Chapter 2

The SMPS Format for Stochastic Linear Programs

*Horand I. Gassmann**

This is a brief introduction to the SMPS format, which can be used to formulate a wide variety of stochastic linear and integer programming problems. A full description of the format can be found in [3] or on the SMPS web page [2]. An earlier version of the format was published by Birge et al. [1].

The use of the format is illustrated with snippets of sample input throughout this chapter. However, the format is very complex, and an exhaustive illustration of every nuance would greatly increase the length of this presentation. For that reason, only the most commonly used features are shown. Additional examples may be gleaned from [2].

The SMPS format was designed to make it easy to convert existing deterministic linear programs into stochastic ones by adding information about the dynamic and stochastic structure. This is done using three text files: the core file, the time file, and the stochastics file (or stoch file). Each file consists of several sections, which must appear in predefined order. The first record of each section is a header record, which is followed by zero or more data records. Empty sections (containing no data) may be omitted. The records in each file are organized into fields according to the MPS line layout [4].

In the MPS format, one must distinguish between header lines and data lines. Header lines contain up to three name fields, normally in these positions:

columns 1–14	first word field
columns 15–24	second word field
columns 40–49	third word field

Data lines contain up to three name fields, two numeric fields, and a code field, normally in these positions:

*School of Business Administration, Dalhousie University, Halifax, NS, B3H 3J5, Canada (horand.gassmann@dal.ca).

columns 2 and 3	code field
columns 5–12	first name field
columns 15–22	second name field
columns 25–36	first numeric field
columns 40–47	third name field
columns 50–61	second numeric field

Names are normally restricted to eight characters and may contain any ASCII symbol; numerical input is limited to twelve characters (including digits, signs, exponents, and a decimal point). Free format input can be used if these limits are too restrictive.

The core file

The core file lists all the deterministic information of the problem, in standard MPS format. Core file information includes the name and type of each constraint and variable, a column-ordered list of coefficients in the constraint matrix, right-hand-side coefficients, and any bounds and ranges on the variables and slacks. The core file also provides placeholders for all the stochastic elements (which must be mentioned in the core file and be given a preliminary value that may or may not be meaningful). The core file thus represents a deterministic problem, which may be thought of as a typical scenario, an average case, a baseline case, or something similar.

Further information about the MPS format may be found in the IBM documentation [4]. The extensions to integer variables and quadratic objectives described there are also supported in SMPS. There also is a network format, based on the NETGEN format by Klingman, Napier, and Stutz [7]. It is even possible to mix LP and network sections in the same file.

The time file

The purpose of the time file is to allow breaking down of the core file scenario into nodes corresponding to the individual stages. The time file contains the following sections:

Section	Purpose
TIME	Gives a name to the problem in name field 2
PERIODS	Gives the name of each time period (in order)
ROWS	(optional) Specifies for each row the period to which it belongs
COLUMNS	(optional) Specifies for each column the period to which it belongs
ENDATA	End of problem data

TIME section

The second name field on the TIME header record contains the name of the problem. This name is verified against the name in the core file.

PERIODS section

The header record contains the value PERIODS in the first name field. If the core file is in time order, then the second name field can be left blank or given the value IMPLICIT. The ROWS and COLUMNS sections are not needed in this case.

Each data record contains the name of a period in the third name field. If the time file is in implicit format, then the first name field contains the name of the first column in this period, and the second name field contains the name of the first row. If the time file is in explicit format, the first two name fields are not used. The objective row must always belong to the first period.

Below is an example of the start of a time file in temporal order. In this example all columns in the core file that appear between COL1 (inclusive) and COL5 (exclusive) make up the first period, named PER1, while COL5 and all the remaining columns belong to period PER2. Similarly, ROW1 and ROW2 (along with the objective) are the rows belonging to the first period; all the others are second-period rows.

Example

```
*23456789 123456789 123456789 123456789 123456789 123456789
PERIODS        IMPLICIT
    COL1       ROW1                        PER1
    COL5       ROW3                        PER2
```

If the core file is not given in temporal order (for instance, if it was generated by an algebraic modeling language), then the keyword EXPLICIT must be placed in the second name field of the header record and the ROWS and COLUMNS sections must be present.

ROWS section

The data records in this section contain the name of a row mentioned in the core file in the first name field and the name of a time period in the second name field. The ROWS section is optional if the core file is given in temporal order.

Example

```
*23456789 123456789 123456789 123456789 123456789 123456789
ROWS
    ROW1       PER1
...
```

COLUMNS section

The data records in this section contain the name of a column mentioned in the core file in the first name field; the second name field contains the name of a time period set up in the preceding PERIODS section. The COLUMNS section is optional if the core file is given in temporal order.

