Algorithmic Operations Research

The Abridged Nested Decomposition Method for Multistage Stochastic Linear Programs with Relatively Complete Recourse

Christopher J. Donohue et John R. Birge

Volume 1, numéro 1, summer 2006

URI : https://id.erudit.org/iderudit/aor1_1art02

Aller au sommaire du numéro

Éditeur(s)

Preeminent Academic Facets Inc.

ISSN

1718-3235 (numérique)

Découvrir la revue

Citer cet article

Donohue, C. J. & Birge, J. R. (2006). The Abridged Nested Decomposition Method for Multistage Stochastic Linear Programs with Relatively Complete Recourse. *Algorithmic Operations Research*, *1*(1), 20–30.

Résumé de l'article

This paper considers large-scale multistage stochastic linear programs. Sampling is incorporated into the nested decomposition algorithm in a manner which proves to be significantly more efficient than a previous approach. The main advantage of the method arises from maintaining a restricted set of solutions that substantially reduces computation time in each stage of the procedure.

Algorithmic Operations Research Vol.1 (2006) 20–30

The Abridged Nested Decomposition Method for Multistage Stochastic Linear Programs with Relatively Complete Recourse

Christopher J. Donohue [a] John R. Birge [b]

[a]Global Association of Risk Professionals, Jersey City, NJ, USA
[b]The University of Chicago Graduate School of Business, Chicago, IL, USA

Abstract

This paper considers large-scale multistage stochastic linear programs. Sampling is incorporated into the nested decomposition algorithm in a manner which proves to be significantly more efficient than a previous approach. The main advantage of the method arises from maintaining a restricted set of solutions that substantially reduces computation time in each stage of the procedure.

Dedicated to the memory of George Dantzig who inspired us to pursue the challenge of finding optimal decisions under uncertainty.

1. Multistage Stochastic Linear Programs

Consider a multistage dynamic decision process under uncertainty that can be modeled as a linear program. Assume that the stochastic elements of the problem can be found in the objective coefficients, the right-hand side values, the technology matrices, or any combination of these. Further, assume that the stochastic elements are defined over a discrete probability space $(\Xi, \sigma(\Xi), \mathrm{P})$, where $\Xi = \Xi^2 \otimes \cdots \otimes \Xi^N$ is the support of the random data in stages two through N, with $\Xi^t = \{\xi_i^t = (h^t(\xi_i^t), c^t(\xi_i^t), T_{\cdot,1}^{t-1}(\xi_i^t), \ldots, T_{\cdot,n^{t-1}}^{t-1}(\xi_i^t))\}$. The stage t nodes of the scenario tree are defined by a realization ξ_i^t of the stage t random parameters and a history of realizations $(\xi_{i^2}^2, \ldots, \xi_{i^{t-1}}^{t-1})$ of the random parameters through stage $t-1$.

A multistage stochastic linear programming problem can then be formulated (Table 1).

$$\min c^1 x^1 + \mathcal{Q}^2(x^1)$$

$$s.t.\ W^1 x^1 = h^1$$

$$x^1 \geq 0,$$

where for any $(t-1)$-stage scenario decision $x^{t-1,k}$, the stage t expected recourse or value function $\mathcal{Q}^t(x^{t-1,k})$ is given as:

$$\mathcal{Q}^t(x_k^{t-1,k}) = \sum_{m \in \mathcal{D}(k)} \frac{p^{t,m}}{p^{t-1,k}} Q^t(x^{t-1,k}, \xi^{t,m}).$$

Let h denote a t-stage scenario in $\mathcal{D}(k)$. Then, the stage t recourse cost obtained with the stage $t-1$ decision $x^{t-1,k}$ and realization $\xi^{t,h} \in \Xi^t$ is another optimization problem given by:

$$\begin{aligned} Q^t(x^{t-1,k}, \xi^{t,h}) &= \min c^t(\xi^{t,h}) x^{t,h} + \mathcal{Q}^{t+1}(x^{t,h}) \\ s.t.\ W^t x^{t,h} &= h^t(\xi^{t,h}) - T^{t-1}(\xi^{t,h}) x^{t-1,k} \\ x^{t,h} &\geq 0. \end{aligned}$$

In stage N, it is assumed that $\mathcal{Q}^{N+1}(x^N) = 0$ for all values of x^N.

Table 1

Ξ^t	support of the stage t random parameters
ξ_i^t	realization of stage t random parameters
$(\xi_{i^2}^2, \ldots, \xi_{i^t}^t)$	t-stage scenario
$\xi^{t,k}$	realization of stage t random parameters in t stage scenario k
$p^{t,k}$	probability of t stage scenario k
$a(k)$	ancestor $(t-1)$-stage scenario of t-stage scenario k
$\mathcal{D}(k)$	set of descendant $(t+1)$-stage scenarios of t-stage scenario k

Email: Christopher J. Donohue [chris.donohue@GARP.com], John R. Birge [john.birge@ChicagoGSB.edu].

A two-stage stochastic program is said to have *relatively complete recourse* if, given any first stage feasible solution x^1, there exists a feasible solution to any realized second stage subproblem with probability one. Similarly, a multistage stochastic linear program is said to have *relatively complete recourse* if, given any feasible solution $(x^1, \ldots, x^{t-1})$ to any $(t-1)$-stage scenario k (defined by the sequence of realizations $(\xi^2, \ldots, \xi^{t-1})$ in stages 2 through $t-1$), there exists a feasible solution to any realized stage t descendant subproblem of k with probability one. Relatively complete recourse simplifies decomposition algorithms since feasibility is ensured.

1.1. *Decomposition Methods*

Decomposition algorithms have proven effective in decreasing times to solve stochastic linear programs over direct solution approaches (see, for example, Birge [2], Gassmann [9], Birge et al. [3]). These algorithms involve separating the problem into subproblems associated with each node in the scenario tree. Given its convexity and piecewise linearity, the expected recourse function $\mathcal{Q}(x)$ in the first stage objective function can be replaced by an unrestricted variable θ^1 which is constrained by a finite set of linear inequality constraints. The goal of decomposition algorithms is to construct only the necessary portions of the expected recourse function with linear constraints and at the same time bound the expected recourse function for all possible first stage decisions. The hope is that the similarities between each of the subproblems and their smaller sizes allow the problem to be solved more efficiently.

