A Collection of Stochastic Programming Problems

Derek Holmes
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor MI 48109-2117

Technical Report 94-11

April 1994

*Supported in part by National Science Foundation under Grant DDM-9215921

A Collection of Stochastic Programming Problems

Derek Holmes
Department of Industrial and Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

April 4, 1994.

Abstract: This technical report describes in some detail a collection of stochastic programming problems. The collection consists of practical problems formulated in the literature and from other sources, most of which are based on real formulations. Problem characteristics and constraint nonzero structures are given, as well as selected optimal solutions. This report only includes problems which are available in machine readable format from the University of Michigan.

Contents

List of Figures

List of Tables

1 Introduction

Over the past decade, a number of practical, real-world stochastic programming problems have appeared in the literature. These problems are generally based on deterministic formulations and modified to include some form of stochasticity. Under the direction of Professor John Birge, the Industrial and Operations Engineering Department at the University of Michigan has undertaken to collect as many of these problems as possible. This technical report will briefly describe the problems that have been collected. Annotated formulations will also be provided wherever possible, as well as the original sources of problems.

This report is arranged in several sections. The first section very briefly describes a general stochastic programming formulation. The following sections classify the collection by their application class. Table 1 shows these application classes and the problems within them. The final section will describes a deterministic equivalent generation package and give details about obtaining the problems in machine readable form.

Class	Problem name
Production and Scheduling Problems	SCFXM1, CEP1, STORM
Expansion and Planning	SC205, SCAGR, SCRS, PLTEXP, SSN, STOCHFOR
Financial Problems	Mulvey and Vladimirou, OPTN, ALM
Other	SCTAP, SCSD, Birge and Louveaux

Table 1: Application Classes

A few acknowledgements must be given before describing the collection. Prof. John Birge has been the driver behind the collection, and has personally collected many of the problems here. Prof. Survajeet Sen from the University of Arizona has also provided many problems, as well as Prof. Hercules Vladimirou of Princeton University. Without their help, this collection would not exist.

This technical report will first review stochastic programming formulations and standardized input formats, and then describe each application shown above.

2 Stochastic Programming Formulations and Formats

Stochastic programs are optimization problems where some of the data is not known with certainty, but instead is described over a multidimensional probability space. Since decisions must generally be made before uncertainty is resolved, stochastic programs alternate decisions with realizations of random data. At each decision point, the cost of the current decision and the expected cost of future decisions is optimized. These observations lead to the general formulation for a multistage stochastic program with recourse. This formulation can be written as:

$$\min_{x_0} f_0(x_0) + E_{\boldsymbol{\xi}_1}[\boldsymbol{f}_1(\boldsymbol{x}_1) + E_{\boldsymbol{\xi}_2|\boldsymbol{\xi}_1}[\boldsymbol{f}_2(\boldsymbol{x}_2) + \ldots + E_{\boldsymbol{\xi}_H|\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_{H-1}}[\boldsymbol{f}_H(\boldsymbol{x}_H)]]]$$

$$\begin{array}{lll} \text{subject to} & g_{i,0}(x_0) \le 0 & i = 1,\ldots,m_0 \\ & \boldsymbol{g}_{i,1}(x_0,\boldsymbol{x}_1) \le 0 & i = 1,\ldots,m_1 \ a.s. \\ & \boldsymbol{g}_{i,t}(\boldsymbol{x}_{t-1},\boldsymbol{x}_t) \le 0 & i = 1,\ldots,m_t, t = 2,\ldots,H \ a.s. \\ & \boldsymbol{x}_t \in X_t & t = 0,\ldots,H. \end{array}$$

where bold-face vectors represent (possibly) stochastic data or decisions. The stochastic data $\boldsymbol{\xi}_t$ is defined over some canonical probability space $(\Xi_t, \sigma(\Xi_t), \mathcal{P}_t)$ and may include constraint data, constraint functional forms and/or objective data.

Most optimization problems to which stochasticity has been applied are linear or integer problems. Even so, the expectations in (2) involve multidimensional integration of implicitly defined functions. To avoid this difficulty, the probability spaces are generally assumed to be discrete. This allows the objective to be written as a finite sum and the constraints to be enforced by replicating them for each element in Ξ_t.

A multistage linear formulation can be derived from (2) to get

$$\min_{x_0} \; cx_0 + E_{\boldsymbol{\xi}_1}(\min_{\boldsymbol{x}^1} \boldsymbol{c}^1\boldsymbol{x}^1 + \ldots E_{\boldsymbol{\xi}_H|\boldsymbol{\xi}_1\cdots\boldsymbol{\xi}_{H-1}}(\min_{\boldsymbol{x}^H} \boldsymbol{c}^H\boldsymbol{x}^H)\ldots)$$

$$\begin{array}{lll} \text{s.t.} & Ax^0 & = b \\ & \boldsymbol{T}^0 x^0 + \boldsymbol{W}^1\boldsymbol{x}^1 & = \boldsymbol{h}^1 \quad a.s. \\ & \vdots & \vdots \\ & \boldsymbol{T}^{H-1}\boldsymbol{x}^{H-1} + \boldsymbol{W}^H\boldsymbol{x}^H & = \boldsymbol{h}^H \quad a.s. \\ & l^0 \le x^0 \le u^0, \boldsymbol{l}^t \le \boldsymbol{x}^t \le \boldsymbol{u}^x & t = 1, \ldots, H \quad a.s. \end{array}$$

where the stochastic elements are defined over a discrete canonical probability space $(\Xi, \sigma(\Xi), P)$. Here, $\Xi = \Xi_1 \otimes \cdots \otimes \Xi_H$, and the S_t elements of Ξ_t are $\{(T_{ts}, W_{ts}, \xi_{ts}, c_{ts}), s = 1, \ldots, S_t\}$.

The requirement that each stochastic constraint must hold almost surely may be written deterministically by defining a set of constraints for each realization. These constraints will relate a decision in period t (which depends on the history of realized data up to that point) to all possible decisions in period $t+1$. A decision made in periods $t+1$ will depend on what happened since period t, so decision variables for each possible realization in period t must be defined. These decisions can be visualized graphically as a tree, where nodes represent decisions and arcs represent stochastic realizations. So, at a node corresponding to some sequence of realizations, a decision maker must solve a stochastic program corresponding to all future events.

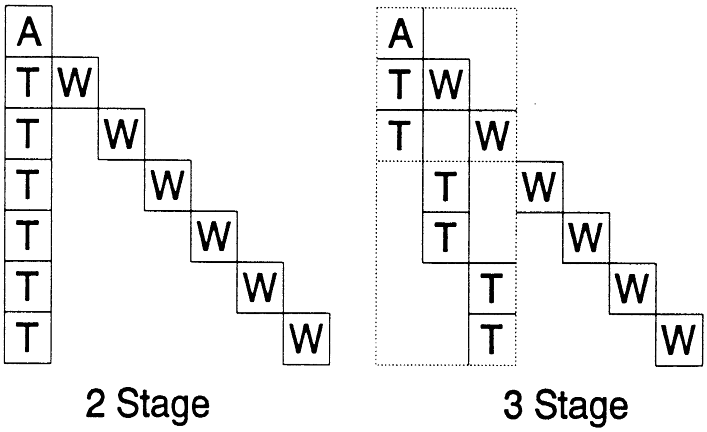

Figure 1: Block structure for deterministic equivalents

The block structure of two deterministic equivalents of 2 is shown in Figure 1. The left hand side shows the block structure for a two stage problem with six realizations, while the right side shows the block structure for a 3 stage problems with 2 scenarios in each stage.

All problems described here are similar to those shown. Deviations from this paradigm will be noted where necessary. The machine readable problems are in one of two forms. The most common is in the electronic format suggested by Birge, et. al. [2], which splits the problem into three files:

1. CORE file: A core file contains the constraint matrices for a "sample path" along a multistage decision tree. The constraint matrix, in MPS format (see, e.g. OSL [16] includes constraints and objectives for all periods.

2. TIME file: A time file splits the CORE file into periods. The time file contains records defining the uppermost lefthand corner of each periods' submatrix for that period's decisions (i.e. W matrix).

3. STOCH file: A stoch file describes the actual probability distributions (generally discrete) for each stochastic data element. The realizations effectively replace the base values in the CORE file. Joint probability distributions are calculated from the (marginal) distributions contained in the STOCH file.

The second format directly forms the deterministic equivalents from the stochastic program. The models are written in the AMPL programming language (Fourer, Gay, and Kernighan, [7]). Where possible, determinisitic equivalent MPS files are also provided.

3 Production Problems

This section contains descriptions of formulations for shorter-term production and scheduling problems. Subproblem dimensions and selected problem solutions are shown in Tables 5 and 6.

3.1 SCFXM1

SCFXM1 is real-world production scheduling problem of unknown origin. The problem is two stage, but can be extended to a multistage problem. The problem originated in Ho and Loute [15], and has been extended by Birge [1], Gassman (MSLIP, [9]) and Birge, et. al. (STOPGEN, [3]). The files that are available in the test sets are summarized in Table 2. A picture of the nonzero structure of the CORE file is shown in Figure 2.

Type	File list	Comments
MSLIP		
CORE	`CORE.6`	
TIME	`TIME2.6`, ..., `TIME4.6`	2, 3, and 4 stage versions
STOCH	`DISTO.6`, ..., `DIST6.6`	3 Stochastic RHSs, 8,27,64,96,128,256 scenarios
STOPGEN		
CORE	`scfxm1.cor`	
TIME	`scfxm12.tim`	2 Stage only
STOCH	`scfxm12r.`x	Stochastic RHSs, $x = 4, 8, 16, 27, 32, 64, 96, 128, 256$ scenarios
STOCH	`scfxm12c.`x	Stochastic Objective, $x = 4$ scenarios
STOCH	`scfxm12b.`x	Stochastic Objective and RHSs, $x = 4, 16, 64$ scenarios

Table 2: SCFXM files

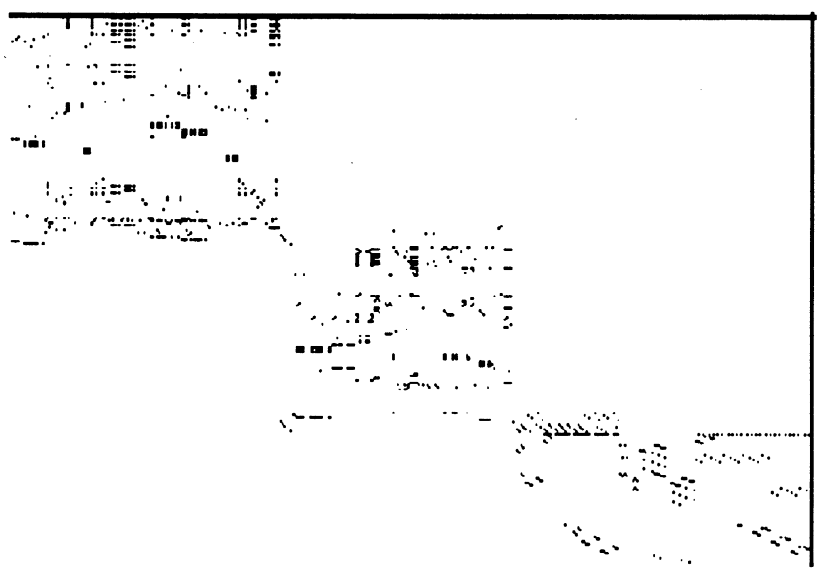

Figure 2: SCFXM1 nonzero structure

3.2 CEP1

CEP1 is a two-stage machine capacity planning problem provided by Survajeet Sen (see, e.g. Higle and Sen, [11], Appendix B.) Their description, without the data, follows:

Formulation

CEP1 is an example of a problem in which a manufacturing plant produces m different part types on any one of n different machines. Machine j has a fixed number of hours, h_j, available per week. The number of hours of new capacity of type j acquired is denoted by x_j and costs c_j dollars per hour. The total usage of machine j should not exceed U_j hours per week. Machine j requires t_j hours of maintenance for every hour of operation. The total scheduled maintenance for all machines is not to exceed T hours per week.

