Dantzig-Wolfe Decomposition for Solving Multi-Stage Stochastic Capacity-Planning Problems

Kavinesh J. Singh* Andy B. Philpott† R. Kevin Wood‡

March 4, 2008

Abstract

We describe a multi-stage, stochastic, mixed-integer-programming model for planning discrete capacity expansion of production facilities. A scenario tree represents uncertainty in the model; a general mixed-integer program defines the operational submodel at each scenario-tree node; and capacity-expansion decisions link the stages. We apply "variable splitting" to two model variants, and solve those variants using Dantzig-Wolfe decomposition. The Dantzig-Wolfe master problem can have a much stronger linear-programming relaxation than is possible without variable splitting, over 700% stronger in one case. The master problem solves easily and tends to yield integer solutions, obviating the need for a full branch-and-price solution procedure. For each scenario-tree node, the decomposition defines a subproblem that may be viewed as a single-period, deterministic, capacity-planning problem. An effective solution procedure results as long as the subproblems solve efficiently, and the procedure incorporates a good "duals stabilization scheme." We present computational results for a model to plan the capacity expansion of an electricity distribution network in New Zealand, given uncertain future demand. The largest problem we solve to optimality has 6 stages and 243 scenarios, and corresponds to a deterministic equivalent with a quarter of a million binary variables.

Key words: Multi-stage stochastic mixed-integer program, column generation, branch-and-price, capacity expansion, Dantzig-Wolfe decomposition

Subject classification: Facilities/equipment planning, capacity expansion, discrete, stochastic: multi-stage. Production/scheduling, planning. Programming, stochastic, integer: column generation, Dantzig-Wolfe decomposition, branch-and-price.

1 Introduction

Research from as early as the 1950s (Masse and Gibrat 1957) suggests that effective capacity planning for industrial facilities must treat uncertainty explicitly. The list of uncertain parameters can include product demands at those facilities, expansion costs, operating costs, and production efficiencies. This paper studies capacity-planning problems in which a sequence of discrete, capacity-expansion decisions must be made over a finite planning horizon, subject to one or more sources of uncertainty.

A deterministic, single-period instance of our model without capacity-expansion decisions can be viewed as an operations-planning model for some type of system. For instance, the system

*Mighty River Power, Auckland, New Zealand, Kavi.Singh@mightyriver.co.nz
†University of Auckland, Auckland, New Zealand, a.philpott@auckland.ac.nz
‡Naval Postgraduate School, Monterey, California, kwood@nps.edu

could represent a single plant with multiple production facilities, each with fixed production capacity, and each producing multiple products. Given known production costs and product demands, the system manager must identify a minimum-cost, capacity-feasible, operational plan to meet those demands. Even this single-period, deterministic problem may be complicated, requiring a high level of modeling fidelity that incorporates both continuous and discrete decision variables.

The full planning problem is more complex, as it spans a multi-period horizon, must incorporate capacity-expansion decisions to accommodate demand growth, and faces uncertainty in demand, costs and possibly other parameters. (Strategic capacity-expansion decisions link the time periods, while operational decisions such as inventory levels do not.) An optimal capacity-expansion plan will (a) enable production to meet demand, and (b) minimize the expected costs of capacity expansion and production over the planning horizon.

We formulate the stochastic capacity-planning problem as a multi-stage, stochastic, mixed-integer program, with uncertain parameters represented through a standard scenario tree (for example, see Ruszczyński and Shapiro 2003, pp. 29-30). Given a finite number of scenarios and their probabilities, this problem can then be stated as a large-scale mixed-integer program (MIP), i.e., a "deterministic equivalent." That model can be solved, in theory, by a commercial optimization code. As we shall see, however, only the smallest real-world instances appear to be tractable with this approach.

We overcome this intractability by applying dynamic column generation to a Dantzig-Wolfe reformulation of the problem (Dantzig and Wolfe 1960, Appelgren 1969). (See Vanderbeck and Wolsey 1996 for a general solution method for Dantzig-Wolfe reformulations of integer programs; see Lübbecke and Desrosiers 2002 for an overview of applications.) The Dantzig-Wolfe master problem contains only binary variables and represents a simplified deterministic equivalent for the problem. The Dantzig-Wolfe subproblems are MIPs and generate columns for the linear-programming (LP) relaxation of the master problem at each node of the scenario tree. Because we use a special "split-variable formulation" of the original model (e.g., Lustig et al. 1991), the master problem exhibits structure that tends to yield strong LP relaxations, and even integer solutions. This makes the master problem, and thus the full problem, particularly easy to solve. When a facility can be expanded at most once over the planning horizon, model simplifications enhance performance. Specially structured subproblems admit stronger formulations that further enhance performance, and "duals stabilization" for the master problem (e.g., du Merle et al. 1999) dramatically reduces the number of columns generated, improving solution times for all problem variants.

The literature on stochastic capacity-planning problems is extensive: Luss (1982) and Van Mieghem (2003) present comprehensive surveys. Manne's seminal paper (Manne 1961), which models demand growth as an infinite-horizon stochastic process, stimulated much research on this area (e.g., Erlenkotter 1967, Manne 1967, Giglio 1970, Freidenfelds 1980). The typical infinite-horizon model cannot incorporate, however, the complex operational constraints that many real-world applications require.

More recent studies incorporate application-specific constraints. For instance, Sen et al. (1994) describe a two-stage model that integrates demand, capacity expansion, and budget constraints, although it incorporates only continuous capacity-expansion decisions and a single capacity-expansion technology. The authors solve the model with the sampling-based, stochastic-decomposition algorithm developed by Higle and Sen (1996).

The assumptions of a discrete probability distribution for uncertain parameters and a finite planning horizon mean that a set of scenarios can represent uncertain outcomes. This results in a standard mathematical-programming problem, but typically at very large scale. This framework enables the incorporation of a detailed operational submodel and many "strategic details," such as a variety of capacity-expansion technologies. Berman et al. (1994) present and solve a scenario-based multi-stage model with a single capacity-expansion technology. Chen et al. (2002) extend this concept to multiple capacity-expansion technologies. Both of these approaches assume continuous capacity-expansion decisions, however.

The modeling of fixed charges and economies of scale adds considerable complexity to a stochastic program. Chen et al. (2002) describe economies of scale for capacity expansions in their multi-stage model, but can only solve a model with linear costs. Eppen et al. (1989), Riis and

Andersen (2002), Riis and Lodahl (2002), and Barahona et al. (2005) all use integer variables to model such effects in the first-stage of a two-stage model. Although more complex, these problem still admit solution through Benders decomposition, because integer variables are confined to the first stage (Laporte and Louveaux 1993).

In recent years, increased computing power and advances in optimization techniques have made it possible to solve multi-stage stochastic integer-programming models. Ahmed et al. (2003) solve such problems with a special branch-and-bound procedure. Ahmed and Sahinidis (2003) and Huang and Ahmed (2005) propose approximation schemes that converge asymptotically to an optimal solution as the planning horizon lengthens.

Dynamic programming, though limited in its ability to integrate practical constraints, appears in a few recent applications. Laguna (1998) solves a two-stage model, which Riis and Anderson (2004) extend to multiple stages. Rajagopalan et al. (1998) present a multi-stage model with deterministic demand, but with uncertainty in the timing of the availability of new capacity-expansion technologies.

