CHAPTER III

A Nested Decomposition Algorithm for SDLP

1. Introduction

Stochastic dynamic linear programs have natural subdivisions corresponding to the separate decisions made in each period under various scenarios. These divisions give SDLP a special structure that can lead to improved efficiency in its solution. In the next three chapters, we present three methods employing distinct optimization techniques, each of which exploits the program's structure. The structure is essential for our development. It is the basis of each technique: decomposition, partitioning, and basis factorization.

These methods have been used extensively in large-scale structured programming, but their application in solving stochastic programs has been limited. In our presentation, we demonstrate how these techniques can be applied to the stochastic version of the multi-stage linear program. We will then show that our algorithms may yield substantial savings over straightforward, "brute-force" techniques by using the program structure effectively in reducing the computational cost.

The first method we present is called a *nested decomposition algorithm* *for SDLP* or *NDSDLP*, because it decomposes a large problem into successively finer subproblems.

The basic principles of optimization that we employ are *outer linearization* and *inner linearization*, as described in Geoffrion [24]. Through outer linearization we optimize over a larger convex region than is feasible and then, by increasing the restrictions obtained from the subproblems, approach optimality within the true feasible region. To perform inner linearization one optimizes over successive subregions of the feasible region and approaches thereby a global optimum.

The method below applies outer linearization to the primal problem of SDLP. It can also be viewed as applying inner linearization to the dual problem. We then have a master subproblem relationship in the primal as in Benders' method [9] or in the dual as in Dantzig-Wolfe decomposition [19]. We also follow a procedure of passing between periods that is similar to the nested decomposition of primal inner linearization as in Glassey [29] and Ho and Manne [32]. The relationship between the two methods is well–known as Kallio and Porteus showed in [38].

We begin our presentation of the nested decomposition algorithm in Section 2, by examining some properties of objective functions and showing the basic master– subproblem relationship for SDLP under the outer linearization scheme. Section 3 presents the analogous development for inner linearization, or Dantzig-Wolfe decomposition. We follow this in Section 4 with a description of a fundamental problem inherent in SDLP, degeneracy. We present suggestions for its resolution and some of its special difficulties in the stochastic framework. Lastly, in Section 5, we present the full algorithm and its strategy in passing through the scenarios.

2. The Master-Subproblem Relationship

The decomposition algorithm we discuss below relies on certain fundamental properties of the multi-stage program under uncertainty. These properties concern the convexity of the objective function and the representation of the solution set as a convex polyhedron. Wets first observed these attributes in [61] and [62]. These are reviewed below.

We begin by formulating the equivalent convex program to SDLP. We then use this formulation to examine how we can find the set of linear constraints that constitute the solution set. This procedure involves inducing feasibility in the subproblems and finding the conditions for optimality. The section concludes with an explanation of how the cutting planes for these operations are constructed.

Every stochastic dynamic linear program (with continuous or discrete distribution of the random variables) is equivalent to a convex program with linear constraints, as we state in the following theorem of Wets:

Theorem 1 *The stochastic dynamic linear program SDLP, as defined in Chapter 1, is equivalent to a program of the following form:*

$$\begin{aligned} &\min c_1 x_1 + Q_1(x_1) \\ &\textit{subject to} \\ &A_1 x_1 = \xi_1 \quad , \\ &x_1 \in D_1, \\ &x_1 \geq 0, \end{aligned} \tag{ECP}$$

where $Q_1(x_1)$ is a convex function and D_1 is a convex polyhedron.

Proof. See Wets [62]. ∎

The $Q_1(x_1)$ of Theorem 1 is defined as

$$Q_1(x_1) = E_{\xi 2}[Q_1(x_2, \xi_2)] \tag{1}$$

where

$$Q_1(x_2, \xi_2) = \{\min[c_2 x_2 + Q_2(x_2)] | A_2 x_2 = \xi_2 + B_1 x_1, x_2 \in D_2, x_2 \geq 0\}. \tag{2}$$

The subsequent $Q_t(x)$ are defined iteratively. We can solve SDLP by repeated solutions of these convex programs, but the objective function and solution set may be hard to find. We present below methods for finding $Q_t(x_t)$ and D_t without explicitly determining the functions.

ECP above is called the *equivalent convex program* of the stochastic program. We note that this theorem holds when ξ has a continuous or discrete distribution.

Before we proceed with this development, we further note that, since every period of SDLP corresponds to an identically formulated optimization problem, we can form a subproblem in every period, t, and for every scenario, j, that is similar to ECP. Let $\hat{j}$ be the immediate ancestor of j, and $\varsigma_t^j = B_{t-1} x_{t-1}^{\hat{j}} + \xi_t^j$, we have

$$z_t^j = \min c_t x_t^j + Q_t(x_t^j)$$

subject to

$$\begin{aligned} A_t x_t^j &= \varsigma_t^j, \\ D_t^j x_t^j &\geq \lambda_t^j. \\ x_t^j &\geq 0, \end{aligned} \tag{$ECP(t,j)$}$$

where we have written the convex polyhedron, D, as $D_t^j x_t^j \geq \lambda_t^j$, and $Q_t(x_t^j)$ is defined as in (1).

ECP(t,j) is used in the subsequent analysis to develop the master subproblem relationships that are encountered in the nested decomposition algorithm. We first want to find a method for constructing D_t^j so that x_t^j will be feasible for its descendant scenarios. Let $\bar{j}$ be a descendant of j in period $t+1$ for a given solution $x_t^{j,0}$ of ECP(t,j). We have

$$\min z_{t+1}^{\bar{j}} = c_{t+1}x_{t+1}^{\bar{j}} + Q(x_{t+1}^{\bar{j}})$$

subject to

$$\begin{aligned} A_{t+1}x_{t+1}^{\bar{j}} &= \xi_{t+1}^{\bar{j}} + B_t x_t^{j,0}, \\ D_{t+1}^{\bar{j}} x_{t+1}^{\bar{j}} &\geq \lambda_{t+1}^{\bar{j}}. \\ x_{t+1}^{\bar{j}} &\geq 0, \end{aligned} \tag{$ECP(T+1,\bar{j})$}$$

Now, if ECP$(t+1,\bar{j})$ has no feasible solution, then by Farkas's lemma, there exists a vector $\left[{}_1\sigma_{t+1}^{\bar{j}}, {}_2\sigma_{t+1}^{\bar{j}}\right]$ such that

$${}_1\sigma_{t+1}^{\bar{j}} A_{t+1} +{}_2 \sigma_{t+1}^{j} D_{t+1}^{\bar{j}} \leq 0, \tag{3}$$

$$(-{}_2\sigma_{t+1}^{j}) \leq 0, \tag{4}$$

and

$${}_1\sigma_{t+1}^{\bar{j}}(\xi_{t+1}^{\bar{j}} + B_t x_t^{j,0}) +{}_2 \sigma_{t+1}^{\bar{j}} \lambda_{t+1}^{\bar{j}} > 0. \tag{5}$$

So, in order for ECP$(t+1,\bar{j})$ to have a feasible solution x_t^j must be chosen such that

$$-({}_1\sigma_{t+1}^{j} B_t)x_t^j \geq +{}_1\sigma_{t+1}^{\bar{j}} \xi_{t+1}^{\bar{j}} +{}_2 \sigma_{t+1}^{\bar{j}} \lambda_{t+1}^{\bar{j}}. \tag{6}$$

This implies the following lemma:

Lemma 1. *For every descendant scenario of j, $\bar{j}$, in period $t+1$ and for all ${}_1\sigma_{t+1}^{\bar{j}}$ and ${}_2\sigma_{t+1}^{\bar{j}}$ satisfying (3) and (4) if x_t^j is feasible in SDLP, then it satisfies the inequality (6).*

Using this result, we can add (6) as an additional constraint to ECP(t, j). We repeat this for each descendant $\bar{j}$ of j. We solve ECP(t, j) again after all of these *cuts* have been added and proceed downward again with a new value, $x_t^{j,1}$. In this manner, we iteratively construct the constraint set for x_t^j. From now on, for clarity of exposition, we will write the equations of the form (6)as

$$-(\sigma_{t+1}^{\bar{j},k} B_t)x_t^j \geq \sigma_{t+1}^{\bar{j},k}(\hat{\xi}_{t+1}^{\bar{j}}), \tag{7}$$

where $k = 1, \ldots, p$.