Part i can be produced on each machine j at a rate of a_{ij} parts per hour at a cost (labor plus tooling cost) of q_{ij}. Each week, an order of ω_i units of part i must be met. If the total demand exceeds the total capacity then the excess parts are obtained from a subcontractor at a premium price, p_i, for part i. It is assumed that $p_i > q_{ij}$ for all parts i and machines j. The weekly demands are treated as i.i.d. random variables with known distribution.

With the objective of minimizing the cost of new capacity plus the expected cost of weekly labor plus retooling, a two stage stochastic program can be formulated as follows:

$$\min \sum_{j=1}^{n} c_j x_j + E[\min \sum_{i=1}^{m} \sum_{j=1}^{m} q_{ij} y_{ij} + \sum_{i=1}^{m} p_i s_i]$$

$$\begin{array}{lll} \text{subject to} & -x_j + z_j \leq h_j & j = 1, \ldots, n, \\ & \sum_{j=1}^{n} t_j z_j \leq T, & \\ & \sum_{j=1}^{n} a_{ij} y_{ij} + s_i \geq \omega_i & i = 1, \ldots, m, \\ & \sum_{i=1}^{m} y_{ij} \leq z_{ij} & j = 1, \ldots, n, \\ & 0 \leq z_j \leq U_j, \quad x_j, y_{ij} s_i \geq 0. & \end{array}$$

The data files and nonzero structure of the CORE file are shown below.

Type	File list	Comments
CORE	`cep1.cor`	
TIME	`cep1.tim`	Two stage program
STOCH	`cep1.sto`	3 stochastic RHSs with 6 realizations each (216 scenarios)

Table 3: CEP1 files

3.3 STORM

STORM is a two period freight scheduling problem described in Mulvey and Ruszczyński [24]. (The problem was provided to the University of Michigan courtesy of Adam Berger). In this model, routes are scheduled to satisfy a set of demands at stage 1, demands occur, and unmet demands are delivered at higher costs in stage 2 to account for shortcomings.

This model has been tested with Stochastic Decomposition, (Sen, Mai, and Higle [25], Regularized Decomposition [24]), and Nested Decomposition [5].) In all cases, the RHS realizations were uniformly distributed about a $\pm$ 20% interval from the reference values in the CORE file.

Data files for this problem are listed in Table 4. Due to the size of the problem, its nonzero structure is shown on a separate page at the end of this technical report. Subproblem sizes and selected solutions are shown in the subsequent tables.

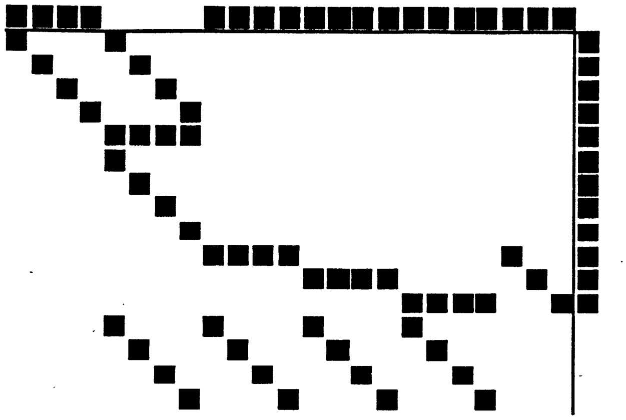

Figure 3: CEP1 nonzero structure

Type	File list	Comments
CORE	`storm1.core`	
TIME	`storm1.time`	Two stage program
STOCH	`storm1.stoch`	Stochastic RHSs in SCENARIO format, 1 scenarios.
	`storm5.stoch`	Stochastic RHSs in SCENARIO format, 5 scenarios.
	`storm10.stoch`	Stochastic RHSs in SCENARIO format, 10 scenarios.
	`storm50.stoch`	Stochastic RHSs in SCENARIO format, 10 scenarios.
Det. Equiv	`storm1.mps`, `storm5.mps`,`storm10.mps`	
Other	`storm.rhs.c`	Code to generate RHS scenarios.

Table 4: STORM files

4 Expansion and Planning Problems

Most stochastic programming problems in the literature have modeled long-term planning problems. These problems lend themselves to stochastic programming, because economic factors which determine the ultimate cost of an expansion plan are always a priori unknown. Since mist economic factors evaolve as stochastic processes, and corrective decisions can be made at any point in time, the models are generally multistage in nature. This section will describe several of these problems.

Problem	A matrix	T matrix	W matrix
SCFXM1	92 x 114	82 x 99	66 x 126
CEP1	9 x 8	7 x 8	7 x 15
STORM	126 x 289	347 x 289	347 x 769

Table 5: Production problem block sizes

Problem	Instance	Objective
SCFXM1	`scfxm12r.4`,`scfxm12r.32`, `scfxm12r.128`	2877.5639, 2877.5639, 2891.7061
CEP1	`cep1.sto`	179.9998
STORM	`storm10.stoch`	15561057.514
	`storm50.stoch`	14668965.782

Table 6: Selected production problem solutions

4.1 SCAGR

SCAGR is a multiperiod dairy farm expansion model. The problem first appeared in deterministic form in a chapter by Swart, Smith, and Holderby [28]. The model was subsequently collected in electronic form by Ho and Loute [15] and modified by Birge [1] and Gassman [9].

SCAGR seeks to maximize the revenue associated with a herd expansion policy in the face of uncertain crop yields. The herd is split into fours groups (corresponding to age), each of which has a unique set of feed requirements. The feed types are corn, silage, hay, or haylage, denoted here with (C, S, A, H). They are either purchased or planted on the total available acreage, which is assumed to be stochastic. A synopsis of the model appears here; refer to [28] for details.

Let the decisions or variables be:

$$
\begin{aligned}
p^b_{it}, p^h_{it} &= \text{Population of bulls (heifers) in group } i \text{, period } t, i = 1, \ldots, 4,\\
b^b_t, b^h_t &= \text{Births of bulls (heifers) in period } t,\\
s^b_{it}, s^h_{it} &= \text{Bulls (heifers) sold in group } i \text{, period } t, i = 1, \ldots, 4,\\
a_{ct} &= \text{Acres planted of crop } c \in \{C, S, A, H\} \text{ in period } t,\\
pu_{ct} &= \text{Units purchased of crop } c \in \{C, S, A, H\} \text{ in period } t,\\
i_{ct} &= \text{Inventory of crop } c \in \{C, S, A, H\} \text{ in period } t,
\end{aligned}
$$

The data for the problem is summarized below.

$$
\begin{aligned}
CS_{ict} &= \text{Feed } c \text{ consumption of group } i \text{ in period } t,\\
SP^h_{it}, SP^b_{it} &= \text{Selling price of bulls (heifers) in group } i \text{, period } t,\\
REV_t &= \text{Milk revenue per head in period } t,\\
UC^h_{it}, UC^b_{it} &= \text{Upkeep cost of bulls (heifers) in group } i \text{, period } t,\\
PLC_{ct}, PUC_{ct}, INVC_{ct} &= \text{Planting, purchasing, and inventory cost of crop } c \text{ in period } t,\\
YLD_{ct} &= \text{Yield per acre planted of crop } c \text{ in period } t,\\
A_t, L_t &= \text{Acres available and minimum amount in alfalfa crops in period } t,\\
CAP_c &= \text{Storage capacity for crop } c.
\end{aligned}
$$

The formulation then is

$$
\max \sum_{t=1}^{T} [\sum_{i=1}^{3} SP^h_{it} s^h_{it} + SP^b_{it} s^b_{it} + REV_t p^h_{4t} \tag{4.1}
$$

$$
- \textstyle\sum_i (UC^h_{it} p^h_{it} + UC^b_{it} p^b_{it}) - \sum_c (INVC_{ct} I_{ct} + PLC_{ct} a_{ct} + PUC_{ct} pu_{ct})], \tag{4.2}
$$

subject to herd expansion equations:

$$
p^b_{1t} + s^b_{1t} = 0.5 p^h_{4t}, \quad p^h_{1t} + s^h_{1t} = 0.5 p^h_{4t}, \tag{4.3}
$$

$$p^h_{2t} = p^h_{1t}, \quad p^b_{2t} = p^b_{1t},$$
$$p^h_{3,t+1} + s^h_{3,t+1} = p^h_{2t}, \quad p^b_{3,t+1} + s^b_{3,t+1} = p^b_{2t}, \quad s^b_{3t} = p^b_{3t},$$
$$p^h_{4,t+1} = p^h_{3t} + 0.7p^h_{4t},$$
$$p^h_{4,t+1} \geq p^h_{4t}, \quad t = 1, \ldots, 4.$$

Crop inventory and planting restrictions are:

$$I_{c,t+1} = I_{ct} + YLD_{ct}a_{ct} - \sum_{i=1}^{4} CS_{ict}p_{it}, \quad c \in \{C, S, A, H\}, t = 1, \ldots, T \tag{4.4}$$

$$0 \leq I_{ct} \leq CAP_c, \quad c \in \{C, S, A, H\}, t = 1, \ldots, T$$
$$\sum_c a_{ct} \leq A_t, \quad a_{At} + a_{Ht} \geq L_t \quad t = 1, \ldots, T$$

There are two ways to model stochastic crop yield. The easiest is to consider the yields to be stochastic, but the most computationally efficient is to consider the number of acres to be stochastic. In some formulations (MSLiP, [9] and STOPGEN [3]), the inventory restriction is considered stochastic.

Data files for this problem are listed in Table 7, and a picture of the problem's nonzero structure is shown in Figure 4. Problem characteristics and a sample solution is summarized in Tables 14 and 15 at the end of this section.

Type	File list	Comments
MSLIP		
CORE	`CORE.4`	
TIME	`TIME2.4`, `TIME3.4`	2, and 3 stage versions
STOCH	`DISTO.4`, ..., `DIST7.4`	3 Stoch. RHSs, 8,27,54,108,324,432,864 scenarios
STOPGEN1		
CORE	`scagr7.cor`	
TIME	`scagr72.tim`	2 Stage only
STOCH	`scagr72r.`x	Stoch. RHSs, $x = 4, 8, 16, 27, 32, 54, 64, 108, 324, 864$ scenarios
STOCH	`scagr72c.`x	Stoch. Objective, $x = 4$ scenarios
STOCH	`scagr72b.`x	Stoch. Objective and RHSs, $x = 4, 16, 64$ scenarios
STOPGEN2		
CORE	`agr7e3.sc` ,..., `agr7e7.sc`	3,4,5,6,7 stage versions.
TIME	`agr73.tim` ,..., `agr77.tim`	3,4,5,6,7 stage versions.
STOCH	`scagr`t`.s2`	2 RHS realiz. per pd.: $t = 3, \ldots, 7$ periods
STOCH	`scagr`t`.s16`	16 RHS realiz. per pd.: $t = 3, \ldots, 7$ periods

Table 7: SCAGR files

4.2 SSN

SSN is a telephone switching network expansion planning problem, discussed in Sen, Doverspike, and Cosares [26]. The model seeks to add capacity (lines) to a network of existing point-to-point connections so as to minimize the number of unfilled requests for service. Demand is defined as the number of requests for connections at a given instance of time, and is stochastic.