Unlike its continuous counterpart, a multi-stage stochastic program with integer variables in all stages does not allow a nested Benders decomposition. In theory, LP-based branch and bound can solve the deterministic equivalent for such a problem, although practical instances usually exceed the capabilities of today's software and hardware. But, new research on solving large deterministic integer programs (IPs) via column generation (e.g., Lübbecke and Desrosiers 2002), has spawned research on solving stochastic IPs with this technique: Lulli and Sen (2004) use branch and price (column generation plus branch and bound) for stochastic batch-sizing problems; Shiina and Birge (2004) use column generation to solve a unit-commitment problem under demand uncertainty; Damodaran and Wilhelm (2004) model high-technology product upgrades under uncertain demand; and Silva and Wood (2006) show how to solve a special class of two-stage problems by branch and price.

We propose a new column-generation approach for solving multi-stage, stochastic, capacity-planning problems: our master problem and subproblems differ substantially from those developed by other researchers. Importantly, the generality of our approach should lend itself to applications in a variety of industries.

Our research relates most closely to that of Ahmed et al. (2003), who present a multi-stage, stochastic, capacity-planning model that incorporates continuous as well as binary capacity-expansion decisions. Ahmed et al. disaggregate the continuous variables using the reformulation strategy of Krarup and Bilde (1977), which enables a strong problem formulation. Our approach differs in three major aspects:

1. We disaggregate binary capacity-expansion decisions rather than continuous ones.

2. Random demand parameters directly determine a facility's capacity requirements in Ahmed et al. (2003), and operational constraints are simple: total installed capacity must meet or exceed demand. (In theory, their model can accommodate more complicated operational constraints.) Our approach incorporates a general operational-level submodel, which meets demand using installed capacity however the modeler deems fit.

3. Ahmed et al. (2003) solve their MIP using an LP-based branch-and-bound algorithm that incorporates a heuristic for finding feasible solutions, whereas we use column generation.

The remainder of this paper develops as follows. Section 2 describes a general, multi-stage, stochastic, capacity-planning model with discrete capacity-expansion decisions, and formulates this problem as a deterministic-equivalent MIP. A reformulation, using the technique of "variable splitting," then enables a Dantzig-Wolfe decomposition whose master problem can have a stronger LP relaxation than the original formulation. Section 3 explores the strength of the decomposition. Section 4 presents a simplified split-variable formulation of a restricted model that allows at most one expansion of each facility over the planning horizon. Section 5 describes a capacity-planning problem for an electricity-distribution network, and uses that for computational studies: we solve both split-variable formulations by Dantzig-Wolfe decomposition, and compare against a potentially competitive, scenario-based decomposition scheme. Section 6 presents conclusions.

2 A Multi-Stage, Stochastic, Capacity-Planning Model

We follow Ahmed et al. (2003) and represent uncertainty using a *scenario tree* $\mathcal{T}$ over T decision *stages*. For simplicity, we think of these stages occurring at evenly spaced increments of time. The scenario tree at each stage t consists of a set of *nodes*, each of which represents a potential state of the world at time t. We denote the complete set of nodes of the scenario tree by $\mathcal{N}$.

For each node $n \in \mathcal{N}$, ϕ_n denotes the probability that the corresponding state of the world occurs. $\mathcal{T}_n$ denotes the *successors* of n which we define to include n itself. Thus, $\mathcal{T}_n$ denotes n plus all nodes "below" n in the tree. $\mathcal{P}_n$ denotes the set of all *predecessors* of n, which we define to include n itself. Thus, $\mathcal{P}_n$ denotes n plus all nodes "above" n in the tree. For any leaf node n in the tree, $\mathcal{P}_n$ defines a *scenario.* Stage 1 comprises only $n = 0$, the root node of $\mathcal{T}$, which is where all scenarios have the same realization (so $\phi_0 = 1$).

We now present the compact formulation of our stochastic capacity-planning model.

Data

$\mathbf{c}_n$	discounted cost vector for expanding capacity of system facilities at scenario-tree node n; this vector has dimension F
$\mathbf{q}_n$	discounted cost vector for operating the system at scenario-tree node n
$\mathbf{u}_0$	vector of initial capacities of facilities
V_n	matrix that converts operating decisions and/or activities into capacity utilization at scenario-tree node n
U_{hn}	non-negative matrix that converts capacity-expansion decisions at scenario-tree node h into available operating capacity at successor node $n \in \mathcal{T}_h$
$\mathcal{Y}_n$	feasible region for operating decisions at scenario-tree node n, with strategic capacity constraints omitted
ϕ_n	probability that the state of the world, defined by $(\mathbf{c}_n, \mathbf{q}_n, V_n, U_{hn}, \mathcal{Y}_n)$, occurs

Variables

Note: Capacity-expansion decisions could be complicated, because we might use various technologies to expand a facility f, and decisions in one time period could affect decisions in another. For simplicity of presentation, the model we describe here assumes that facility f can be expanded at scenario-tree node n or not, but can be expanded multiple times over the planning horizon.

$\mathbf{x}'_n$	vector of binary decisions for capacity expansion of facilities at scenario-tree node n. Specifically, $x'_{fn} = 1$ if facility f is expanded at node n, 0 otherwise.
$\mathbf{y}_n$	vector of continuous and/or discrete operating decisions at scenario-tree node n

Formulation

$$\textbf{CF}: \quad z^*_{\text{CF}} = \min \quad \sum_{n \in \mathcal{N}} \phi_n \left(\mathbf{c}_n^\top \mathbf{x}'_n + \mathbf{q}_n^\top \mathbf{y}_n \right) \tag{1}$$

$$\text{s.t.} \quad V_n \mathbf{y}_n \le \mathbf{u}_0 + \sum_{h \in \mathcal{P}_n} U_{hn} \mathbf{x}'_h \;\; \forall\, n \in \mathcal{N}, \tag{2}$$

$$\mathbf{y}_n \in \mathcal{Y}_n \;\; \forall\, n \in \mathcal{N}, \tag{3}$$

$$\mathbf{x}'_n \in \{0,1\}^F \;\; \forall\, n \in \mathcal{N}. \tag{4}$$

Note: By convention, if "AB" denotes a MIP, then "AB-LP" denotes that model's LP relaxation. Also, z^*_{AB} ($z^*_{\text{AB-LP}}$) denotes the optimal objective value to AB (AB-LP).

With the exception of ϕ_n, parameters subscripted by n in the model indicate potentially random quantities. Constraints (3) represent generic relationships between the operational variables $\mathbf{y}_n$, independent of all $\mathbf{x}'_h$. These constraints may also include random effects. For instance, our application includes, among other constructs, flow-balance constraints with random demands.

Constraints (2) ensure that adequate capacity exists to satisfy the operational requirements $V_n \mathbf{y}_n$ at node n. The matrices U_{hn} can model lags between when capacity-expansion decisions are executed and when capacity becomes available, and, more generally, can model capacity expansions that get larger or smaller over time after installation.

Constraints (2) and (3) can handle a general operational model at each node of the scenario tree. If a set of discrete capacity-expansion decisions adequately models continuous capacity expansions with fixed charges, the "(SCAP)" model of Ahmed et al. (2003) may be viewed as an instance of CF: this instance sets $\mathbf{q}_n = \mathbf{0}$ and defines constraints (3) as $\mathbf{y}_n = \mathbf{d}_n$, where $\mathbf{d}_n$ represents demands at node n.

Capacity-planning problems like CF typically have weak LP relaxations, and that makes them difficult to solve. The scale imposed by a scenario tree, especially when some components of $\mathbf{y}_n$ must be integer, exacerbates this difficulty. On the other hand, an optimization model over $\mathbf{y}_n \in \mathcal{Y}_n$, for a single node n, might be relatively easy to solve as a MIP. This structure suggests some form of decomposition.

