sorts of single period and inter-period linking constraints that one typically finds in multi-stage mathematical programming formulations.

We must obtain solutions that do not require foreknowledge and that will be feasible independent of which scenario is ultimately realized. In particular, lacking prescience, only solutions that are implementable are practically useful. Solutions that are not admissible, on the other hand, may have some value because while some constraints may represent laws of physics, others may be violated slightly without serious consequence.

We refer to a solution that satisfies constraints for all scenarios as *admissible*. We refer to a solution vector as *implementable* if for all pairs of scenario s and s' that are indistinguishable up to time t, $x_i(s,t') = x_i(s',t')$ for all $1 \leq t' \leq t$ and each i in each $N(t)$. We refer to the set of all implementable solutions as $\mathcal{N}_{\mathcal{S}}$ for a given set of scenarios, $\mathcal{S}$.

To achieve admissible and implementable solutions, the expected value minimization problem then becomes:

$$\min \sum_{s \in \mathcal{S}} [\Pr(s) f(s; X(s))] \qquad (\mathrm{P})$$

subject to

$$X \in \Omega_s$$

$$X \in \mathcal{N}_{\mathcal{S}}.$$

Formulation (P) is known as a stochastic mathematical program. If all decision variables are continuous, we refer to the problem simply as a stochastic program. If some of the decision variables are discrete, we refer to the problem as a stochastic mixed-integer program.

In practice, the parameter uncertainty is stochastic programs is often encoded via a *scenario tree*, in which a node specifies the parameter values $b(s,t)$, $c(s,t)$, and $A(s,t)$ for all $t \in \mathcal{T}$ and $s, s' \in \mathcal{S}$ such that s and s' are indistinguishable up to time t.

Grid Generation Expansion: Formulation

We consider a grid generation expansion problem formulation introduced by Shan Jin and Sarah Ryan, collaborators at Iowa State University. We now briefly describe the stochastic programming formulation for this problem, summarizing the more detailed descriptions available in [3, 4]. We draw our notation from those references, retaining consistency to facilitate cross-referencing.

The core decision variables in the generation expansion problem (GEP) are the number of generators of type $g \in \mathcal{G}$ built in year $y \in \mathcal{Y}$. We denote these quantities by U_{gy}, whose values are constrained to take non-negative integer values; one cannot build a fraction of a nuclear power plant. The U_{gy} represent the so-called first stage (here and now) decision variables in the stochastic programming formulation of the GEP. Each generator $g \in \mathcal{G}$ costs a certain value c_g to construct (capital cost) per MW of power capacity installed. Each generator type g has a maximum MW generation capacity m_g per installed unit, and a maximum total number of units $u_g^{\max}$ that can be built over the planning time horizon. The number of existing generators of type g is denoted u_g.

Each year $y \in \mathcal{Y}$ of the planning horizon in split into a number of sub-periods, representing qualitatively different types demand, i.e., base, shoulder, or peak. We let $t \in \mathcal{T}$ denote a sub-period index, T_y the set of sub-periods in year y, and Y_t the year of which a sub-period t is a member. Clearly, $|\mathcal{Y}| \leq |\mathcal{T}|$. The total number of hours h_t in each sub-period t could potentially vary over time, but is not treated stochastically in this particular formulation of the GEP. There is a penalty cost p (units in \$/MWh) for failing to serve any portion of demand. Finally, an annual interest rate r is imposed to yield cost discounting.

Each scenario $s \in \mathcal{S}$ in the SP formulation of the GEP specifies the probability of occurrence π_s, the demand per hour d_{ts} for each sub-period (in MW), and the generation cost l_{gts} for each generator type g and time period t (in \$/MWh). Given an assignment of first stage decision variables U_{gy}, the second stage (scenario-specific) decision variables consist of (1) the amount E_{ts} of unmet demand in each sub-period t for scenario s and (2) the load L_{gts} generated by generators of type g in sub-period t for scenario s.