The formulation supposes there is an existing network connecting m pairs of points. Let $R(i), i = 1, \ldots, m$, be the set of routes that can be used to connect those points. The total new capacity that can be allocated

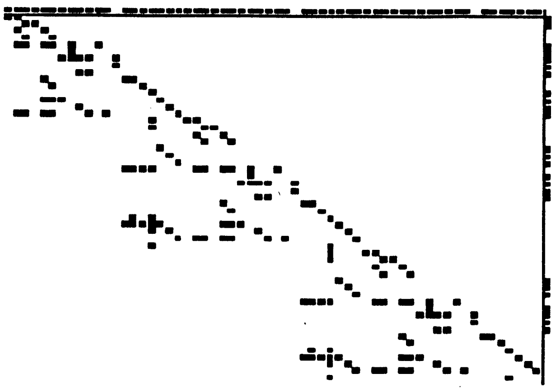

Figure 4: SCAGR nonzero structure

is denoted b, and the vector of existing capacities is e. Letting the decisions variables be

x_j	$j = 1, \ldots, n$	additional capacity added to link j
f_{ir}	$r \in R(i), i = 1, \ldots, m$	number of connections associated with pair i using route r
s_i	$i = 1, \ldots, m$	Number of unserved requests of pair i,

the two stage problem is

$$\min_x E[\min_s \sum_{i=1}^{m} s_i]$$

subject to

$$\begin{aligned} \sum_j x_j &\leq b \\ \sum_i \sum_{r \in R(i)} A_{ir} f_{ir} &\leq x + e, \\ \sum_{r \in R(i)} f_{ir} + s_i &= \boldsymbol{d}_i, \quad i = 1, \ldots, m, \\ f_{ir}, a_i, x_j &\geq 0, \end{aligned}$$

where A_{ir} is the capacity that can be added to pair i using route r, and $\boldsymbol{d}$ is the stochastic call demand. The network model includes 31 nodes, 86 demand pairs, and 82 stochastic demands. There are 89 links and 709 routes. The demands are independent. The model has been approximately solved using Stochastic Decomposition in [26], (using 5-10 samples per r.v.), and using nested decomposition by the author. Data files for this problem are listed in Table 8, and a picture of the problem's nonzero structure is down in Figure 5. Again, problem characteristics and a sample solution is summarized in the e last two tables at the end of this section.

4.3 SC205

SC205 is a dynamic multisector planning problem adapted by Ho and Loute [15] from Manne's ETA model [21]. The history of the model is unclear, but SC205 appears to be model whose goal is to determine a free market's ability to supply energy to an economy. (The row and column names are generic, and provide no information.) Since the model appears to be a feasibility problem, it probably was formulated to find equilibrium points where shadow prices of energy sources are equal to their marginal costs of supply.

Type	File list	Comments
CORE	`ssn.cor`	Core file
TIME	`ssn.time`	Two stage problem only
STOCH	`ssn.sto`	5-10 Outcomes per r.v.

Table 8: SSN files

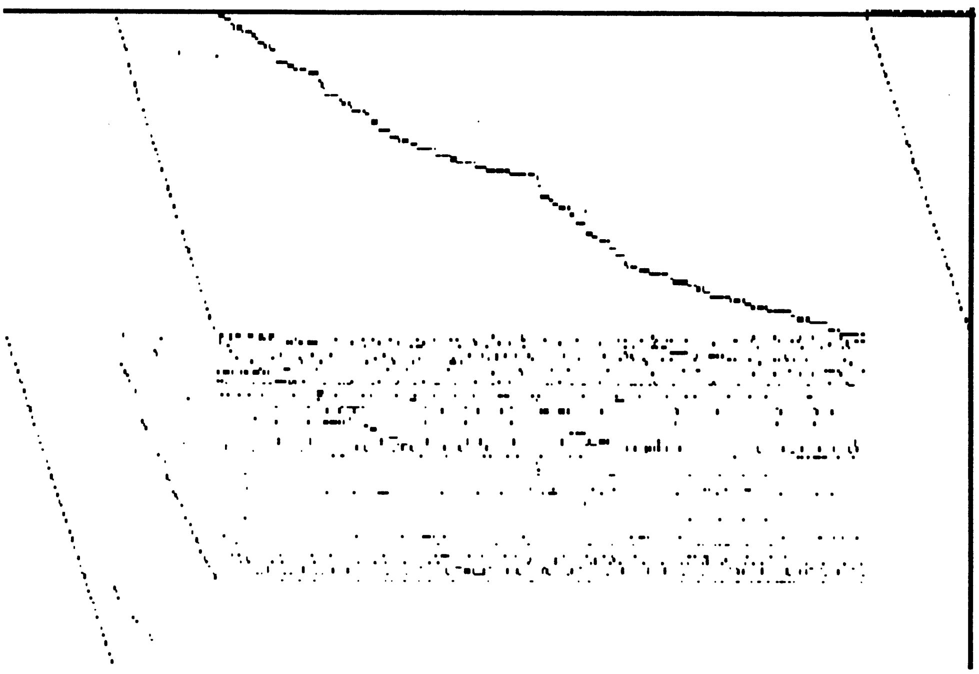

Figure 5: SSN nonzero structure

Stochastic versions of this problems have been created by Birge [1], Gassman [9] and in the STOPGEN set [3]. The data files available for this problem include Again, the nonzero structure of the CORE file is shown in Figure 6.

4.4 SCRS

SCRS8, collected by Ho [13] and first modified to include stochasticity by Birge [1], is a technological assessment model for the transition from nonrenewable to renewable energy sources. The model is a linearized version of the ETA model (Manne, [21]), which seeks to find the minimal cost strategy to meet future energy demands.

Since SCRS does not appear to exactly match the ETA model presented in [21], the formulation is described in words. For specific information, refer to the reference.

The objective of the model is to minimize the expected present value of (1) annual conservation, substitution, and energy costs, added to (2) annual investment costs in new sources of energy, appropriately weighted by the technology's service life. THere are 6 types of constraints:

1. Energy capacity constraints, that equates the current period's production and capacity to the previous period's plus or minus changes in capacity.

2. Bounds on the rates at which new technologies can be introduced.

Type	File list	Comments
MSLIP		
CORE	`CORE.4`	
TIME	`TIME2.4`, `TIME3.4`	2, and 3 stage versions
STOCH	`DISTO.4`, ..., `DIST7.4`	3 Stoch. RHSs, 8,27,54,108,324,432,864 scenarios
STOPGEN1		
CORE	`sc205.cor`	
TIME	`sc2052.tim`, `sc2053.tim`	2 and 3 Stages only
STOCH	`sc2052r.`x	Stochastic RHSs, $x = 4, 8, 16, 27, 32,$ $50, 64, 100, 160, 200, 400, 800$ scenarios
STOPGEN2		
CORE	`2053.cor` ,..., `20510.cor`	3,...,10 stage versions.
TIME	`2053.tim` ,..., `20510.tim`	3,...,10 stage versions.

Table 9: SC205 files

3. Supplies for electric and nonelectric energy must be no less than final demands.
4. Supplies for intermediate uses of energy must be no less than demands.
5. Cumulative sums of fuel resources must equal consumption less production.
6. Upper bounds on cumulative resource extraction.

The data files available from the University of Michigan were collected from Gassman, Birge, and STOPGEN [3]. (Filenames are shown in Table 10. Again, the nonzero structure is shown in Figure 7.

Type	File list	Comments
MSLIP		
CORE	`CORE.4`	
TIME	`TIME2.4`, `TIME3.4`	2, and 3 stage versions
STOCH	`DISTO.4`, ..., `DIST7.4`	3 Stoch. RHSs, 8,27,54,108,324,432,864 scens.
STOPGEN1		
CORE	`scrs.cor`	
TIME	`scrs2.tim`, `scrs3.tim`	2 and 3 Stages only
STOCH	`scrs2r.`x	Stoch. RHSs, $x = 4, 8, 16, 32, 64, 128, 256, 512$ scens.

Table 10: SCRS8 files

4.5 PLTEXP

PLTEXP (Sims [27]) is a stochastic capacity expansion model inspired by manufacturing flexibility research in Graves and Jordan [10]. The model is similar to SSN in that it tries to allocate new production capacity across a set of plants so as to maximize profit subject to uncertain demand. However, instead of enumerating a set of possible expansions, PLTEXP explicitly models the expansion options with binary variables.

Formulation

Suppose that a manufacturing system has I plants and J products. Each plant has manufacturing capacity $REGCAP_i$ and overtime capacity $OTCAP_i$. Capacity for a given product within each plant is limited to α_{ij} percent of the total capacity. Let the capital costs, regular operating cost and overtime operating cost for product j in plant i be CC_{ij}, RG_{ij}, and OT_{ij}. Suppose the amortization factor is β, and the capital budget

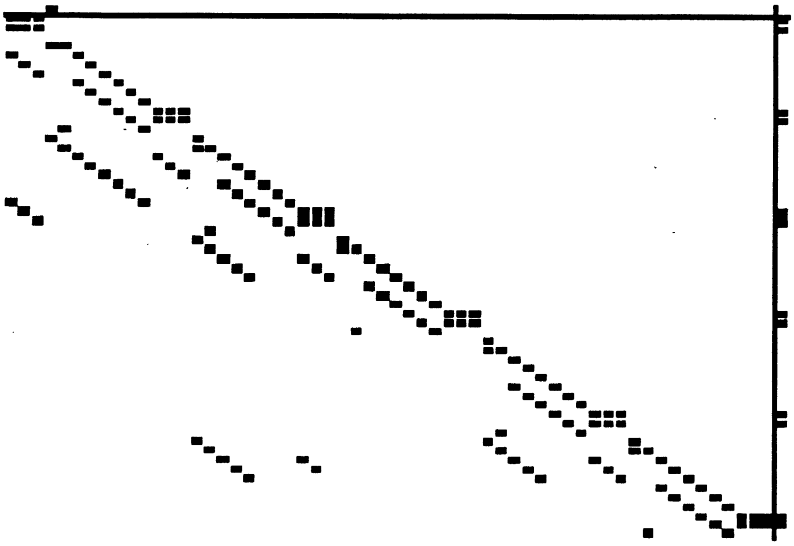

Figure 6: SC205 nonzero structure

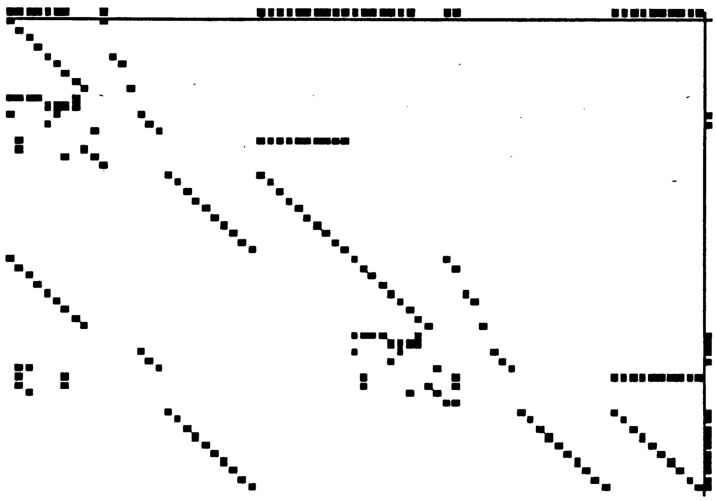

Figure 7: SCRS nonzero structure

is $BUDG$. Further, assume that demand is discretely distributed and has elements $(d_{js}, p_s), s = 1, \ldots, S$, and that each unit sold brings in REV_j dollars.