2.1 A Split-variable Reformulation and Dantzig-Wolfe Decomposition

The classical approach to solving multi-stage stochastic linear programs uses nested Benders decomposition (e.g. Birge and Louveaux 1997, pp. 234-236), but integer variables in the subproblems make this impracticable. Our approach exploits Dantzig-Wolfe decomposition (Dantzig and Wolfe 1960) as extended to integer variables by Appelgren (1967). As we shall later discuss, a straightforward Dantzig-Wolfe decomposition of CF could lead to a master problem that provides a weak lower bound on z^*_{CF}. To address this difficulty, we apply decomposition to the following, *split-variable reformulation.*

$$
\begin{aligned}
\textbf{SV}: \quad z^*_{\mathrm{SV}} \;=\; \min \quad & \sum_{n\in\mathcal{N}} \phi_n \left(\mathbf{c}_n^\top \mathbf{x}'_n + \mathbf{q}_n^\top \mathbf{y}_n\right) && (5)\\
\text{s.t.} \quad & \mathbf{x}_{hn} \le \mathbf{x}'_h \;\; \forall\, n\in\mathcal{N},\; h\in\mathcal{P}_n, && (6)\\
& V_n \mathbf{y}_n \le \mathbf{u}_0 + \sum_{h\in\mathcal{P}_n} U_{hn}\mathbf{x}_{hn} \;\; \forall\, n\in\mathcal{N}, && (7)\\
& \mathbf{y}_n \in \mathcal{Y}_n \;\; \forall\, n\in\mathcal{N}, && (8)\\
& \mathbf{x}'_n \in \{0,1\}^F \;\; \forall\, n\in\mathcal{N}, && (9)\\
& \mathbf{x}_{hn} \in \{0,1\}^F \;\; \forall\, n\in\mathcal{N},\; h\in\mathcal{P}_n. && (10)
\end{aligned}
$$

The proof of the following proposition is obvious.

Proposition 1 Suppose $U_{hn} \ge 0 \,\forall\, n\in\mathcal{N},\; h\in\mathcal{P}_n$. Then $(\mathbf{x}'_n, \mathbf{y}_n)_{n\in\mathcal{N}}$ is feasible for CF if and only if there exists $(\mathbf{x}_{hn})_{n\in\mathcal{N},h\in\mathcal{P}_n}$ such that $(\mathbf{x}'_n, (\mathbf{x}_{hn})_{h\in\mathcal{P}_n}, \mathbf{y}_n)$ is feasible for SV. That is, CF and SV are essentially equivalent, and $z^*_{\mathrm{CF}} = z^*_{\mathrm{SV}}$. ■

In SV, for each node n, and for each of its predecessor nodes $h\in\mathcal{P}_n$, we define a new vector of *split variables* $\mathbf{x}_{hn}$ that indicate whether capacity expansions of facilities at scenario-tree node h contribute towards meeting the capacity requirement at node n. Here, one may think of $\mathbf{x}_{hn}$ as *requests* for capacity expansions at nodes $h\in\mathcal{P}_n$ which, if granted, will jointly satisfy capacity requirements at node n. Constraints (7) accumulate such requests. The variables $\mathbf{x}'_n$ determine actual capacity expansions at node n and can be viewed as capacity *grants*. Thus, the natural interpretation of constraints (6) is that variables $\mathbf{x}_{hn}$ request capacity and variables $\mathbf{x}'_h$ grant capacity. (As an alternative, looking "down the tree" from node n, one may split $\mathbf{x}'_n$ into variables $\mathbf{x}_{nh}$, which indicate whether a capacity-expansion decision at node n is exploitable, non-exclusively, at successor node h. This equivalent interpretation can be formalized by rewriting constraints (6) as $\mathbf{x}_{nh} \leq \mathbf{x}'_n \; \forall n \in \mathcal{N},\ h \in \mathcal{T}_n$.)

The split-variable reformulation has some similarities to the reformulation that Krarup and Bilde (1977) use to strengthen lot-sizing models, and to the variable-disaggregation-based reformulation used by Ahmed et al. (2003) for strengthening stochastic capacity-expansion models. Our model differs from those in that the split variables $\mathbf{x}_{hn}$ are binary and force binary capacity-expansion decisions $\mathbf{x}'_n$. In contrast, Ahmed et al. disaggregate continuous variables that force both continuous and binary capacity-expansion decisions. (We do not consider continuous capacity expansions.) With this disaggregation, demand provides an explicit lower bound on each facility's capacity, and this leads to tighter constraints and a stronger model.

The aim of our variable-disaggregation reformulation and solution methodology is to obtain a tighter approximation of the convex hull of the feasible solutions to an IP. In this general respect, our approach relates to cutting planes for 0-1 IPs, and particularly to the "lift-and-project" techniques described by Sherali and Adams (1990), Lovasz and Schrijver (1991), Balas, Ceria and Cornúejols (1993), Sherali, Adams and Driscoll (1998), and Lasserre (2001).

Variable splitting is a common technique used in stochastic programming to enable the decomposition of certain models. The conventional application of this approach decomposes a model by scenarios. The decomposed model can then be solved by a variety of approaches such as Lagrangian relaxation plus branch and bound (Carøe and Schultz 1999), the branch-and-fix coordination scheme (Alonso-Ayuso et al. 2003), or branch and price (Lulli and Sen 2004). Applied to CF, for each node $n \in \mathcal{N}$, this *scenario decomposition* would split variables $\mathbf{x}'_n$ and $\mathbf{y}_n$ into variables for the stage t associated with n and all scenarios s that are indistinguishable at n. Thus, the split variables here would be $\mathbf{x}'_{ts}$ and $\mathbf{y}_{ts}$. Because all split variables for a particular node n correspond to the same realization of the random parameters, their values must be equal: "non-anticipativity constraints" impose this condition (e.g., Birge and Louveaux 1997, p. 25). Our formulation uses relaxed, yet still valid non-anticipativity constraints (6). Lagrangian relaxation of these (relaxed) constraints enables a *nodal decomposition*, i.e., a decomposition by scenario-tree node.

Dentcheva and Römisch (2004) show that the duality gap achieved using Lagrangian relaxation to implement a scenario decomposition of a problem is no greater than that resulting from the nodal decomposition. This makes nodal decomposition less attractive. On the other hand, the number of non-anticipativity constraints in scenario decomposition can be huge, as they must be imposed on all variables at each non-leaf node. Furthermore, subproblem size increases proportionally to the number of stages. Indeed, for this reason, scenario decomposition becomes intractable for the class of capacity-planning problems that we study here. We confirm this with some computational experiments in section 5.

2.2 Dantzig-Wolfe Reformulation of SV

The capacity-expansion constraints (6) in SV link capacity expansions across successors of a scenario-tree node; these are "complicating constraints" to what are otherwise a set of simpler (sub)problems, one for each scenario-tree node n. (Subproblem n includes split variables $\mathbf{x}_{hn}$ indexed over $h \in \mathcal{P}_n$, but these variables are not linked across subproblems. They may be viewed, therefore, as alternative capacity-expansion choices for subproblem n alone.) Thus, we can use decomposition to partition the constraints of the split-variable formulation into two sets: the set of linking (complicating) constraints (6), and the set of constraints specific to scenario-tree node n, for which we define

$$\mathcal{X}_n = \left\{ (\mathbf{x}_{hn})_{h \in \mathcal{P}_n} \mid V_n \mathbf{y}_n \leq \mathbf{u}_0 + \sum_{h \in \mathcal{P}_n} U_{hn} \mathbf{x}_{hn},\ \mathbf{x}_{hn} \in \{0,1\}^F \ \forall\, h \in \mathcal{P}_n,\ \mathbf{y}_n \in \mathcal{Y}_n \right\}. \tag{11}$$

In what follows, we find it convenient in some situations to replace the notation $(\mathbf{x}_{hn})_{h \in \mathcal{P}_n}$ with the more "vector-oriented" notation $(\mathbf{x}_{nn} \cdots \mathbf{x}_{0n}) \equiv (\mathbf{x}_{nn}\ \mathbf{x}_{p(n)n}\ \mathbf{x}_{p(p(n))n} \cdots \mathbf{x}_{0n})$, where $p(n)$ denotes the direct predecessor of node n.