We know how to find the constraint set of ECP(t, j) and we must now find a method of constructing the convex function, $Q(x_t^j)$. To do this, we first observe that ECP(t, j) is equivalent to

$$\min z_t^j = c_t x_t^j + \theta_t^j \tag{8}$$

subject to

$$\begin{aligned} A_t x_t^j &= \xi_t^j, \\ -(\sigma_{t+1}^{\bar{j},k} B_t)x_t^j &\geq \sigma_{t+1}^{\bar{j},k}(\hat{\xi}_{t+1}^{\bar{j}}); k = 1, \ldots, p, \\ Q(x_t^j) &\geq \theta_t^j, \\ x_t^j &\geq 0. \end{aligned}$$

With this formulation, we consider that $x_t^{j,0}$ is again a solution to ECP(t, j) and that ECP$(t+1, \bar{j})$ is feasible for all $\bar{j}$. Next let

$$\pi(x_t^{j,0}), \bar{j} = (\pi_1(x_t^{j,0}, \bar{j}); \pi_2(x_t^{j,0}, \bar{j})) \tag{9}$$

be the optimal dual prices vector for each ECP$(t+1, \bar{j})$, given $x_t^{j,0}$ a solution to ECP(t, j), and define

$$\begin{aligned} \pi_1(x_t^{j,0}) &= \mathrm{E}_{\xi_{t+1}}[\pi_1(x_t^{j,0}, \bar{j})], \\ \rho_1(x_t^{j,0}) &= \mathrm{E}_{\xi_{t+1}}[\pi_1(x_t^{j,0}, \bar{j}) \cdot (\xi_{t+1}^{\bar{j}})] \end{aligned} \tag{10}$$

and

$$\rho_2(x_t^{j,0}) = \mathrm{E}_{\xi_{t+1}}[\sum_{k=1}^{p} \pi_2^k(x_t^{j,0}, \bar{j}) \cdot (\sigma_{t+1}^{\bar{j},k}(\hat{\xi}_{t+2}^{\bar{j}})]. \tag{11}$$

Now, for $\pi(x_{t+1}^j, \bar{j})$ optimal in ECP$(t+1, \bar{j})$ for any x_{t+1}^j, the following inequality must hold

$$\pi_1(x_t^j, \bar{j})(\xi_{t+1}^{\bar{j}} + B_t x_t^j) + \sum_{k=1}^{p} \pi_2^k(x_t^j, \bar{j}) \sigma_{t+2}^{\bar{j},k}(\tilde{\xi}_{t;2}^{\bar{j}}) \geq$$

$$\pi_1(x_t^{j,0}, \bar{j})(\xi_{t+1}^{\bar{j}}; B_t x_t^j) + \sum_{k=1}^{p} \pi_2^k(x_t^j, \bar{j}) \sigma_{t+2}^{j,k}(\xi_{t+2}^{\bar{j}}, \tag{12}$$

and, since $\pi(x_t^j, \bar{j})$ is optimal, we have

$$Q(x_t^j, \bar{j}) = \pi_1(x_t^j, \bar{j})(\xi_{t+1}^{\bar{j}}) + B_t x_t^j) + \sum_{k=1}^{p} \pi_2^k(x_t^j, \bar{j}) \cdot \sigma_{t+1}^{\bar{j},k}(\hat{\xi}_{t+1}^{\bar{j}}). \tag{13}$$

Therefore, by taking expectations,

$$Q(x_t^j) \geq \rho_1(x_t^{j,0}) + \rho_2(x_t^{j,0}) + \pi_1(x_t^{j,0}) \cdot B_t x_t^j. \tag{14}$$

Letting $\rho(x_t^{j,0}) = \rho_1(x_t^{j,0}) + \rho_2(x_t^{j,0})$ in (14), we have the following lemma.

Lemma 2. *If (x_t^j, θ_t^j) is a feasible solution to ECP(t, j) written as in (8), then $\theta_t^j \geq \rho(x_t^{j,0}) + (\pi_1(x_t^{j,0}) B_t) x_t^j$.*

This lemma enables us to form additional constraints, as

$$(\pi_1^\ell \cdot B_t) x_t^j + \theta_t^j \geq \rho^\ell \tag{15}$$

for successive l, where $\pi_1^l = \pi_1(x_t^{j,l})$ and $\rho^l = \rho(x_t^{j,l})$. These cuts are then also added to ECP(t, j) whenever we find that a solution $(x_t^{j,l}, \theta_t^{j,l})$ is such that

$$\theta_t^{j,\ell} < \rho^\ell + \pi_1^\ell B_t x_t^{j,\ell}. \tag{16}$$

When (15) is solved for all π_1^l, and ρ^l, then we have achieved master-subproblem optimality between ECP(t, j) and its descendants, ECP$(t, \bar{j})$.

We have shown how the master problems and subproblems can be constructed in SDLP by using the fundamental results in Lemma 1 and Lemma 2. We will present below the basic algorithm for finding master-suboptimality. This algorithm follows closely from Benders [9] and has been presented in the two-stage case by Van Slyke and Wets [56]. It is an outer linearization scheme because the feasible regions D_t^j and convex functions Q_t^j are successively approximated by the inequalities in (7) and (15). We call this procedure OLSDLP, for outer linearization of SDLP. This exposition includes Step 2′, the case of an unbounded solution in ECP(t, j). The justification for this procedure can be found in Van Slyke and Wets [56] for a deterministic problem. We omit details here, because, in general, we will use the algorithm with upper bounded variables and no unbounded solution will be possible.

OLSDLP

Step 1. Solve the current form of ECP(t, j), using Phase I and Phase II of the simplex method:

$$\min c_t x_t^j + \theta_t^j$$

$$\text{subject to}$$

$$\begin{aligned} A_t x_t^j &= \xi_t^j, \\ -(\sigma_{t+1}^{\bar{j},k} B_t) x_t^j &\geq \sigma_{t+1}^{\bar{j},k}(\hat{\xi}_{t+1}^{\bar{j}}), k = 1, \ldots, p, \\ -(\pi_1^\ell \cdot B_t) x_t^j + \theta_t^j &\geq \rho^\ell, \ell = 1, \ldots, q, \\ x_t^j &\geq 0, \end{aligned} \tag{17}$$

where we set $p = q = 0$ initially and let $\theta_t^j = -\infty$, if $q = 0$. If (17) is infeasible, stop. If (17) is feasible and unbounded, go to Step 2′. If (17) is feasible and bounded, go to Step 2.

Step 2. For $(x_t^{j,i}, \theta_t^{j,i})$ a solution of (17), solve the Phase 1 problem of ECP$(t+1, \bar{j})$:

$$\min \omega^{\bar{j}} = ev + eu^+$$

subject to

$$\begin{aligned} A_{t+1} x_{t+1}^{\bar{j}} + Iv &= \xi_{t+1}^{\bar{j}} + B_t x_t^{j,i}, \\ D_{t+1}^{\bar{j}} x_{t+1}^{\bar{j}} + Iu^+ - Iu^- &= \lambda_{t+1}^{\bar{j}} \\ x_{t+1}^{\bar{j}}, v, u^+, u^- &\geq 0. \end{aligned} \tag{18}$$

For each $\bar{j}$, such that $\omega^{\bar{j}} > 0$ in (18), use the resultant multipliers to build a cut of the form in (7). Add these cuts to (17) and increment p. If $\omega^{\bar{j}} > 0$ for any $\bar{j}$, go to Step 1; otherwise, go to Step 3.

Step 2′. From (17), we obtain an unbounded ray, $x_t^{j,l^o} + \lambda y_t^{j,l^o}$, for $\lambda \geq 0$. Now, solve (18) for each $\bar{j}$, but replace $\xi_{t+a}^{\bar{j}} + B_t x_t^{j,l}$ by $B_t y_t^{\bar{j},l^o}$. If $\omega^{\bar{j}} > 0$, for any $\bar{j}$, add cuts as in (7) and return to Step 1. If $\omega^{\bar{j}} = 0$ for all $\bar{j}$, solve ECP$(t+1, \bar{j})$ for all $\bar{j}$ with the same replacement. Let $\bar{z}^l(t+1, \bar{j})$ be the expected value of the objective functions and compute π^l and ρ^l.

Next, solve (18) with $x_t^{j,l}$ for each $\bar{j}$. If $\omega^{\bar{j}} > 0$ for any $\bar{j}$, add the feasibility *cuts* and return to Step 1. If $\omega^{\bar{j}} = 0$ for all $\bar{j}$, then we check if $c_t y_t^{j,l} + \bar{z}^l(t+1, \bar{j}) < 0$. If so, the objective function is unbounded, stop. If $c_t y_t^{j,l} + \bar{z}^l(t+1, \bar{j}) \geq 0$, then we must eliminate $y_t^{\bar{j},l}$ as a feasible direction. We do this by using the π^l and ρ^l found above in forming a constraint of the form (15) and adding it to (17). In this case, we return to Step 1.

