Subtree decomposition for multistage stochastic programs

Shane Dye*
University of Canterbury, New Zealand

February 21, 2005

Keywords: Stochastic programming, scenario tree, decomposition.

Extended Abstract

A number of methods for solving multistage stochastic linear programs with recourse decompose the deterministic equivalent [4] to form subproblems based on scenarios (*e.g.*, Rockafellar and Wets [6] and Mulvey and Ruszczyński [5]). Other methods use forms of Benders decomposition to form subproblems based on nodes of the scenario tree (*e.g.*, Birge [1] and Gassmann [2]). Both types of algorithms can be conceptualized as a decomposition of the scenario tree [4] of the stochastic program. Also, under both of these approaches the number and size of the subproblems is predefined: each scenario or node corresponds to a subproblem.

This research describes an algorithmic approach which allows more flexibility in the structure and size of the subproblems. Two existing decomposition approaches are used as a basis for applying the principles. The deterministic equivalent is reformulated in two ways in order to meet the needs of two underlying decomposition approaches. We use the concept of the *extension* of a node set, which adds the predecessor of each node to the node set.

A *subtree cover* of the scenario tree is a set covering of the scenario nodes with the property that for each node set the subgraph induced by its extension is connected.

A *subtree partition* of the scenario tree is a partition of the scenario nodes with the property that the subgraph induced by each node set is connected.

Note the different node sets that are required to induce subtrees for the above. Different subtree covers and subtree partitions will lead to different subtree decompositions for the same problem.

0.1 Subtree partition decomposition

In its most straightforward form, the nested Benders decomposition approach decomposes the scenario tree into subproblems corresponding to individual nodes. The influence of a node's successors is applied through additional constraints added to the node's subproblem. These constraints are passed towards the root, while partial solutions are passed in the other direction to direct the generation of constraints. Nested Benders decomposition is used as the basis for a more general decomposition.

For any subtree partition of the scenario tree form the corresponding *partition extensive form* of a stochastic program by re-indexing the deterministic equivalent so that variables are additionally labelled with the index of the node subset containing their corresponding node. This re-indexing exposes a new underlying tree structure, called the *partition tree*.

The partition extensive form can be viewed as a stochastic linear program with the partition tree as the underlying scenario tree. Stages and probabilities may be assigned in a straightforward manner. Since we have a stochastic linear program with an underlying scenario tree, nested Benders may be directly applied. In this way, all results for nested Benders apply directly to the

*Shane.Dye@canterbury.ac.nz

partition extensive form. This also means that the subtree partition algorithm may be implemented as a *layer* between the model (or modelling software) and a nested Benders solver (for instance MSLiP [2]).

A subtree cover decomposition

The scenario decomposition approach retains the dynamic structure of a deterministic version of the problem. The scenario tree is expanded with each scenario allocated copies of the nodes that form it with the variables and constraints corresponding to these nodes. *Nonanticipativity* constraints are added to ensure that one common decision is made for all of the nodes in the expanded tree that correspond to the same node of the original tree. These nonanticipativity constraints are then relaxed so that the problem decomposes into subproblems corresponding to each of the scenarios. This effectively *splits* those variables corresponding to nodes which have more than one scenario passing through them. Some scheme is then used to iteratively tighten the relaxation of the nonanticipativity constraints. The scenario decomposition approach is, also, generalized.

For any subtree cover of the scenario tree of a stochastic linear program, form the corresponding *cover extensive form* by adding copies of the linking variables which appear in constraints corresponding to different node subsets. Subtree weights are used to apportion the objective coefficients amongst the various copies of each variable. Additional constraints are added to ensure that the various copies of each variable have the same value. These constraints are called *implementability* constraints (adapting its use from [6]) to differentiate them from the *nonanticipativity* which may be present in the deterministic equivalent formulation.

The decomposition is effected by relaxing the implementability constraints, to form a number of subproblems, then applying some scheme to progressively tighten the relaxation. Applying the diagonal quadratic decomposition of Ruszczyński [7] as a subtree decomposition is straightforward.

This scheme is essentially an augmented Lagrangian approach. One factor affecting the convergence rate of augmented Lagrangian is the dimension of the multiplier vector. By allowing an arbitrary subtree cover we allow greater control over this dimension.

Subproblem creation

One important issue is how to take the original stochastic program (as presented to a solver) and determining which variables and constraints will play the various roles required.

The procedure outlined below determines the composition of the subproblems for both decompositions. It assumes that the deterministic equivalent is provided together with the scenario tree and a subtree partition or subtree cover. Each variable in the linear program is associated with a node of the scenario tree. These assumptions are consistent with the input standard for stochastic programs [3].

1. All variables are assigned a stage index corresponding to the stage of their associated scenario tree node.

2. Constraints are assigned a scenario tree node corresponding to any one of the youngest variables with non-zero coefficient, appearing in the constraint. Variables appearing in constraints with variables from more than one node are labelled as linking variables.

3. Each node set, S, in the subtree partition or subtree cover defines a subproblem consisting of:

   (a) all constraints whose node is in S.

   (b) all variables with a non-zero coefficient in one of those constraints.

References

[1] John R. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Oper. Res.*, 33(5):989–1007, 1985.

[2] H. Gassmann. MSLiP: a computer code for the multistage stochastic linear programming problem. *Mathematical Programming*, 47:407–423, 1990.

[3] H.I. Gassmann and E. Schweitzer. A comprehensive input format for stochastic linear programs. *Annals of Operations Research*, 104:89–125, 2001.

[4] P. Kall and S.W. Wallace. *Stochastic Programming*. Wiley, Chichester, 1994.

[5] John M. Mulvey and Andrzej Ruszczyński. A new scenario decomposition method for large-scale stochastic optimization. *Oper. Res.*, 43(3):477–490, 1995.

[6] R.T. Rockafellar and Roger J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Math. Oper. Res.*, 16(1):119–147, 1991.

[7] Andrzej Ruszczyński. On convergence of an augmented Lagrangian decomposition method for sparse convex optimization. *Math. Oper. Res.*, 20(3):634–656, 1995.