If we rewrite $\mathcal{X}_n$ as the finite, enumerated set $\mathcal{X}_n = \left\{ (\widehat{\mathbf{x}}_{nn} \cdots \widehat{\mathbf{x}}_{0n})^j \mid j \in \mathcal{J}_n \right\}$, we can then express any element of $\mathcal{X}_n$ through

$$(\mathbf{x}_{nn} \cdots \mathbf{x}_{0n}) = \sum_{j \in \mathcal{J}_n} (\widehat{\mathbf{x}}_{nn} \cdots \widehat{\mathbf{x}}_{0n})^j w_n^j,\ \sum_{j \in \mathcal{J}_n} w_n^j = 1,\ w_n^j \in \{0,1\} \ \forall\, j \in \mathcal{J}_n. \tag{12}$$

Each vector $(\widehat{\mathbf{x}}_{nn} \cdots \widehat{\mathbf{x}}_{0n})^j$ represents a collection of capacity-expansion requests from nodes $h \in \mathcal{P}_n$; satisfying these requests will ensure feasible system operation at node n. Hence, we refer to each collection as a *feasible expansion plan* (FEP).

Without loss of generality, we may assume that each FEP has associated with it at least one optimal operational plan $\widehat{\mathbf{y}}_n^j$, i.e., $\mathcal{J}_n$ simultaneously indexes FEPs and operational plans at scenario-tree node n. Thus, we can attach the operational costs $\mathbf{q}_n^\top \widehat{\mathbf{y}}_n^j$ to the w_n^j, and substitute for $(\mathbf{x}_{nn} \cdots \mathbf{x}_{0n})$ using (12) to obtain the Dantzig-Wolfe reformulation of SV. We denote this multi-scenario, column-oriented master problem as "SV-MP."

For each scenario node n, SV-MP contains a group of columns with index set $\mathcal{J}_n$. Each $j \in \mathcal{J}_n$ corresponds to an FEP. For simplicity, we assume that SV-MP is always feasible, i.e., $\mathcal{J}_n \neq \emptyset$ for any n. The formulation for SV-MP follows, with previously defined notation omitted:

Sets and Indices

$j \in \mathcal{J}_n$ FEPs for scenario-tree node n

Data

$\widehat{\mathbf{x}}_{hn}^j$ binary vector representing capacity-expansion requests at scenario-tree node h that form part of FEP j for node n

$\widehat{\mathbf{y}}_n^j$ operational plan used at scenario-tree node n with FEP j

Variables

$\mathbf{x}'_n$ binary decision vector for capacity expansion of facilities at scenario-tree node n

w_n^j 1 if FEP j is selected for scenario-tree node n, 0 otherwise

Formulation

$$\textbf{SV-MP}: \quad z^*_{\text{SV-MP}} = \min \sum_{n \in \mathcal{N}} \phi_n \mathbf{c}_n^\top \mathbf{x}'_n + \sum_{n \in \mathcal{N}} \sum_{j \in \mathcal{J}_n} \phi_n \mathbf{q}_n^\top \widehat{\mathbf{y}}_n^j w_n^j \qquad \text{[dual variables]} \quad (13)$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{J}_n} \widehat{\mathbf{x}}_{hn}^j w_n^j \leq \mathbf{x}'_h \quad \forall\, n \in \mathcal{N},\ h \in \mathcal{P}_n, \qquad [\boldsymbol{\pi}_{hn}] \quad (14)$$

$$\sum_{j \in \mathcal{J}_n} w_n^j = 1 \quad \forall\, n \in \mathcal{N}, \qquad [\boldsymbol{\mu}_n] \quad (15)$$

$$w_n^j \in \{0,1\} \quad \forall\, n \in \mathcal{N},\ j \in \mathcal{J}_n,$$

$$\mathbf{x}'_n \in \{0,1\}^F \quad \forall\, n \in \mathcal{N}.$$

SV-MP's objective function (13) minimizes expected capacity-expansion costs plus expected operational costs. Constraints (14) ensure that no FEP is chosen for any node without sufficient capacity having been installed (granted). "Convexity constraints" (15) select exactly one FEP for each scenario-tree node n.

Naturally, the cardinality of $\mathcal{J}_n$ in SV-MP will be huge, so we solve SV-MP using *dynamic column generation*. First, we create a restricted master problem (SV-RMP) which is identical to SV-MP, except that each set $\mathcal{J}_n$ now represents a modest-sized subset of all the FEPs at scenario-tree node n. We solve the LP relaxation of SV-RMP (SV-RMP-LP), which replaces $w_n^j \in \{0,1\}$ and $\mathbf{x}'_n \in \{0,1\}$ by $w_n^j \geq 0$ and $\mathbf{x}'_n \geq 0$, respectively. (The convexity constraints imply satisfaction of $w_n^j \leq 1$ and $\mathbf{x}'_n \leq 1$.) Given a solution to SV-RMP-LP, we extract dual variables, and attempt to generate new columns corresponding to FEPs with negative reduced costs, by solving optimization subproblems (e.g., Barnhart et al. 1998, Lübbecke and Desrosiers 2002).

The subproblem at scenario-tree node n is

$$\textbf{SV-SP}(n): \quad z^*_{\text{SV-SP}(n)} = \min \quad \phi_n \mathbf{q}_n^\top \mathbf{y}_n - \sum_{h \in \mathcal{P}_n} \widehat{\boldsymbol{\pi}}_{hn}^\top \mathbf{x}_{hn} - \widehat{\mu}_n \tag{16}$$

$$\text{s.t.} \quad V_n \mathbf{y}_n \leq \mathbf{u}_0 + \sum_{h \in \mathcal{P}_n} U_{hn} \mathbf{x}_{hn}, \tag{17}$$

$$\mathbf{y}_n \in \mathcal{Y}_n, \tag{18}$$

$$\mathbf{x}_{hn} \in \{0,1\} \;\; \forall\, h \in \mathcal{P}_n. \tag{19}$$

The cycle of solving SV-RMP-LP, extracting duals, and generating new columns repeats until no columns price favorably, i.e., no columns with negative reduced cost can be found and so we have solved SV-MP-LP to optimality. If the optimal solution to SV-MP-LP happens to be integer, then we have solved SV-MP, and thus SV. If not, we may resort to a branch-and-price algorithm, which generates columns within a branch-and-bound procedure (Savelsbergh 1997), or settle for solving the SV-RMP as an IP in the hope of obtaining a good integer solution.

3 Strength of the Decomposition

Dantzig-Wolfe decomposition of a large LP replaces the direct solution of a large-scale problem with a sequence of solutions of smaller, easier-to-solve problems, which are coordinated through a master problem. This indirect approach helps to solve certain large MIPs, also. Furthermore, decomposition of a MIP may improve solution efficiency by defining a master problem whose LP relaxation is stronger than the relaxation of the original MIP.

