CUSTOMIZED BLOCK STRUCTURES IN ALGEBRAIC MODELING LANGUAGES: THE STOCHASTIC PROGRAMMING CASE [1]

Emmanuel Fragnière [*], **Jacek Gondzio** [**,2] and
Robert Sarkissian [**]

[*] *HEC, Department of Management, University of Lausanne, BFSH1,*
1015 Dorigny-Lausanne, Switzerland
Fax: +41 21 692 33 05
e-mail: Emmanuel.Fragniere@hec.unil.ch
[**] *LOGILAB, HEC, Section of Management Studies, University of Geneva, 102 Bd. Carl–Vogt, CH-1211 Genève 4, Switzerland*
Fax: +41 22 705 81 04
e-mail: Jacek.Gondzio@hec.unige.ch
Robert.Sarkissian@hec.unige.ch

Abstract: Extracting complex block structures from an anonymous mathematical program is a difficult task. It is however a mandatory step to exploit them with adequate algorithmic techniques. Moreover, most economic models are usually built with an Algebraic Modeling Language (AML) which loose any structure. The recently developed concept SET (Structure Exploiting Tool) responds to these needs. This approach relates directly to the "semantic" of the original model that was used to generate the corresponding anonymous mathematical program. Examples from stochastic programming are presented.

Keywords: Decomposition methods, Modelling, Parallel algorithms, Stochastic programming, Structured programming.

1. INTRODUCTION

Many economic models based on mathematical programming may display complex nested block structures. Unfortunately, it remains difficult to exploit these structures within algebraic modeling languages.

Thanks to the concept SET (Structure Exploiting Tool) (Fragnière *et al.*, 1997), this task can be easily performed directly from an algebraic modeling language. To illustrate these ideas one presents the structures of deterministic equivalent formulations of multistage stochastic problems that are generated with the GAMS modeling language. A parallel interior-point based decomposition method has been linked to GAMS using that framework.

One focuses here on a particular formulation of multistage stochastic programming models, namely their deterministic equivalents programming problems. These problems are interesting since they combine two dimensions: time and uncertainty. In order to extract and exploit their nested block structures within the algebraic modeling language, one employs the concept SET that is made up of two distinctive parts: SPI (Structure Passing Interface) which extracts the structure desired by the modeler and an appropriate SES (Structure Exploiting Solver) which exploits the extracted form.

[1] This research was supported by the Fonds National de la Recherche Scientifique Suisse, grants #12-42503.94 and #1214-049696.96/1.

[2] On leave from the Systems Research Institute, Polish Academy of Sciences, Newelska 6 , 01-447 Warsaw, Poland

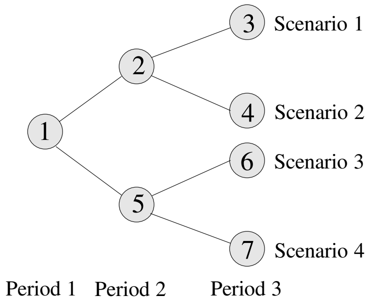

Fig. 1. A simple event tree

The SES (Structure Exploiting Solver) hooked to the GAMS modeling language is an interior-point based decomposition method (Gondzio and Vial, 1997). It has been specially implemented for this presentation to solve a dual block-angular structure. That is the block structure displayed by a stochastic programming model. The master problem is solved by ACCPM (Goffin and Vial, 1990; Gondzio *et al.*, 1996). The subproblems are solved with HOPDM (Gondzio, 1995). Experiments are realized on a IBM RISC 6000 machine and on a cluster of PC's.

To the best of ones knowledge, this is the first time that by an efficient management of multiple sets complex nested structures have been exploited within the algebraic modeling language.

This paper is organized as follows. In Section 2, one explains the links that exist between the multistage stochastic programming problem and nested dual block-angular structures. In Section 3, one recalls the concept SET (Structure Exploiting Tool) that is used for the first time to extract nested block structures. In Section 4, one shows that completely different structures can be extracted from the same model by manipulating multiple sets. In Conclusion, one indicates the kind of results provided with this approach.