If the decision variables are

$$\begin{aligned} y_{ij} &= 1 \text{ if product } j \text{ is produced at plant } i, 0 \text{ otherwise,} \\ r_{ijs} &= \text{regular production of product } j \text{ in plant } i \text{ under scenario } s, \text{ and} \\ o_{ijs} &= \text{overtime production of product } j \text{ in plant } i \text{ under scenario } s, \end{aligned}$$

the deterministic formulation is

$$\max \sum_i \sum_j -\beta CC_{ij} y_{ij} + \sum_s \sum_i \sum_j p_s((REV_j - RG_{ij}) r_{ijs} + (REV_j - OT_{ij}) o_{ijs})$$

subject to

$$\begin{aligned} \sum_i r_{ijs} + o_{ijs} &\leq d_{js}, \quad j = 1, \ldots, J, s = 1, \ldots, S \\ \alpha_{ij}(r_{ijk} + o_{ijs}) &\leq y_{ijs}(REGCAP_i + OTCAP_i), \quad i = 1, \ldots, I, j = 1, \ldots, J, s = 1, \ldots, S, \\ \sum_j \alpha_{ij} r_{ijs} &\leq REGCAP_i, \quad i = 1, \ldots, I, s = 1, \ldots, S, \\ \sum_j \alpha_{ij} o_{ijs} &\leq OTCAP_i, \quad i = 1, \ldots, I, s = 1, \ldots, S, \\ \sum_i \sum_j CC_{ij} y_{ij} &\leq BUDG, \end{aligned}$$

$$y_{ij} \in \{0, 1\}, r_{ijs}, o_{ijs} \geq 0, \quad i = 1, \ldots, I, j = 1, \ldots, J, s = 1, \ldots, S.$$

The test problem is in AMPL [7] format, and has two data sets. The files are shown in Table 11, and a picture of the nonzeros for the smaller test problem is shown in Figure 8.

Type	File list	Comments
Model	`PLTEXP.mod`	AMPL model file.
Data	`PLTEXP.dat1`	3 plants, 3 products, 3 scenarios.
Data	`PLTEXP.dat2`	6 plants, 8 products, 3 scenarios.
Data	`PLTEXP.dat3`	8 plants, 12 products, 5 scenarios.

Table 11: PLTEXP files.

A multistage linear relaxation for the capacity expansion problem has also been developed by the author of this technical report. The model captures the same costs as the two-stage problem, with the exception that capital costs for assigning a product to a plant are only incurred at startup. For brevity, we only describe the stochastic formulation here. (Further details are in Birge, et. al. [5]).

Let

$$\begin{aligned} y_{ijt} &= 1 \text{ if product } j \text{ is produced at plant } i \text{ in period } t, 0 \text{ otherwise,} \\ r_{ijt} &= \text{regular production of product } j \text{ in plant } i \text{ in period } t, \\ o_{ijt} &= \text{overtime production of product } j \text{ in plant } i \text{ in period } t, \\ \delta^+_{ijt}, \delta^-_{ijt} &= \textit{Change in production of product } i \textit{ from plant } j \textit{ in period } t, \end{aligned}$$

and define a "cost" SC_{ij} to be the cost of eliminating production of product i from plant j. The formulation becomes

$$\max \sum_i \sum_j -\beta CC_{ij} y_{ij1} + (REV_j - RG_{ij}) r_{ij1} + (REV_j - OT_{ij}) o_{ij1}$$

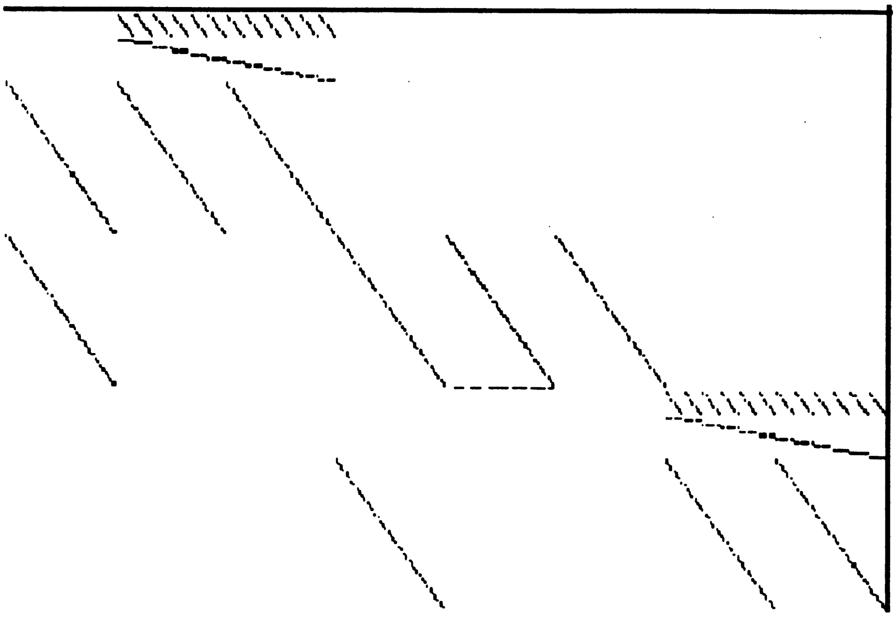

Figure 8: PLTEXP nonzero structure

$$+\sum_{t>1}\sum_i\sum_j -\beta CC_{ij}\delta^+_{ijt} + SC_{ij}\delta^-_{ijt} + (REV_j - RG_{ij})r_{ijt} + (REV_j - OT_{ij})o_{ijt}$$

subject to

$$\begin{aligned}
\delta^+_{ijt} - \delta^-_{ijt} - y_{ijt} - y_{ij(t-1)} &= 0, \quad i=1,\dots,I, j=1,\dots,J, t>1,\\
\sum_i r_{ijt} + o_{ijt} &\le d_{jt}, \; j=1,\dots,J, t=1,\dots,T\\
\alpha_{ij}(r_{ijt}+o_{ijt}) &\le y_{ijt}(REGCAP_i + OTCAP_i), \quad i=1,\dots,I, j=1,\dots,J, t=1,\dots,T,\\
\sum_j \alpha_{ij} r_{ijt} &\le REGCAP_i, \quad i=1,\dots,I, t=1,\dots,T,\\
\sum_j \alpha_{ij} o_{ijt} &\le OTCAP_i, \quad i=1,\dots,I, t=1,\dots,T,\\
\sum_i\sum_j CC_{ij} y_{ij1} &\le BUDG_1,\\
\sum_i\sum_j CC_{ij} delta^+_{ijt} &\le BUDG_t, \quad t=1,\dots,T,
\end{aligned}$$

$$y_{ijt} \in \{0,1\}, r_{ijt}, o_{ijt}, \delta^+_{ijt}.\delta^-_{ijt} \ge 0, \quad i=1,\dots,I, j=1,\dots,J, t=1,\dots,T.$$

4.6 PGP2

PGP2 is an electrical capacity expansion problem developed by Louveaux and Smeers [19]. (The problem was donated in digital format by Survajeet Sen, University of Arizona, and by H. I. Gassman) The two-stage stochastic program finds the minimal cost strategy for investing in various types of power plants.

Formulation

Suppose there are n technologies available to produce electricity and j power "modes" or classes of power demands over a duration T^t_j. Further, assume the costs and demands are stochastic. Define the other data as follows:

$$\begin{aligned}
g_i^t &= \text{Existing capacity of technology } i \text{ in time } t,\\
a_i, L_i &= \text{Availability (as a percentage) and Lifetime for technology } i,\\
\Delta_i &= \text{Construction delay for technology } i,\\
c_i^t, q_i^t &= \text{Unit investment (production) cost for } i \text{ at time } t,\\
d_j^t &= \text{Demand of power mode } j \text{ in time } t.
\end{aligned}$$

The decision variables are

$$\begin{aligned}
x_i^t, s_i^t &= \text{New (total) capacity of technology } i \text{ in time } t,\\
y_{ij}^t &= \text{Capacity of } i \text{ effectively used in time } t,
\end{aligned}$$

The stochastic model is

$$\min_{x,s} E[\sum_{t=1}^{N}(\sum_{i=1}^{n} c_i^t s_i^t + \sum_{i=1}^{n}\sum_{j=1}^{k} q_i^t T_j^t y_{ij}^t)]$$

subject to

$$s_i^t = s_i^{t-1} + x_i^t - x_i^{t-L_i} \ a.s.,$$

$$\sum_{i=1}^{n} y_{ij}^t = d_j^t \ a.s.,$$

$$\sum_{j=1}^{k} y_{ij}^t \leq a_i(g_i^t + s_i^{t-\Delta_i}) \ a.s.$$

$$a_n(g_n^t + s_n^{t-1} + x_n^t) \geq \sum_{j=1}^{m} d_j^t - \sum_{i=1}^{n-1} a_i(g_i^t + s_i^{t-\Delta_i}), \ a.s.$$

$$x, y, s \geq 0.$$

Since the electronic format of this problem is available from two sources, it has two names. To keep data files exactly as they were used in research papers, both versions have been kept. The data files are shown in Table 12 and the nonzero structure of a core file is shown in Figure 9.

Type	File list	Comments
MSLIP		
CORE	CORE.LOU	
TIME	TIME2.LOU	2 stage only
STOCH	DIST0.LOU, ..., DIST7.LOU	3 Stoch. RHSs, 1,8,20,40,80,160,320 scens.
STOCH	DIST8.LOU	3 Stoch. RHSs, 1 Stoch. Obj. 1280 scenarios.
STOPGEN1		
CORE	pgp2.cor	
TIME	pgp2.tim	2 Stages only
STOCH	pgp2.sto	3 Stoch. RHSs, 576 scenarios.

Table 12: PGP2/LOU files

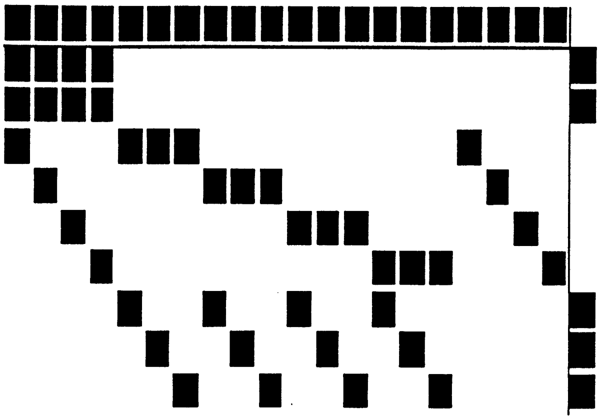

Figure 9: PGP2/LOU nonzero structure

4.7 STOCHFOR

STOCHFOR is a multistage model developed by Gassman [8] to determine logging levels to maximize output. The problem considers the risk of fire and of other environmental hazards. (The electronic form was donated by H. Gassman).