The stoch file

The stoch file allows the solver to build a deterministic equivalent, which requires information about the random variables. If all the random variables are finitely distributed, this task amounts to the construction of an event tree. The event tree can be described in three different ways: scenario by scenario, node by node, or implicitly using marginal distributions. Implicit descriptions are also available if the random variables are continuously distributed, where explicit descriptions would obviously fail.

The stoch file may contain the following sections:

Section	Purpose
STOCH	Gives a name to the problem in name field 2
SIMPLE	For problems with simple recourse (optional)
ROBUST	For quadratic penalties (optional)
PLINQUAD	Piecewise linear-quadratic penalties (optional)
CHANCE	For chance constraints (optional)
ICC	For integrated chance constraints (optional)
SCENARIOS	Gives the values of stochastic elements one scenario at a time
NODES	Gives the values of stochastic elements one node at a time
INDEP	Marginal distribution of independent random variables
BLOCKS	Marginal distribution of a random vector
DISTRIB	Distribution of auxiliary random variables
ENDATA	End of problem data

The three ways of describing the event tree are mutually exclusive. That is, a stoch file cannot contain SCENARIOS, NODES, and INDEP/BLOCKS sections simultaneously. INDEP, BLOCKS, and DISTRIB sections may be mixed freely, although the order must be such that enough information is available to construct the event tree during a single pass through the file.

STOCH section

The second name field on the STOCH header record contains the name of the problem. This name is verified against the name in the core file.

SIMPLE section for simple recourse

Simple recourse problems use very special recourse matrices, which should not have to be set up by the user explicitly. This section allows the user to set up appropriate penalties for violating a constraint. The corresponding slack and surplus variables along with their constraint coefficients are automatically generated by the system.

In most applications the recourse costs will be deterministic, but if the situation calls for stochastic recourse costs, the user can reference the generated column names in a distribution section later. For that reason the SIMPLE section precedes the INDEP, BLOCKS, and DISTRIB sections.

Example

```
*23456789 123456789 123456789 123456789 123456789 123456789
SIMPLE
    SIMPLE1    ROW1       10.0                        5.0
    SIMPLE1    ROW2       50.0
```

Assume that ROW1 is an E-type row (equality constraint) and ROW2 has type G greater than or equal to contraint. Then each unit of shortage in ROW1 incurs a penalty of 10.0 units in the objective, and each unit of surplus has a penalty of 5. Surplus in ROW2 is penalized by 50 units.

Similar mechanisms exist to specify purely quadratic penalties (as used in robust optimization; see [8]) and the piecewise linear-quadratic penalties of [5].

CHANCE section for chance-constrained problems

There are individual (one-dimensional) and joint (multidimensional) chance constraints, with slightly differing syntax. CC1 is a placeholder for this particular set of constraints.

Example

```
*23456789 123456789 123456789 123456789
CHANCE          INDIV
  G CC1         ROW1          0.95
  L CC1         ROW2          0.10
CHANCE          JOINT
 JG CC1                       0.98
    ROW3
    ROW4
 JL CC1                       0.001
    ROW5
    ROW6
    ROW7
```

Let us assume that all rows mentioned above (ROW1–ROW7) are set up as G-type rows in the core file, with the algebraic equivalents

$$\sum_j a_{ij}x_{ij} \geq b_i, \quad i = 1, \ldots, 7.$$

Then the four chance constraints defined by this construct are

$$\Pr\left(\sum_j a_{1j}x_{1j} \geq b_1\right) \geq 0.95,$$

$$\Pr\left(\sum_j a_{2j}x_{2j} \geq b_2\right) \leq 0.10,$$

$$\Pr\left(\sum_j a_{3j}x_{3j} \geq b_3 \wedge \sum_j a_{4j}x_{4j} \geq b_4\right) \geq 0.98,$$

$$\Pr\left(\sum_j a_{5j}x_{5j} \geq b_5 \wedge \sum_j a_{6j}x_{6j} \geq b_6 \wedge \sum_j a_{7j}x_{7j} \geq b_7\right) \leq 0.001.$$

ICC section for integrated chance constraints

This type of constraint was introduced by Klein Haneveld [6]. Assume that a deterministic version of the problem features the constraint

$$\sum_j a_j x_j \leq b.$$

Then an integrated chance constraint is of the form

$$\mathrm{E}\left(\sum_j a_j x_j \,\middle|\, \sum_j a_j x_j > b\right) \Delta\, d,$$

where Δ is an arbitrary relation ($\geq$, $\leq$, or $=$). The format is very similar to that of the individual chance constraints. (Details can be found in [2].)