The L-shaped algorithm was developed for two-stage stochastic linear problems (proposed in Dantzig [7]) by Van Slyke and Wets [13]. This algorithm is an outer linearization of the second stage primal recourse problem with additional steps taken to ensure feasibility for all possible second stage subproblems. The Nested Decomposition algorithm (Birge [2]) extended the L-Shaped algorithm for multistage stochastic linear programs.

As in the L-Shaped algorithm, the Nested Decomposition (ND) algorithm replaces the stage $t+1$ expected recourse function in each subproblem with a free variable $\theta^{t,k}$, and then constrains $\theta^{t,k}$ with successive linear approximations of $\mathcal{Q}^{t+1}(x^{t,k})$. The linear approximations, or optimality cuts, serve as lower bounds on the expected recourse functions for all feasible solutions of the stage t subproblem.

At node k in stage t, the following subproblem is solved:

$$
\begin{aligned}
Q^t(x^{t-1,a(k)}, \xi^{t,k}) &= \min c^t(\xi^{t,k})x^{t,k} + \theta^{t,k} \\
\text{s.t. } W^t x^{t,k} &= h^t(\xi^{t,k}) - T^{t-1}(\xi^{t,k})x^{t-1,a(k)} \\
D_i^{t,k} x^{t,k} &\geq d_i^{t,k} i = 1, \ldots, r^{t,k} \qquad (1) \\
E_i^{t,k} x^{t,k} + \theta^{t,k} &\geq e_i^{t,k} i = 1, \ldots, s^{t,k} x \qquad (2) \\
x^{t,k} &\geq 0 \\
\theta^{t,k} &\text{ unrestricted.}
\end{aligned}
$$

Constraints (1) and (2) are feasibility and optimality cuts, respectively. The feasibility cuts constrain the set of stage t solution to those which have feasible recourse over the remainder of the planning horizon. The optimality cuts constrain the expected recourse approximation $\theta^{t,k}$ to a value which is a lower bound on the actual expected recourse function value. Both types of cuts are generated successively during the algorithm.

Both the L-Shaped and ND algorithms begin with no feasibility or optimality cuts in any of the subproblems, and with $\theta^{t,k}$ restricted to zero for all values of k and t. The first stage subproblem is solved. Each of the second stage subproblems is then considered, using the current first stage solution to adjust the right hand side. The algorithm continues forward through the scenario tree in this manner.

If a stage t subproblem $Q^t(\hat{x}^{t-1,a(k)}, \xi^{t,k})$ is infeasible for a particular $(t-1)$-stage scenario solution $\hat{x}^{t-1,a(k)}$, an additional constraint is added to the stage $t-1$ subproblem which removes $\hat{x}^{t-1,a(k)}$ from the set of feasible solutions. By duality, the infeasibility of $Q^t(\hat{x}^{t-1,a(k)}, \xi^{t,k})$ implies that there exists a direction $\hat{\pi}^{t,k}$ along which the dual problem becomes unbounded and which satisfies $\hat{\pi}^{t,k}(h^t(\xi^{t,k}) - T^{t-1}(\xi^{t,k})\hat{x}^{t-1,a(k)}) > 0$. Thus, $\hat{x}^{t-1,a(k)}$ is removed from the set of feasible stage $t-1$ solutions of $Q^{t-1}(\bullet, \xi^{t-1,a(k)})$ by adding the constraint,

$$
D_{r^{t-1}+1}^{t-1,a(k)} x^{t-1,a(k)} \geq d_{r^{t-1}+1}^{t-1,a(k)},
$$

where $D_{r^{t-1}+1}^{t-1,a(k)} = \hat{\pi}^{t,k} T^{t-1}(\xi^{t,k})$, $d_{r^{t-1}+1}^{t-1,a(k)} = \hat{\pi}^{t,k} h^t(\xi^{t,k})$, and r^{t-1} is the current number of feasibility cuts in the stage $t-1$ subproblem.

If the first stage subproblem is infeasible, then the problem is infeasible and the algorithm terminates.

Typically, once the forward pass has solved each subproblem in the scenario tree, the process of developing optimality cuts begins. Starting in stage $N-1$, the algorithm begins a backward pass through the scenario

tree. At subproblem k in stage t, a lower bound on the stage $t+1$ expected recourse function $\mathcal{Q}^{t+1}(x^{t,k})$ associated with the current node is established as a linear function of $x^{t,k}$ and the weighted sum of the optimal dual multipliers, $(\pi^{t+1,m}, \delta^{t+1,m}, \sigma^{t+1,m})$, from each of the stage $t+1$ descendant nodes. Suppose the current solution at this node is $\tilde{x}^{t,k}$ and the expected recourse approximation value is $\tilde{\theta}^{t,k}$. The lower bound is formed as follows:

$$\begin{aligned}\theta^{t,k} \geq \sum_{m\in\mathcal{D}(k)} \frac{p^{t+1,m}}{p^{t,k}} \big(&\pi^{t+1,m}(h^{t+1}(\xi^{t+1,m}) \\ &- T^t(\xi^{t+1,m})x^{t,k}) + \delta^{t+1,m}d^{t+1,m} \\ &+\sigma^{t+1,m}e^{t+1,m}\big) \\ = \bar{\theta}^{t,k} \quad &\text{for } x^{t,k} = \tilde{x}^{t,k}.\end{aligned}$$

A constraint of the form of (2) is then established by letting

$$E_i^{t,k} = \sum_{m\in\mathcal{D}(k)} \frac{p^{t+1,m}}{p^{t,k}} \pi^{t+1,m} T^t(\xi^{t+1,m}) \text{ and}$$

$$\begin{aligned}e_i^t = \sum_{m\in\mathcal{D}(k)} \frac{p^{t+1,m}}{p^{t,k}} \big(&\pi^{t+1,m} h^{t+1}(\xi^{t+1,m}) \\ &+\delta^{t+1,m}d^{t+1,m} + \sigma^{t+1,m}e^{t+1,m}\big)\end{aligned}$$

for $i = s^{t,k}+1$. If the current expected recourse function approximation is no longer valid (i.e., $\tilde{\theta}^{t,k} < \bar{\theta}^{t,k}$), then this linear constraint (*optimality cut*) is added to this node's subproblem. The process continues until the latest first stage optimality cut is not added to the first stage subproblem, at which point the problem is solved.

