INFORMS Journal on Computing
Vol. 16, No. 3, Summer 2004, pp. 291–299
ISSN 0899-1499 | EISSN 1526-5528 | 04 | 1603 | 0291

DOI 10.1287/ijoc.1030.0037

On a New Collection of Stochastic Linear Programming Test Problems

K. A. Ariyawansa
Department of Mathematics, Washington State University, Pullman, Washington 99164-3113, USA, ari@wsu.edu

Andrew J. Felt
Department of Mathematics and Computing, University of Wisconsin-Stevens Point, Stevens Point, Wisconsin 54481-3897, USA, afelt@uwsp.edu

The purpose of this paper is to introduce a new test-problem collection for stochastic linear programming that the authors have recently begun to assemble. While there are existing stochastic programming test-problem collections, our new collection has three features that distinguish it from existing collections. First, our collection is web-based with free public access, and we intend to enrich it as new test problems become available. Indeed, we encourage submissions of new test problems. Second, along with the collection we provide documentation of the problems, so that researchers can quickly find information about each family without reading through the original source. Third, all of the data in our collection are provided in SMPS (Birge et al. 1987, Gassmann and Schweitzer 2001) format. In this paper, we provide an introduction to the stochastic linear program, give a brief description of each problem family currently in the test-problem collection, and describe the documentation that accompanies the collection.

Key words: probability, stochastic model applications; programming, stochastic; analysis of algorithms
History: Accepted by Jan Karel Lenstra; received May 2002; revised November 2002, January 2003; accepted February 2003.

1. Introduction

Stochastic programming (Kall and Wallace 1994, Prékopa 1995) has grown in importance in recent decades, because advances in computing power and algorithm development have brought us to a point where realistic problem instances can be solved in a reasonable amount of time. With strong interest in solving such problems and in finding more efficient solution techniques, there has arisen a need for a collection of stochastic programming test problems. A high-quality test-problem collection would not only assist algorithm development by providing a standard set of problems with which to challenge new algorithm implementations, but also the teaching of stochastic programming.

One of the most popular forms of stochastic programming problems is the multistage stochastic linear program with recourse (MSSLP). See §2 for a precise statement of the MSSLP problem. While MSSLPs are growing in popularity, many of the applications are proprietary, and therefore the models are not publicly available. Test-problem collections of MSSLPs exist (King 1988, Holmes 1997). However, they need to be enriched with newer applications. In fact, in some cases, the application associated with the existing test case is not known. Also, it would be helpful if the original applications were described in the notation of the original model, and related to a unified notation such as in §2. In addition, it would be desirable if the data for the test problems is available in SMPS (Birge et al. 1987, Gassmann and Schweitzer 2001), the (emerging) standard for specifying input to software for MSSLPs.

To address the above needs, we have collected a group of 11 problem families from a variety of settings and literature sources. Here and in the rest of the paper we use the following terminology. A problem statement where symbols are used to represent problem data, such as the one given by Equations (1)–(4) for MSSLPs, may be thought of as representing a *problem class*. A problem class is thus a set, and each of its members is uniquely identified by a complete specification of numerical values for the data symbols. We refer to such members as *problem instances*. A *problem family* is a subset of the problem class such that data specifying all problem instances in the family share a common mathematical structure. It is often the case that all possible problem instances that may arise from a specific real-world application belong to a problem family.

The 11 families in the collection all represent MSSLPs, but of various structures and sizes, with randomness occurring in different parts in different problems. For some families, problem instances were explicitly stated in the literature. In other cases, we created the instances based solely on the family description in the literature, and in some cases, there is not yet any instance in the collection.

The problem instances that are available as an *INFORMS Journal on Computing* Online Supplement (see http://joc.pubs.informs.org/) to this paper are all in SMPS format (Birge et al. 1987, Gassmann and Schweitzer 2001). Included is a chapter of problem descriptions. Each section of the chapter covers a single problem family. At the beginning of each section, we give a citation to the original application, a brief description of the problem structure, and, if applicable, the names of the SMPS files for the associated problem instances. We then present a description of the original application, a concise problem statement and, if available, an example of an instance given by the model authors. We have attempted to stay as close to the authors' notation as possible in these subsections. Additionally, where feasible, we present a notational reconciliation, which shows how to transform the notation of the problem into that in §2.

Currently, all problem instances contain only random variables with discrete and finite probability distributions. The primary reason for this is that the authors' first use of the test-problem collection was to test the performance of algorithms for such problems. Moreover, most existing algorithms eventually work with such distributions.

It is the intention of the authors to add families of problems to the collection as new application areas become prominent and to add test-problem instances as they become available. We are especially interested in adding problems with continuous or infinite probability distributions. The collection will continue to be freely available to the stochastic programming community.

In that spirit, we encourage colleagues to submit new problem data with an accompanying description. Such submissions should include the following:

1. description of the application and problem notation;
2. problem statement, in the same notation;
3. numerical example, if practical;
4. reconciliation to the notation of §2;
5. data files in SMPS format for each instance; and
6. optimal solutions for each instance and example.

The rest of the paper is organized as follows. In §2, we present a standard form for the MSSLP and indicate some common simplifications and extensions. In §3, we provide a short description of each of the problem families in the test collection. Section 4 briefly describes the documentation format for a problem family in the chapter that is distributed with the test collection. Finally, we have some concluding remarks in §5.