Formulation

Suppose that a forest is divided into K age-types. Age-type 1 is the youngest, and age-type K includes all trees not in the other types. Let $s_t = (s_{t,1}, \ldots, s_{t,K})$ be the area of the forest covered in each age class in period t. Let the decision variables be the areas $x_t = (x_{t,1}, \ldots, x_{t,K})$ which are to be harvested and $\boldsymbol{p}_k$ is the proportion of timber destroyed by fires or other diseases. The destruction rate $\boldsymbol{p}_k$ is stochastic, and is an n-tile discretization from historical data. [1] (For details, refer to [8].) Let $y = (y_1, \ldots, y_K)$ and $\nu = (\nu_1, \ldots, \nu_K)$ be the timber yields and values of standing timber for each class.

The objective is to maximize the expected harvest over some long horizon. To reduce the problem, end effects are modeled explicitly. If δ is a discount rate over the planning period, the objective of the stochastic program is to

$$\max \delta^1 y^T x_1 + E[\delta^2 y^T x_2 + E[\cdots + E[\delta^{T+1} \nu^T s_{T+1}]]],$$

subject to availability constraints

$$x_t \le s_t, \quad t = 1, \ldots, T,$$

subject to inventory constraints, which assume that reforestation takes place immediately,

$$\begin{aligned} s_1^{t+1} &= \sum_{k=1}^{K} \boldsymbol{p}_k (s_k^t - x_k^t) + x_k^t, \\ s_k^{t+1} &= (1 - \boldsymbol{p}_{k-1})(s_{k-1}^t - x_{k-1}^t), \quad k = 2, \ldots, K-1, \\ s_K^{t+1} &= (1 - \boldsymbol{p}_{K-1})(s_{K-1}^t - x_{K-1}^t) + (1 - \boldsymbol{p}_K)(s_K^t - x_K^t), \end{aligned}$$

and finally subject to relative limits on the change of the harvest:

$$\alpha y^T x_{t-1} \le y^T x_t \le \beta y^T x_{t-1}.$$

Finally, all variables are nonnegative.

The files that are available for this problem were donated by H. Gassman, and are shown in Table 13. Again, a picture of the nonzero structure follows, and selected solutions and sizes appear in Tables 14 and 15.

[1] The assumption p_k is the same for all t is necessitated by the lack of data availability.

Type	File list	Comments
MSLIP		
CORE	CORE.F	
TIME	TIME7.F	7 stage version only.
STOCH	DISTO.F, ..., DIST6.F	Block structure, 0,6,12,14,16,17,18 blocks.

Table 13: STOCHFOR files

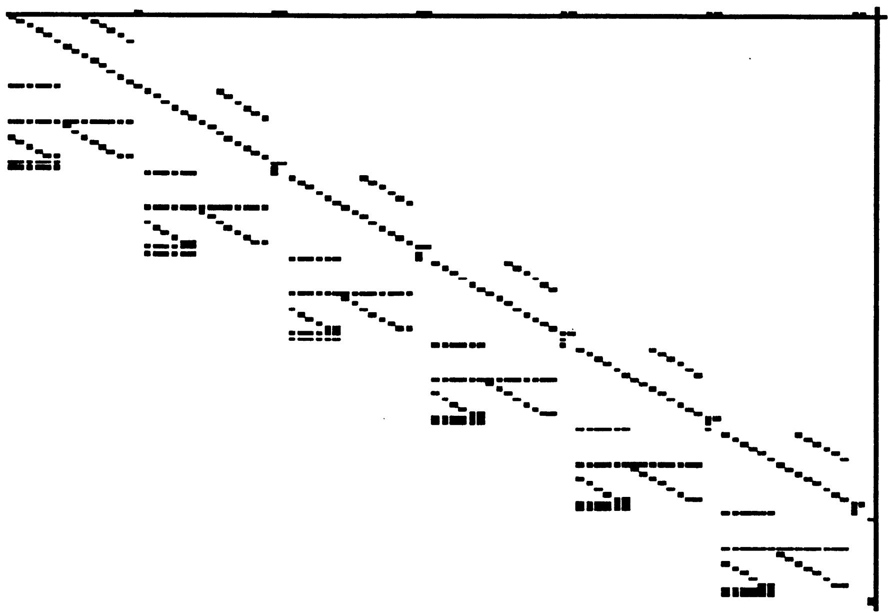

Figure 10: STOCHFOR nonzero structure

5 Financial Problems

The most fertile field stochastic programming has been applied to in the last few years is financial engineering and its related subfields. Applications in this area have (in the author's opinion) furthered the operations research community's acceptance of stochastic programming more than any other area. (A good example is the 1993 Edelman Award, in which a stochastic financial optimization model won second prize.) The most active researchers in this field have been Hercules Vladimirou, John Mulvey, and W. T. Ziemba. Profs. Vladimirou and Mulvey have graciously shared some of their models for public consumption.

5.1 Network Models of Mulvey and Vladimirou

Mulvey and Vladimirou [23], [22] have created several two-stage network programs for finding the optimal investment strategy to meet cash flow needs in the face of uncertain investment returns. Although the formulations are two-stage in the formal sense, the second stage groups together multiple time periods.

The goal of the model is to allocate funds to asset categories in each of four periods. Flows of funds between assets in a period have fixed marginal costs. Asset returns from one period to the next are modeled using stochastic multipliers on "arcs" from between periods. A scenario in this case is a realization of all future asset returns.

Formulation

The nodes in this formulation will represent the various asset categories in each period. $\mathcal{V}_1$ will be the set

Problem	A matrix	T matrix	W matrix
SCAGR	15 x 20	19 x 20	19 x 20
SSN	1 x 89	175 x 89	175 x 706
SC205	13 x 14	22 x 14	22 x 22
SCRS	28 x 37	28 x 37	28 x 38
PLTEXP(3)	125 x 288	221 x 288	221 x 480
PGP2	2 x 4	7 x 4	7 x 16
STOCHFOR	15 x 15	17 x 15	17 x 16

Table 14: Planning problem block sizes

Problem	Instance	Objective
SCAGR7	`scagr72r.4,scagr72r.64`	-832902.1516,-832902.1515
	`scagr72c.4`	-832941.3274
	`scagr72b.4`	-832941.3274
SSN	`ssn.sto`	11.46
SC205	`sc2052r.4,sc2052r.32, sc2052r.100`	-60.4230, -55.3877, -10.0705
	`sc2053r.8`	-66.6667
SCRS	`scrs82.r4, scrs82.r32, scrs82.r128`	123.1766, 123.1766, 128.3694
PLTEXP	`PLTEXP.dat1`	901.13636
	`PLTEXP.dat2`	2313.36818
	`PLTEXP.dat3`	7359.115
PGP2	`pgp2.sto`	446.44
STOCHFOR	`DIST1.F`	-43613.4625
	`DIST3.F`	-43381.046

Table 15: Selected planning problem solutions

of assets in the first period, and $\mathcal{V}_2$ will be the set of nodes (assets in each period) in the second stage. Let $\mathcal{A}_1^d$ be the set of arcs representing flows between assets in the first period, and $\mathcal{A}_1^s$ be the arcs representing the asset returns between the first and second periods. Let $\mathcal{A}_2$ be the flows between arcs in the second and subsequent periods. Also, let Δ_i^+ (Δ_i^-) be the set of arcs outgoing (incoming) arcs connected to node i.

Let $v_{ij} \in \mathcal{A}_1^d \cup \mathcal{A}_1^s$ be the variables representing flows on first stage arcs. Let $y_{ij}(s)$ be the flow between two nodes in the second stage under scenario $s \in S$. Definitions of the data are

$$\begin{aligned} r_{ij} &= \text{Marginal cost of funds transferal between assets } i \text{ and } j \text{ in the first stage,} \\ b_i &= \text{External supply or demand into asset } i, \\ p_s &= \text{Probability of scenario } s, \\ \xi_{ij}(s) &= \text{Asset returns or marginal costs between assets in scenario } s, \\ d_i(s) &= \text{Stochastic supply/demand at node } i \text{ in scenario } s. \end{aligned}$$

The objective will be to minimize a convex continuously differentiable utility function $f_s(v, y(s))$ subject to network constraints on funds flow. The problem is to

$$\min \sum_{s \in S} p_s f_s(v, y(s))$$

subject to first period fund movement constraints

$$\sum_{(i,j) \in \Delta_i^+} v_{ij} - \sum_{(j,i) \in \Delta_i^-} r_{ji} v_{ji} = b_i, i \in \mathcal{V}_1,$$

and second stage fund movement constraints

$$\sum_{\substack{(i,j)\in\Delta_i^+\\(i,j)\in\mathcal{A}_1}} v_{ij} - \sum_{\substack{(j,i)\in\Delta_i^-\\(j,i)\in\mathcal{A}_1^d}} r_{ji}v_{ji} - \sum_{\substack{(j,i)\in\Delta_i^-\\(j,i)\in\mathcal{A}_1^s}} \xi_{ji}(s)v_{ji}+$$

$$\sum_{\substack{(i,j)\in\Delta_i^+\\(j,i)\in\mathcal{A}_2}} y_{ij}(s) - \sum_{\substack{(j,i)\in\Delta_i^-\\(j,i)\in\mathcal{A}_2}} \xi_{ji}(s)y_{ji}(s) = d_i(s)$$

for all $s \in S$ and $i \in \mathcal{V}_2$. Each variable is also bounded from above and below by given data.

Both deterministic equivalent and compact (MSLIP) forms of each problem are available. The instantiations run from 18 scenarios to 80 scenarios, and have deterministic equivalents of up to 22,000 columns. Their filenames are shown below.

Type	File list	Comments
MSLIP		
CORE	`LP0.core` ... `LP9.core`	
TIME	`LP0.time` ... `LP9.time`	
STOCH	`LP0.stoch` ... `LP9.stoch`	Block structure, 18,52,80,72,70,48,40,60,36 scenarios
Det. Equiv.	`deter0.mps` ... `deter8.mps`	

Table 16: Mulvey and Vladimirou Network files

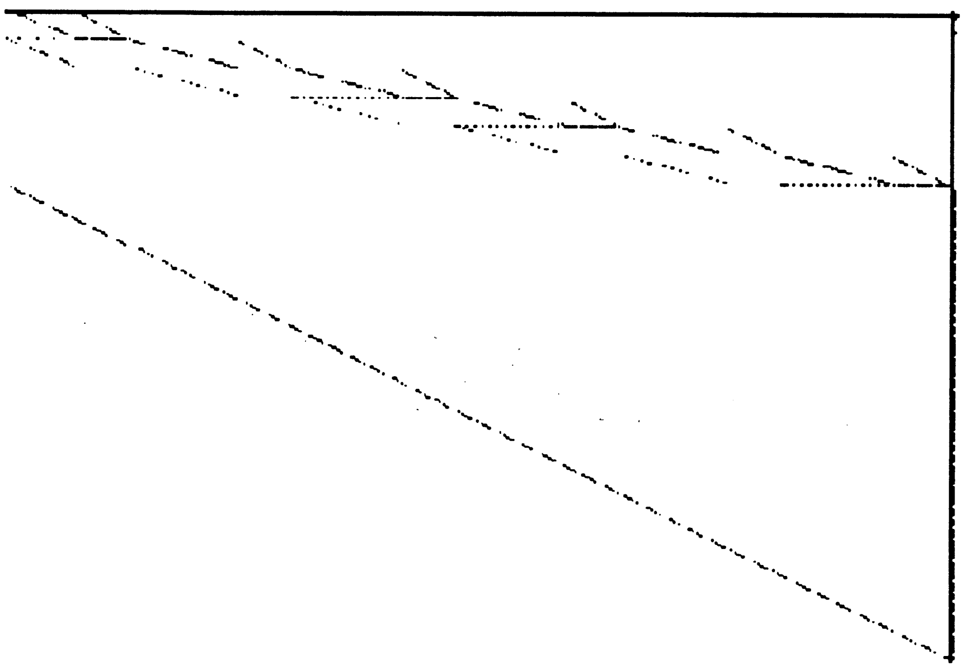

Figure 11: Mulvey and Vladimirou typical nonzero structure, including bounds

5.2 OPTN - An option selection model

OPT is a multistage stochastic programming model for selecting a minimal (expected) cost option portfolio. The portfolio must guarantee that an acceptable level of dollar revenue will be obtained given that a known quantity of a foreign currency must be exchanged in the future. The model was first developed by Klaassen, et. al. [17] and was digitized and slightly modified by the author.