Given the presented parameters (deterministic and uncertain) and decision variables, the objective of the GEP SP is to minimize the investment cost plus the expected operating cost (accounting for any penalties for unmet demand), expressed mathematically as follows:

$$\sum_{y\in\mathcal{Y}} \frac{\sum_{g\in\mathcal{G}}(c_g m_g U_{gy}) + \sum_{s\pi_s}(\sum_{t\in T_y}(\sum_{g\in\mathcal{G}}(h_t l_{gts} L_{gts}) + p h_t E_{ts}))}{(1+r)^{y-1}} \tag{1}$$

On a per-year basis, the cost is split into investment and operating costs, which are discounted based on the annualized interest rate r.

The constraints on the GEP SP are relatively straightforward. First, the total demand must equal the load generated plus the unserved demand, as follows:

$$\sum_{g\in\mathcal{G}} L_{gts} + E_{ts} = d_{ts} \quad \forall t \in \mathcal{T}, s \in \mathcal{S} \tag{2}$$

Second, we must ensure that the load delivered in each sub-period t by generators of type g is consistent with the total quantity of generators installed at that time. This logical condition is imposed via the following constraint:

$$L_{gts} \le n_g^{\max}(u_g + \sum_{y\le Y_t} U_{gy}) \quad \forall g \in \mathcal{G}, t \in \mathcal{T}, s \in \mathcal{S} \tag{3}$$

Finally, it is necessary to enforce the limit on the aggregate number of generators built, of each type, over the planning horizon:

$$\sum_{y\in\mathcal{Y}} U_{gy} \le u_g^{\max} \quad \forall g \in \mathcal{G} \tag{4}$$

While this GEP formulation appears innocuous, it is in practice extremely difficult to solve directly. The difficulty is driven in part by the number of scenarios used to approximate the stochastic process driving the future costs of power generation and the inter-dependencies across the time periods. Further, the discrete nature of the U_{gy} yield a large-scale stochastic mixed-integer program, which are notoriously difficult for even small numbers of scenarios.

Grid Generation Expansion: Test Case

The specific test instance we consider is fully described in [4]. The instance considers six different types of generators and a planning horizon of 10 years. Each year is split into 3 sub-periods, with identical numbers of hours across the different years of the planning horizon. The uncertain parameters – demand and generation costs – are driven by specific stochastic process models. In the case of demand, the process simply models growth as one of a small, medium, or large fraction of current demand. In the case of generation cost, the stochastic process is driven by uncertainty in fuel prices.

A single, large scenario tree was generated by simulating the stochastic process through various sample paths. Specifically, the 10-year planning horizon yields a scenario tree 9 stages deep (discounting the first year) with a branching factor of 3 – yielding a total of $3^9 = 19683$ scenarios. In comparison to models appearing in the literature, such a deep scenario tree is exceptionally large, and as a consequence, the corresponding GEP SP instance is extremely difficult to solve. In particular, we observe that [3] report several weeks of run-time using a sophisticated decomposition strategy were required to solve the instance to optimality.

The Multiple Replication Procedure

The key question under consideration is: Given a particular set of scenarios for a stochastic program, what is the confidence interval on the cost optimality? In other words, how much might we expect the optimal cost to change should a different set of scenarios be used? This issue is critical in practice – especially in electrical grid – due to the down-side risk associated with unexpected jumps in operations or planning costs. We also note that any optimal solution to a stochastic program generated with a sampled set of scenarios is necessarily optimistic with respect to the complete or "true" set of scenarios. Thus, the issue of down-side risk or increased cost is always present – the only question is the degree to which this is the case.

To address this question, we consider the Multiple Replication Procedure (MRP), first introduced by Mak, Morton, and Wood. We now briefly describe the procedure, referring to [5] for the full derivation and details. We suppose we are given a solution $\hat{y}$, a confidence level $0 < \alpha < 1$ for a $1 - \alpha$ confidence level, and n_g sets of scenarios, $\tilde{\mathcal{E}}^i,\ i = 1, \ldots, n_g$ such that each set has equal probability.