As part of testing the algorithms developed by Ariyawansa and Jiang (1996), we have also developed C routines that convert SMPS data into a data structure suitable for implementing algorithms for stochastic programs. These routines are available as open source in the Online Supplement to this paper available from this journal's website (http://joc.pubs.informs.org/).

2. A Standard Form for the Multistage Stochastic Linear Program with Recourse

In this section we state a standard form for the multistage stochastic linear program with recourse (MSSLP). The form we state is intended to represent the mathematical statements of the MSSLP for which algorithms, especially those based on cutting-plane notions, are usually formulated.

We begin by describing the underlying probability structure. In this paper, random variables will be represented in boldface while their realizations will be denoted by the same symbols in normal face. We have N sequential discrete time stages with Stage 1 representing the present. Time Stages $2, 3, \ldots, N$ occur in the future sequentially in that order, at which realizations of random variables $\boldsymbol{\xi}_2, \boldsymbol{\xi}_3, \ldots, \boldsymbol{\xi}_N$ become available respectively. Let $\xi_2, \xi_3, \ldots, \xi_N$ be realizations of $\boldsymbol{\xi}_2, \boldsymbol{\xi}_3, \ldots, \boldsymbol{\xi}_N$ respectively. We denote the present by ξ_1 (a realization of $\boldsymbol{\xi}_1$) and for $t = 1, 2, \ldots, N$ refer to $\sigma_t := (\xi_1, \xi_2, \ldots, \xi_t)$ as a *partial scenario*. It is customary in stochastic programming to refer to $\sigma_N := (\xi_1, \xi_2, \ldots, \xi_N)$ as a *scenario*, and our term partial scenario reflects the fact that at Stage t we know only realizations $\xi_1, \xi_2, \ldots, \xi_t$ of a scenario. Note that a partial scenario is not unique to a scenario. We let $\mathcal{S}_t$ denote the set of partial scenarios σ_t for $t = 1, 2, \ldots, N$. The distribution of $\boldsymbol{\xi}_{t+1}$ given σ_t is known for all $\sigma_t \in \mathcal{S}_t$ for $t = 1, 2, \ldots, N-1$.

We now describe the decision structure associated with an MSSLP. For $t = 1, 2, \ldots, N$, decision x_t is made at Stage t after the realization of $\boldsymbol{\xi}_t$ becomes available. The decision x_1 needs to be taken at present from the set $\{x_1\colon A_1 x_1 = b_1, x_1 \geq 0\}$ so that the sum of a direct cost $c_1^\top x_1$ and the expected value $\mathcal{Q}_2(x_1, \xi_1)$ of future decisions to be taken are minimized. The decision x_t for $t \geq 2$ represents a recourse decision that may be taken, depending on the decisions $x_1, x_2, \ldots, x_{t-1}$ that have been taken, partial scenario σ_{t-1} that has been observed, and realization ξ_t of $\boldsymbol{\xi}_t$ just observed, so that the sum of a direct cost $c_t^\top x_t$ and the expected cost $\mathcal{Q}_{t+1}(\chi_t, \sigma_t)$ of the future recourse decisions to be taken are minimized, where $\chi_t := (x_1, x_2, \ldots, x_t)$.

Since there are no future stages after Stage N, we let $\mathcal{Q}_{N+1}(\chi_N, \sigma_N) := 0$ for all χ_N and all σ_N. For $t \leq$

$N-1$, $\mathcal{Q}_{t+1}(\chi_t, \sigma_t)$ is defined recursively in a specific manner. We now state the following standard form for the MSSLP that includes this recursive definition:

$$\begin{aligned} \text{minimize } & c_1^\mathsf{T} x_1 + \mathcal{Q}_2(x_1, \xi_1) \\ \text{subject to } & A_1 x_1 = b_1 \\ & x_1 \geq 0 \end{aligned} \tag{1}$$

where $\mathcal{Q}_2(x_1, \xi_1)$ is defined (with $\sigma_1 := \xi_1$ and $\chi_1 := x_1$) by the recursion

$$\mathcal{Q}_{t+1}(\chi_t, \sigma_t) := E[Q_{t+1}(\chi_t, \sigma_t, \boldsymbol{\xi}_{t+1}) \mid \sigma_t]; \tag{2}$$

$$\begin{aligned} & Q_{t+1}(\chi_t, \sigma_t, \xi_{t+1}) \\ & := \inf_{x_{t+1}} \Big\{ c_{t+1}(\xi_{t+1})^\mathsf{T} x_{t+1} + \mathcal{Q}_{t+2}(\chi_{t+1}, \sigma_{t+1}) \colon A_{t+1}(\xi_{t+1}) x_{t+1} \\ & \qquad = b_{t+1}(\xi_{t+1}) - \sum_{i=1}^{t} T_{(t+1)i}(\xi_{t+1}) x_i, \; x_{t+1} \geq 0 \Big\} \\ & \qquad \text{for } t = 1, 2, \ldots, N-1. \end{aligned} \tag{3}$$