Formulation

The goal of the model is to exchange D foreign currency units at time T. Suppose N options are available over T periods, and S exchange rate scenarios have been identified. The N options are assumed to have strike prices E_n.

The authors develop a set of scenarios for future spot prices by assuming the standard lognormal diffusion process. The diffusion process has a drift which is a function of the forecast foreign risk free rate of interest and the domestic (US) rate of interest, and a dispersion which is a function of the foreign currency's volatility[2].

Option prices P_{tsn} are determined in [17] using a form of the Black-Sholes model, which relates option price with interest rates and volatility measures. (All options are assumed to be European puts.) For each scenario s, let F_s be the final exchange rate in period T (scenario s). Suppose that the US interest rate in period t for scenario s is USi_{st}. If the decision variables x_{tsn} are the quantities of option j purchased at time t in scenario s (nonanticipativity is implicit in the model), the basic formulation is to minimize the expected cost of exchange

$$\min \sum_{t=0}^{T-1} \sum_{s=1}^{S} p_s (1 + USi_{ts})^{-t} \sum_{n=1}^{N} x_{tsn} P_{tsn}$$

subject to (1) a restriction on the choice of options:

$$\sum_{t=0}^{T-1} \sum_{n=1}^{N} x_{tsn} \leq 1 \quad s = 1, \ldots, S,$$

and (2) the effective exchange rate must be at least the target exchange rate Q_s:

$$F_s + \sum_{t=0}^{T-1} \sum_{n=1}^{N} x_{tsn}((E_n - F_s)^+ - (1 + USi_{ts})^{T-t} P_{tsn}) \geq Q_s \quad s = 1, \ldots, S,$$

and all $x_{tsn} \geq 0$.

Since most optimization packages do not have cumulative normal functions (which are required by the Black-Sholes model to determine option prices), the data files are generated by a *Mathemtica* [29] file which writes directly the option prices. The file names are shown in Table 17 and the nonzero structure is shown in Figure 12

Type	File list	Comments
Mathematica	opt3.m	Package that generates the input data
AMPL	opt3.all	Includes model and data
AMPL	opt3.mps	AMPL-processed version.

Table 17: OPT files

5.3 ALM: Asset Liability Management

Kusy and Ziemba [18] developed a multistage stochastic formulation called ALM to try to balance a bank's income stream from a set of assets against a set of liabilities. The assets are loans and investments with uncertain returns and varying levels of risk, and the liabilities are depositor's withdrawals from demand accounts.

The details of the simple recourse formulation are beyond the scope of this technical report. The model has been translated by Kristen Missle and Todd Mueller of the University of Michigan into AMPL format. The model files and sample data sets are shown below. The only picture of the constraint matrix nonzeros for this problem is too large to graph, and hence is omitted.

[2] Foreign exchange rates were generated heuristically by the authors of [17].

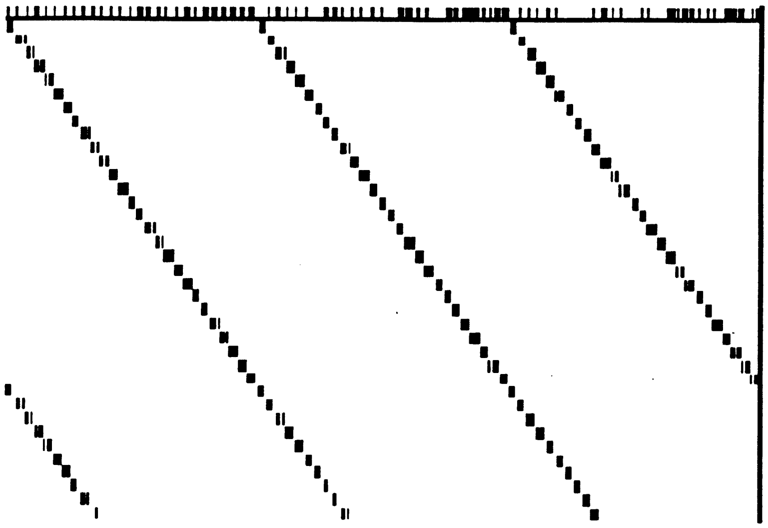

Figure 12: OPT nonzero structure

Type	File list	Comments
AMPL model	`assliab.mod`	Notation follows that in [18].
AMPL data	`assliab.dat2`, `assliab.dat3`	2 and 3 stage data files.

Table 18: ALM asset liability management files

6 Other models

6.1 SCTAP1: Traffic Assignment

SCTAP1 is a stochastic version of a two stage traffic assignment problem given in (Ho, [14]). The objective of the problem is to find a set of traffic flows over a network with multiple sources and one sink that minimizes some cost function. The cost function is a utility function associated with traffic congestion. The flow of traffic is governed by an "exit function" for each arc which relates the amount of traffic entering and leaving an arc in a given period. This function is (in general) a nondecreasing, continuous concave function, which is linearized to form a problem in standard LP form.

Formulation

Suppose that traffic flows from several nodes in a set $\mathcal{N}$ into one destination $n \in \mathcal{N}$ over a set of arcs $\mathcal{A}$. Flow is split into $t = 0, \ldots, T$ time periods. Let $A(q)$ and $B(q)$ be the set of arcs leaving (entering) node

Problem	A matrix	T matrix	W matrix
Mulvey and Vladimirou (`LP.0`)	15 x 32	105 x 32	105 x 301
OPT		96 x 2916[1]	
ALM		5698 x 5664[1]	

Table 19: Financial problem block sizes

Problem	Instance	Objective
Mulvey and Vladimirou	LP.0,LP.1, LP.2	-2.045920, -2.557564, -1.829869
	LP.3,LP.4, LP.5	-2.107791, -1.428035, -2.261224
	LP.6,LP.7, LP.8	-2.307987, -2.228568, -2.794296
OPT	opt2.all	42.4083
ALM	assliab.dat2	829.9939598

Table 20: Selected financial problem solutions

$q \in \mathcal{N}$. Define the other data as follows:

$$\begin{aligned} F_t(q) &= \text{External input to node } q \text{ in period } t. \\ x_{tj} &= \text{Traffic or flow on arc } j \text{ at start of period } t. \\ h_{tj}(x_{tj}) &= \text{Cost of flow } x_{tj}. \\ d_{tj} &= \text{Traffic admitted to arc } j \text{ in period } t. \\ g_j(x_{tj}) &= \text{Traffic exiting from arc } j \text{ in period } t. \end{aligned}$$

For the piecewise linearization, let λ_{tj}^k be interpolation weights on $1, \ldots, K(j)$ grid points defined for each arc j such that each is nonnegative and $\sum_{j=1}^{K(j)} \lambda_{tj}^k = 1$. Define c_j^k, g_j^k, and h_{tj}^k to be data such that

$$x_{tj} \equiv \sum_{k=1}^{K(j)} c_j^k \lambda_{tj}^k, \qquad g_j(x_{tj}) \equiv \sum_{k=1}^{K(j)} g_j^k \lambda_{tj}^k, \qquad h_{ij}(x_{tj}) \equiv \sum_{k=1}^{K(j)} h_{ij}^k \lambda_{tj}^k.$$

The objective is to minimize the total cost

$$\min \sum_{t=0}^{T} \sum_{j \in \mathcal{A}} \sum_{k=1}^{K(j)} h_{tj}^k \lambda_{tj}^k$$

subject to flow equations on each node and in each time period except the last:

$$\sum_{k=1}^{K(j)} c_j^k \lambda_{(t+1)j}^k = \sum_{k=1}^{K(j)} (c_j^k - g_j^k) \lambda_{tj}^k + d_{tj}, \quad t = 0, \ldots, T-1, j \in \mathcal{N}.$$

The previous constraint only considers flow within the network. The next constraint also accounts for external flows into and out of each node except the sink:

$$\sum_{j \in A(q)} d_{tj} = F_t(q) + \sum_{j \in B(q)} \sum_{k=1}^{K(j)} g_j^k \lambda_{tj}^k, \quad t = 0, \ldots, T-1, q \in \mathcal{N}, q \neq n.$$

An initial condition spreading existing traffic over the network is

$$\sum_{k=1}^{K(j)} c_j^k \lambda_{0j}^k = R_j, \quad j \in \mathcal{N}.$$

Finally, each variable is nonnegative and the sums of λ's must add to one for each arc.

A stochastic programming formulation of the deterministic model presented above first appeared in (Birge, [1]). The two stage version assumes that the external traffic inputs at each stage are stochastic.

Files available for this problem are shown in Table 21. A picture of the nonzero structure for the deterministic problem follows.

Type	File list	Comments
MSLIP		
CORE	`CORE.5`	
TIME	`TIME5.x`, `TIME5.x`	2, and 3 stage versions
STOCH	`DIST5.x`, ..., `DIST5.x`	3 Stoch. RHSs, 8,27,54,108,324,432,864 scenarios
STOPGEN1		
CORE	`sctap1.cor`	
TIME	`sctap12.tim`, `sctap13.tim`	2 Stage and 3 stage partitions of CORE file
STOCH	`sctap12r.`x	Stoch. RHSs, $x = 4, 8, 8b, 16, 27, 32, 54, 64, 108, 216, 480$ scenarios
STOCH	`sctap12c.`x	Stoch. Obj., $x = 4, 16, 64$ scenarios
STOCH	`sctap12b.`x	Stoch. Obj. and RHSs, $x = 4, 16, 64$ scenarios

Table 21: SCTAP1 files

6.2 SCSD: Design of a multistage truss

SCSD is a multistage model due to (Ho [12]) which seeks to find the lowest weight truss design given a set of admissible joints in the plane and a load which the truss must bear. The structure is formed by placing bars between pairs of joints, and must satisfy constraints on maximum stress in each bar.

Formulation

Following the presentation in [12], we present a two stage formulation only. The multistage extension is straightforward. Suppose $P_i = (P_{ix}, P_{iy})$ are the horizontal and vertical forces for each point in a list of N points representing possible joints in space. There are $K = (N-1)N/2$ possible bars given these points. Let δ_k and γ_k be the cosines of the angles each bar forms with the x and y planes. The variables are u_k and s_k, which represent the force carried by each bar and its cross-sectional area.