Step 3. Solve ECP$(t+1,\bar{j})$ for each $\bar{j}$. Compute $\bar{z}_{t+1}^{\bar{j},i} = E_{\xi_{t+1}}\left[z_{t+1}^{\bar{j}}(x_{t+1}^{j})\right]$, π_1^i, and ρ^i. If $\bar{z}_{t+1}^{\bar{j},i} \leq \theta^i$, stop. ECP$(t, j)$ is solved. Otherwise, generate a cut of the form (15) add it to (17), and return to Step 1.

This algorithm terminates in a finite number of steps as we state in the following theorem.

Theorem 2. *The algorithm, OLSDLP, for finding the solution of ECP(t, j), results in either an infeasibility criterion, an unbounded solution, or an optimal solution in a finite number of steps.*

Proof. Every iteration of the algorithm results in the addition of a constraint of the form (7) or (15) to the optimization in (17). Since there are a finite number of bases for each ECP$(t+1, \bar{j})$, the number of these constraints is finite. They also cannot be repeated since $x_{t+1}^{j,i}$ would already have had to satisfy that constraint. Therefore, the algorithm terminates after a finite number of steps. ∎

The finiteness of OLSDLP can also be maintained if we delete the cuts in (17) that are slack after each iteration. This is true because the objective function in (17) is monotonically decreasing as new cuts are introduced. We also need only keep at most $m(t+1)+1$ cuts because the solution of ECP(t, j) has at most $m(t) + m(t+1)+1$ basic variables, as Murty showed in [45]. We will return to this important property in more detail in Chapter V in our discussion of the local basis method for SDLP.

We have seen how a master problem at period t in SDLP can be solved by outer linearization using subproblems at period $t+1$. As we stated above, this development is completely analogous to applying inner linearization to the dual. In the next section, we demonstrate the relationship between OLSDLP and a Dantzig-Wolfe decomposition form of inner linearization.

3. The Relationship to Dantzig-Wolfe Decomposition

Dantzig and Madansky [18] in their fundamental paper on programming under uncertainty first proposed that two-stage stochastic linear programs could be solved by applying Dantzig-Wolfe decomposition to the dual of the stochastic linear program. In the context of our development here, we want to apply this decomposition to the linear subproblem of SDLP at period t and scenario j. We call this program LP(t,j). (Note that in this case we must assume that ξ_{t+1} has a discrete distribution as in SDLP.) The problem we address is then

$$\min z_t^j = c_t x_t^j + p^1 c_{t+1} x_{t+1}^{\bar{j},1} + \cdots + p^k c_{t+1} x_{t+1}^{\bar{j},k}$$

subject to

$$\begin{aligned} A_t x_t^j &= \xi_t^j + B_{t-1} x_{t-1}^{\hat{j}} \\ -B_t x_t^j + A_{t+1} x_{t+1}^{\bar{j},1} &= \xi_{t+1}^{\bar{j},1} \\ -B_t x_t^j + A_{t+1} x_{t+1}^{\bar{j},k} &= \xi_{t+1}^{\bar{j},k} \\ x_t^j, x_{t+1}^{\bar{j},i} &\geq 0, \quad i = 1, \ldots, k. \end{aligned}$$

This is the program for one section of SDLP. We have removed the constraints before period t and after $t+1$.

The dual of LP(t,j) can be written as

$$\max u(\xi_t^j + B_{t-1} x_{t-1}^{\hat{j}}) + p^1 v^1 \xi_{t+1}^{\bar{j},1} + \cdots + p^k v^k \xi_{t+1}^{\bar{j},k}$$

subject to

$$\begin{aligned} uA_t - p^1 v^1 B_t \quad \cdots \quad p^k v^k B_t &\leq c_t, \\ v^1 A_{t+1} &\leq c_{t+1},\ i = 1, \ldots, k. \end{aligned} \qquad (DP(t,j))$$

Next, let $\{\pi^l\} = \{(\pi_1^l, \ldots, \pi_k^l)\}$, where $l = 1, \ldots, q$, be the set of all possible combinations of k extreme points of $P = \{\pi | \pi A_{t+1} \leq c_{t+1}\}$, and let $\{\sigma^m\}$, where $m = 1, \ldots, p$, be the set of extreme rays of P. Define also $\overline{\pi}^l = \sum_{i=1}^{k} p^i \pi_i^l$, $\overline{\rho}^l = \sum_{i=1}^{k} p^i \pi_i^l \xi_t^{\overline{j},i}$ and $\gamma^{m,i} = \sigma^m \xi_t^{\overline{j},i}$. DP$(j,t)$ can be written in Dantzig-Wolfe decomposition form as:

$$\max u(\xi_t^j + B_{t-1} x_{t-1}^{\hat{j}}) + \sum_{\ell=1}^{q} \lambda^\ell \overline{\rho}^\ell + \sum_{m=1}^{p} \sum_{i=1}^{k} \nu^{m,i} p^i \gamma^{m,i}$$

subject to

$$\begin{aligned} uA_t - \sum_{\ell=1}^{q} \lambda^\ell \overline{\pi}^\ell B_t - \sum_{m=1}^{p} \sum_{i=1}^{k} \mu^{m,i} p^i \sigma^m B_t &\leq c_t, \\ \sum_{\ell=1}^{q} \lambda^\ell &= 1, \\ \lambda^\ell \geq 0;\ \mu^{m,i} \geq 0, \quad &\text{for all } i, l, \text{ and } m. \end{aligned} \tag{DWD(t,j)}$$

Now, we take the dual of DW(t,j) and use x_t^j and θ as multipliers. The result is

$$\min c_t x_t^j + \theta$$

subject to

$$\begin{aligned} A_t x_t^j &= \xi_t^j + B_{t-1} x_{t-1}^{\hat{j}}; \\ (-\sigma^m B_t) x_t^j &\geq \sigma^m \xi^{j,i}; \quad i = 1, \ldots, k;\ m = 1, \ldots, p; \\ (-\overline{\pi}^\ell B_t) x_t^j + \theta &\geq \overline{\rho}^\ell; \quad \ell = 1, \ldots, q; \\ x_t^j &\geq 0. \end{aligned} \tag{DWD(t,j)}$$

DWD(t,j) has the same form as (17) except that we have not included the extra constraints that enter into the subproblems of ECP$(t+1, \overline{j})$. The feasibility criteria correspond to the subproblems' proposing an extreme ray to the master problem in DW(t,j), and the optimality cuts on θ correspond to extreme point proposals. Optimality in the outer linearization corresponds then to the absence of better proposals from the subproblems in the Dantzig-Wolfe approach. We state these results in the following lemma:

Lemma 3. *The outer linearization of the primal problem $ECP(t, j)$ in (17) is equivalent to solving the dual of Dantzig-Wolfe inner linearization as applied to the dual problem of $ECP(t, j)$, $DP(t, j)$.*

Having completed the analysis of this basic algorithm, we would like to show how it is implemented in solving the entire program, SDLP. Further complications enter into OLSDLP because of possible degeneracy in the subproblems. In the next section, we discuss these difficulties and how they relate to stochastic programs. We also propose ways for resolving them.

4. The Degeneracy Problem

One weakness of decomposition techniques is that much of the work to optimize subproblems can be wasted, because the final inputs from the sub to the master differ so much from the initial ones. This can lead to many iterations from master to subproblem that a method with more interaction between the problems might be able to avoid. The next two methods we present have a more unified framework, and hence, fit this description. In this section, we will show how to make OLSDLP more responsive in the subproblems to changes in the master.

One property that might cause unnecessary iterations in decomposition schemes is the fact that excess columns in the basis of the master problem (that is,more than those required to meet the original set of constraints) cause degeneracy in the solution of the subproblems. Dantzig and Abrahamson

[1] first observed (and then proved) this property in their experiments with a dual nested decomposition algorithm for deterministic multi-stage linear programs. They also noticed that repeated sub-optimization changed the basis in the master problem very slightly, and they theorized that some efficiency may be gained by allowing the subproblems to determine some of the values of first period basic variables. This is possible because of the subproblem degeneracy.