Each cut can be uniquely assigned to an optimal basis of a subproblem, which has a finite number of bases; thus, both the L-Shaped and ND algorithms terminate finitely. Further, both algorithms terminate with an optimal solution (if one exists) since termination in both only occurs if $\theta^1 = \mathcal{Q}^2(x^1)$ or the problem is infeasible or unbounded.

1.2. *Computational Improvements*

Various techniques have been explored for improving the computational efficiency of decomposition algorithms. After solving each subproblem in a particular stage in the course of the ND algorithm, the choice of which adjacent stage to solve next does not disrupt the convergence of the algorithm. Hence, different sequencing protocols have been suggested. The protocol described above, referred to as the "fast-forward, fast-back" procedure, involves continuing in the current direction until the process cannot proceed in that direction. Wittrock [15] argues that by changing direction as seldomly as possible, the procedure most effectively propogates information throughout the tree. Alternate strategies include the "fast-forward" procedure and the "fast-back" procedure. The "fast-forward" procedure (Birge [2]) only proceeds from stage t to stage $t-1$ when all current solutions in stages $t, \ldots, N$ are optimal. The "fast-back" procedure (Gassmann [9]) only proceeds from stage t to stage $t+1$ when all current solutions in stages $1, \ldots, t$ are optimal. Results from implementations of the ND algorithm (Gassmann [9], Birge et. al. [3]) suggest that the "fast-forward, fast-back" protocol generally works most effectively.

To improve the quality of each iteration, Birge and Louveaux [4] propose the multicut version of the ND algorithm. In the general decomposition algorithm mentioned, for a given stage t decision x^t, all possible realizations in stage $t+1$ are optimized in order to obtain their optimal simplex multipliers. These multipliers are then aggregated in order to generate one cut. A single θ variable is used to approximate the expected recourse function value, and its value is constrained by these aggregated cuts. Instead, Birge and Louveaux [4] suggest that more information from a node's descendants may be gained by disaggregating optimality cuts. The resulting multicut version uses a θ variable corresponding to each descendant realization of ξ^{t+1} and constrains each by cuts generated from that realization multiplied by its probability. The obvious disadvantage to the multicut version is the much more rapid increase in the size of each subproblem, but the big advantage is the increase in the information that is being passed back in each iteration. The hope in this approach is that the increase in information will decrease the number of iterations needed to converge at the first stage and that this savings will outweigh the added effort needed to solve each subproblem.

For problems with stochastic elements found only in the right hand sides and the technology matrices, the stage t recourse function $Q^t(x^{t-1}, \xi^t)$ is also a convex function of the random vector ξ^t; convexity of the recourse function when the technology matrix is random follows since, for the given stage $t-1$ solution x^{t-1}, the technology matrix is found in the right hand side of the problem. Hence, a lower bound on the stage t expected recourse function $\mathcal{Q}^t(x^{t-1})$ can be established by solving only the recourse function with the expected value of ξ^t, $Q^t(x^{t-1}, \bar{\xi}^t)$. In particular, if the random

parameters in each stage are independently distributed, a lower bound can be established by solving the deterministic problem where the random parameters in each stage are replaced by their expected values. The cuts generated in each stage of the expected value problem are valid cuts to the true expected recourse function, and so, can be passed to each node in that stage in the true scenario tree. Solving the original problem can then begin with this additional information. This helps primarily with reducing the number of iterations needed for convergence. Computation times have been reduced by as much as 40% using this technique (Donohue et. al. [8])

Techniques have also been developed to improve computational efficiency within subproblems by taking advantage of similarities. Assuming that the objective cost coefficients are not stochastic, the stage t subproblems only differ in their right hand sides when no cuts have been added or the same cuts have been added to every problem. Two techniques have been proposed in the literature for solving linear programs with multiple right hand sides, sifting (Gartska and Rutenberg [10]) and bunching (Walkup and Wets [14]). After solving a subproblem with a particular right hand side, these methods identify other subproblems for which the current basis is optimal. The goal of these methods is to minimize the number of full simplex pivots which must be performed to solve all the subproblems in the current stage.

1.3. *Pereira and Pinto Method*

For multistage stochastic linear programs with relatively complete recourse and a modestly large number of N-stage scenarios, Pereira and Pinto [11] developed an algorithm which incorporates sampling into the general framework of the Nested Decomposition algorithm. The goal is to minimize the curse of dimensionality by eliminating a large portion of the scenario tree in the forward pass of the algorithm. The algorithm was successfully applied to multistage stochastic water resource problems in South America.

The multistage stochastic linear programs considered are assumed to have relatively complete recourse with finite optimal objective value. Assume that the stochastic elements are defined over a discrete probability space ($\Xi, \sigma(\Xi)$,P), where $\Xi = \Xi^2 \otimes \cdots \otimes \Xi^N$ is the support of the random data in stages two through N, with $\Xi^t = \{\xi_i^t = (h^t(\xi_i^t), c^t(\xi_i^t), T_{\cdot,1}^{t-1}(\xi_i^t), \ldots, T_{\cdot,n^{t-1}}^{t-1}(\xi_i^t),\ i = 1, \ldots, M^t)\}$. Further, assume that the random parameters are serially independent. Thus, the probability of a particular stage t realization ξ_i^t is constant from all possible $(t-1)$-stage scenarios.

The strategy of the Pereira and Pinto algorithm is to use sampling to generate an upper bound on the expected value (over an N-stage planning horizon) of a given first stage solution and to use decomposition to generate a lower bound. The algorithm terminates when the two bounds are sufficiently close.