2. NESTED STRUCTURES IN STOCHASTIC PROGRAMS

Generally, economic models include several dimensions such as space, time and uncertainty. Multistage stochastic-programming problems form a class of mathematical programs that contains two kinds of dimensions: time and uncertainty. In its simplest form the stochastic process is represented as an event tree describing the unfolding of the uncertainty over the period of planning (see Figure 1).

A path from the root to an extremity of the event tree represents a scenario. With each scenario a given probability is associated. At each node of the event tree, a set of constraints and an economic function which involves variables specific to that node and also specific to the predecessor nodes, are defined. For instance, node 1 of the tree in Figure 1 corresponds to the first stage and associated decisions are identical for scenarios 1, 2, 3 and 4. Note that in stage-2 decisions

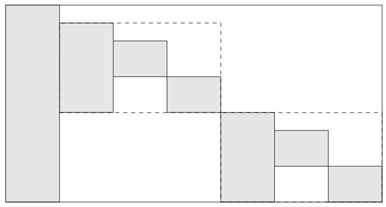

Fig. 2. The constraints matrix of the corresponding mathematical program

of scenario 1 and 2 are identical. In the same way decisions of scenario 3 and 4 are identical. If the ordering of time and randomness dimensions is maintained during the generation of the mathematical program, the Jacobian matrix of the multistage stochastic problem displays a nested dual-block structure, as shown in Figure 2. Links between the nested dual block-angular structure and the algebraic formulation of the original model can be easily established referring to the general mathematical formulation of multistage stochastic programming problems (see (Ruszczyński, 1997)).

From the algebraic formulation provided by the user, one can find an interesting block structure. Unfortunately, the latter cannot be exploited. Indeed, assuming that the model is written with an algebraic modeling language such as GAMS (Brooke *et al.*, 1992) or AMPL (Fourer *et al.*, 1993) the structure easily identified in the algebraic formulation will be probably lost when the corresponding mathematical program is sent to the solver, because each algebraic modeling language uses its own ordering scheme to make the translation between the algebraic formulation and the complete mathematical program. They indeed produce an anonymous mathematical program. In the next section a tool based on the concept SET (Fragnière *et al.*, 1997) is presented to extract efficiently dual block-angular structure from the multistage stochastic program formulated with the algebraic modeling language.

3. NESTED BLOCK STRUCTURE OF MATHEMATICAL PROGRAM

SET is used here for the first time to retrieve nested block angular structure from the algebraic modeling language. This is done in two steps. First, SPI (Structure Passing Interface) enables retrieving the desired structure from the anonymous mathematical program built by the algebraic modeling language. Second, SPI passes the special structure of the problem to a SES (Structure Exploiting Solver). SET can be integrated with any Algebraic Modeling Language (AML) as shown in Figure 3.

The model tested in this section is a simple financial planning model suggested by Birge (1985). Possibilities of investment at each period consist of 4 securities and cash. There are no transaction costs. The initial capital and the desired wealth at the end of