SCENARIOS section

This section describes an explicit event tree scenario by scenario. There is a unique node in the event tree that belongs to the first period. This node is sometimes called the *root node*. Scenarios are identified with the leaf nodes of the tree. One could think of the scenarios as paths from the root node to the leaves, but it is easier to deal with them in the following simplified manner. One scenario (the root scenario) represents a path from the root node to one of the leaves. All other scenarios branch from a parent scenario at some time before the final period. Information up to the branch period is shared between a scenario and its parent scenario; only after the branch occurred will there be duplicate information. This point of view is advantageous because it allows for a reduction of redundancy in the tree, which compresses the size of the stoch file.

Each scenario is identified by its name, the name of its predecessor in the event tree, the period in which the branch occurred, and the probability of seeing this particular scenario. Any data items not specifically mentioned in the stoch file are inherited by each scenario from its parent scenario, which in turn may inherit them from its parent scenario, and so on. The base scenario inherits its data from the core file.

Example

```
*23456789 123456789 123456789 123456789 123456789
SCENARIOS
 SC SCEN_A     'ROOT'     0.3              PERIOD1
```

```
    COL2        ROW3        1.0
UP  BOUND1      COL5        1.0
SC  SCEN_B      SCEN_A      0.2             PERIOD3
UP  BOUND1      COL5        2.0
SC  SCEN_C      SCEN_A      0.1             PERIOD2
    COL2        ROW3        2.0
UP  BOUND1      COL5        2.0
```

This is a description of the distribution of two stochastic elements: the matrix coefficient in position COL2/ROW3 and the upper bound on variable COL5, which we assume to occur in stages 2 and 3, respectively. In scenario *A* both elements take a value of 1.0, while in scenario *B* the upper bound on COL5 is 2.0. (Both scenarios share the information in stage 2, including the coefficient in position COL2/ROW3, as well as the optimal decisions.) In scenario *C* both elements have value 2.0. Scenario A happens with probability 0.3, scenario B with 0.2, and C with 0.1. Since these add to less than one, there is at least one more scenario in this tree.

NODES section

This section describes the construction of an event tree one node at a time, which is very convenient when the depth of the tree is not uniform (containing so-called trap states) or when the number of rows or columns associated with a particular period depends on the history of the data process. It is possible to treat such stochastic problem dimensions in a scenario-wise description by introducing a number of empty rows and columns, but these phantom elements (phantom rows, phantom columns, and phantom nodes) introduce unnecessary overhead into the problem formulation. The NODES section provides an alternate way to describe the event tree, without the use of phantom elements.

For an example of how the NODES section can be used, refer to Gassmann and Schweitzer [3] or the SMPS web page [2].

INDEP section

This section describes the modeling of independent random variables. The SMPS format allows for three different forms of the distribution: discretely distributed, possessing a special distribution with no more than two parameters, or accessing a user-defined subroutine. The header record distinguishes among these forms by placing appropriate keywords into the second name field (see table below). The third keyword may contain the value ADD, MULTIPLY, or REPLACE to indicate how the value of the random variable is to modify the information found in the core file. The default is REPLACE.

The two-parameter distributions in Table 2.1 are supported.

Example 1

```
*23456789 123456789 123456789 123456789 123456789 123456789
INDEP           DISCRETE
    COL1        ROW8        6.0             PERIOD2     0.5
    COL1        ROW8        8.0             PERIOD2     0.5
```

Table 2.1. *Types of random variables.*

Keyword in second name field of header record	Distribution	First numeric field	Second numeric field
DISCRETE	Discrete distribution	Value	Probability*
UNIFORM	Continuous uniform on $[a, b]$	a	b
NORMAL	Normal distribution	mean μ	variance σ^2
GAMMA	Gamma distribution (on $[0, \infty)$) $f(x) = \frac{1}{b\Gamma(c)}(\frac{x}{b})^{c-1}e^{x/b}$	scale b	shape c
BETA	Beta distribution (on $[0, 1]$) $f(x) = \frac{x^{n-1}(1-x)^{m-1}}{B(n,m)}$	n	m
LOGNORM	Lognormal distribution	$\mu = \mathrm{E}(\ln L)$	$\sigma^2 = \mathrm{Var}(\ln L)$
(any other)	User-defined subroutine	(not used)	(not used)

*Possible values for each random element must be listed in sequence, and the probabilities must sum to 1.

In this example, the entry COL1/ROW8 takes value 6.0 with probability 0.5 and 8.0 with probability 0.5.

Example 2

```
*23456789 123456789 123456789 123456789 123456789 123456789
INDEP          DISCRETE                    ADD
    COL1       ROW8        -1.0            PERIOD2     0.5
    COL1       ROW8         1.0            PERIOD2     0.5
```

Assuming that the core file contains a value of 7.0 for the coefficient COL1/ROW8, the entry COL1/ROW8 will again take value 6.0 with probability 0.5 and 8.0 with probability 0.5.