As in the Nested Decomposition algorithm, each iteration of the Pereira and Pinto algorithm begins by solving the first stage subproblem. Then H N-stage scenarios are sampled. Let x_k^t and ξ_k^t denote the stage t solution vector and the stage t random parameter realization, respectively, in sampled scenario k. The forward pass through the sampled version of the scenario tree solves the following subproblem, for stages $t = 2, \ldots, N$ and scenarios $k = 1, \ldots, H$.

$$\begin{aligned} Q^t(x_k^{t-1}, \xi_k^t) &= \min c^t(\xi_k^t)x_k^t + \theta_k^t \\ s.t.\ W^t x_k^t &= h^t(\xi_k^t) - T^{t-1}(\xi_k^t)x_k^{t-1} \\ E_i^t x_k^t + \theta_k^t &\geq e_i^t \quad i = 1, \ldots, K^t, \\ x_k^t &\geq 0 \\ \theta_k^t\, &unrestricted. \end{aligned} \tag{3}$$

First note that since the problems under consideration have relatively complete recourse, feasibility cuts are not needed. The constraints (3) represent optimality cuts which are successively added during the course of the algorithm. These cuts represent lower bounds on the expected recourse function in stage t for all values of x^t. K^t denotes the number of optimality cuts that have been added to the stage t subproblem. In the first forward pass, there are no optimality cuts. Hence, $K^t = 0$, and θ_k^t is constrained to equal zero. Optimality cuts are never generated for the stage N subproblems, so θ^N is dropped from those subproblems. The first stage subproblem also has this formulation, although the problem has no stochastic elements and $T^0 = 0$.

The total objective values for all of the sampled scenarios are collected as follows to generate a confidence interval for an upper bound on the actual expected recourse function value. Let z_k denote the total objective value for scenario k; then,

$$z_k = c^1 x_k^1 + \sum_{t=2}^{N} c^t(\xi_k^t)x_k^t. \tag{4}$$

Note that x_k^1 is the same for all values of k. For the given first stage solution x_k^1, the expected recourse function

value $\mathcal{Q}^2(x_k^1)$ is a function of the random parameters in stages two through N. By assumption, the recourse cost for any N-stage scenario, given first stage solution x^1, is finite. Thus, the expected recourse cost, $\mathcal{Q}^2(x^1)$, and the variance of the recourse cost are finite. Also, the H sampled scenarios are independent observations of these random parameters. Hence, by the Central Limit Theorem (see, for example, [1]), for H large enough, a statistical estimate of the expected objective value of the first stage solution is given by:

$$\bar{z} = \frac{1}{H}\sum_{k=1}^{H} z_k. \tag{5}$$

The uncertainty of the estimate $\bar{z}$ is then measured by the standard deviation of the estimate,

$$\sigma_z = \sqrt{\left(\frac{1}{H^2}\sum_{k=1}^{H}(\bar{z} - z_k)^2\right)}. \tag{6}$$

Using these values, a confidence interval for the actual value of $\bar{z}$ can be constructed. For example,

$$[\bar{z} - 2\sigma_z, \bar{z} + 2\sigma_z] \tag{7}$$

represents a 95% confidence interval for $\bar{z}$. Note that $\bar{z}$ is a statistical estimate of the first stage costs and the expected recourse costs, given the current first stage solution x^1. Since the current first stage solution is feasible but not necessarily optimal, Condition 7 represents a confidence interval for an upper bound on the optimal objective value of the given stochastic program.

Once the forward pass has solved all N stages for all H sampled scenarios and assuming $K^t \geq 1$ for $t1, \ldots, N-1$, the stopping criterion is checked. From the discussion of the Nested Decomposition algorithm, we know that the current first stage objective value $c^1x_k^1 + \theta^1$ is a lower bound on the total expected cost over the duration of the planning horizon. Therefore, if the current first stage optimal objective value, $c^1x_k^1+\theta^1$, lies in the confidence interval of the upper bound on $\bar{z}$, the current solution is declared optimal, and the algorithm terminates; otherwise, the backward pass through the scenario tree begins.

The backward pass proceeds as in the Nested Decomposition algorithm. Starting in stage N with the current stage $N-1$ solution to scenario k, x_k^{N-1}, the subproblem $Q^N(x_k^{N-1}, \xi^N)$ is solved *for all possible stage N realizations* ξ^N. Let M^t denote the number of distinct realizations of the stage t random parameters, and let $\pi_{i,k}^t$ denote the optimal dual solution vector to the stage t subproblem $Q^t(x_k^{t-1}, \xi_i^t)$. As in the Nested Decomposition algorithm, an optimality cut on the stage $t-1$ expected recourse function is derived from these optimal dual values as follows, for each sampled scenario k,

$$\mathcal{Q}^t(x^{t-1}) \geq \sum_{i=1}^{M^t} prob(\xi_i^t)\pi_{i,k}^t \left(h^t(\xi_i^t) - T^{t-1}(\xi_i^t)x^{t-1} + e^t\right). \tag{8}$$

For $t = N$, the inequality follows by duality. For $t < N$, the inequality follows by duality and the inductive argument that the stage $t+1$ optimality cuts are lower bounds on the stage $t+1$ expected recourse function. To generate a cut of the form $Ex + \theta \geq e$, let

$$E_{K^{t-1}+1}^{t-1} = \sum_{i=1}^{M^t} prob(\xi_i^t)\pi_{i,k}^t \left(-T^{t-1}(\xi_i^t)\right),$$
$$e_{K^{t-1}+1}^{t-1} = \sum_{i=1}^{M^t} prob(\xi_i^t)\pi_{i,k}^t \left(h^t(\xi_i^t) + e^t\right).$$

Since the problems considered have serial independence, the expected recourse function in all stage t subproblems is identical. This allows all cuts generated for stage t (regardless of which scenario it was generated from) to be placed in all stage t subproblems. As each cut is added to the stage t subproblem, the value of K^t is increased by one.

Once a new optimality cut has been added to the first stage subproblem, the first iteration is completed and the forward pass begins again.