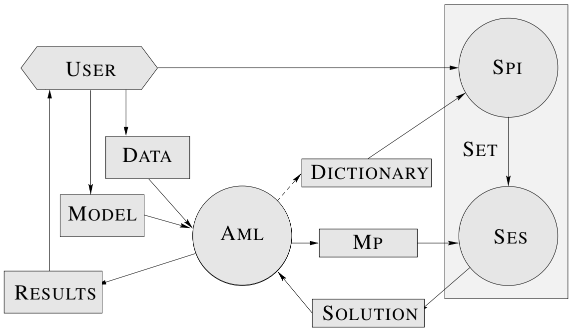

Fig. 3. Integrating SET with an AML

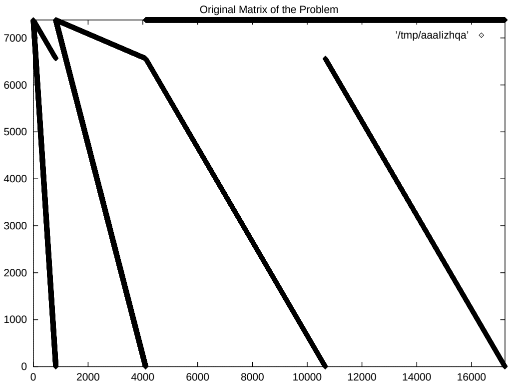

Fig. 4. Jacobian matrix generated by GAMS

the planning period are defined exogenously. Prices of securities have been computed by a multivariate log-normal random generator implemented in MATLAB. Prices are assumed to be independent between periods. The time scale dimension is defined over T periods. The uncertainty dimension is defined at each node for N realizations. The corresponding event tree is symmetric and involves N different branches at each node excepted for the terminal nodes. Hence the total number of scenarios is equal to N^T. The objective function is a piecewise linear function. It corresponds to the expected utility of all scenarios (a scenario is penalized strongly when the goal is not reached). The model is written in the GAMS modeling language. In Figure 4, the Jacobian matrix (constraints matrix) generated by GAMS when using its own ordering scheme is presented. No apparent block structure can be identified from it. Then passing a few information about time and uncertainty dimensions SPI produces directly from GAMS a nested dual block angular structure as presented in Figure 5.

4. CUSTOMIZED FORMS FROM THE SAME MODEL

When several dimensions (index sets) are present in the algebraic model, many different structures can be potentially found in the same model. An important feature of this approach is that it leaves to the user the possibility of choosing completely different customized structures from the same model. Hence in real-life complicated models, the user can choose the

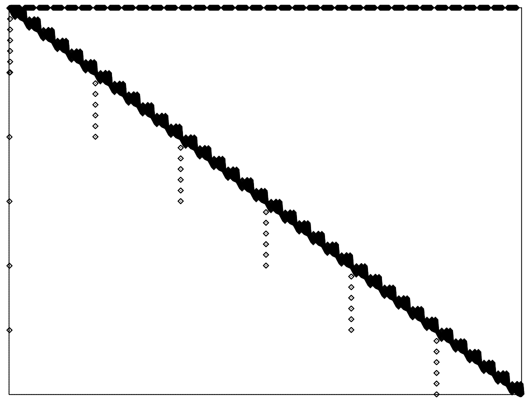

Fig. 5. Reordered Jacobian matrix displaying a nested dual block angular structure

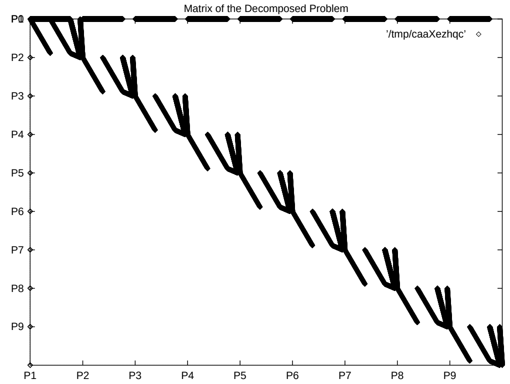

Fig. 6. Dual block angular structure

most favorable structure making the problem easier to be solved by the specialized solver. It is indeed well known that the structure exploiting solvers give better results depending on the sparsity of sub-blocks involved. For instance, the decomposition algorithm usually performs more efficiently when the master problem is small and there is a considerable number of hopefully similar subproblems.

Returning to the financial planning model. By retaining only the randomness dimension at the level of the first period, it is possible to extract a dual block angular structure as shown in Figure 6. One notices that the sub-blocks are now quite different from those displayed in Figure 2. On the other hand, by retaining only the time dimension one could also display a staircase (i.e. dynamic) structure from the same model. Finally, by an efficient management of both dimensions at the time it is also possible to adjust the sizes of the master problem and of the subproblems by moving columns or variables from different stages to other stages as the example shown in Figure 7. The resulting matrix is presented in Figure 8.