To show degeneracy in SDLP, we will again refer to ECP(t, j) as written in (17) and will use ECP$(t+1, \bar{j})$ as

$$\min z_{t+1}^{\bar{j}} = c_{t+1} x_{t+1}^{\bar{j}}$$

subject to

$$\begin{aligned} A_{t+1} x_{t+1}^{\bar{j}} &= \xi_{t+1}^{\bar{j}} + B_t x_t^j, \\ x_t^{\bar{j}} &\geq 0. \end{aligned} \tag{19}$$

where we have dropped the $Q_{t+1}^{\bar{j}}$ and $D_{t+1}^{\bar{j}}$ from ECP$(t+1, \bar{j})$ for the sake of clarity. Equation (19) is the form of ECP$(t+1, \bar{j})$ used before any of its subproblems have been encountered. The degeneracy result is included in the following lemma.

Lemma 4. *If a constraint of the form (7) is binding at the optimal solution $x_t^{j,*}$, to (17), then every feasible primal basic solution of (19) with right-hand side, $\xi_{t+1}^{\bar{j}} + B_t x_t^{j,*}$, is degenerate.*

Proof. For the binding cut, we have

$$(-\sigma_{t+1}^{\bar{j}} B_t) x_t^{j,*} = \sigma_{t+1}^{\bar{j}} \xi_{t+1}^{\bar{j}}. \tag{20}$$

Now, let $A^{\overline{B}}_{t+1}$ be a feasible basis for (19). By applying $z^{\overline{j}}_{t+1}$ to this, we find

$$\sigma^{\overline{j}}_{t+1} A^{\overline{B}}_{t+1} x^{\overline{B}}_{t+1} = \sigma^{\overline{j}}_{t} (\xi^{\overline{j}}_{t+1} + B_t x^{j,*}_t) = 0. \tag{21}$$

We have $\sigma^{\overline{j}}_{t+1} A^{\overline{B}}_{t+1} \leq 0$, but $\sigma^{\overline{j}}_{t+1} A^{\overline{B}}_{t+1} \neq 0$ for $A^{\overline{B}}_{t+1}$ a basis, hence, there exists some $x^{\overline{B}}_{t+1,j} = 0$, proving the result. ∎

This lemma implies that a degeneracy will occur in any subproblem that has forced a tight feasibility cut on the master problem. The constraints in θ which enter the optimality conditions can also cause degeneracies in the subproblems. The difficulty with these degeneracies, however, is that they may enter in any subproblem, and we may not be able to determine which one. We state this degeneracy result in the following lemma.

Lemma 5. *If two constraints of the form (12) are binding at the optimal solution $(x^{j,*}_t, \theta^{j,*}_t)$ to (17), then every solution of (19) for all $\overline{j}$ which satisfies the optimality criterion, $\overline{z}^{\overline{j}}_{t+1} \geq \theta^{j,*}_t$, includes a degenerate solution for some $\overline{j}$.*

Proof. Let the binding constraints be

$$-(\pi^1 B_t) x^{j,*}_t + \theta^{j,*}_t = \rho^1, \tag{22}$$

and

$$-(\pi^2 B_t) x^{j,*}_t + \theta^{j,*}_t = \rho^2. \tag{23}$$

Let the optimal set of bases for $\{\overline{j}^1, \ldots \overline{j}^k\}$ be $\{A^{B_{1,k}}_{t+1}, \ldots, A^{B_{1,k}}_{t+1}\}$. Associated with these bases are prices $\{\pi^{1,*}, \ldots, \pi^{*,k}\}$. These prices may be the same as those for one of (22) or (23). Without loss of generality assume they are identical with the prices in (23). They must be distinct from (22) because, in the progression of the algorithm, (23) must have been violated, when (22) was satisfied and before (23) was added to the constraint set of (17). Now, letting

$$\pi^1 = \sum_{i=1}^{k} p^i \pi^{1,i} \text{ and } \rho^1 = \sum_{i=1}^{k} p^i \pi^{1,i} \xi_{t+1}^{\bar{j}i},$$

we have

$$\sum_{i=1}^{k} p^i \pi^{1,i} A_{t+1}^{B_i} x_{t+1}^{\bar{j}i} = \sum_{i=1}^{k} p^i \pi^{1,i} (\xi_{t+1}^{\bar{j}i} + B_t x_t^{j,*}), \tag{24}$$

and

$$\sum_{i=1}^{k} p^i \pi^{i,*} A_{t+1}^{B_i} x_{t+1}^{\bar{j}i} = \sum_{i=1}^{k} p^i \pi^{j,*} (\xi_{t+1}^{\bar{j}i} + B_t x_t^{j,*}). \qquad (25).$$

So, by assumption

$$\theta_t^{j,*} = \sum_{i=1}^{k} p^i \pi^{j,*} (\xi_{t+1}^{\bar{j}i} + B_t x_t^{j,*})$$

and $\pi^{i,*} A_{T+1}^{B_1} = c^{B_1}$, and $\pi^{1,i} A_{t+1}^{B_i} \leq c^{B_i}$ for $\pi^{1,i}$ feasible for all i. Therefore, we obtain

$$\sum_{i=1}^{k} p^i \pi^{j,*} A_{t+1}^{Bi} x_{t+1}^{\bar{j}i} = \sum_{i=1}^{k} p^i \pi^{1,i} A_{t+1}^{bi} x_{t+1}^{\bar{j}i} \tag{26}$$

or

$$\sum_{i=1}^{k} p^i (c^{Bi} - \pi^{1,i} A_{t+1}^{Bi}) x_{t+1}^{\bar{j}i} = 0. \tag{27}$$

Now, $p^i \geq 0$, $x_{t+1}^{\bar{j}i} \geq 0$, and $c^{Bi} - \pi^{1,i} A_{t+1}^{Bi} \leq 0$, but we must have $c^{Bi} \neq \pi^{1,i} A_{t+1}^{Bi}$ for $\pi^{i,*}$ unique; therefore, there exists some $\bar{j}i$ such that $x_{t+1}^{\bar{j}i}$ is degenerate. Hence, the result. ∎

The degeneracy we have shown implies that the subproblems are too restricted to alter the direction of the solution to the master problem. Dantzig and Abrahamson [1] have proposed remedying this difficulty in the multi-stage deterministic model by passing columns forward from the master to the subproblem and allowing the subproblems to determine the weights of these surplus columns in the master problem. We present below an application of this technique to SDLP and discuss its weaknesses in the case of stochastic programs.

We assume an optimal solution, $x_t^{j,*}$, of (17) is partitioned as

$$x_t^{j,*} = (x_t^B, x_t^s) \tag{28}$$

where x_t^B corresponds to a square nonsingular submatrix of A_t, A_t^B, and x_t^s corresponds to A_t^s. We can write x_t^B in terms of x_t^s as

$$x_t^B = (A_t^B)^{-1}(\xi_t^j + B_{t+1}x_{t+1}^{\hat{j}}) - (A_t^B)^{-1}A_t^s x_t^s. \tag{29}$$

So, we can let x_t^s vary as long as $x_t^B \geq 0$. We then can write (19) for each $\bar{j}$ as

$$\min c_{t+1} x_{t+1}^{\bar{j}}$$

subject to

$$\begin{aligned} A_{t+1}x_{t+1}^{\bar{j}} = \xi_{t+1}^{\bar{j}} &+ B_t^B(A_t^B)^{-1}(\xi_t^j + B_{t-1}x_{t-1}^{\hat{j}}) \\ &- B_t^B(A_t^B)^{-1}A_t^S x_t^S + B_t^S x_t^S, \\ x_{t+1}^{\bar{j}} \geq 0,\ & x_t^S \geq 0, \end{aligned} \tag{30}$$

or, letting
$\hat{B}_t^S = B_t^S - B_t^B(A_t^B)^{-1}A_t^S$ and $\hat{\xi}_{t+1}^{\bar{j}} = \xi_{t+1}^{\bar{j}} + B_t^B(A_t^B)^{-1}(\xi_t^j + B_{t-1}x_{t-1}^{\hat{j}})$,,
as

$$\min c_{t+1} x_{t+1}^{\bar{j}}$$

subject to

$$\begin{aligned} A_{t+1}x_{t+1}^{\bar{j}} - \hat{B}_T^S x_t^S &= \xi_{t+1}^{\bar{j}}, \\ x_{t+1}^{\bar{j}} \geq 0,\ x_t^S &\geq 0, (x_t^B \geq 0), \end{aligned} \tag{31}$$

where we have added the constraint $x_t^B \geq 0$, parenthetically, because we must check in optimizing (31) that this is not violated. If so, we would fix the variables at zero and proceed.