Finite convergence of this algorithm follows from the finite convergence of the Nested Decomposition algorithm, since the scenarios from which the optimality cuts are generated are resampled each iteration. Since the accuracy of the optimal solution depends on the accuracy of the estimated upper bound, the performance of the algorithm depends on the number of scenarios sampled in each iteration.

2. The Abridged Nested Decomposition Algorithm

The Pereira and Pinto algorithm does effectively resolve the curse of dimensionality, especially for narrow and long scenario trees. Pereira and Pinto considered 10-stage problems with only two possible realizations in stages two through ten, giving a total of 512 possible 10-stage scenarios. By sampling only 50 scenarios in

each iteration, the algorithm significantly reduced the effort needed to solve these problems.

The algorithm does not, however, seem well-designed for bushier trees, where the number of realizations in each stage is, say, twenty or more. In order to get a reliable estimate of the true population expected recourse function, the Central Limit Theorem generally requires that the number of scenarios sampled be at least thirty or more. While this presents little problem in the forward pass, the amount of work required in the backward pass to solve all realizations in each stage thirty or more times might be exhausting, especially for problems with four or more stages. Further, this fails to recognize that many of the scenarios may be giving similar solutions in stage t, making the need to resolve all subproblems in stages $t+1$ through N superfluous. Finally, the end result of all this work is a single optimality cut in the first stage subproblem. Since each iteration could be expensive, the need for several optimality cuts for the first stage to converge could make the algorithm more cumbersome than intended.

The new protocol proposed here, which we refer to as the Abridged Nested Decomposition algorithm, also involves sampling in the forward pass, but the forward pass does not proceed forward from all solutions of the realizations sampled in each stage. Instead, the stage t solutions from which to proceed are also sampled.

The scenario tree in Figure 1 highlights the new protocol. As in the Pereira and Pinto algorithm, the new protocol begins by solving the first stage subproblem, again with no optimality cuts initially. From the set of second stage realizations, F^2 realizations are then sampled and these subproblems are solved. The goal is to obtain a good sample of second stage solution values, without solving all realizations in the second stage. The darkened second stage nodes in the diagram correspond to the realizations sampled and solved. From the F^2 solution values, the algorithm proceeds forward from only B^2 ($\leq F^2$) values. The values, from which the algorithm branches forward, are referred to as branching values. A branching value may be a current stage t solution value or some combination of several current stage t solution values (more details in next section). Nodes 1 and 2 correspond to the second stage solutions from which forward branching occurs. The figure is drawn as shown to highlight the idea that branching may not occur from the same node or from any specific node in each iteration.

From each of the B^{t-1} stage $t-1$ branching solutions, F^t stage t realizations are again selected and

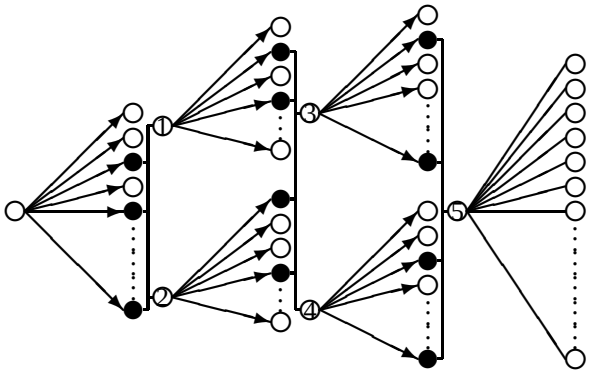

Fig. 1. Abridged Scenario Tree

solved. From the $(B^{t-1} * F^t)$ stage t solution values, the algorithm proceeds forward from only B^t values. Typically, the initial value of B^t will be relatively small (< 5) to allow a rapid forward pass.

Once the branching values for stage $N-1$ have been selected, the backward pass begins. For each branching solution in stage $N-1$, all possible realizations in stage N are solved. The optimal dual values are aggregated to generate an optimality cut on the stage $N-1$ expected recourse function, as in equation (8). Again, because of serial independence, all optimality cuts generated from the stage N subproblems are added to the stage $N-1$ subproblem. The process is repeated for all branching solutions in stages $N-2$ down to stage 1.

As the second stage subproblems are solved in the backward pass, let $(\tilde{x}_k^2, \tilde{\theta}_k^2)$ denote the optimal solution for second stage realization ξ_k^2 in Ξ^2. Let

$$\bar{\theta}^1 = \sum_{\xi_k^2 \in \Xi^2} prob(\xi_k^2)(c^2(\xi_k^2)\tilde{x}_k^2 + \tilde{\theta}_k^2).$$

Since $c^2(\xi_k^2)\tilde{x}_k^2$ is the second stage cost for solution $\tilde{x}_k^2$ and $\tilde{\theta}_k^2$ is a lower bound on the second stage recourse cost for solution $\tilde{x}_k^2$, $\bar{\theta}^1$ represents a lower bound on the expected recourse function value for the current first stage solution, $\tilde{x}^1$. Recall that the current first stage approximation of the expected recourse function value at $\tilde{x}^1$, $\tilde{\theta}^1$, is also a lower bound on the expected recourse function at $\tilde{x}^1$. The value of $\bar{\theta}^1$ may differ from the value of $\tilde{\theta}^1$, however, since $\bar{\theta}^1$ includes the additional information that has been gained in the current iteration. This additional information also ensures that $\bar{\theta}^1$ is always greater than or equal to $\tilde{\theta}^1$.

The Nested Decomposition algorithm terminates when $\bar{\theta}^1 = \tilde{\theta}^1$, since this implies that in each subproblem, the current expected recourse approximation value, $\tilde{\theta}_k^t$, is exact at a current solution value $\tilde{x}_k^t$. Unfortunately, since the Abridged Nested Decomposition algorithm does not consider all scenarios in each iteration, the same claim does not hold. The solution of a particular second stage subproblem may not change in the forward and backward pass of an iteration simply because this solution is not selected as a branching value, and so, the opportunity to evaluate the second stage expected recourse function value at that solution is not given. However, if that solution had been selected as a branching solution, as it always would be in the Nested Decomposition algorithm, an optimality cut might have been generated which would change that subproblem solution.