The objective is to minimize the overall weight of the structure. The objective of the deterministic version is formed by noting that any optimal solution will be fully stressed, so that $|u_k| = \sigma s_k$, where σ is the yield stress of the material forming each bar. So, the objective can be written as the density of the material (ρ) times the cross sectional area times the length of each bar (L_k), or

$$\min W = \frac{\rho}{\sigma} \sum_{k=1}^{K} L_k (u_k^+ - u_k^-)$$

subject to $2N$ equilibrium equations, which can be written as

$$\sum_{k=1}^{K} a_{lk}^1 (u_k^+ - u_k^-) = P_{lx}, \quad l = 1, \ldots, N,$$

$$\sum_{k=1}^{K} a_{lk}^2 (u_k^+ - u_k^-) = P_{ly}, \quad l = 1, \ldots, N,$$

$$u_k^+, \ u_k^- \geq 0,$$

where $a_{lk}^1 = \pm\delta_k$ and $a_{lk}^2 = \pm\gamma_k$ if bar k leaves (enters) joint i, and are both zero in any other case. Here, P_{lx} and P_{ly} are set to the external forces on the truss.

The stochastic version of the problem considers the set of external loads placed on the truss to be random. Arbitrarily, the weights of each bar can be considered random. The model can also be extended to multiple stages representing multiple levels in the truss.

Again, files available for this problem and a nonzero picture are shown in the next table and figure.

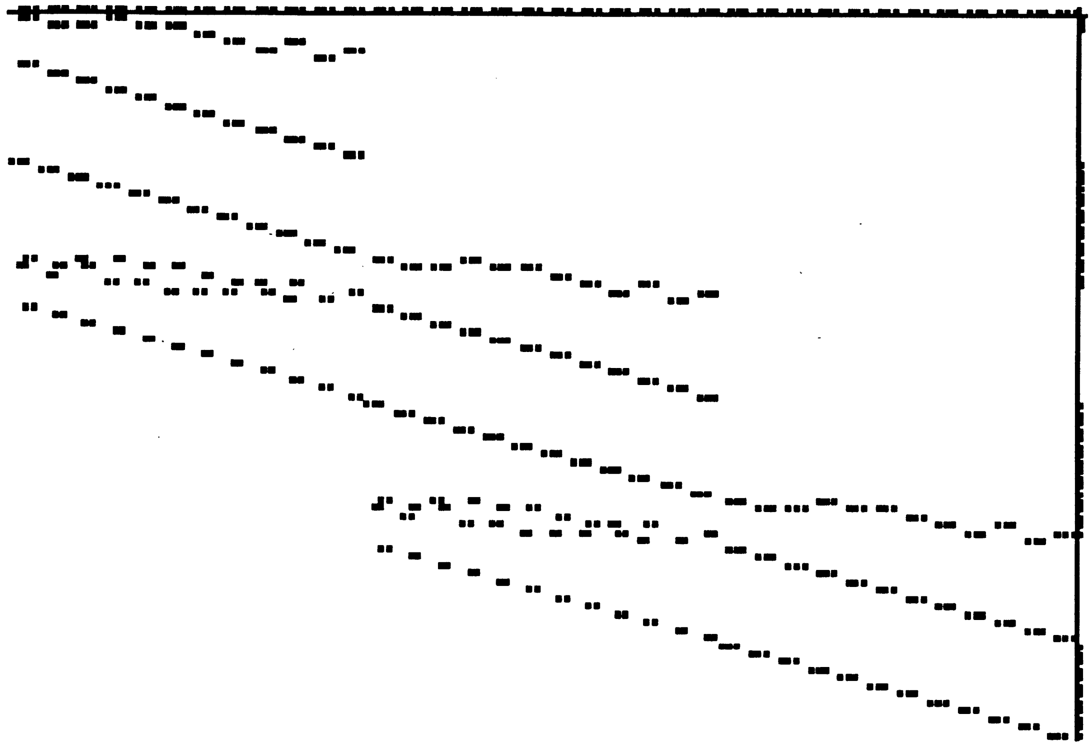

Figure 13: SCTAP nonzero structure

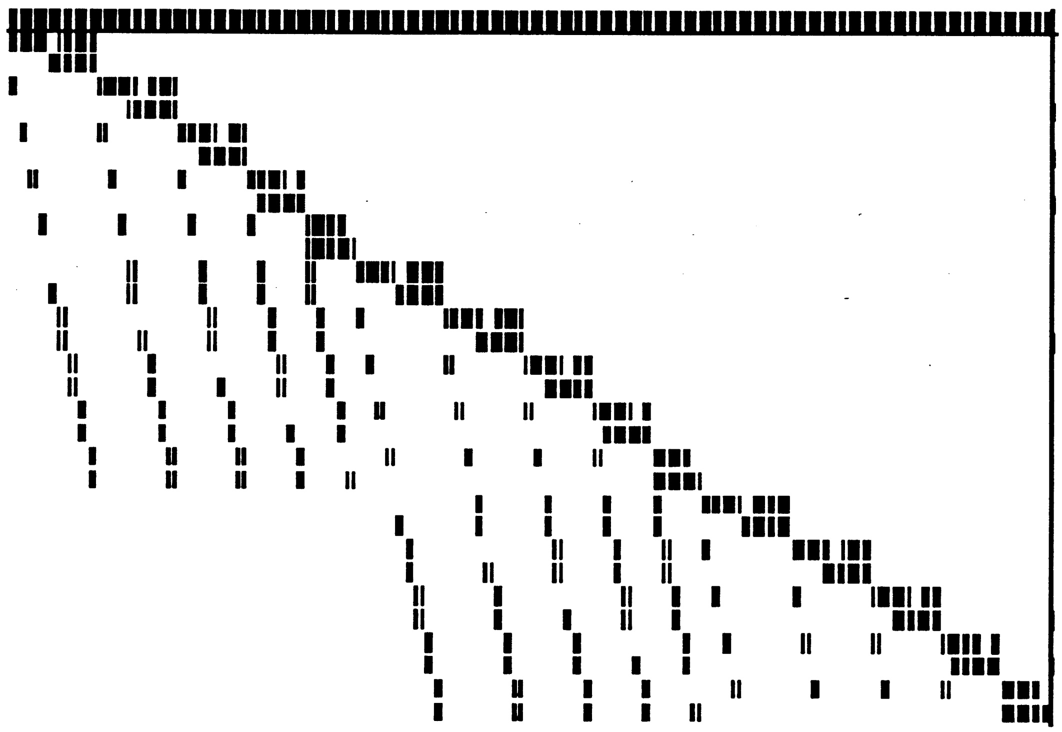

Figure 14: SCSD8 nonzero structure

Type	File list	Comments
MSLIP		
CORE	CORE.5	
TIME	TIME5.x, TIME5.x	2, and 3 stage versions
STOCH	DIST5.x, ..., DIST5.x	3 Stoch. RHSs, 8,27,54,108,324,432,864 scenarios
STOPGEN1		
CORE	scsd8.cor	
TIME	scsd82.tim, scsd83.tim	2 Stage and 3 stage partitions of CORE file
STOCH	scsd82r.x	Stochastic RHSs, $x = 4, 8, 8b, 16, 27, 32, 54, 64, 108, 432$ scenarios
STOCH	scsd82c.x	Stochastic Objective, $x = 4, 16, 64$ scenarios
STOCH	scsd82b.x	Stochastic Objective and RHSs, $x = 4, 16, 64$ scenarios

Table 22: SCSD8 files

Problem	A matrix	T matrix	W matrix
SCTAP	30 x 48	60 x 48	60 x 96
SCSD	10 x 70	10 x 70	10 x 70

Table 23: Miscellaneous problem block sizes

6.3 Examples and Problems from Birge and Louveaux

Two examples of stochastic programming formulations, as well as a selection of formulation exercises are given in Chapter 1 of Birge and Louveaux [4]. Solutions to these exercises are available in AMPL format. Describing each problem in detail is beyond the scope of this technical report, but the next table gives brief descriptions and associated filenames. In the following, each file contains both data and model files for the recourse problems and expected value problems.

7 STOCHTOMPS: A deterministic equivalent problem generator.

The STOPGEN package includes a set of test files and a program which reads in stochastic programming problems in standard format and outputs an MPS file that describes the deterministic equivalent problem. This program, called **stochtomps**, is especially useful in cases where special algorithms are not available or in cases where models are in preliminary development. (*Note:* **stochtomps** *is a research code. The code*

Problem	Instance	Objective
SCTAP	sctap12r.4, sctap12r.32, sctap12r.128	280.5, 354.00, 298.00
	sctap12c.4, sctapc.16	236.25, 326.400
	sctap12b.4, sctapb.16	242.375, 280.800
SCSD	scsd82r.4, scsd82r.16, scsd82r.108	15.50, 15.9995, 23.7000
	scsd82c.4, sctapc.16	15.00, 15.00
	scsd82b.4, sctapb.16	15.25, 27.50

Table 24: Selected miscellaneous problem solutions

Filename(s)	Description
aac.all	Dantzig's ([6]) Aircraft Allocation Problem
fvl.all	Louveaux's power generation expansion model.
p1.all	Airline load management problem.
p2.all	Blending problem.
p3.all	Lake water management problem.
p4.all	A(nother) power generation expansion model.
p5.all	Forestry management problem.
p6.all	Hospital staff planning problem.
p7.all	Automobile racing control problem.
p8.all	Olympic decathlon strategy planning model.

Table 25: Book Problems and files

has been tested on several problems, but may not be 100% correct. Use at your own risk.)

stochtomps uses the input routines written for MSLiP version 7 (Gassman, [9]) to input CORE, STOCH, and TIME files for multistage stochastic linear programs. The **stochtomps** "back-end" forms the deterministic equivalent by writing appropriate submatrices from MSLiP's internal tree structure.

Several options for the output format are possible. The options are summarized in Table 26. Some

	Option	Description
1	Full Formulation	The multistage deterministic equivalent.
2	Split Formulation	(Two-stage only). A deterministic equivalent with explicit nonanticipativity constraints. (See Lustig, et. al. [20])
3	Period subproblems	Last period subproblems of the form $\min\{q^T y \| Wy = \xi_{ts}\}$. Problems are output in MPS format sequentially into one output file.
4	Complete period subproblems	Subproblems of the form $\min\{c^T x + q^T y \| Tx + Wy = \xi_{ts}\}$. Problems are output in MPS format sequentially into one output file.
5	All period subproblems	Same as (3), but all periods' subproblems are output. Problems are output in MPS format sequentially into one output file.

Table 26: Program stochtomps output formats.

of the options form several small LP problems (in MPS format) in the same file. A shell script called **splitprobs** included in the **src** directory can be used with the last three options to split the output file into its component problems.

As mentioned earlier, the sizes of the deterministic equivalent can become quite large. Since the MPS format limits each variable and row to 8 alphanumeric characters, there is an absolute limit to size of the deterministic equivalent program. **stochtomps**'s default is to replace the last two (of eight) characters of each row or column identifier read in from the CORE file with an alphanumeric sequence corresponding to MSLiP's node number. Alternatively, specifying the **-n** option replaces the column and row names with (sequential) numbers. Doing so allows **stochtomps** to fit in more characters for the node number. For example, if there are 100 rows and columns, then only 2 alphanumeric characters are necessary to identify each row and column, and 6 alphanumeric characters can be used for node numbers. This corresponds to a maximum of $36^6 = 2176782336$ nodes.