Here ξ_{t+1} is a realization of $\boldsymbol{\xi}_{t+1}$ given σ_t; vectors $c_{t+1}(\xi_{t+1})$, $b_{t+1}(\xi_{t+1})$ and matrices $A_{t+1}(\xi_{t+1})$, $T_{(t+1)i}(\xi_{t+1})$ $(i = 1, 2, \ldots, t)$ are realizations of random variables dependent on $\boldsymbol{\xi}_{t+1}$; $\sigma_{t+1} := (\sigma_t, \xi_{t+1})$, and $\chi_{t+1} := (\chi_t, x_{t+1})$ for $t = 1, 2, \ldots, N-1$; and

$$\mathcal{Q}_{N+1}(\chi_N, \sigma_N) := 0 \tag{4}$$

for all χ_N and σ_N. Note that the input data for the MSSLP in form (1)–(4) above consist of:

first stage deterministic data c_1, b_1, A_1, and
for all $\sigma_{t-1} \in \mathcal{S}_{t-1}$ the conditional distribution of $(\boldsymbol{c}_t(\boldsymbol{\xi}_t), \boldsymbol{b}_t(\boldsymbol{\xi}_t), \boldsymbol{A}_t(\boldsymbol{\xi}_t), \boldsymbol{T}_{ti}(\boldsymbol{\xi}_t)$ $(i = 1, 2, \ldots, t-1))$ given σ_{t-1},
for $t = 2, 3, \ldots, N$. (5)

An important special case of (1)–(4) is the case where the distribution of $\boldsymbol{\xi}_{t+1}$ is independent of $\sigma_t \in \mathcal{S}_t$ for $t = 2, 3, \ldots, N-1$. Many MSSLPs in applications belong to the independent case, and sometimes (see Birge and Louveaux 1997, §3.5, for example) only the independent case is explicitly stated. Since we request contributors of test problems to our collection to provide a notational reconciliation to our standard form, we state this special case of (1)–(4) for convenient reference:

$$\begin{aligned} \text{minimize } & c_1^\mathsf{T} x_1 + \mathcal{Q}_2(x_1) \\ \text{subject to } & A_1 x_1 = b_1 \\ & x_1 \geq 0 \end{aligned} \tag{6}$$

where $\mathcal{Q}_2(x_1)$ is defined (with $\chi_1 := x_1$) by the recursion

$$\mathcal{Q}_{t+1}(\chi_t) := E[Q_{t+1}(\chi_t, \boldsymbol{\xi}_{t+1})]; \tag{7}$$

$$\begin{aligned} & Q_{t+1}(\chi_t, \xi_{t+1}) \\ & := \inf_{x_{t+1}} \Big\{ c_{t+1}(\xi_{t+1})^\mathsf{T} x_{t+1} + \mathcal{Q}_{t+2}(\chi_{t+1}) \colon A_{t+1}(\xi_{t+1}) x_{t+1} \\ & \qquad = b_{t+1}(\xi_{t+1}) - \sum_{i=1}^{t} T_{(t+1)i}(\xi_{t+1}) x_i, \; x_{t+1} \geq 0 \Big\} \\ & \qquad \text{for } t = 1, 2, \ldots, N-1. \end{aligned} \tag{8}$$

Here ξ_{t+1} is a realization of $\boldsymbol{\xi}_{t+1}$; vectors $c_{t+1}(\xi_{t+1})$, $b_{t+1}(\xi_{t+1})$ and matrices $A_{t+1}(\xi_{t+1})$, $T_{(t+1)i}(\xi_{t+1})$ $(i = 1, 2, \ldots, t)$ are realizations of random variables dependent on $\boldsymbol{\xi}_{t+1}$; $\chi_{t+1} := (\chi_t, x_{t+1})$ for $t = 1, 2, \ldots, N-1$; and

$$\mathcal{Q}_{N+1}(\chi_N) := 0 \tag{9}$$

for all χ_N. The data for the MSSLP in form (6)–(9) above consist of:

first stage deterministic data c_1, b_1, A_1, and the distribution of $(\boldsymbol{c}_t(\boldsymbol{\xi}_t), \boldsymbol{b}_t(\boldsymbol{\xi}_t), \boldsymbol{A}_t(\boldsymbol{\xi}_t), \boldsymbol{T}_{ti}(\boldsymbol{\xi}_t)$ $(i = 1, 2, \ldots, t-1))$ for $t = 2, 3, \ldots, N$. (10)

We provide an example in just enough detail to illustrate the difference between the general and the independent cases. It is a small three-stage problem that is based on the problem of Louveaux and Smeers (1988). The challenge is to plan investments in electrical generation plants ($x_1 \in \mathbb{R}^4$) of four types so that uncertain future demand can be met at the least expected cost. Each plant type is operated under one of three total output conditions: High, medium, and low. The recourse decisions $x_2 \in \mathbb{R}^{12}$ and $x_3 \in \mathbb{R}^{12}$ in Stages 2 and 3, respectively, represent output levels for each of the four plant types when operating under each of three operating modes.

A random variable describes the highest level, or "peak," demand for each of the second and third stages. In the model, these demands are right-hand-side coefficients in $\boldsymbol{b}_2$ and $\boldsymbol{b}_3$. Two cases are described in Figure 1, a dependent case and an independent case. Conditional probabilities are shown along the arcs, with the realized demands at the nodes. Notice that in the independent case, the distribution of the Stage 3 demand is independent of the value of the Stage 2 demand, whereas that is not true in the dependent case.