Although it cannot be used as a termination criterion, the relative closeness of $\bar{\theta}^1$ and $\tilde{\theta}^1$ can be used as an indication that the solution is converging. Hence, given that $\tilde{\theta}^1$ is within a relative tolerance of $\bar{\theta}^1$, a termination test is given which employs sampling to generate a statistical estimate of an upper bound.

An N-stage scenario $(\xi_1^2, \ldots, \xi_1^N)$ is randomly selected. The current first stage subproblem is solved. Let the solution be x_1^1. Then, starting with $t = 2$, the current stage t subproblem is solved, given sampled stage t realization ξ_1^t and stage $t-1$ solution x_1^{t-1}. Let the solution be denoted x_1^t. This is repeated for $t = 3, \ldots, N$. The entire process is repeated for H different N-stage scenarios.

The Central Limit Theorem can be invoked again to establish a statistical estimate of an upper bound on the optimal objective value. As in Equations (4), (5), and (6), the total value of each scenario is recorded, the total values are averaged, and a confidence interval around the average value is established by calculating the standard deviation. If the current first stage objective value, $c^1\tilde{x}^1 + \tilde{\theta}^1$, falls within that confidence interval, the algorithm terminates; otherwise, the number of nodes solved in each stage in each forward pass, F^t, and the number of nodes from which we branch in each stage, B^t, can be increased, and a new forward pass begins.

The increase in both F^t and B^t after each failed termination test helps to achieve convergence as the increase in these values increases the amount of information being brought back to the first stage subproblem. Eventually, the entire scenario tree could be considered in each forward and backward pass, at which point the algorithm would be identical to the Nested Decomposition algorithm. The hope of the algorithm, though, is that significant tree expansion will not be needed, thereby allowing much faster iterations than those of the Nested Decomposition algorithm, while passing back enough valuable information about the expected recourse function that the number of additional iterations needed will not be significant.

2.1. *Branching Selections*

In order for this algorithm to be effective, the solution values from which to branch in each stage must be selected carefully. The following theorem shows that valid branching values exist which are not necessarily current stage t solution values.

Theorem 1 *Consider a multistage stochastic linear program with relatively complete recourse. Let $\tilde{x}_i^t$ be any feasible solution to the stage t subproblem with realization $\xi_i^t \in \Xi^t$, $1 \le i \le M^t$. Further, let*

$$\tilde{x} = \sum_{i=1}^{M^t} w_i^t \tilde{x}_i^t \quad \textit{where} \quad \sum_{i=1}^{M^t} w_i^t = 1,\ 0 \le w_i^t \le 1 \ (i = 1, \ldots, M^t);$$

then there exists a feasible completion in stages $t+1, \ldots, N$ from $\tilde{x}$.

Proof. By contradiction. Suppose that there exists a convex combination of the feasible solutions such that $\mathcal{Q}^{t+1}(\tilde{x}) = \infty$ (i.e., there does not exist a feasible completion from $\tilde{x}$).

We know that $\mathcal{Q}^{t+1}(x)$ is a convex, piecewise linear function of x. Thus,

$$\sum_{i=1}^{M^t} w_i^t \mathcal{Q}^{t+1}(\tilde{x}_i^t) \ge \mathcal{Q}^{t+1}\Big(\sum_{i=1}^{M^t} w_i^t \tilde{x}_i^t\Big),$$

by Jensen's Inequality, convexity,

$$= \mathcal{Q}^{t+1}(\tilde{x}), \quad \text{by definition,}$$

$$= \infty, \quad \text{by assumption.}$$

This implies that $\mathcal{Q}^{t+1}(\tilde{x}_i^t) = \infty$ for at least one i value, which contradicts the assumption of relatively complete recourse. ■

Thus, any convex combination of the current stage t solution values can be chosen as a possible branching value, including the expected value of the current solution values. This implies that the $B^{t-1} * F^t$ possible solution values in stage t can be gathered into several groups, and the expected value of the solution values

Step 0: For $t = 1, \ldots, N-1$, set $K^t = 0$, and add the constraint $\theta^t = 0$ to the stage t subproblem. Choose initial values for F^t and B^t for $t = 2, \ldots, N-1$. Go to Step 1.

Step 1: Solve the first stage problem. Let $\tilde{x}^1$ be the current optimal solution and $\tilde{\theta}^1$ be the current expected recourse approximation value. Let $\tilde{z}^1$ be the current optimal objective value. Let $\tilde{x}^1$ be the first stage branching value. Go to Step 2.

Step 2: **FORWARD PASS.**
For $t = 2, \ldots, N-1$,
For $m = 1, \ldots, B^{t-1}$,
For $k = 1, \ldots, F^t$,
Solve stage t subproblem, given sampled realization ξ_k^t and the m^{th} stage $t-1$ branching value.
Select B^t branching values.
Go to Step 3.

Step 3: **BACKWARD PASS.**
For $t = N, \ldots, 2$,
For $m = 1, \ldots, B^{t-1}$,
For $i = 1, \ldots, M^t$,
Solve stage t subproblem, given realization ξ_i^t and the m^{th} stage $t-1$ branching value. Let $(\pi_{i,m}^t, \sigma_{i,m}^t)$ denote the optimal dual vector values.
Compute

$$E^{t-1} = \sum_{i=1}^{M^t} p_k^t \pi_{i,m}^t T^{t-1}(\xi_i^t),\ e^{t-1} = \sum_{i=1}^{M^t} p_k^t \left(\pi_{i,m}^t h^t(\xi_i^t) + \sigma_{i,m}^t e_i^t\right)$$

The new cut is then: $E^{t-1}x^{t-1} + \theta^{t-1} \geq e^{t-1}$.
If the constraint $\theta^{t-1} = 0$ appears in the stage $t-1$ subproblem, then remove it. Increment K^{t-1} by one and add the new cut to the stage $t-1$ subproblem. If $t = 2$, then the updated first stage expected recourse function upper bound is: $\bar{\theta}^1 = e^1 - E^1\tilde{x}^1$. If $\tilde{\theta}^1$ is within a relative tolerance of $\bar{\theta}^1$, then go to Step 4. Otherwise, go to Step 1.