In the class of problems we consider, there are several possible approaches to constructing such a decomposition. The simplest approach decomposes CF directly, that is, without first applying the split-variable reformulation. This decomposition expresses feasible points for the master problem "CF-MP" as convex combinations of extreme points of $\mathrm{conv}(\mathcal{Y}_n)$, for $n \in \mathcal{N}$, the convex hulls of the subproblems' feasible regions $\mathcal{Y}_n$. If each subproblem is simply an LP, then $\mathrm{conv}(\mathcal{Y}_n) = \mathcal{Y}_n$ and $z^*_{\text{CF-MP-LP}} = z^*_{\text{CF-LP}}$. (Recall our convention: CF-MP-LP denotes the LP relaxation of the master problem for the Dantzig-Wolfe decomposition of CF.) On the other hand, if $\mathrm{conv}(\mathcal{Y}_n) \subset \mathcal{Y}_n$—for example, when the subproblem is an IP whose LP relaxation does not have integer extreme points—then the resulting master problem can have a tighter relaxation than that of the original MIP (Barnhart et al. 1998).

We have implemented a direct Dantzig-Wolfe decomposition of CF. In our test-problem instances, the mixed-integer subproblems for this decomposition need not have, but typically do have, naturally integer LP solutions. This results in no strengthening of the MIP through decomposition. For example in the smallest instance $z^*_{\text{CF-LP}} = z^*_{\text{CF-MP-LP}} = 201{,}017$, whereas the optimal integer solution has $z^*_{\text{CF-IP}} = 444{,}149$. (See Table 1, but note that we denote "CF-LP" by "CF-DE-LP" there.)

A second approach to decomposition would use the split-variable formulation, but with the integrality of variables $\mathbf{x}_{hn}$ relaxed. Let "SVR" denote this model. Typically, SVR does have a stronger LP relaxation than CF, and thus we expect $z^*_{\text{SVR-LP}} > z^*_{\text{CF-LP}}$. Furthermore, the split variables typically induce fractionation in the LP relaxation of any subproblem, so no subproblems have intrinsically integer solutions to their LP relaxations. Thus, we also expect $z^*_{\text{SVR-MP-LP}} > z^*_{\text{SVR-LP}}$.

The subproblems used in a Dantzig-Wolfe decomposition of SVR would have fewer binary variables than the SV subproblems, so they would normally solve faster: this is another benefit of the SVR formulation. But the columns returned to SVR-MP would represent the fractions of capacity-expansion options used in the subproblems—compare this to the 0s and 1s in the columns returned in the decomposition of SV—and thus we would expect SVR-MP-LP to be weaker than SV-MP-LP. Indeed, if (*a*) subproblem variables have no costs associated with them, and (*b*) the maximum fraction of capacity utilization in any subproblem is ρ, $0 < \rho < 1$, then it is easy to construct instances in which $z^*_{\text{SVR-MP-LP}} \leq \rho z^*_{\text{SV-MP-LP}}$.

Computational tests with Dantzig-Wolfe decomposition for SVR confirm the observations made above. For the "smallest problem instance" referred to above, $z^*_{\text{SVR-MP-LP}} = 363,079$. This is certainly better than $z^*_{\text{CF-MP-LP}} = 201{,}017$, but is still far from $z^*_{\text{SV-MP-LP}} = z^*_{\text{CF-IP}} = 444{,}149$. So, SVR-MP-LP may solve solver faster than SV-MP-LP—solution times are 27.6 seconds versus 55.9 seconds, respectively, for this instance—but after solving SVR-MP-LP, the non-zero optimality gap means that we would need to implement and apply a branch-and-price algorithm to guarantee an optimal solution. In contrast, SV-MP-LP has an integer solution in all problem instances we have tested, and thus the branch-and-price step is avoided.

We do find it remarkable that every one of our computational tests yields an optimal integer solution for SV-MP-LP. (Fractional intermediate solutions are not unusual.) Because the constraint matrix for this problem has coefficients that are either 0, 1 or −1 (when placed in standard form), it is easy to see that fixing the w_n^j to binary values leads to binary solutions for $\mathbf{x}'_n$ even when the latter variables are allowed to be continuous. Furthermore, for each node n in the scenario tree, the submatrix corresponding to the variables w_n^j has a perfect-matrix structure (Padberg 1974). These perfect submatrices prevent fractional solutions from occurring within a single block of variables w_n^j, $j \in \mathcal{J}_n$, thus making it less likely for fractional solutions to occur in SV-MP-LP. (See Ryan and Falkner 1988 for an account of this effect in set-partitioning problems.) On the other hand, the complete constraint matrix for SV-MP-LP may lack the perfect-matrix property because of constraints on the $\mathbf{x}'_n$ that link its (perfect) submatrices. Consequently, the interaction between these submatrices can give rise to fractional solutions as we show in Section 5.

4 At Most One Capacity Expansion of a Facility

The general model SV allows the expansion of a facility's capacity more than once over the planning horizon. However, in some industries, planning for multiple expansions makes little sense, because associated fixed charges are large, or "setups" have highly undesirable side effects.

This section therefore studies a version of SV that restricts each facility to being expanded at most once over the planning horizon. With this change, SV becomes:

$$\textbf{SV1}': \quad z^*_{\text{SV1}'} = \min \sum_{n \in \mathcal{N}} \phi_n \left(\mathbf{c}_n^\top \mathbf{x}'_n + \mathbf{q}_n^\top \mathbf{y}_n \right) \tag{20}$$

$$\text{s.t.} \quad \mathbf{x}_{hn} \le \mathbf{x}'_h \ \ \forall n \in \mathcal{N},\ h \in \mathcal{P}_n, \tag{21}$$

$$V_n \mathbf{y}_n \le \mathbf{u}_0 + \sum_{h \in \mathcal{P}_n} U_{hn} \mathbf{x}_{hn} \ \ \forall n \in \mathcal{N}, \tag{22}$$

$$\sum_{h \in \mathcal{P}_n} \mathbf{x}'_h \le \mathbf{1} \ \ \forall n \in \mathcal{N}, \tag{23}$$

$$\mathbf{y}_n \in \mathcal{Y}_n \ \ \forall n \in \mathcal{N}, \tag{24}$$

$$\mathbf{x}'_n \in \{0,1\}^F \ \ \forall n \in \mathcal{N}, \tag{25}$$

$$\mathbf{x}_{hn} \in \{0,1\}^F \ \ \forall n \in \mathcal{N},\ h \in \mathcal{P}_n. \tag{26}$$

The model SV1$'$ simplifies further if we assume that the matrix U_{hn} is deterministic and does not evolve with the scenario tree, that is, $U_{hn} = U \ \forall n \in \mathcal{N},\ h \in \mathcal{P}_n$. In this case, we can transform SV1$'$ into an equivalent formulation with fewer variables:

$$\textbf{SV1}: \quad z^*_{\text{SV1}} = \min \sum_{n \in \mathcal{N}} \phi_n \left(\mathbf{c}_n^\top \mathbf{x}'_n + \mathbf{q}_n^\top \mathbf{y}_n \right) \tag{27}$$

$$\text{s.t.} \quad \mathbf{x}_n \le \sum_{h \in \mathcal{P}_n} \mathbf{x}'_h \ \ \forall n \in \mathcal{N}, \tag{28}$$

$$V_n \mathbf{y}_n \le \mathbf{u}_0 + U \mathbf{x}_n \ \ \forall n \in \mathcal{N}, \tag{29}$$

$$\sum_{h \in \mathcal{P}_n} \mathbf{x}'_h \le \mathbf{1} \ \ \forall n \in \mathcal{N}, \tag{30}$$

$$\mathbf{y}_n \in \mathcal{Y}_n \ \ \forall n \in \mathcal{N}, \tag{31}$$

$$\mathbf{x}'_n \in \{0,1\}^F \ \ \forall n \in \mathcal{N}, \tag{32}$$

$$\mathbf{x}_n \in \{0,1\}^F \ \ \forall n \in \mathcal{N}. \tag{33}$$

The following proposition implies the equivalence of SV1$'$ and SV1.