The most common standard for specifying input data for stochastic programs is SMPS (Birge et al. 1987, Gassmann and Schweitzer 2001). This standard was first proposed by Birge et al. (1987) as an extension to the MPS standard for specifying input data for (deterministic) linear programs, and it has recently

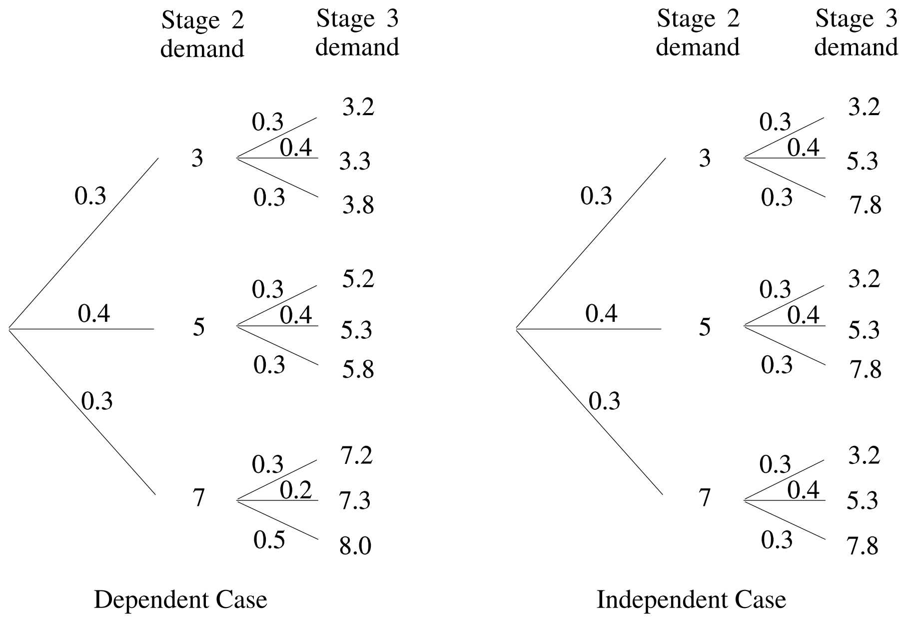

Figure 1 Example

been extended by Gassmann and Schweitzer (2001) to allow inputs to larger classes of problems to be specified. The standard is still evolving and has been implemented to varying degrees in several pieces of software.

A problem instance in SMPS format is a set of three files, the so-called "core" file, "time" file, and "stoch" file. One scenario of the entire problem (i.e., all stages) is specified in the core file. The core file is in MPS format, and the information it contains is arranged as if the problem were a single, large, deterministic linear program with decisions from all stages occurring at once. Indeed, it is often impossible, given just the core file, to guess which variables and constraints belong in which time stage.

That information is contained in the time file. This file allows the problem to be parsed into time stages, so that a deterministic problem in the form (1)–(4) or (6)–(9) may be obtained.

Finally, the stoch file describes the distribution(s) of the random variables in the problem.

Often, the instances of a single problem family all share the same core file and, sometimes, even the same time file. They usually have unique stoch files. The problem instances in this collection are expressed in SMPS format.

As the above description of the ideas behind the SMPS standard indicates, input data for an MSSLP specified according to SMPS is not explicitly in forms (5) or (10). On the other hand, algorithms for the MSSLP (especially those based on cutting-plane notions) are stated as sequences of operations on data in forms (5) or (10). Thus, for implementing algorithms, computer routines are necessary for reading SMPS input data and placing the data into appropriate data structures. In a companion effort, we have released as an Online Supplement to this paper, available from this journal's web site (http://joc.pubs.informs.org/), open source C routines that perform this task. At present our conversion routines do not implement all the features of the SMPS standard. (Gassmann 2001 has also released a set of utility routines for reading SMPS data.)

We conclude this section by highlighting three features covered by the standard forms (1)–(4) and (6)–(9) and the SMPS standard, but not by many formulations of the MSSLP found in the computational stochastic programming literature or by the current version of our input conversion routines.

1. The MSSLP is often stated in the literature with the term $\sum_{i=1}^{t} T_{(t+1)i}(\xi_{t+1})x_i$ in (3) or (8) replaced by the term $T_{t+1}(\xi_{t+1})x_t$. Note that the resulting "staircase" structure amounts to setting $T_{(t+1)i}(\xi_{t+1}) := 0$ for $i = 1, 2, \ldots, t-1$, and relabeling $T_{(t+1)t}(\xi_{t+1})$ by $T_{t+1}(\xi_{t+1})$. This allows only a "one-stage-lag," whereas the standard forms (1)–(4) and (6)–(9) allow "multistage-lags." These multistage-lags also allow the possibility of problem instances with scenarios that allow no recourse actions at some stages. This can be achieved for such scenarios by setting appropriate problem coefficients equal to zero.