This problem of checking for feasibility of x_t^B enters both the deterministic and stochastic programs, but, in the stochastic program, any of the subproblems of type (31) may determine x_t^s. Since x_t^s enters the program in which a degeneracy is caused (filling the degenerate variable's position in the basis), we must know for which $\bar{j}$ the program (31) will have some x_t^s basic. This would be possible if all the x_t^s corresponded to tight *feasibility* (type (7)) cuts, because then the degeneracy would occur in the corresponding scenarios which generated those cuts. For the *optimality* cuts (type (12)), however, degeneracy, where x_t^s is basic, can be in any scenario.

To apply this column passing technique, in general, to SDLP, we can formulate the following alternative form of (17). It includes constraints to keep x_t^B feasible as well as the remaining additional cuts.

$$\min \tilde{c}_t^S x_t^S + \theta_t^j$$

$$\text{subject to}$$

$$\begin{aligned} -(A_t^B)^{-1}A_t^s x_t^s &\geq -(A_t^B)^{-1}(\xi_t^j + B_{t-1}x_{t-1}^{\hat{j}}), \\ -(\tilde{\sigma}_t^{\bar{j},k} B_t^s)x_t^s &\geq [\sigma_t^{j,k}(\tilde{\xi}_{t+1}^{\bar{j}})], k = 1,\ldots,p, \\ -(\tilde{\pi}_1^{\ell} B_t^s)x_t^s + \theta_t^j &\geq \tilde{\rho}^{\ell}, \ell = 1,\ldots,q, \\ x_t^s &\geq 0, \end{aligned} \tag{32}$$

where $\tilde{c}_t^s = c_t^B(A_t^B)^{-1}A_t^s + c_t^s$, and the other quantities under tilde are correspondingly defined to reflect the substitution of (29) for x_t^B.

The program (32) is a second master problem that we can use to determine the optimal values of x_t^s given x_t^B. Our proposal then is to follow

OLSDLP with (32) in place of (17). We would do this after solving the subproblems of the form (31), which have generated tight feasibility cuts for (17) and in which we know degeneracies must occur.

The use of this method of passing some columns for tight feasibility cuts and then determining the other x_t^s by using (32) in OLSDLP depends on the difficulty of solving the first master problem. If the repeated solution of (17) has indicated that some variables, x_t^B, are persistent in the basis, then the solution of (32) could obtain the optimal values without involving a reoptimization of (17). Furthermore, if the number of surplus variables, x_t^s is small, (32) may become significantly easier to solve than (17). This alternative method then is one that must be adopted to the individual problem and its requirements. The complications of creating an additional optimization problem in (32) may outweigh the savings in solving this smaller problem.

5. The Complete Methods

We are prepared now to present the algorithm NDSDLP for the entire stochastic problem. This method involves repeated use of the algorithm OLSDLP, and proceeds through the tree of possible scenarios in SDLP in a forward and backward manner. Our presentation here does not include the resolution of the degeneracy problem as discussed in Section 4, but this may be added as a modification to OLSDLP. The algorithm follows.

NDSDLP

Step 0. Set up a problem of the form (17) with no extra constraints for each scenario j in each period t of SDLP.

Step 1. Solve (17) for period T (written 17-1). Use the result x_1^*, and solve

(17) for each node in period 2 ((17-2j) for $j = 1, \ldots, k_2$). If any subproblem is infeasible, add a feasibility constraint of type (7) to (17-1). Solve (17-1) again and proceed to period 2.

After each subproblem at $t = 2$ is feasible, proceed to solve (17-3j) for $j = 1, \ldots, k_3$. Again, for any infeasibilities, pass back cuts. Continue in this manner to period T, until there is a feasible solution for (17-tj) for all t and j. If an infeasible solution to (17-1) ever results, then stop–the problem is infeasible. Else, the result is a feasible primal solution of SDLP.

Step 2. Solve OLSDLP for every scenario in period $T - 1$. This implies master-suboptimality for the last period. Set $T = T - 2$ and go to Step 3.

Step 3. If $t = 1$, go to Step 4. Otherwise, solve OLSDLP for every scenario in period t. For some scenario j at t, this may involve resolving OLSDLP for its descendants $\bar{j}$ at $t + 1$ in order to get optimality for the subproblems. We say a subproblem is "solved", in terms of the algorithm, when OLSDLP applied to it ends in master-suboptimality.

After OLSDLP has been solved for each j at t, set $t = t - 1$ and repeat Step 3.

Step 4. Solve OLSDLP for the original master problem at period 1. This program may again involve resolving the subproblems. If OLSDLP at 1 results in an unbounded or infeasibility termination, then stop–SDLP is accordingly unbounded or infeasible. If OLSDLP ends with master-suboptimality, then stop–the current solution is optimal for SDLP.

This algorithm follows an iterative procedure as in dynamic programming. We pursue this relationship more closely in Chapter VI. NDSDLP also terminates in a finite number of steps as we state in the following theorem.

Theorem 3. *The method, NDSDLP, terminates in a finite number of steps with an optimal solution to SDLP or the unbounded or infeasibility conditions from OLSDLP.*

Proof. From Theorem 2, we know that each implementation of OLSDLP must terminate in a finite number of steps. Since the algorithm proceeds backwards after each period's scenarios are solved by OLSDLP, the terminal conditions in Step 4 must be met in a finite number of steps. ∎

Several improvements can be made to NDSDLP to aid in its efficiency. We have already mentioned the second decomposition possible in OLSDLP as a resolution to the degeneracy problem. We may also want to proceed between the periods without completely satisfying master-suboptimality. This modification may help efficiency, but it must be done carefully in order to avoid any excessive number of iterations among the periods.

Another possibility for speeding the search for a feasible primal solution in Step 1 is that SDLP may have inequality constraints. In this case, the subproblem at period t is

$$\min c_t x_t$$

subject to

$$\begin{aligned} A_t x_t &\geq \xi_t^{\bar{j}} + B_{t-1} x_{t-1}, \\ x_t &\geq 0, \end{aligned} \tag{33}$$

for all $\bar{j}$. Hence, to find feasibility for all $\bar{j}$, we need only solve the Phase I problem

$$\min ev$$

subject to

$$\begin{aligned} A_t x_t - Iu + Iv &= \alpha_t + B_{t-1} x_{t-1} \\ x_t &\geq 0, \end{aligned} \tag{34}$$

where $\alpha_t = (\alpha_{t,i} \mid \alpha_{t,i} = \max_{\bar{j}} \xi^{j}_{t,i})$, thus $\alpha_t \geq \xi^{\bar{j}}_t$ for all $\bar{j}$. A feasible solution to (33) implies that each subproblem $\bar{j}$ at period t is feasible for x_{t-1}.

By solving (33) in Step 2 of OLSDLP, instead of solving the Phase I problem (18) of each $\bar{j}$, we eliminate many unnecessary optimizations. This would greatly aid the efficiency of NDSDLP for SDLP's that have inequality constraints. These programs are quite common in practice and, thus, (33) should prove most valuable.

This modification and our presentation of NDSDLP above demonstrate some of the possibilities for solving SDLP by concentrating on the optimization of smaller subproblems. In Chapter VII, we discuss the computational aspects of the algorithm more carefully. In the next two chapters, we present other methods that rely upon subproblem optimization, but maintain closer ties between the sub- and master problem.

Chapter IV

A Piecewise Strategy

1. Introduction

The decomposition algorithm, NDSDLP, in the last chapter broke the stochastic program SDLP into a sequence of master–subproblem relations. A drawback to NDSDLP could come from wasted effort in optimizing subproblems without affecting the master problems' inputs. The approach in this chapter, the *piecewise strategy* for SDLP, or *PCSDLP*, also separates the program into master and subproblems, but it allows for only one optimization of the subproblems without the master's involvement. This method, which maintains some ties among the separate scenarios, forms a bridge between the decomposition approach in Chapter III and the local basis factorization of Chapter V.

The method is called "piecewise" because it relies on the piecewise linear property of the objective function. Piecewise methods in general (see Geoffrion [27]) follow an optimizing trajectory across the regions of the feasible set. For a convex function, an optimization is performed on each region that leads either to a boundary or interior solution. If the solution is interior, then that point is optimal. If a boundary point is optimal, one optimizes on the adjacent region and repeats the process. (See Figure 1.) If no direction in an adjacent region is improving, then the current point again is optimal.

The piecewise strategy has been applied to large-scale linear programming through a method, called "partitioning", for which, J. B. Rosen [53] has been most responsible. Our use of the strategy in this chapter will be to exploit the repetitions of the blocks of SDLP and to form an algorithm that can adapt to different scenarios and combine them adequately. In Section 2, we present the basic master–subproblem algorithm for this method and show where efficiencies can be made in its implementation. Section 3 then states the strategy for the full program and presents the difficulties that may occur with PCSDLP.