Step 4: **SAMPLING STEP**
Let $x_k^1 = \tilde{x}^1$, for $k = 1, \ldots, H$.
For $k = 1, \ldots, H$,
Generate N-stage sample scenario, $(\xi_k^2, \ldots, \xi_k^N)$.
For $t = 2, \ldots, N$,
Given stage $t-1$ solution x_k^{t-1} and realization ξ_k^t, solve the stage t subproblem. Let x_k^t denote the optimal solution.
Using Equations (4), (5), and (6), obtain a confidence interval on the expected objective value of the current first stage solution. If $c^1\tilde{x}^1 + \tilde{\theta}^1$ is in the confidence interval, stop with $\tilde{x}^1$ as the optimal solution. Else, increase F^t and B^t for stage $t = 2, \ldots, N$ and go to Step 1.

Fig. 2. Abridged Nested Decomposition Algorithm for Relatively Complete Programs

within each group can be used as that group's branching value. This method for selecting branching values could prove effective, since the branching values would then represent the values which stage t can expect from those stage $t-1$ subproblems, rather than just one possible solution value.

Other branching values might include current solution values whose distance from the current average solution value is greatest. Using these values may prove effective, as this helps to generate optimality cuts which restrict solution values from outlying solution values. This may be important to achieve convergence.

Finally, choosing current solution values randomly to be branching values may also be effective. Unlike the other selection techniques, this strategy gives an unbiased sample of stage t solution values.

3. Implementation & Results

In this section, the results of a computational study are reported. The computational efficiency of the Pereira

and Pinto algorithm is compared to that of the new Abridged Nested Decomposition algorithm.

3.1. *Implementation Description*

The code ND.PP follows the algorithm developed by Pereira and Pinto. The code ND.Abridged follows the new Abridged Nested Decomposition algorithm discussed in Section 2.. Both ND.PP and ND.Abridged were written in C. Both work interactively with CPLEX's callable library for mathematical programming and were run on Sun SPARC 20 workstations.

For ND.PP, the sample size (H) is thirty for each problem. For ND.Abridged, the number of stage t subproblems solved in the forward pass from each stage $t-1$ branching value (F^t) is set initially between ten and fifteen. The number of stage t branching values selected (B^t) is set initially to one. The upper bound estimate is calculated whenever the current first stage expected recourse approximation value $\tilde{\theta}^1$ is within a relative tolerance of 10^{-3} of $\bar{\theta}^1$. After the upper bound is calculated, if the current first stage optimal objective value, $c^1\tilde{x}^1 + \tilde{\theta}^1$, fails to be within one standard deviation of the statistically estimated upper bound, $\bar{z}$, then for $t = 2, \ldots, N-1$, the value of B^t is increased by one (if possible). If B^t is now larger than F^t, F^t is also increased by one (if possible).

While B^t equals one, the average solution value of the current stage t solution values is used as the branching value. For $B^t > 1$, the set of $B^{t-1} * F^t$ current stage t solution values are partitioned into $\lceil \frac{B^t}{3} \rceil$ groups. The average solution value within each group is used as one branching value. The current solution value within each group most distant from the group's average value is used as another branching value. The remaining $B^t - 2 * \lceil \frac{B^t}{3} \rceil$ branching values are chosen randomly from all current stage t solution values, excluding those already chosen.

3.2. *Test Problem Set Description*

The Dynamic Vehicle Allocation (DVA) Problem with Uncertain Demand represents the situation which arises when a carrier must manage a fleet of vehicles in an environment of uncertain future demand while maximizing expected profits over a given planning horizon. In each time period, the carrier receives requests to have loads moved between various pairs of sites. The carrier can accept or decline each request. Since each request is to have a load moved between a specific origin and a specific destination, the effective demand is, therefore, along arcs instead of at nodes as in a transportation problem. In the vehicle allocation literature, this is represented as a capacity for loaded movement. In addition, the carrier can dispatch empty vehicles between two sites in anticipation of future requests out of the destination site. The carrier knows shipping requests that exist in the current time period, but is uncertain about the shipping demands in future periods; the carrier has information about the distribution of possible demand scenarios, perhaps based upon past demand realizations. Much of the work done on the DVA problem has been initiated and developed by Powell (see [12] for a review of the problem and methods).

All of the problems tested are DVA problems of various sizes. All of the random demands are assumed to be independent, so serial independence of the random parameters is given. The probability distributions on demand between sites were derived using historical data from a national transportation company. Further, note that the carrier is not committed to take any of the loads; thus, the option of leaving all of the vehicles stationary for the duration of the planning horizon is a feasible option which has finite cost. Hence, the problem has relatively complete recourse and serially independent random parameters, so the Abridged Nested Decomposition algorithm can be used to solve these problems.

The naming convention used for all problems is DVA.$x.y.z$, where x denotes the number of sites, y denotes the number of stages, and z denotes the number of distinct realizations per stage. The DVA.8.$y.z$, DVA.12.$y.z$ and DVA.16.$y.z$ problems have 16, 24, and 32 nodes connected by 72, 168 and 244 arcs, respectively, in each stage. The fleet sizes are 50, 120 and 140, respectively.

3.3. *Results*

The results of the comparison between the two algorithms are given in Table 2. All times are given in seconds. Test runs lasting longer than 40,000 seconds ($>$ 10 hours) were terminated with the objective value in the last iteration reported.