2. A common assumption made in stating the MSSLP is uniformity of dimensions of random vectors and matrices for a given stage. In the case of forms (1)–(4) and (6)–(9), this amounts to assuming that $c_t(\xi_t) \in \mathbb{R}^{n_t}$, $b_t(\xi_t) \in \mathbb{R}^{m_t}$, $A_t(\xi_t) \in \mathbb{R}^{m_t \times n_t}$, and $T_{ti}(\xi_t) \in \mathbb{R}^{m_t \times n_i}$ $(i = 1, 2, \ldots, t-1;\ t \geq 2)$, where m_t and

n_t are deterministic for $t = 1, 2, \ldots, N$. The latest version of the SMPS standard (Gassmann and Schweitzer 2001) allows some forms of stochastic dimensions. Indeed, it is to allow for this flexibility that we have tacitly avoided indicating vector and matrix dimensions in forms (1)–(4) and (6)–(9).

3. The distributions of random variables in forms (1)–(4) and (6)–(9) may be continuous or discrete. While discrete distributions are commonly assumed, the SMPS standard (Gassmann and Schweitzer 2001) supports sampling from continuous distributions.

3. The Test-Problem Collection

The 11 families of problems in the test problem collection were drawn from the literature. In selecting the families, we attempted to choose a variety in problem structure, application, size, and solution time.

A summary of information about the test problem collection is provided in Table 1. For each family, its name, a reference to its original source, the number of time stages, and the size of each time stage are given. The column titled "No. of scenarios" indicates the number of scenarios for each problem instance in the family. The number of entries in this column indicates the number of problem instances available. For example, the "environ" family has nine instances, with random realizations ranging from 5 to 32,928. The problem family *bonds* has ten instances available of various sizes and time stages, and we use "var" in Table 1 to indicate this. See §3.11 for more details.

Some families of problems already had problem instances available. These were *stocfor, electric*, and *bonds*. For other families, we were unable, after contacting the authors, to obtain specific problem instances in SMPS format. For those problem families—*airlift, chem, environ, assets, cargo*, and *phone*—we created problem instances based on the corresponding literature source. There is still no problem instance for the *currency* and *RY* problem families.

The instances of *stocfor* available in this test problem collection may be different than those available at the URL cited in Gassmann (1989). We have changed the data in our instances since downloading them.

Table 1 Problem Sizes for the Test-Problem Collection

Family name	Original source	No. of time stages	Stage: sizes	No. of scenarios
airlift	Midler and Wollmer (1969)	2	1: 2×4 2: 6×8	25 25
stocfor	Gassmann (1989)	7	1: 15×15 2: 17×16 3: 17×16 4: 17×16 5: 17×16 6: 17×16 7: 17×16	1 64 512
electric	Louveaux and Smeers (1988)	2	1: 2×4 2: 7×12	3
currency	Klaassen et al. (1990)	4	—	—
RY	Cariño and Ziemba (1998); Cariño et al. (1998)	Multiple	—	—
chem	Subrahmanyam et al. (1994)	2	1: 38×37 2: 45×43	2
environ	Fragnière (1995)	2	1: 48×49 2: 48×49	5 5 15 1,200 1,875 3,780 5,292 8,232 32,928
assets	Mulvey and Vladimirou (1991); Mulvey and Ruszczyński (1995)	2	1: 5×13 2: 5×13	100 32,768
cargo	Mulvey and Ruszczyński (1995)	2	1: 14×52 2: 74×202	2^n for $n = 0, 4, 5, \ldots, 15$
phone	Sen et al. (1994)	2	1: 1×8 2: 23×84	32,768
bonds	Frauendorfer et al. (1997)	var	var	var

The remainder of this section consists of a very short introduction to each of the problem families.

3.1. Airlift Operations Scheduling

This is due to Midler and Wollmer (1969) and is a two-stage, mixed integer linear stochastic problem.

In scheduling monthly airlift operations, demands for specific routes can be predicted. Actual requirements will be known in the future, and they may not agree with predicted requirements. Recourse actions are then required to meet the actual requirements. The actual requirements are expressed in tons, or any other appropriate measure, and they can be represented by a random variable. Aircraft of several different types are available for service. Each of these types of aircraft has its own restriction on number of flight hours available during the month.

The recourse actions available include allowing available flight time to go unused, switching aircraft from one route to another, and buying commercial flights. Each of these has its associated cost, depending on the type(s) of aircraft involved.

The two instances of this family in the collection are both two-stage with probability distributions containing 25 realizations. Random variables appear in the right-hand side of Stage 2. The instances are based on numerical examples given by Midler and Wollmer (1969).

3.2. Forest Planning

This is due to Gassmann (1989) and is a multistage, linear stochastic problem.

The job of a long range forest planner is to decide what parts of the forest will be harvested at what time. Important criteria for such a decision are the age of the trees and the likelihood that trees left standing will be destroyed by fire.

Gassmann creates a set of K age classifications of equal length (e.g., 20 years) and places each portion of the forest into one of the classes, according to the age of the trees within. He also divides the future planning horizon into T rounds, each with a time length equal to that of each age classification. That is, in one time round, any trees that are not destroyed or harvested will move to the next age class.

The decision at each time stage is how much of each classification to harvest, and the risk comes from random fire damage. The three instances for this family originated with Gassmann and have been slightly changed. Each has seven time stages. Random variables occur in the right-hand sides of Stages 2 through T. The three instances have 1, 64, and 512 scenarios, respectively.

3.3. Electrical Investment Planning

This is due to Louveaux and Smeers (1988) and is a two-stage, linear stochastic problem.