Problem sizes are also limited by MSLiP's internal limits on memory allocation. These limits can be changed by changing DATA statements in the header file **comsm2.f** and recompiling the **stochtomps** program.

The next subsection gives details on making the **stochtomps** program and retrieving the included test problems. The subsection following that includes instructions on how to use the **stochtomps** program.

7.1 Unpacking and making STOPGEN

STOPGEN is designed for UNIX systems, but can be modified for other systems. The package consists of four files: **readme.1st**, **readme.probs**, **STOPGEN.Z** (or STOPGEN2.Z), and **getstopgen**. The readme files give instructions for unpacking STOPGEN and up-to-date notes on changes made to the package. **STOPGEN.Z** is a compressed file which contains all data, and **getstopgen** is a batch file which uncompresses the data.

Typing **getstopgen** or **sh getstopgen** from a UNIX prompt unpacks several files into the current directory. One directory made is ./src. Changing to that directory and typing **make** compiles the **stochtomps** program, which can be moved to data directories (as necessary) to create each instance.

Another shell file retrieved using the **getstopgen** script is **getsto**. Typing **getsto** from a UNIX prompt uncompresses all of the data files in the STOPGEN package. Two additional shell scripts (**putsto** and **putstopgen**) are provided to recompress the source and data files into their original data files.

7.2 Running the Deterministic Equivalent Problem generator.

To run **stochtomps**, type the program name and any necessary command line switches. The prototype for the program is shown below

```
stochtomps [specs] [core] -fm {format} {-n} -so {output}
                -t {time} -st {stoch} -l {log}  -su {summary}
```

where

Switch	Required?	Description
specs	Yes	A null specs file, included in the src directory.
core	Yes	A CORE filename.
-fm format	No	Format of output, format is taken from numbers given in the previous table.
-n	No	Replaces CORE row and column names with sequential numbers. The default is to replace the last 2 (of 8) characters with node identifiers. (See discussion).
-so output	No	Output filename. Default is the screen.
-t time	No	TIME data file filename.
-st stoch	No	STOCH data file filename.
-l log	No	Log filename. Default is the screen.
-su summary	No	Summary filename (Not currently used). Default is the screen.

Table 27: Program stochtomps runtime options.

The program uses UNIX sorting routines. Porting the program to non-UNIX platforms will require substitution of another external sorting subprogram.

7.3 Obtaining STOPGEN and other files.

STOPGEN is available via anonymous ftp from the University of Michigan. The anonymous ftp receiver is called freebie.engin.umich.edu. Below is an annotated sample of the commands necessary to get STOPGEN and other problems described in the technical report.

```
dice% ftp freebie.engin.umich.edu <-- internet address where probs are
Connected to mingin.engin.umich.edu.
220 University of Michigan, CAEN, mingin.engin.umich.edu FTP server (Version 5.6
0a / ULTRIX 4.2) ready.  Email problems to ftp_support@caen.engin.umich.edu
  "anonymous" allows anyone to access the problems --|
                                                     |
Name (freebie.engin.umich.edu:dholmes): anonymous  <--
331 Guest login ok, send your email address as password.
```

```
Password:        <-- Any text will do.
230 Guest login ok, access restrictions apply.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
bin
etc
pub                            <--- This directory
226 Transfer complete.
ftp> cd pub
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
EMSA+MAS
Apollo.Mail
 ...... (Several deleted) .....
tmp
incoming
stoprobs      <---- This subdirectory
misc
226 Transfer complete.
ftp> cd stoprobs
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
networks          <--|
stopgen              |
mslip_probs          |
ampl                 |-- These are the problems we have so far.
stoprobs.README      |
misc              <--|
226 Transfer complete.
ftp> get stoprobs.README     <--- Use "get" to get an ASCI
200 PORT command successful.
150 Opening ASCII mode data connection for stoprobs.README (6132 bytes).
226 Transfer complete.
6251 bytes received in 0.1776 seconds (34.37 Kbytes/s)
ftp> cd stopgen
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
makefile
STOPGEN.Z              <--- Any .Z file is compressed and must be
readme.1st                  transferred in binary for
readme.probs
226 Transfer complete.
ftp> get makefile
200 PORT command successful.
150 Opening ASCII mode data connection for makefile (928 bytes).
226 Transfer complete.
965 bytes received in 0.0492 seconds (19.16 Kbytes/s)
```

```
ftp> binary             <-- To transfer a binary file, must be in bin.
200 Type set to I.          mode.
ftp> get STOPGEN.Z      <-- Once in binary mode, use "get" as usual..
200 PORT command successful.
150 Opening BINARY mode data connection for stodata.Z (117103 bytes).
226 Transfer complete.
117103 bytes received in 0.8852 seconds (129.2 Kbytes/s)
ftp> cd ..
250 CWD command successful.
ftp> ascii              <-- To transfer a text file, must be in text
200 Type set to A.          mode.
ftp> get readme.1st
200 PORT command successful.
150 Opening ASCII mode data connection for readme.1st (3628 bytes).
226 Transfer complete.
3717 bytes received in 0.07912 seconds (45.88 Kbytes/s) <- File Xfer d
ftp> get readme.probs
200 PORT command successful.
150 Opening ASCII mode data connection for readme.probs (3628 bytes).
226 Transfer complete.
3717 bytes received in 0.07912 seconds (45.88 Kbytes/s) <- File Xfer d
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
networks
stopgen
mslip_probs
ampl
stoprobs.README
misc
226 Transfer complete.
ftp> quit              ------------- All files now transferred
221 Goodbye.            |
dice% ls               \_/
makefile        STOPGEN.Z        readme.1st   readme.probs
dice%
```

8 Conclusion

This technical report has described several electronically available stochastic programming problems collected from the literature and from activities undertaken at the University of Michigan (under the direction of John Birge). Any inquiries regarding these problems should be directed to the author (**dholmes.engin.umich.edu**) or to John Birge (**jrbirge.engin.umich.edu**). This report will be updated as necessary.

References

[1] J. R. Birge, 1985. "Decomposition and partitioning methods for multistage stochastic linear programs," *Operations Research*, 33, pp. 989-1007.

[2] J. R. Birge, M.A.H. Dempster, H. I. Gassman, E. A. Gunn, A. J. King, and S. W. Wallace, 1987. "A Standard forinput format for multiperiod stochastic linear programs," *COAL newsletter*, 17, pp. 1-20.

[3] J. R. Birge, H. I. Gassman, and D. Holmes, 1991. STOPGEN stochastic linear program deterministic equivalent problem generator.

[4] J. R. Birge and F. Louveaux, 1994. **Introduction to Stochastic Programming**, manuscript.

[5] J. Birge, C. Donohue, D. Holmes, and O. Svintsitski, 1993. "A Parallel Implementation of the Nested Decomposition Algorithm for Multistage Stochastic Linear Programs," submitted to Annals of Operations Research.

[6] G. Dantzig, 1963. **Linear Programming and Extensions**, Princeton University Press, Princeton, NJ.

[7] Fourer, R., D. M. Gay, and B. W. Kernighan, 1992. **AMPL: A Modeling Language for Mathematical Programming**, Scientific Press, San Francisco, CA.

[8] H. I. Gassman, 1989. "Optimal harvest of a forest in the presence of uncertainty," *Canadian Journal of Forest Research*, Vol 19, pp. 1267-1274.

[9] H. I. Gassman, 1990. "MSLiP: A computer code for the multistage stochastic linear programming problem." *Mathematical Programming*, 47, pp. 407-423.

[10] S. C. Graves and W. C. Jordan, 1991. "Principles of the Benefits of Manufacturing Process Flexibility," General Motors Laboratories Research Publication.

[11] J. L. Higle, and S. Sen, 1990. "Finite Master Programs in Regularized Stochastic Decomposition," Dept. of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, 85721.

[12] J. K. Ho, 1975. "Optimal Design of Multistage Structures: A Nested Decomposition Approach," *Computers and Structures*, Vol. 5, pp 249-255.

[13] J. K. Ho, 1977. "Nested Decomposition of a Dynamic Energy Model," *Management Science* Vol. 23, No 9. pp. 1022-1026.

[14] J. K. Ho, 1980. "A successive Linear Optimization Approach to the Dynamic Traffic Assignment Problem," *Transportation Science*, 14(4), pp. 295-305.

[15] J.K. Ho and E. Loute, 1981. "A Set of Linear Programming Test Problems," *Mathematical Programming 20* 245-250.

[16] International Business Machines Corp., 1991. "Optimization Subroutine Library Guide and Reference, Release 2," document SC23-0519-02, International Business Machines Corp.

[17] P. Klaassen, J. F. Shapiro, and D. E. Spitz, 1990. "Sequential Decision Models for Selecting Currency Options," IFSRC Report No. 133-90, International Financial Services Research Center, Massachussets Institute of Technology, Cambridge, MA.

[18] M. I. Kusy and W. T. Ziemba, 1986. "A Bank Asset and Liability Management Model," *Operations Research*, 34(3), pp. 356-376.

[19] F. Louveaux and Y. Smeers, 1988. "Optimal investments for electricity generation: a stochastic model and a test-problem," in **Numerical Techniques for Stochastic Optimization**, Springer-Verlag, Berlin, pp. 33-64.

[20] I. Lustig, J. Mulvey, and T. Carpenter, 1991. "Formulating stochastic programs for interior point methods," **Operations Research** 39:5, pp. 757-770.

[21] A. S. Manne, 1975."U.S. Options for the transition from oil and gas to synthetic fuels," Discussion Paper No. 26F, Public Policy Program, John F. Kennedy School of Government, Harvard University January, 1975.

[22] J. M. Mulvey, and H. Vladimirou, 1989. "Stochastic network optimization model for investment planning," *Annals of Operations Research.* Vol. 20, p. 187.

[23] J. M. Mulvey, and H. Vladimirou, 1991. "Applying the Progressive Hedging Algorithm to Stochastic Generalized Networks." *Annals of Operations Research.* Vol. 31, pp. 399-424.

[24] J. M. Mulvey, and A. Ruszczy/'nski, 1992. "A New Scenario Decomposition Method for Large Scale Stochastic Optimization," Technical Report SOR-91-19, Dept. of Civil Engineering and Operations Research, Princeton Univ. Princeton, N.J. 08544

[25] S. Sen, J. Mai, and J. L. Higle, 1993. " Solution of Large Scale Stochastic Programs with Stochastic Decomposition Algorithms," Technical Report, SIE Dept., University of Arizona, Tucson, AZ, 85721.

[26] S. Sen, R. D. Doverspike, and S. Cosares, 1992. "Network Planning with Random Demand," Dept. of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, 85721.

[27] M. J. Sims, 1992. "Use of a stochastic capacity planning model to find the optimal level of flexibility for a manufacturing system," Senior Design Project, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109.

[28] W. Swart, C. Smith, and T. Holderby, 1975. "Expansion Planning for a Large Dairy Farm," Chapter 8 in **Studies in Linear Programming**, H. Salkin and J. Saha, eds. North Holland, Amsterdam.

[29] S. Wolfram, 1992. *Mathematica*, **A System for Doing Mathematics by Computer,** Addison Wesley, New York, NY.

STORM Nonzero Structure