The Abridged Nested Decomposition algorithm (ND.Abridged) significantly outperformed the Pereira and Pinto algorithm (ND.PP) on all problems tested. For problems where a comparison can be made, the Abridged Nested Decomposition runtimes are, on average, twelve times faster than the Pereira and Pinto runtimes. Furthermore, as the size of the problems in-

Table 2

Problem	ND.Abridged Time	ND.PP Time	ND.Abridged Obj. Value	ND.PP Obj. Value
DVA.8.4.30	46.2	121.0	-12192.81	-12298.14
DVA.8.4.45	32.0	338.9	-12395.10	-12363.43
DVA.8.4.60	89.8	485.6	-12255.98	-12251.13
DVA.8.4.75	72.0	1486.5	-12243.91	-12150.55
DVA.8.5.30	79.2	648.0	-12845.88	-12788.99
DVA.8.5.45	164.0	1480.1	-13027.60	-12961.76
DVA.8.5.60	99.3	2032.7	-12925.55	-12855.18
DVA.8.5.75	139.8	1754.3	-12938.41	-12848.27
DVA.12.4.30	131.2	755.0	-32826.07	-32855.58
DVA.12.4.45	141.7	2259.6	-32771.80	-32741.75
DVA.12.4.60	341.0	5309.9	-32773.34	-32766.86
DVA.12.4.75	258.4	5059.6	-32845.11	-32819.66
DVA.12.5.30	560.4	4139.7	-39389.62	-39384.02
DVA.12.5.45	672.7	6342.1	-39375.73	-39447.76
DVA.12.5.60	539.1	7001.2	-39435.64	-39549.73
DVA.12.5.75	1561.8	24502.1	-39499.48	-39479.29
DVA.16.4.30	1049.6	8353.2	-21663.45	-21679.82
DVA.16.4.45	1209.1	$>$ 40000	-21792.31	-21775.15
DVA.16.4.60	3739.1	$>$ 40000	-21839.29	-21981.07
DVA.16.4.75	3753.7	$>$ 40000	-21817.53	-21862.53
DVA.16.5.30	600.6	9712.4	-22452.53	-22557.43
DVA.16.5.45	1658.4	$>$ 40000	-22552.15	-22515.51
DVA.16.5.60	3576.1	$>$ 40000	-22603.35	-22798.30
DVA.16.5.75	3504.0	$>$ 40000	-22576.36	-22705.12

CPU Time Comparison of Pereira and Pinto Algorithm and Abridged Nested Decomposition Algorithm

crease, the rate of increase in runtimes for Pereira and Pinto algorithm is noticeably steeper than that of the Abridged Nested Decomposition algorithm.

4. Conclusion

We have presented a method for solving multi-stage stochastic programs that incorporates sampling into nested decomposition. The resulting algorithm has advantages, as seen in the computational results, over previous approaches in reducing the size of the tree required to generate new value-function bounds. The convergence results require full subproblem solutions at each stage to ensure valid lower bounds and serial independence to ensure that the value function only depends on the current state and not prior history, but each of these assumptions may be relaxed in various ways.

The complete subproblem solution requirement for the lower bound may be relaxed to use a sample, but the sample needs to be chosen consistently with a corresponding convergence result that is somewhat different. The analysis above still applies if the upper bound is sampled from the sample distribution that generates the lower bound. In this case, the convergence is then with given confidence for the lower-bound sample distribution. As that lower bound sample increases, the overall result then approaches an optimal value.

The serial independence assumption can be relaxed in some cases by re-formulation (e.g., in a portfolio optimization problem, by replacing prices that depend on previous year's values with returns that are serially independent). In other cases, the recourse or value function $\mathcal{Q}^t$ may be written as a function of the state x^t and a set of parameters v^t that determine the future probability distributions. In those cases, the Abridged Nested Decomposition algorithm can include separate approximations for different values of v^t. For low-dimensional v^t, this approach may remain computationally efficient.

References

[1] Billinsley, P. *Probability and Measure.* New York: John Wiley & Sons, 1995.

[2] Birge, J. R. . "Decomposition and Partitioning Methods for Multi-Stage Stochastic Linear Programs." *Operations*

Research **33** (1985), 989–1007.

[3] Birge, J. R., C. J. Donohue, D. F. Holmes, and O. G. Svintsiski. "A Parallel Implementation of the Nested Decomposition Algorithm for Multistage Stochastic Linear Programs." *Mathematical Programming* **75** (1996) 327–352.

[4] Birge, J. R., and F. Louveaux. "A Multicut Algorithm for Two-Stage Stochastic Linear Programs." *European Journal of Operations Research* **34** (1988), 384–392.

[5] Birge, J. R., and R. J-B. Wets. "Designing Approximation Schemes for Stochastic Optimization Problems, in Particular Stochastic Programs with Recourse." *Mathematical Programming* **27** (1986), 54–102.

[6] Birge, J. R., and R. J-B. Wets. "Sublinear Upper Bounds for Stochastic Programs with Recourse." *Mathematical Programming* **43** (1989), 131–149.

[7] Dantzig, G. B., "Linear programming under uncertainty," *Management Science* **1** (1955), 197–206.

[8] Donohue, C. J., J. R. Birge, and D. F. Holmes. "Implementation of the Nested Decomposition Algorithm for Multistage Stochastic Linear Programs." VII International Conference on Stochastic Programming, Nahariya, Israel. 28 June 1995.

Received 12 November 2005; accepted 15 December 2005

[9] Gassmann, H. I.. "MSLiP: A Computer Code for the Multistage Stochastic Linear Programming Problem." *Mathematical Programming* **47** (1990), 407–423.

[10] Gartska, S., and D. Rutenberg. "Computation in Discrete Stochastic Programs with Recourse." *Operations Research* **21** (1973), 112–122.

[11] Pereira, M. V. F., and L. M. V. G. Pinto. "Multistage Stochastic Optimization Applied to Energy Planning." *Mathematical Programming* **52** (1991), 359–375.

[12] Powell, W. B. "A Comparative Review of Alternative Algorithms for the Dynamic Vehicle Allocation Problem." In *Vehicle Routing: Methods and Studies.* Ed. B. Golden, A. Assad. North-Holland, 1988, pp. 249–291.

[13] Van Slyke, R., and R. J-B. Wets. "L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming." *SIAM Journal on Applied Mathematics* **17** (1969), 638–663.

[14] Wets, R. J-B. "Large Scale Linear Programming Techniques." In *Large scale linear programming techniques.* Ed. Y. Ermoliev and R. J-B. Wets. Berlin: Springer- Verlag, 1988.

[15] Wittrock, R. "Dual Nested Decomposition of Staircase Linear Programs." *Mathematical Programming Study* **24** (1985), 65–86.