2. The Master–Subproblem Relationship

The method proceeds by performing two–period optimizations for successively larger problems. In this section, we present the two–period problem, in which, the first, parent, scenario forms the master problem for its direct descendants, the subproblems. Without loss of generality, we assume that this optimization takes place between the first and second periods.

The first period problem we solve is

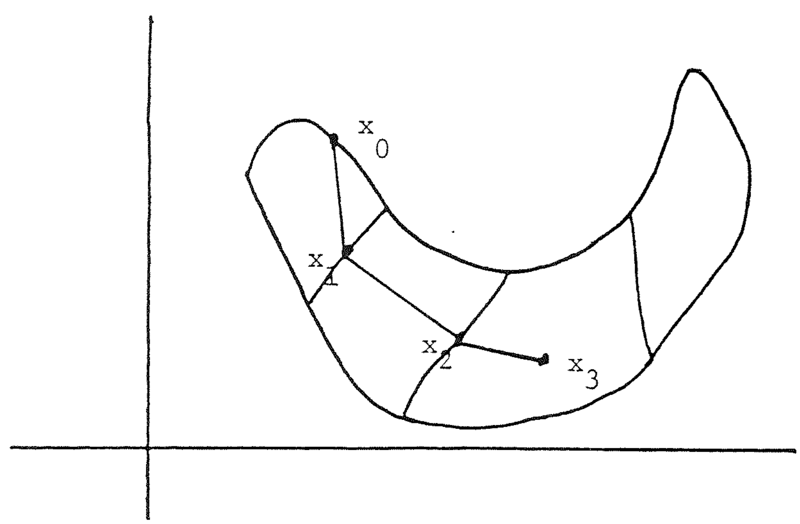

Figure 1. The piecewise path begins at x_0 and leads to x_3.

$$\begin{aligned} \min \varsigma^0(x_1) &= c_1 x_1 \\ \text{subject to} \quad A_1 x_1 &= b_1, \\ x_1 &\geq 0. \end{aligned} \tag{1}$$

If (1) is infeasible, the program is infeasible, and we stop. If (1) is unbounded, then we follow Step $2'$ of NDSDLP to remove this case. If we succeed, we return with the cuts that eliminate the unbounded ray and resolve (1). Now, for x_1^0, an optimal solution to (1), we want to find $Q(x_1^0)$ as ECP of Chapter III, where

$$Q(x_1^0) = \mathrm{E}_{\xi_2}[\, \min c_2 x_2 \text{ s.t. } A_2 x_2 = \xi_2 + B_1 x_1^0, x_2 \geq 0]. \tag{2}$$

Here, if there exists ξ_2^j such that there is no feasible solution in the jth scenario, then we form a cut as in (3.7) and add it to (1) as part of its constraint matrix. We continue until each subproblem is feasible. Next, associated with each ξ_2^j, there exists an optimal basis, A_2^B, for the problem in $Q(x_1^0)$. We write $Q(x_1^0)$ as

$$Q(x_1^0) = \sum_{j=1}^{k_2} p^j c_2((A_2^{B_j})^{-1} \xi_2^j + (A_2^{B_j})^{-1} B_1 x_1^0). \tag{3}$$

The function $Q(x_1)$ from (3) is linear for all x_2 feasible for $A_2^{B_j}$. Thus,

$$Q(x_1) = \sum_{j=1}^{k_2} p^j c_2((A_2^{B_j})^{-1} \xi_2^j + (A_2^{B_j})^{-1} B_1 x_1) \tag{4}$$

for all x such that

$$x_2 = (A_2^{B_j})^{-1} \xi_2^j + (A_2^{B_j})_1^{-1} x_1 \geq 0, \tag{4a}$$

where we have assumed a single lower bound for all x_2 at 0. In general, and, for most practical purposes, both lower and upper bounds should be included, augmenting the constraints in (4a). In implementations, this is especially important for artificial variables, where lower and upper bounds coincide.

Now, we can write $\varsigma^0(x_1)$ in (1) as

$$\varsigma^0(x_1) = (c_1 + \sum_{j=1}^{k_2} p^j c_2^{B_j} (A_2^{B_j})^{-1} B_1) x_1 + \sum_{j=1}^{k_2} p^j c_2^{B_j} (A_2^{B_j})^{-1} \xi_2^j, \qquad (5)$$

for all x_1 such that $(A_2^{B_j})^{-1}\xi_2^j + (A_2^{B_j})^{-1} B_1 x_1 \geq 0$ for all $j = 1, \ldots, k_2$.

We note , for optimal multipliers π^j in subproblem j, that $\pi^j = c_2^{B_j}(A_2^{B_j})^{-1}$, which makes computations in (5) somewhat easier. We also note that each $A_2^{B_j}$ need not be unique, so we may use one basis several times without checking (4a) for each j. We will return to this idea below, but, for the current exposition, we will use constraints (4a) explicitly. Thus, for x_1 restricted as in (4a), we look for an improving direction along the current linear section of $Q(x_1)$ by solving

$$\min \varsigma^1(x_1) = [c_1 + \textstyle\sum_{j=1}^{k_2} p^j c_2 (A_2^{B_j})^{-1} B_1] x_1$$

subject to

$$\begin{aligned} A_1 x_1 &= b_1, && (6a) \\ (A_2^{B_j})^{-1} B_1 x_1 &\geq \rho^j, j = 1, \ldots, k_2, && (6b) \\ x &\geq 0, && \end{aligned} \qquad (6)$$

where $\rho^j = (A_2^{B_j})^{-1}\xi_2^j$.

From (6), we obtain an optimal solution, x_1^1, and optimal objective value, $\varsigma^1(x_1^1)$. This is the initial point of our feasible path. We next look at the set of constraints in (6b) which are active. We define

$$\mathcal{T} = \{(i,j) : [(A_2^{B_j})^{-1} B_1](i,*) \cdot x_1^1 = \rho_i^j\}, \qquad (7)$$

where ρ_i^j is the ith component of the vector ρ^j. $\mathcal{T}$ includes the rows and scenarios that generated a binding cut. It corresponds to the set of degeneracies in the subproblems of NDSDLP and is directly related to the surplus columns described in the next chapter. We next order the pairs $(i, j) \in \mathcal{T}$ lexicographically and write $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_p\}$.

Now, if $\mathcal{T} = \emptyset$, then x_1^0 is a solution of (6), so, by our convexity result in Lemma 2.1, $z(x_1^0) = c_1 x_1^0 + Q(x_1^0) \leq c_1 x_1 + Q(x_1)$ for all feasible x_1. Hence, x_1^0 is an optimal solution of this two–stage SDLP.

For $\mathcal{T} \neq \emptyset$, we consider subproblem j for $\tau_1 = (i, j)$. Here, we have

$$x_2^{B_j} = (A_2^{B_j})^{-1} \xi_2^j + (A_2^{B_j})^{-1} B_1 x_1 \tag{8}$$

and

$$x_2^{B_j}(i) + (A_2^{B_j})^{-1}(i, *) A_2^N x_2^N = 0, \tag{9}$$

where x_2^N is the non–basic partition of the variables, x_2. We now want to force $x_2^{B_j}(i)$ out of the basis, so we can follow a new path in the adjacent feasible region. To maintain optimality, we find the entering variable as in the Dual Simplex Method, (see Dantzig [16])

$$\frac{\overline{c}_2^N(s)}{-\overline{A}_2^N(i, s)} = \min_{-\overline{A}_2^N(i,j) > 0} \frac{\overline{c}_2^N(j)}{\overline{A}_2^N(i, j)}, \tag{10}$$

where $\overline{c}_2^N$ and $\overline{A}_2^N$ are the representations of c_2^N and A_2^N relative to the basis, $A_2^{B_j}$.

We can now pivot out $x_2^{B_j}(i)$ and replace it with $x_2^N(s)$. In doing this, we keep $A_2^{B_j}$ and call the new basis $A_2^{B_j,*}$, an *auxiliary basis*, where

$$(A_2^{B_j,*})^{-1} = \eta(i, s)^{-1} (A_2^{B_j})^{-1}, \tag{11}$$

for $\eta(i,s)$, the elementary matrix corresponding to the pivot of $x_2^N(s)$ into the basis in position i. This new basis is used to restrict x_1 in (4a).