5. CONCLUSION

Algebraic modeling languages loose important information the solver could have taken advantage of. SET (Structure Exploiting Tool) is a new framework for

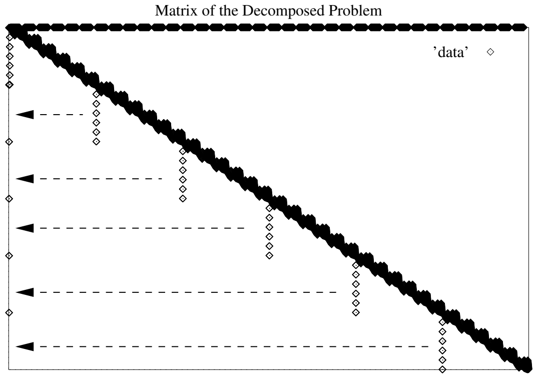

Fig. 7. Moving columns from the second stage to the first stage

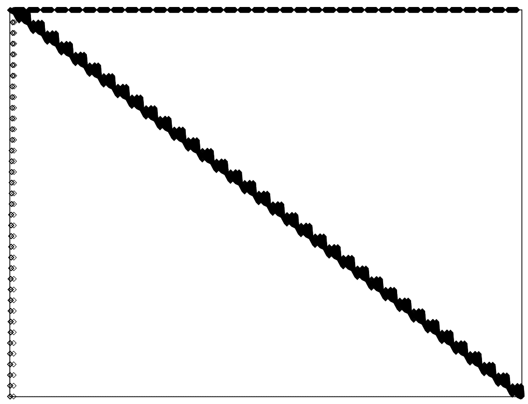

Fig. 8. Resulting dual block angular structure

retrieving this information from the algebraic formulation. It can handle any structure of the model, e.g. decomposable. One focused in this paper on multistage stochastic programming problems since they contain two dimensions: time and uncertainty. Some models of this type are tested with a cutting-edge IPM-based decomposition solver on a sequential and parallel computing environments. Some of the experiments that will be presented are shown in Table 1. One notices that the largest problem (137,258 rows and 338,339 columns) is solved in less than one hour (including the time spent in the generation of the LP by GAMS, its partitioning, and its solution by the decomposition algorithm).

Table 1. CPU times in seconds, IBM RS6000

Problem	Rows	Columns	Time
STOC1	1094	3279	19
STOC2	5462	15018	87
STOC3	19532	50781	338
STOC4	55988	139968	938
STOC5	137,258	338,339	3411

6. REFERENCES

Birge, J. R. (1985). Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research* **33**, 989–1007.

Brooke, A., D. Kendrick and A. Meeraus (1992). *GAMS: A User's Guide*. The Scientific Press. Redwood City, California.

Fourer, R., D. Gay and B. Kernighan (1993). *AMPL: A Modeling Language for Mathematical Programming*. The Scientific Press. San Francisco, California.

Fragnière, E., J. Gondzio, R. Sarkissian and J.-P. Vial (1997). Structure exploiting tool in algebraic modeling languages. Technical report. Logilab, University of Geneva. 102 Bd Carl–Vogt, CH-1211.

Goffin, J.-L. and J.-P. Vial (1990). Cutting planes and column generation techniques with the projective algorithm. *Journal of Optimization Theory and Applications* **65**, 409–429.

Gondzio, J. (1995). HOPDM (version 2.12) – a fast LP solver based on a primal-dual interior point method. *European Journal of Operational Research* **85**, 221–225.

Gondzio, J. and J.-P. Vial (1997). Warm start and ϵ-subgradients in the cutting plane scheme for block-angular linear programs. Technical report. Logilab, University of Geneva. 102 Bd Carl-Vogt, CH-1211. To appear in *Computational Optimization and Applications*.

Gondzio, J., O. du Merle, R. Sarkissian and J.-P. Vial (1996). ACCPM - a library for convex optimization based on an analytic center cutting plane method. *European Journal of Operational Research* **94**, 206–211.

Ruszczyński, A. (1997). Decomposition methods in stochastic programming. In: *Mathematical programming* (Thomas M. Liebling and Dominique de Werra, Eds.). Vol. 79. pp. 333–353. North-Holland.