Louveaux and Smeers (1988) consider the challenge of planning investments in the electricity generation industry. While the model is, in general, multistage, the specific example given in the study of Louveaux and Smeers (1988) is two-stage. This family has but one very small two-stage instance, with three random realizations for a random variable in the right-hand side of Stage 2.

3.4. Selecting Currency Options

This is due to Klaassen et al. (1990), and is a multistage, nonstaircase, linear stochastic problem.

The situation described by Klaassen et al. (1990) involves a U.S. multi-national corporation (MNC), which has significant forecasted revenues in a foreign currency (FC). If the *exchange rate* (in $US/FC) goes down, the MNC would face declining revenue versus the forecast. To protect, or hedge, against this undesirable possibility, the MNC may choose to purchase *options* that guarantee a certain exchange rate at some point in the future. The guaranteed exchange rate is called the *strike price*.

This model is multistage, nonstaircase, with random variables in the cost, left-hand side and right-hand side. There is yet no instance for this family in the collection, although Klaassen et al. (1990) provide the data to construct one.

3.5. Financial-Planning Model

This is due to Cariño and Ziemba (1998) and Cariño et al. (1998, 1994) and is a multistage, linear stochastic problem.

Cariño and Ziemba (1998) describe a model created for the Yasuda Fire and Marine Insurance Co., Ltd. (Yasuda Kasai) of Tokyo by the Frank Russell Company (Russell) of Tacoma, Washington. The model is a comprehensive investment, liability, and risk-planning tool. It is a multistage linear stochastic model with a steady-state condition imposed on the last stage.

The complexity of the model is such that it cannot be completely described in article format. The model presented is therefore a simplification of the original (Cariño and Ziemba 1998), although it is much more detailed than the abbreviated model presented in an earlier paper (appendix of Cariño et al. 1994). The model contains random variables in the left-hand side and right-hand side. There is yet no instance of this family in the collection.

3.6. Design of Batch Chemical Plants

This is due to Subrahmanyam et al. (1994) and is a multistage, mixed integer linear stochastic problem.

Subrahmanyam et al. (1994) describe the design of a batch-type chemical plant to produce products for which the future demand is unknown. We present only half of the problem given in Subrahmanyam et al. (1994), the "Design SuperProblem."

The decisions include how many plants to build, of what type, when to build them, and how to operate them. Therefore, the problem has some integer decision variables. A single two-stage instance is included in the collection. The instance is based on data provided by Subrahmanyam et al. (1994) and has random variables in the second stage cost and right-hand side coefficients. The joint probability distribution has two realizations.

3.7. Energy and Environmental Planning

This is due to Fragnière (1995) and is a multistage, linear stochastic problem.

The model by Fragnière (1995) assists the Canton of Geneva in planning its energy supply infrastructure and policies. The model is based on the MARKAL (market allocation) model. This is quite an extensive model, containing a great degree of realism. Included is the possibility that emissions of greenhouse gases will be required to decrease. This possibility is expressed in a discrete probability distribution for a right-hand side coefficient.

The model includes equilibrium constraints, capacity-expansion constraints, demand constraints, production constraints, and environmental constraints. Energy is supplied by many different technologies, including hydropower, cogeneration, fossil fuels, urban-waste incineration, and imported electricity. Demands are also classified by technology. Examples are electricity for industrial use, gas furnaces in existing houses, and wood stoves in new houses.

The problem created by Fragnière (1995) for the Canton of Geneva is extremely large and complex, and the input data format is not SMPS. Therefore, we have created nine of our own instances for this family. The numbers in these instances are based on the authors' judgment, not actual economic data. The number of scenarios ranges from 5 to 32,928.

3.8. Network Model for Asset or Liability Management

This is due to Mulvey and Vladimirou (1991) and is a two-stage, linear stochastic problem. See also Mulvey and Ruszczyński (1995).

The management of assets or liabilities can be regarded as a network problem, where the asset categories are represented by nodes and transactions are represented by arcs. The purchase or sale of an asset usually has fixed, deterministic associated costs, while the return on an investment from one stage to the next is usually unknown.

Mulvey and Vladimirou (1991) did not provide data for the numerical examples that they discuss (Mulvey 1999), so we have created two instances, each with two stages. There are five nodes in each stage: Checking, savings, certificate of deposit (CD), cash, and loans. Random coefficients are found in the objective left-hand side and right-hand side of the second stage. The smaller problem has 100 random realizations, while the larger problem has 37,500 realizations.

3.9. Cargo Network Scheduling

This is due to Mulvey and Ruszczyński (1995) and is a two-stage, mixed integer linear or nonlinear stochastic problem.

Mulvey and Ruszczyński(1995) provide a two-stage network problem for scheduling cargo transportation. The flight schedule is completely determined in stage one, and the amounts of cargo to be shipped are uncertain. The recourse actions are to determine which cargo to place on which flights. Transshipment, getting cargo from node m to node n by more than one flight on more than one route, is allowed. When a transshipment is made, cargo must be unloaded at some intermediate node, so that it may be loaded onto a different route going through the same node. Such nodes are called *transshipment nodes*. Any undelivered cargo incurs a penalty. Random variables appear on the right-hand side only.

Mulvey and Ruszczyński (1995) did not provide data for the numerical examples that they discuss (Mulvey 1999). Therefore, we have created some examples from a four-node network. All flights have two legs. That is, including the airport of origin, there are three airports in each flight. Currently, there are 13 instances in the collection, with 2^n scenarios for $n = 0, 4, 5, \ldots, 15$.