Proposition 2 Suppose that $U_{hn} = U \ \forall n \in \mathcal{N},\ h \in \mathcal{P}_n$. Then, there exists $(\mathbf{x}_{hn})_{h \in \mathcal{P}_n}$ with $(\mathbf{x}'_n, (\mathbf{x}_{hn})_{h \in \mathcal{P}_n}, \mathbf{y}_n)$ being feasible for SV1$'$ if and only if there exists $\mathbf{x}_n$ such that $(\mathbf{x}'_n, \mathbf{x}_n, \mathbf{y}_n)$ is feasible for SV1.

Proof. Suppose $(\mathbf{x}'_n, (\mathbf{x}_{hn})_{h\in\mathcal{P}_n}, \mathbf{y}_n)$ is feasible for SV1$'$. Let $\mathbf{x}_n = \sum_{h\in\mathcal{P}_n} \mathbf{x}_{hn}$. To show that $(\mathbf{x}'_n, \mathbf{x}_n, \mathbf{y}_n)$ is feasible for SV1, it suffices to check that constraints (28), (29) and (33) are satisfied. Constraints (21) imply (28), and constraints (22) give (29). Moreover, $\mathbf{x}_n$ is binary because of (21) and (23).

Conversely, if $(\mathbf{x}'_n, \mathbf{x}_n, \mathbf{y}_n)$ is feasible for SV1, then let $\mathbf{x}_{hn} = \mathbf{x}'_h$ for all $h \in \mathcal{P}_n$. All constraints of SV1$'$ hold trivially, except for (22). These constraints are satisfied because

$$V_n\mathbf{y}_n \le \mathbf{u}_0 + U\mathbf{x}_n \le \mathbf{u}_0 + U\sum_{h\in\mathcal{P}_n}\mathbf{x}'_h = \mathbf{u}_0 + U\sum_{h\in\mathcal{P}_n}\mathbf{x}_{hn}.$$

This completes the proof. ■

We can now formulate a Dantzig-Wolfe decomposition of SV1, analogous to that of section 2.2, by defining

$$\mathcal{X}_n = \left\{\mathbf{x}_n \mid V_n\mathbf{y}_n \le \mathbf{u}_0 + U\mathbf{x}_n,\ \mathbf{x}_n \in \{0,1\}^F,\ \mathbf{y}_n \in \mathcal{Y}_n\right\},$$

and by expressing $\mathbf{x}_n$ through $\widehat{\mathbf{x}}^j_n$, $j \in \mathcal{J}_n$, which denote the enumerated feasible solutions in $\mathcal{X}_n$:

$$\mathbf{x}_n = \sum_{j\in\mathcal{J}_n}\widehat{\mathbf{x}}^j_n w^j_n,\ \sum_{j\in\mathcal{J}_n} w^j_n = 1,\ w^j_n \in \{0,1\}\ \forall\, j \in \mathcal{J}_n.$$

This gives a simpler master problem

$$\begin{aligned}
\textbf{SV1-MP}:\quad z^*_{\text{SV1-MP}} \;=\; \min\quad & \sum_{n\in\mathcal{N}}\phi_n\mathbf{c}_n^\top\mathbf{x}'_n + \sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{J}_n}\phi_n\mathbf{q}_n^\top\widehat{\mathbf{y}}^j_n w^j_n && \text{[dual variables]} && (34)\\
\text{s.t.}\quad & \sum_{j\in\mathcal{J}_n}\widehat{\mathbf{x}}^j_n w^j_n \le \sum_{h\in\mathcal{P}_n}\mathbf{x}'_h \quad \forall\, n\in\mathcal{N}, && [\boldsymbol{\pi}_n] && (35)\\
& \sum_{h\in\mathcal{P}_n}\mathbf{x}'_h \le \mathbf{1}\ \ \forall\, n\in\mathcal{N}, && && (36)\\
& \sum_{j\in\mathcal{J}_n} w^j_n = 1\ \forall\, n\in\mathcal{N}, && [\mu_n] && (37)\\
& w^j_n \in \{0,1\}\ \forall\, n\in\mathcal{N},\ \ j\in\mathcal{J}_n, && && \\
& \mathbf{x}'_n \in \{0,1\}^F\ \forall\, n\in\mathcal{N},
\end{aligned}$$

and a simpler subproblem

$$\begin{aligned}
\textbf{SV1-SP}(n):\quad z^*_{\text{SV1-SP}(n)} \;=\; \min\quad & \phi_n\mathbf{q}_n^\top\mathbf{y}_n - \widehat{\boldsymbol{\pi}}_n^\top\mathbf{x}_n - \widehat{\mu}_n && (38)\\
\text{s.t.}\quad & V_n\mathbf{y}_n \le \mathbf{u}_0 + U\mathbf{x}_n, && (39)\\
& \mathbf{y}_n \in \mathcal{Y}_n, && (40)
\end{aligned}$$

$$\mathbf{x}_n \in \{0,1\}^F. \quad (41)$$

Recall that SV-SP(n) includes binary variables $\mathbf{x}_{hn}$ for all nodes $h \in \mathcal{P}_n$. In contrast, SV1-SP(n) incorporates only binary variables $\mathbf{x}_n$. Thus, the number of binary variables in SV1-SP(n) reduces by a factor of $|\mathcal{P}_n|$, which can make this subproblem easier to solve. The reader will note that constraints (36) for non-leaf nodes are redundant. However, we include these constraints in SV1-MP because, for reasons we cannot explain, the Dantzig-Wolfe algorithm tends to solve faster that way.

5 Computational Results

This section applies the SV and SV1 formulations to instances of a model for planning the capacity expansion of an electricity-distribution network subject to uncertain demand. The details of this class of models have been described in Singh (2004), so we give only a brief description. A distribution network is the local, low-voltage part of the electricity system that connects customers to the long-distance, high-voltage transmission system which, in turn, connects to generating plants. The distribution network may be viewed as connecting to the transmission system, via a substation, at a single point or "source." (In reality, it may connect to several points.) For each demand realization (i.e., at each scenario-tree node), the distribution network of interest must operate in a radial (tree) configuration, so that power flows from the source to each demand point along a unique path of power lines. Typically, each line has a switch at either end that can be open or closed and, although the full network has an underlying mesh structure, it is always operated in a radial configuration by opening and closing these switches.

We model the underlying mesh structure of the network as a connected, undirected graph $G = (\mathcal{V}, \mathcal{E})$ consisting of a set of *vertices* $i \in \mathcal{V}$ and a set of *edges* $e \in \mathcal{E}$ such that $e = (i, j)$, where $i, j \in \mathcal{V}$ and $i \neq j$. A vertex represents either a supply point, a demand point, a junction, or a switching point; an edge represents (a) a route along which a line connecting the adjacent vertices has already been installed and may or may not be replaced with a higher-capacity line (or "cable"), or (b) it represents a new route along which a new line may be installed. In case (a), the initial capacity of an edge equals the capacity of the line installed in the corresponding route; in case (b) the initial capacity is zero. (References to an edge refer to the corresponding line.)

Power may flow in either direction along a line, and to model this we create a directed version of G, denoted $G' = (\mathcal{V}, \mathcal{K})$. The set of vertices in G' is the same as in G, but $\mathcal{K}$ replaces each edge $e = (i, j)$ with two antiparallel, directed *arcs* (i, j) and (j, i). For edge $e = (i, j)$, we define $\mathcal{K}_e = \{(i, j), (j, i)\}$, so we may also write $\mathcal{K} = \cup_{e \in \mathcal{E}} \{\mathcal{K}_e\}$. We model the power source as a single vertex $i_0 \in \mathcal{V}$. (Note: By allowing negative arc flows, we could use undirected-graph constructs in this formulation. However, the directed-graph formulation appears to have a stronger LP relaxation; see Magnanti and Wolsey 1995.)