The MRP procedure is then defined as follows:

1. For each $i = 1, \ldots, n_g$ compute the gap statistic:

$$G^i = F(\hat{y}, \tilde{\mathcal{E}}^i) - \min_y F(y, \tilde{\mathcal{E}}^i)$$

2. Compute the average gap statistic $\bar{G}$ and the sample standard deviation s_G

The approximate (1-α)-level confidence interval for the optimality gap is then given by

$$\left[0,\ \bar{G}\frac{t_{n_g-1,\alpha} s_G}{\sqrt{n_g}}\right]$$

where $t_{n_g-1,\alpha}$ is the α tail value for a t-distribution with $n_g - 1$ degrees of freedom.

In practice, we typically assume a total of N scenarios are available, generated either via a computer simulation or stochastic process. We then partition these N scenarios as follows, for use in the MRP procedure. We first use $\hat{n} < N$ solutions as the scenarios over which we optimize to get $\hat{y}$, drawn at random without replacement. The remaining samples are then divided up (again at random, without replacement) into n_g samples of size n.

Software Infrastructure

Sandia has developed and actively maintains an environment for both modeling and solving stochastic mixed-integer programs. This software module is called PySP, and is distributed as part of a larger optimization software project known as Coopr (`https://software.sandia.gov/trac/coopr`). Coopr (also developed and maintained by Sandia) is an open-source project, jointly hosted at Sandia and as part of IBM'S COIN-OR (`http://www.coin-or.org`) open-source optimization software project. All Coopr software is written in Python, to both facilitate rapid prototyping and to minimize the barrier to entry for users that are not experts in computer programming.

PySP uses the Coopr Pyomo module to express the deterministic base scenario of an optimization problem, in addition to the scenario tree structure. PySP provides alternative methods for solving the resulting extensive form, either directly (e.g., via a commercial MIP solver) or through decomposition approaches such as Progressive Hedging [6]. PySP can be deployed on a cluster, and has a number of mechanisms to support parallel solution of stochastic programs. All algorithms in PySP are generic, operating on any stochastic programming model expressed using the PySP/Pyomo modeling mechanisms.

As part of this effort, we developed a capability in PySP to execute the Mak, Morton, and Wood MRP procedure, given a generic stochastic programming model expressed in PySP. This script – named *computeconf* – (for Compute Confidence) is now generally available as part of the Coopr distribution. We have used PySP and the *computeconf* script to compute all of the results described below. This capability has since been more broadly used by research groups at the University of Texas and the University of California Davis.

Experimental Results

Using the test case described above, we randomly sample 1000 scenarios. This down-sampling forms the basis for our execution of the MRP procedure, using the GEP SP formulated in our PySP modeling and solver framework. We then execute the MRP procedure for various combinations of the key parameters $\hat{n}$ and n_g, i.e., the number of scenarios used to obtain the baseline solution and the number of groups into which the remaining scenarios are split, respectively. Clearly, there are trade-offs between larger $\hat{n}$ (allowing for a more

$\hat{n}$	n_g	Reference Solution Cost	0.99 Confidence Interval Width
140	5	17.457B	41.646M
140	40	17.549B	429.054M
420	10	17.528B	160.694M
420	40	17.884B	606.710M

Table 1. Results of the MRP procedure for expected cost minimization of the GEP stochastic program. Cost units are USD.

accurate reference solution) and more n_g (allowing for more samples and therefore a tighter confidence interval).

A sampling of results for the case of expected cost minimization are shown in Table 1. We first observe the remarkable stability of the reference solution cost, obtained over a variety of initial samples. This is a strong initial indicator of solution stability and tight confidence intervals, which are confirmed by the MRP computation. In the worst case, the confidence interval width is only 606.710M (USD) out of a total investment of 17.884B (USD). While the gap is significant in absolute terms, the relative proportion is quite small (less than 5%). For an investment planning problem, such a gap is acceptable, especially given potential inaccuracies in both the model and the stochastic process used to generate the scenario sets. More comprehensive results are available, although the presentation of the parameter trends with respect to accuracy is beyond the present scope.

Additionally, we considered a variant of the GEP SP in which the objective was to minimize down-side risk, as expressed via the well-known concept of Conditional Value-at-Risk or CVaR [7]. Here, we find that the confidence interval widths relative to the case of expected cost minimization are roughly 5 to 10 times larger. This is somewhat expected, due to the emphasis of the metric on extreme events. However, the magnitude is surprising, and has practical implications for the difficulty of accurately minimizing risk-oriented performance metrics.