3.10. Telecommunication-Network Planning

This is due to Sen et al. (1994) and is a two-stage, mixed integer linear stochastic problem.

The service of providing private lines to telecommunication customers is one with which most people are not familiar. Such service is used by large corporations between business locations for high-speed, private data transmission. Private lines are generally used for a much longer duration than public switched service, and they generally carry more capacity per connection.

A manager of such a network must be constantly looking to the future, deciding where and how much to expand capacity. In this problem formulation, the "how much" is decided beforehand, to some extent, by the imposition of an overall budget. Within the constraints of the budget, expansion is not penalized. The goal is to minimize the unserved requests, while staying within budget.

Such networks are usually very interconnected, so that for any point-to-point demand pair, there is usually more than one route that may service the demand. Each route is made of one or more direct links.

The resulting model is a two-stage network model with a stochastic (right-hand side) demand variable in the second stage. We have created an instance with $2^{15} = 32{,}768$ scenarios and six nodes.

3.11. Bond Investment Planning

This is due to Frauendorfer et al. (1997) and is a multistage, linear stochastic problem.

Frauendorfer et al. (1997) describe a suite of test problems for multistage stochastic programming, based on bond investments.

Many business ventures are financed by lending bonds, and many of these ventures also purchase bonds. There is risk in such dealings, as returns on bonds fluctuate, and earnings from the business ventures are uncertain. This risk cannot be modeled by assuming a mean rate of return. Therefore, the situation is a good one for the application of stochastic programming.

The model is multistage with random variables in the objective and right-hand side. The test problems are denoted SGPFmYn, where $m \in \{3, 5\}$ and $n \in \{3, 4, 5, 6, 7\}$. The number of time stages is n, while m is the maximum maturity of the bond pool. The number of scenarios for the SGPF3Yn instances are 25, 125, 625, 3125, and 15,625. Those for the SGPF5Yn instances are the same. The problems were obtained from Birge's POSTS web site (Holmes 1997).

4. Collection Documentation

Along with the data representing the specific problem instances, we are making available as part of an Online Supplement to this paper a manuscript. The Online Supplement is available from this journal's website (http://joc.pubs.informs.org/). The goal of the manuscript is to provide a standard reference where researchers can quickly find the information they need without spending a lot of time studying the original literature source.

To this end, the document contains a section of identical format for each problem family. Each section contains four subsections. Here, we briefly describe and give the motivation for each subsection.

4.1. Description

The real-world application is described, using the notation of the original source. This allows the researcher to connect the problem instances they are solving to the applications from which they came. Researchers who want to draw conclusions about the types of problems for which certain algorithms are most suited may find this section helpful.

Practitioners with similar applications can also use the description to help them model their own situations and more easily choose the right algorithm, based on performance on a similar application.

4.2. Problem Statement

The problem statement shows the mathematical LP (or NLP) in a single statement. Often, this is simply a collection of all the equations from the Description subsection. The notation is still that of the original source.

This subsection can be useful for those who are perhaps not interested in the real-world application but simply want to see what structure the stochastic linear program has. Many of the original sources do not state the problem in a single statement. In such a case, considerable time can be spent reading the original source and writing the mathematical problem.

4.3. Numerical Example

In this subsection, specific instances are discussed. If practical, all of the problem data and an optimal solution are given for each instance. Where this is impractical, reference is made to specific data files.

In each case, an attempt was made to use data from the original source. In some cases, not all of the problem data were included in the original article, and in some of those cases, the original authors were unable to provide original data. In many of these cases, we created data in an attempt to provide a problem instance in the spirit of the original source.

This subsection uses the notation of the original source.

4.4. Notational Reconciliation

This subsection details how one would transform the problem from the notation of §4.2 to the notation of §2. This might be useful for those who do not want to learn the notation of the original source or for computer programmers who wish to transform each problem to a standard data format.

One of the reasons for providing the documentation is to lend cohesion to the test-problem collection. To understand the performance of our algorithms, researchers need more than just the test data. Being able quickly to look up problem background, notation, and structure can be an efficient means to a deeper insight on algorithm performance.

5. Conclusion

The initial motivation for the present work came from the need to test carefully the new polynomial interior cutting-plane algorithms for stochastic programs developed by Ariyawansa and Jiang (1996). The design of our test-problem collection was motivated and influenced by the elegant work of Moré (1990) and Averick et al. (1991) on test-problem collections for several classes of (deterministic) nonlinear optimization problems and by the impact such collections have had on the development of software for such problem classes (Moré and Toraldo 1991, Lin and Moré 1999, Benson et al. 2001).

As with all such projects, this is an evolving effort. The test collection will continue to change as new applications become prominent and as data formats improve. While at present all the test problems in our collection have discrete distributions with a finite number of realizations, we anticipate adding test problems with continuous distributions. We encourage the stochastic-programming community to assist us by providing their problem descriptions and data. It is through the process of updating the test problem collection that its relevance may be maintained.

Acknowledgments

The work of the first author was supported in part by the U.S. Army Research Office under Grant DAAD19-00-1-0465. The second author was supported in part by Washington State University under a research assistantship during the summer of 2000. The authors appreciate the constructive comments given by the referees.