We now present a compact formulation of the stochastic multi-stage capacity-planning model for radial distribution networks.

Sets and Indices

$i \in \mathcal{V}$	vertices in the distribution network.
$e \in \mathcal{E}$	edges in the network
$k \in \mathcal{K}$	antiparallel arcs corresponding to edges in $\mathcal{E}$
$k \in \mathcal{K}_e$	pair of antiparallel arcs representing edge e
$l \in \mathcal{L}_{en}$	technologies (power cables) available for expanding capacity of edge e at scenario-tree node n
i_0	power-source vertex ($i_0 \in \mathcal{V}$)

Data [units]

A_{ik}	1 if $k = (j, i)$, -1 if $k = (i, j)$, and 0 otherwise
C_{eln}	discounted cost of expanding capacity of edge e using technology l at scenario-tree node n [\$]
D_{in}	demand ("load") at vertex i at scenario-tree node n [MVA]
ϕ_n	probability associated with scenario-tree node n
U_{e0}	initial capacity of edge e [MVA]
U_{elhn}	capacity on edge e gained from installing technology l at scenario-tree node h which becomes available for use at successor node n [MVA]
$\bar{U}_e$	upper bound representing the maximum possible power flow on edge e [MVA]

Variables [units]

x'_{eln}	1 if technology l is chosen for expanding edge e at scenario-tree node n, and 0 otherwise
y_{kn}	non-negative power flow on arc k at scenario-tree node n [MVA]
r_{kn}	1 if arc k is active (part of the operating radial configuration) at scenario-tree node n, and 0 otherwise

Formulation

$$\text{CF-E:} \quad z^*_{\text{CF-E}} = \min \quad \sum_{n \in \mathcal{N}} \phi_n \sum_{e \in \mathcal{E}} \sum_{l \in \mathcal{L}_{en}} C_{eln} x'_{eln} \tag{42}$$

$$\text{s.t.} \quad y_{kn} \le U_{e0} + \sum_{h \in \mathcal{P}_n} \sum_{l \in \mathcal{L}_{en}} U_{elhn} x'_{elh} \;\; \forall\, e \in \mathcal{E},\; k \in \mathcal{K}_e,\; n \in \mathcal{N} \tag{43}$$

$$\sum_{k \in \mathcal{K}} A_{ik} y_{kn} = D_{in} \;\; \forall\, i \in \mathcal{V},\; n \in \mathcal{N}, \tag{44}$$

$$\sum_{k \in \mathcal{K}: A_{ik}=1} r_{kn} = 1 \;\; \forall\, i \in \mathcal{V} \backslash \{i_0\},\; n \in \mathcal{N}, \tag{45}$$

$$\sum_{k \in \mathcal{K}} r_{kn} = |\mathcal{V}| - 1 \;\; \forall\, n \in \mathcal{N}, \tag{46}$$

$$y_{kn} \le \bar{U}_e r_{kn} \;\; \forall\, e \in \mathcal{E},\; k \in \mathcal{K}_e,\; n \in \mathcal{N}, \tag{47}$$

$$y_{kn} \ge 0 \;\; \forall\, k \in \mathcal{K},\; n \in \mathcal{N}, \tag{48}$$

$$r_{kn} \in \{0,1\} \;\; \forall\, k \in \mathcal{K},\; n \in \mathcal{N}, \tag{49}$$

$$x'_{eln} \in \{0,1\} \;\; \forall\, e \in \mathcal{E},\; l \in \mathcal{L}_{en},\; n \in \mathcal{N}. \tag{50}$$

The objective function (42) minimizes the expected discounted cost of capacity expansions, because operational costs are zero. Constraints (43) ensure that the flow through any edge does not exceed the edge's total capacity (initial plus additional capacity acquired at predecessor scenario-tree nodes); these constraints correspond to constraints (2) in CF. Note that $U_{e0} = 0$ for potential routes. Constraints (44) represent the standard Kirchhoff current-balance (flow-balance) constraints at each vertex i. Constraints (45) and (46) enforce the requirement of a radial operating configuration. Constraints (47) ensure that flow is permitted on an arc k if and only if arc k is part of the radial configuration in scenario-tree node n, i.e., if and only if $r_{kn} = 1$. Observe that constraints (44-49) are the operational constraints corresponding to constraints (3) in CF.

The binary variables, and the capacity-expansion and radial-configuration constraints in CF-E result in a difficult MIP. The split-variable reformulation and Dantzig-Wolfe decomposition approach leads to subproblems SV-SP(n) (or SV1-SP(n)) that also incorporate such variables and constraints, and are therefore challenging, albeit simpler, MIPs in their own right. A "super-network model" for any subproblem provides a stronger LP relaxation for that subproblem. This model replaces certain sets of vertices and edges with simpler constructs involving "super-vertices" and "super-edges" which reduces the number of binary variables, and exploits some problem-specific valid inequalities; see Singh et al. (2007) for details. We make use of this strengthened formulation in all of the tests reported here.

We report results for seven problem instances, which differ by the number of stages in a binary scenario tree (five problems) and the number of stages in a ternary scenario tree (two problems). All problem instances derive from data for an actual distribution network in Auckland, New Zealand. The network comprises 152 vertices, most of which are demand points, and 182 edges. For this network, the distribution company provided data that define:

1. network connectivity in terms of existing and potential routes (edges) and vertices;
2. the current demand D_{i0} at each vertex i;
3. the capacity of each existing route;
4. the capacity made available on each route by installing a new line, if allowed (only a single type of cable is ever specified, so at most one technology and thus capacity is available for capacity expansion of any route); and
5. the cost of installing each new cable.

All problem instances have a single capacity-expansion technology (a cable) and are designed so that an optimal solution always exists in which no edge is expanded more than once over the planning horizon. This allows us to apply both SV1 and SV formulations and make direct comparisons.

Demand is the only stochastic parameter in our problems. For any problem instance, each demand scenario occurs with equal probability. In a problem instance with a binary scenario tree, each scenario-tree node, except the root node, is randomly allocated a demand growth factor α_n, $1 < \alpha_n < 2$. Let the current demand D_{i0} for each vertex i correspond to node-0 demands, i.e., root-node demands, and recall that $p(n)$ denotes the direct-predecessor node of each non-root, scenario-tree node. Then, the demands at all other scenario-tree nodes are computed as follows:

For (each stage $t = 2$ to T){

For (each scenario-tree node n in stage t){

For (each vertex $i \in V$) $D_{in} \leftarrow \alpha_n D_{i,p(n)}$;

}

}

Demands in the ternary scenario tree are computed in a similar fashion, except that a growth factor α_{in}, $1 < \alpha_{in} < 2$, is randomly generated for each vertex i and each scenario-tree node except the root node, and "$D_{in} \leftarrow \alpha_n D_{i,p(n)}$" above is replaced by "$D_{in} \leftarrow \alpha_{in} D_{i,p(n)}$." Observe that the ternary scenario trees provide a sterner test for our algorithms as the demand does not grow at a uniform rate for each vertex.

We have implemented and tested our algorithms on a desktop computer with a Pentium IV 2.6 GHz processor and 1 GB of RAM. We generate all models, and implement all decomposition algorithms within the Mosel algebraic modeling system, version 1.24, from Dash Optimization. The restricted master problems are solved, as LPs or IPs, with Xpress Optimizer, version 14.24, also from Dash Optimization, but the MIP subproblems and the deterministic-equivalent models are solved with CPLEX, version 9.0 from ILOG, Inc.