Example 3

```
*23456789 123456789 123456789 123456789 123456789 123456789
INDEP          UNIFORM
    COL1       ROW8         8.0            PERIOD2     9.0
INDEP          NORMAL
    RHS        ROW9        10.0            PERIOD2     4.0
```

In this example the random entry COL1/ROW8 is uniformly distributed over the interval [8.0, 9.0], and the right-hand side in ROW9 is normally distributed with mean 10.0 and variance 4.0.

The SMPS format provides for a mechanism by which the user can generate the realizations by accessing a user-defined subroutine. Any modifier on the header record different from the keywords for univariate or multivariate distributions, that is, other than DISCRETE, UNIFORM, NORMAL, GAMMA, BETA, LOGNORM, MVNORMAL, LINTRAN, is taken to be the name of a subroutine, which must be provided and distributed by the user.

BLOCKS section

This section allows the user to set up multidimensional random vectors that may have correlated components but are independent of other random elements in the problem. The SMPS format allows for discrete distributions, multivariate normal distributions, and user-defined subroutines.

Example

```
*23456789 123456789 123456789 123456789
BLOCKS          DISCRETE
 BL BLOCK1      PERIOD2     0.5
    COL1        ROW6       83.0
    COL2        ROW8        1.2
 BL BLOCK1      PERIOD2     0.2
    COL2        ROW8        1.3
 BL BLOCK1      PERIOD2     0.2
    COL1        ROW6       84.0
 BL BLOCK1      PERIOD2     0.1
    COL1        ROW6       84.0
    COL2        ROW8        0.0
```

In this example, the two matrix coefficients ROW6/COL1 and ROW8/COL2 take values of (83, 1.2), (83, 1.3), (84, 1.2), and (84, 0) with respective probability 0.5, 0.2, 0.2, and 0.1. (Note that the second and third realizations of BLOCK1 mention only one of the two elements; the other element's value is inherited from the first realization.)

Sometimes the user may want to specify stochastic coefficients determined by underlying random phenomena (factors) of lower dimension. A simple form of this relationship is an affine transformation, which can be written as $v = Hu + c$, where the matrix H and the vector c are deterministic and u is a random vector. This device is also available in SMPS. For further details as well as examples, consult Gassmann and Schweitzer [3] or the SMPS web page [2].

DISTRIB section

This section describes the use of random variables (univariate and multivariate) that do not correspond directly to stochastic elements within the current problem. This device is needed to adequately model the linear transformations of the last section, since the MPS data record does not allow sufficient fields to specify correlations between random elements. The main purpose of a DISTRIB section is to provide an alias (a single name) by which the random variable can be referenced in a later BLOCKS section.

There may be many DISTRIB sections; it is assumed that each of them defines random variables independently of the others. Conversely, each new group of random variables must be introduced by a new DISTRIB header line.

Example

```
*23456789 123456789 123456789 123456789 123456789 123456789
DISTRIB        MVNORMAL
 BL BL1
    U1                      0.0                       1.0
    U2                      0.0                       1.0
 CV
    U1         U2           0.5
```

This example defines a bivariate normal random vector whose components have mean 0, variance 1, and covariance 0.5. The vector can be referred to as BL1, its first component as U1, and its second component as U2.

Linking constraints

Most constraints in stochastic programming are of the "replicating" kind. This means that each constraint contains variables from at most one node in each period. The constraint is thought to belong to the latest period in which it has variables and is replicated for each realization of the random elements.

In some special instances (for example, to model risk constraints in financial applications) it may be useful to link together variables belonging to different nodes in the same period. (For an example, consult [9].) The easiest way to include such linking or global constraints in a stochastic program is to assign the constraint to the first period. (This is best done using the explicit form of the time file.)

Solver capabilities and error recovery

The SMPS format is extremely flexible and can define a wide variety of problems, far exceeding the capability of any existing stochastic programming solver. It should not be assumed, therefore, that a given problem can be solved with a particular solver nor that in fact there exists any solver capable of solving the problem. However, any SMPS parser should recover gracefully when it encounters a section or construct its solver is not prepared to deal with.

Acknowledgments

This research was supported by a grant from the National Science and Engineering Research Council of Canada (NSERC). The author thanks Stein W. Wallace for many helpful comments.

Bibliography

[1] J. R. Birge, M. A. H. Dempster, H. I. Gassmann, E. A. Gunn, A. J. King, and S. W. Wallace, *A standard input format for multiperiod stochastic linear programs*, Math. Program. Soc. Committee Algorithms Newsletter, 17 (1987), pp. 1–19.