References

Ariyawansa, K. A., P. L. Jiang. 1996. Polynomial cutting plane algorithms for two-stage stochastic linear programs based on ellipsoids, volumetric centers and analytic centers. Technical report, Department of Pure and Applied Mathematics, Washington State University, Pullman, WA. http://www.wsu.edu.

Averick, B. M., R. G. Carter, J. J. Moré. 1991. The Minpack-2 test problem collection (preliminary version). Technical report ANL/MCS-TM-150, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL.

Benson, S. J., L. C. McInnes, J. J. Moré. 2001. A case study in the performance and scalability of optimization algorithms. *ACM Trans. Math. Software* **27** 361–376.

Birge, J., M. Dempster, H. Gassmann, E. Gunn, A. King, S. Wallace. 1987. A standard input format for multiperiod stochastic linear programs. *COAL Newslett.* (17) 1–19. http://www. mgmt.dal.ca.

Birge, J. R., F. R. Louveaux. 1997. *Introduction to Stochastic Programming*. Springer-Verlag, New York.

Cariño, D. R., T. Kent, D. H. Myers, C. Stacy, M. Sylvanus, A. L. Turner, K. Watanabe, W. T. Ziemba. 1994. The Russell-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming. *Interfaces* **24** 29–49.

Cariño, D. R., D. H. Myers, W. T. Ziemba. 1998. Concepts, technical issues, and uses of the Russell-Yasuda Kasai financial planning model. *Oper. Res.* **46** 450–462.

Cariño, D. R., W. T. Ziemba. 1998. Formulation of the Russell-Yasuda Kasai financial planning model. *Oper. Res.* **46** 433–449.

Fragnière, E. 1995. *Choix énergétiques et environnementaux pour le Canton de Genève*. Ph.D. thesis, Thèse No. 412. Départment HEC, Université de Genève, Geneva, Switzerland.

Frauendorfer, K., C. Marohn, M. Schürle. 1997. SG-portfolio test problems for stochastic multistage linear programming (II). Technical report, Institute of Operations Research, University of St. Gallen, St. Gallen, Switzerland.

Gassmann, H. I. 1989. Optimal harvest of a forest in the presence of uncertainty. *Canadian J. Forest Res.* **19** 1267–1274. See also http://www.netlib.org.

Gassmann, H. I. 2001. Utility routines for working with the SMPS input format for stochastic linear programs. http://www.mgmt.dal.ca.

Gassmann, H., E. Schweitzer. 2001. A comprehenseve input format for stochastic linear programs. *Ann. Oper. Res.* **104** 89–125.

Holmes, D. 1997. A (PO)rtable (S)tochastic programming (T)est (S)et (POSTS). http://users.iems.nwu.edu.

Kall, P., S. Wallace. 1994. *Stochastic Programming*. Wiley, Chichester, U.K.

King, A. 1988. Stochastic programming problems: Examples from literature. Ermoliev, Y., R. J-B. Wets, eds. *Numerical Techniques for Stochastic Optimization*, Chap. 30, Springer-Verlag, New York, 543–567.

Klaassen, P., J. F. Shapiro, D. E. Spitz. 1990. Sequential decision models for selecting currency options. Technical report IFSRC No. 133-90, International Financial Services Research Center, Massachusetts Institute of Technology, Cambridge, MA.

Lin, C., J. J. Moré. 1999. Newton's method for large bound-constrained optimization problems. *SIAM J. Optim.* **9** 1100–1127.

Louveaux, F. V., Y. Smeers. 1988. Optimal investments for electricity generation: A stochastic model and a test-problem. Ermoliev, Y., R. J-B. Wets, eds. *Numerical Techniques for Stochastic Optimization*, Chap. 24, Springer-Verlag, New York, 445–453.

Midler, J. L., R. D. Wollmer. 1969. Stochastic programming models for scheduling airlift operations. *Naval Res. Logist. Quart.* **16** 315–330.

Moré, J. J. 1990. A collection of nonlinear model problems. *Computational Solution of Nonlinear Systems of Equations*, Vol. 26, *Lectures in Applied Mathematics*. American Mathematical Society, Providence, RI, 723–762.

Moré, J. J., G. Toraldo. 1991. On the solution of large quadratic programming problems with bound constraints. *SIAM J. Optim.* **1** 93–113.

Mulvey, J. M. 1999. Re: Stochastic LP examples. Personal e-mail from author.

Mulvey, J. M., A. Ruszczyński. 1995. A new scenario decomposition method for large-scale stochastic optimization. *Oper. Res.* **43** 477–490.

Mulvey, J. M., H. Vladimirou. 1991. Applying the progressive hedging algorithm to stochastic generalized networks. *Ann. Oper. Res.* **31** 399–424.

Prékopa, A. 1995. *Stochastic Programming*, Vol. 324, *Mathematics and Its Applications*. Kluwer Academic Publishers Group, Dordrecht, The Netherlands.

Sen, S., R. D. Doverspike, S. Cosares. 1994. Network planning with random demand. *Telecomm. Systems* **3** 11–30.

Subrahmanyam, S., J. F. Pekny, G. V. Reklaitis. 1994. Design of batch chemical plants under market uncertainty. *Indust. Engrg. Chemical Res.* **33** 2688–2701.