We now can formulate our auxiliary problem, for $\mu = 1$,

$$\min \varsigma^{\mu,*}(x_1) = [c_1 + \textstyle\sum_{j=1}^{k_2} p^j c_2^{B_j}(A_2^{B_j})^{-1}B_1]x_1$$

subject to

$$\begin{aligned} A_1x_1 &= b_1, \\ (A_2^{B_j})^{-1}B_1x_1 &\geq \rho^{j,*}, j = 1,\ldots,k_2, \\ x_1 &\geq 0. \end{aligned} \tag{12}$$

In (12), we have changed the objective function from (6) and some of the constraints (6b). The variables x_1^1 still form a feasible basis, but they may no longer be optimal.

We proceed to price out the cost row in (12) with the new parameters and to check optimality. If x_1^1 is still optimal, then we drop $A_2^{B_j,*}$ and return to $\mathcal{T}$ for τ_2, again find an entering variable, and form the auxiliary problem (12). As long as we cannot improve on x_1^1 for τ_l, we try τ_{l+1}. If we find $l+1 > p$, then no direction can improve on x_1^1, hence, it is optimal.

If we find that, for any τ_l, x_1^1 is not optimal in (12), then we optimize (12) and obtain x_1^2. We set $\varsigma^2(x_1^2) = \varsigma^{1,*}(x_1^2)$, replace $A_2^{B_j}$ by $A_2^{B_j,*}$, form a new set, $\mathcal{T}$, of the tight constraints, and proceed again with τ_1 in search of an improving direction.

We follow the above procedure to obtain a sequence of decreasing objective values, $\varsigma^1(x_1^1) > \varsigma^2(x_1^2) > \cdots > \varsigma^\mu(x_1^\mu)$, until we cannot improve our current solution. The algorithm we have described has the following steps :

PCSDLP(2)

Step 0. Find an optimal bounded feasible solution of (1), or terminate if infeasible or if Step2′ of NDSDLP implies unboundedness.

Step 0'. Find an optimal feasible basis, $A_2^{B_j}$, for each subproblem j in $Q(x_1^0)$ by applying feasibility cuts (3.7) to (1).

Step 1. Form the program (6) using the set of bases, $\mathcal{B} = \{A_2^{B_j}\}$. Solve (6) and obtain $\varsigma^1(x_1^1)$ and x_1^1. Form $\mathcal{T}$. If $\mathcal{T} = \emptyset$, stop, x_1^0 is optimal. If $\mathcal{T} \neq \emptyset$, set $l = 1$, $\mu = 1$, and, go to Step 2.

Step 2. If $l > p$, stop, x_1^μ is optimal. For $\tau_l = (i, j)$, find the entering variable s in scenario j by (10). Form the auxiliary basis, $A_2^{B_j,*}$, and the program (12) for $\varsigma^{\mu,*}(x_1)$. Using x_1^μ as a starting solution, solve (12) and obtain x_1^*. If $\varsigma^{\mu,*}(x_1^*) = \varsigma^{\mu,*}(x_1^\mu)$, set $l = l+1$ and return to Step 2. If $\varsigma^{\mu,*}(x_1^*) < \varsigma^{\mu,*}(x_1^\mu)$, go to Step 3.

Step 3. Update $\mathcal{T}$ and $\mathcal{B}$. Set $x_1^{\mu+1} = x_1^*$, $l = 1$, $\mu = \mu + 1$, and go to Step 2.

The following theorem states the finiteness of PCSDLP.

Theorem 1. *The method described above, PCSDLP, terminates in a finite number of steps with an infeasible, unbounded, or optimal solution to the two–stage ($T = 2$) form of the program, SDLP.*

Proof. From Chapter III, we know that Steps 0 and 0' must terminate in a finite number of steps. After Step 2, the solution to (6) and (12) must be feasible in SDLP because primal feasibility of the last solution is maintained. It is bounded because (6) and (12) are more restrictive than (1).

$\mathcal{T}$ is finite since the number of constraints (12b) is finite and each improving solution corresponds to a new set of bases, $\mathcal{B}$. Since there are a finite number of possible basis set combinations, the algorithm must terminate. ∎

3. A Method for Reduced Basis Storage Requirements

Efficiency and storage requirements in the solution of (12) can be significantly improved, if we do not include redundant constraints that occur with duplicated bases, $A_2^{B_j}$. To do this, begin by checking in Step 0′ for a repetition of $A_2^{B_j}$. We start with $l = 1$ and increase l, letting each distinct new basis be $A_2^{B_l}$. We obtain $\mathcal{B} = \{A_2^{B_l} | l = 1, \ldots, q\}$. We also define

$$p(l) \equiv \sum_{J(l)} p^j \tag{13}$$

where $J(l) = \{$ all scenarios j with optimal bases, $A_2^{B_l}\}$.

Now, when we construct the constraints (12b), we define

$$\rho^{l,*} = \max_{j \in J(l)} \rho^{j,*}, \tag{14}$$

and, for each component i, we store $\hat{j}(l,i)$ for every l, where $\rho^{\hat{j},*}(i) \geq \rho^{j,*}(i)$, for all $j \in J(l)$. Thus, (12) becomes

$$\min \varsigma^{\mu,*}(x_1) = [c_1 + \textstyle\sum_{l=1}^{q} p(l) c_2^{B_l,*} (A_2^{B_l,*})^{-1} B_1] x_1$$

subject to

$$A_1 x_1 = b_1, \tag{15a}$$

$$(A_2^{B_l,*})^{-1} B_1 x_1 \geq \rho^{l,*}, l = 1, \ldots, q, \tag{15b}$$

$$x_1 \geq 0.$$

(15)

Now, in Step 2, write the elements of $\mathcal{T}$ as $\tau_\lambda = (i, \hat{j}(l))$. Each time a new auxiliary basis, $A_2^{B_j,*}$, is investigated, if $A_2^{B_j,*} \in \mathcal{B}$, then we adjust $p(l)$ and possibly $\rho^{l,*}$, but do not change the coefficient matrix. If $A_2^{B_j,*} \not\in \mathcal{B}$, then we must add another set of constraints to (15). At Step 3, we update $\mathcal{B}$, the associated probabilities, $p(l)$, and update $\mathcal{T}$ using the pair, $(i, \hat{j}(l))$.

This modification can significantly reduce the number of constraints since a large number of scenarios may have the same basis. We could again use the Garstka–Rutenberg procedure (see Chapter I) to find the probability that each basis is optimal in Step $0'$ without solving the individual problems. This effect combined with the smaller size of (15) can lead to greater computational efficiencies.

Another efficiency can be gained from using a method similar to the column passing technique of Chapter III. During the algorithm, we may observe that one set of variables, $\{x_1^B\}$, remains in the optimal basic set, $\{x_1^\mu\}$, while the other variables are chosen from a set, $\{x_1^S\}$. If the columns of the set $\{x_1^B\}$ have full rank, we can take a square non–singular submatrix, A_1^B, from (15) and find

$$x_1^B = (A_1^B)^{-1} b_1 - (A_1^B)^{-1} A_1^S x_1^S. \tag{16}$$

We then eliminate x_1^B from (15) and obtain

$$\min \varsigma^{\mu,*}(x_1^S) = \qquad \tilde{c}_1^s x_1^S$$

subject to

$$\begin{aligned} -\tilde{A}_1^s x_1^S &\geq -(\widetilde{A_1^B})^{-1} b_1, \\ ((\widetilde{A_2^{B_l,*}})^{-1} B_l) x_1^S &\geq \tilde{\rho}^{l,*}, l = 1, \ldots, q, \\ x_1^S &\geq 0, \end{aligned} \tag{17}$$

where tilde indicates that $\tilde{c}_1^s$, $\tilde{A}_1^s$, $((\widetilde{A_2^{B_l,*}})^{-1} B_l)$, and $\tilde{\rho}^{l,*}$ are defined relative to x_1^B through substitution of (16) into the program (12). The definitions are completely analogous to those in (3.32).

The optimization procedure can then continue with the reduced problem (17) to find the optimal values x_1^S, given x_1^B. Using the result of (17) in (15) would then determine optimality. When decisions can be narrowed to choices among a few variables, this modification may again prove effective in improving efficiency.

The algorithm, PCSDLP(2), is stated for two–period optimizations. It does not require the subproblems to be reoptimized after Step $0'$. By maintaining primal feasibility, it is always on a feasible path to the solution and may eliminate the problems of wandering among multiple suboptimal points. In the next section, we present the implementation of PCSDLP(2) for general multi–stage programs.