Finally, we briefly discuss the stability of the MRP results, although this is rarely addressed in the literature. Distinct executions of the MRP procedure lead to different partitioning of the base set of N scenarios into the core $\hat{n}$ set and the n_g validation groups. In theory, different replications of the MRP could lead to significantly different results. In practice, at least for the GEP problem we investigate, the discrepancies across multiple replications of the MRP (controlling for all other parameters) is minimal.

Future Research Directions

We have demonstrated the ability to effectively compute cost confidence intervals to a difficult stochastic mixed-integer program, that of electrical grid generation expansion. In this particular instance, the total number of scenarios required to achieve reasonable confidence intervals on cost optimality is surprisingly small – in the case in which we deal with expected cost minimization. However, two key open research challenges remain.

The first challenge involves our ability to compute tight cost confidence intervals when risk-oriented metrics such as CVaR are considered. As expected, confidence interval widths grow relative to the case of cost minimization. An obvious answer to this question is simply to use more scenarios in each validation "bundle", but that approach explodes the computationally difficulty of the bundles. More rigorous statistical techniques, along the lines of importance sampling, should provide a pathway to the use of a smaller number of samples while simultaneously achieving tighter confidence intervals.

The second challenge relates to our understanding of the relationship between confidence interval width, sample size, and problem structure. While the results for our generation expansion problem are satisfying (i.e., positive, in the sense that fewer samples than expected were required to achieve tight confidence intervals), this result is far from universal. For example, we have executed the same MRP procedure on a stochastic programming formulation involving unit commitment with wind generators, with qualitatively different results. In this particular case, we find that "standard" problem sizes considered in the literature yield very unstable estimates of expected cost. This finding implies significantly more scenarios will be required in the case of unit commitment (a critical electrical grid operations problem), which already poses a significant computational challenge at the scale considered in the literature.

Conclusions

A range of core operations and planning problems for the national electrical grid are naturally formulated and solved as stochastic programming problems, which minimize expected costs subject to a range of uncertain outcomes relating to, for example, uncertain demands or generator output. A critical decision issue relating to such stochastic programs is: How many scenarios are required to ensure a specific error bound on the solution cost? Scenarios are the key mechanism used to sample from the uncertainty space, and the number of scenarios drives computational difficultly. We have explored this question in the context of a long-term grid generation expansion problem, using a bounding procedure introduced by Mak, Morton, and Wood. We showed experimental results using problem formulations independently minimizing expected cost and risk. Our results indicate that we can use a surprisingly small number of scenarios to yield tight error bounds in the case of expected cost minimization, which has key practical implications. In contrast, error bounds in the case of risk minimization are significantly larger, suggesting more research is required in this area. Future research into more effective methods with application to risk minimization and other grid operations and planning problems is required.

References

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

[2] CPLEX. `http://www.cplex.com`, July 2010.

[3] Shan Jin. Long term power generation planning under uncertainty. Master's thesis, Iowa Sate University, Department of Industrial and Manufacturing Systems Engineering, 2009.

[4] Shan Jin and Sarah M. Ryan. Power generation expansion planning under uncertainties. In *Proceedings of the 2009 Industrial Engineering Research Conference*, 2009.

[5] W. Mak, D.P. Morton, and R.K. Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Annals of Operations Research*, 24(1–2):47–56, 1999.

[6] R.T. Rockafellar and R.J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.

[7] R. Schultz and S. Tiedemann. Conditional value-at-risk in stochastic programs with mixed-integer recourse. *Mathematical Programming*, 105(2–3):365–386, February 2005.

[8] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory*. Society for Industrial and Applied Mathematics, 2009.

[9] Stein W. Wallace and William T. Ziemba, editors. *Applications of Stochastic Programming*. Society for Industrial and Applied Mathematics, 2005.

DISTRIBUTION:

1	MS	1318	Jean-Paul Watson, 01465
1	MS	1318	Mark D. Rintoul, 01465
1	MS	0899	Technical Library, 9536 (electronic copy)