4. The Complete Solution Strategy

The PCSDLP method follows a procedure very similar to NDSDLP in its passing through the scenarios from period to period. In fact, both of these methods can be seen as local approximations of a dynamic programming scheme, which we present in greater detail in Chapter 6. PCSDLP even begins by finding a feasible primal solution through NDSDLP, but PCSDLP never allows for primal infeasibility or non–optimality in a subproblem after a single optimization.

First, we set up subproblems for each node as in Step 0 of NDSDLP. Next, we find a feasible primal solution by Step 1 of NDSDLP, passing feasibility cuts as we proceed through the periods from 1 to T. After finding this feasible solution, we start by applying PCSDLP(2) to the master–subproblem relations at period $T-1$. Primal feasibility is then maintained throughout the optimization.

For each scenario j in period $T-1$, we solve

$$
\begin{aligned}
&\min\ c_{T-1}x^{j}_{T-1}+\sum_{\bar{j}=1}^{\bar{k}_T} p^{\bar{j}} c_T x^{\bar{j}}_T \\
&\text{subject to} \\
&A_{T-1}x^{j}_{T-1} \qquad\qquad\qquad\qquad = \xi^{j}_{T-1} + B_{T-2}\hat{x}^{\hat{j}}_{T-2}, \\
&-B_{T-1}x^{j}_{T-1} \qquad\qquad +A_T x^{\bar{j}}_T = \xi^{\bar{j}}_T,\ \bar{j}=1,\ldots,\bar{k}_T, \\
&\qquad\qquad x^{j}_{T-1} \geq 0, x^{\bar{j}}_T \geq 0,\ \bar{j}=1,\ldots,\bar{k}_T,
\end{aligned}
\tag{18}
$$

by using PCSDLP(2).

For a solution to (18), define

$$\beta^{\bar{j}} = \{i : x^{\bar{j}}_T(i) \text{ is basic in subproblem } \bar{j}\}$$

and

$$\gamma^{\bar{j}} = \{i : i \text{ is basic in row } l \text{ and } (l,\bar{j}) \in \mathcal{T}\}.$$

Now, with the solution from PCSDLP(2), we want to find the optimal basis for the full problem (18). This larger matrix will form the basis of a subproblem for period $T-2$. We first include the set of basic variables in the master problem, $\{x^{B_j}_{T-1}\}$. The basic variables from the subproblems will be chosen as

$$X^{j}_T = \{x^{\bar{j}}_T(i) : i \in \beta^{\bar{j}} \bigcap \overline{\gamma}^{\bar{j}}\}. \tag{19}$$

This definition eliminates the degenerate variables from the basic set. Since the elements of X^{j}_T are the only non–zero variables in an optimal feasible solution to (18), if (18) is not degenerate, then the union of $\{x^{B_j}_{T-1}\}$ and X^{j}_T must form a basic set of variables in (18).

If (18) is degenerate, then we must check whether the columns corresponding to $\{x^{B_j}_{T-1}\}$ and X^{j}_T span the solution space. If not, we add columns from those corresponding to

$$\overline{X}^{j}_T = \{x^{\bar{j}}_T(i) : i \in \beta^{\bar{j}} \bigcap \gamma^{\bar{j}}\}. \tag{20}$$

If a column of $x_T^j(i) \in \overline{X}_T^j$ is independent of the columns in the present basis, then it is added until full rank is a achieved. We know that we must obtain a basis since the union of all columns for $\{x_{T-1}^{B_j}\}$, X_T^j, and $\overline{X}_T^j$ spans the space.

We thus obtain a basis for all j in $T-1$. We call this basis, $D_{T-1}^{B_j}$, and we further define

$$D_{T-1} = \begin{pmatrix} A_{T-1} & & \\ -B_{T-1} & A_T & \\ \cdot & & \cdot \\ -B_{T-1} & & A_T \end{pmatrix} \tag{21}$$

$$d_{T-1} = (c_{T-1}, p^1 c_T, \ldots, p^{\overline{k}_T} c_T), \tag{22}$$

$$\psi_{T-1}^j = (\xi_{T-1}^j, \xi_T^1, \ldots, \xi_T^{\overline{k}_T})^T, \tag{23}$$

and

$$y_{T-1}^j = (x_{T-1}^j, x_T^{\overline{j}_1}, \ldots, x_T^{\overline{j}_{\overline{k}_T}}) \tag{24}$$

Now, the problem for PCSDLP(2) is

$$\min \varsigma_{T-2}^{\hat{j}} = c_{T-2} x_{T-2}^{\hat{j}} + \sum_{j=1}^{\overline{k}_{T-1}} p^j d_{T-1} y_{T-1}^j \tag{25}$$

subject to

$$A_{T-2} x_{T-2}^{\hat{j}} = \xi_{T-2}^{\hat{j}} + B_{T-1} x_{T-1}^{\hat{\hat{j}}}, \tag{25a}$$

$$-\tilde{B}_{T-2} x_{T-2}^{\hat{j}} + D_{T-1} = \psi_{T-1}^j,\ j = 1, \ldots, \overline{k}_{T-1}, \tag{25b}$$

$$x_{T-2}^{\hat{j}} \geq 0, y_{T-1}^j \geq 0,\ j = 1, \ldots, \overline{k}_{T-1},$$

where $\tilde{B}_{T-2}$ is the matrix B_{T-2} augmented by zeroes to correspond with D_{T-1}^j.

To begin PCSDLP(2) for (25), we substitute constraints of the form in (6b) for (25b) and enter Step 1 in PCSDLP(2). We never reoptimize the subproblems, but look for feasibility maintaining pivots in both periods $T-1$

and T. In general, after solving the two-stage problem for each scenario j in period t, we would again find the bases, $D_t^{B_j}$, and construct a two-stage problem for $t-1$ as in (25) by combining x_t^j with $(y_{t+1}^{\bar{j}_1}, \ldots, y_{t+1}^{\bar{j}_{\bar{k}_{t+1}}})$. PCSDLP(2) would begin by optimizing

$$\min \varsigma^1(x_{t-1}^{\hat{j}}) = [c_{t-1} + \textstyle\sum_{j=1}^{\bar{k}_{t-1}} p^j d_t^{B_j} (D_t^{B_j})^{-1} B_1] x_{t-1}^{\hat{j}} \tag{26}$$

subject to

$$\begin{aligned} A_{t-1} x_{t-1}^{\hat{j}} &= \xi_{t-1}^{\hat{j}} + B_{t-2} x_{t-2}^{\hat{\hat{j}}}, \\ (D_t^{B_j})^{-1} \tilde{B}_{t-1} x_{t-1}^{\hat{j}} &\geq \rho^j,\ j = 1, \ldots, \bar{k}_{t-1}, \\ x_{t-1}^{\hat{j}} &\geq 0. \end{aligned}$$

The dual pivoting operations could then be performed in any of periods t through T.

PCSDLP continues by combining master problems with subproblems and following these iterations back to period 1. This process uses the basis structure depicted in Figure 2b. of Chapter I, in which, we view each scenario as starting a new problem. The steps we have described for PCSDLP follow.

PCSDLP

Step 1. Follow Step 0 and Step 1 of NDSDLP to obtain a feasible solution to SDLP. Set $t = T - 1$ and set up a program of the form (12) for each scenario j in $T-1$. Set $j = 1$.

Step 2. Follow Steps 1, 2, and 3 of PCSDLP(2) for the problem at node (j, t). If $j = k_t$, go to Step 3. If $j < k_t$, set $j = j + 1$ and return to Step 2.

Step 3. If $t = 1$, stop, x_1^μ is optimal. If $t > 1$, combine the master and subproblems of each scenario j at period t and form programs as in (26) to initiate PCSDLP(2). Set $t = t - 1$ and go to Step 3.

The finite termination of this method is guaranteed by the finiteness of

PCSDLP(2) and our passing back one period in each encounter with Step 3. We state this as a Corollary to Theorem 1.

Corollary. *PCSDLP terminates in a finite number of steps with an infeasible or unbounded solution from the procedures of NDSDLP or with an optimal bounded feasible solution to SDLP.*

PCSDLP's greatest potential improvement over NDSDLP is, as we have emphasized, its maintenance of primal feasibility and subproblem optimality. This advantage over the possible suboptimizations and infeasibilities in NDSDLP must be discounted, however, by the growth of the bases, D_t^B, in the subproblems. Their larger size may lead to a greater number of computations in performing pricing and the minimum ratio test in (1). We can, however, gain efficiency with a compact factorization of D_t^B. The following chapter describes such a technique and its application to the full problem, SDLP. This local basis method could then be used in conjunction with PCSDLP to gain still greater efficiency.