Solver settings remain constant throughout all tests. All MIPs are solved with default parameter settings except that Gomory cuts are turned off and a moderate level of probing is used (CPX_PARAM_PROBE = 2). All subproblems are solved to optimality and the deterministic-equivalent problems are solved with a relative optimality tolerance of 1.0%. A time limit of 7,200 seconds is applied in some tests.

Observe that any (nontrivial) instance of RMP-LP will be infeasible unless one feasible column (FEP) exists for each scenario-tree node. We could use the classical "Phase I" approach to find an initial feasible solution, but it is simpler to guarantee such a solution by seeding the master problem with one FEP for each scenario-tree node. Except for trivially infeasible problems, an FEP for each node that requires all possible capacity expansions will surely be feasible, so those generate our initial columns.

Any such FEP translates into a column in RMP-LP that has coefficients of 1 in the capacity-expansion constraints for each facility, a coefficient of 1 in the convexity constraint for the corresponding scenario-tree node, and 0s elsewhere. Note that our application imposes no operational costs, so these initial columns, as well as the columns generated later, all have cost coefficients of 0.

Given an initial feasible solution, the basic decomposition algorithms for SV and SV1 repeat the following *major iteration* until no column prices favorably:

Solve the master problem for a new set of dual variables;
For (each stage $t = 1$ to T) {
    For (each scenario-tree node n in stage t) {
        Solve the subproblem for node n given the current set of dual variables;
        If (the corresponding master-problem column prices favorably)
            Add the column to the master problem;
    }
}

We note that the master problem could be re-solved after each new column is added. Although the master problem is only a linear program, and re-solving may actually reduce the number of major iterations required to solve the problem, we have not found the extra computational effort to be worthwhile. Furthermore, defining a major iteration this way simplifies computation of lower bounds on z^*, as discussed below.

The scenario-decomposition algorithm works similarly, except that (a) the master problem solves for the optimal Lagrangian multipliers for the scenario decomposition, and (b) each scenario subproblem is solved once in each major iteration (rather than each scenario-tree subproblem). In practice, our nodal decomposition does not need to be embedded in a branch-and-bound algorithm, so we have not implemented a branch-and-bound stage for the scenario decomposition. That is, we are only solving the LP relaxation of the scenario decomposition.

Let "SVx" denote either SV or SV1. While solving SVx by Dantzig-Wolfe decomposition, a lower bound $\underline{z}_{\text{SVx-MP-LP}}$ on $z^*_{\text{SVx-MP-LP}}$ is readily available. In particular, using the arguments in Wolsey (1998, p. 189), it is easy to show that

$$\underline{z}_{\text{SVx-MP-LP}} = z_{\text{SVx-MP-LP}} + \sum_{n \in \mathcal{N}} \delta_n \leq z^*_{\text{SVx-MP-LP}}, \tag{51}$$

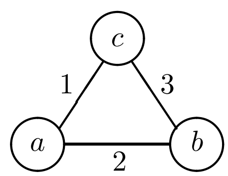
where $z_{\text{SVx-MP-LP}}$ and δ_n denote the optimal objective values for RMP-LP and SP(n) for SVx at the current iteration, respectively. Note that this lower bound is only valid when "full pricing" is invoked, that is, after a major iteration has been completed, and all subproblems SP(n), $n \in \mathcal{N}$ have been solved to optimality using the same set of dual variables. At any particular iteration, it is easy to compute an upper bound $\overline{z}$ on $z^* = z^*_{\text{CF}}$ by solving the integer RMP (RMP-IP) with the existing set of columns, assuming this is feasible. We define the (relative) optimality gap for the master problem, "MP-Gap," as $100\% \times (\overline{z} - \underline{z}_{\text{SVx-MP-LP}})/\underline{z}_{\text{SVx-MP-LP}}$. MP-Gap gives an optimality check on our algorithm which can be used to terminate the Dantzig-Wolfe decomposition early if it reaches a tolerable level.

Observe that when the solution to RMP-LP is fractional, we must solve RMP-IP to obtain $\overline{z}$, which can be expensive if carried out after every major iteration. Thus, for the overall efficiency of the algorithm, the number of such checks should be limited. As an empirical rule, when allowing a non-zero optimality gap, we start checking MP-Gap at the first iteration when the gap between the RMP-LP objective and the lower bound, "LP-Gap," reaches 80% of the prespecified termination tolerance. For instance, for a termination tolerance of 5%, we start checking MP-Gap when LP-Gap reaches 4%. After the first check, we re-solve RMP-IP with a branch-and-bound algorithm only when RMP-LP yields fractional solutions for five consecutive iterations. We demonstrate the effect of termination tolerances on solution times later.

Unfortunately our Dantzig-Wolfe master problems suffer from severe dual degeneracy, and this slows convergence of a conventionally implemented decomposition algorithm. To improve convergence speed, we apply "duals stabilization" to the sequence of RMP-LP solutions, and compare two different methods: du Merle et al. (1999) describe the first, which we call "du Merle stabilization"; the other simply generates interior-point dual solutions by solving RMP-LP using an interior-point algorithm. The latter technique has been used by a number of researchers studying column-generation algorithms, e.g., Desrosiers et al. (2002). For lack of a better phrase, we call this technique "interior-point duals stabilization."

The optimal solutions of SV-MP-LP are invariably integral in our test problems. Consequently, we have not required a full branch-and-price solution procedure. It is interesting to note, however, that it is possible to devise problem instances with fractional optimal solutions. Such an example can be constructed in a network with three edges that connect a single supply vertex to two demand vertices; see Figure 1. The problem has two stages and three equally likely scenarios. In the first stage (scenario-tree node 0) the demand is zero, and in each scenario in the second stage a different vertex is chosen to be the supply vertex and the others each have demand of 1. We assume that a, b, and c are the supply vertices in scenario-tree nodes 1, 2, and 3 respectively. (By adding a dummy supply vertex it is easy to modify this example so that the supply vertex remains constant across scenarios, as we assume in CF-E.)

Figure 1: An example network with a fractional optimal solution for SV-MP-LP.



To construct a fractional optimal solution for this network, we suppose that the capacity on each edge can be expanded using two technologies $l = 1, 2$ with increments of 1 or 2 units, but technology 1 is available only in stage 1, and technology 2 is only available in stage 2. In other words, the cost of expansion in stage 1 is $C_{e10} = 1$, $C_{e20} = \infty$, for each edge e, and in stage 2 this cost is $C_{e1n} = \infty$, $C_{e2n} = 2$, for each edge e, and scenario node $n = 1, 2, 3$. A fractional optimal solution for CF has $x_{eln} = 0$ everywhere, except $x_{110} = x_{210} = x_{310} = 0.5$, and

$$\begin{array}{ll} x_{221} = 0.5 & \text{(in scenario 1 where } a \text{ is the supply node);} \\ x_{322} = 0.5 & \text{(in scenario 2 where } b \text{ is the supply node);} \\ x_{123} = 0.5 & \text{(in scenario 3 where } c \text{ is the supply node).} \end{array}$$

The total (expected) cost of this plan is 2.5. In each scenario this solution provides a capacity of 1.5 for an edge incident on the supply vertex and capacity of 0.5 on the other edges so that a feasible flow exists in the second stage.

In terms of the variables of SV-MP-LP, each scenario-node apart from the root node generates three FEPS. For scenario-node 1 we obtain:

$$(\widehat{\mathbf{x}}_{11}\ \widehat{\mathbf{x}}_{01})^1 = (0,\ 0,\ 0,\ 1,\ 1,\ 0\ )$$
$$(\widehat{\mathbf{x}}_{11}\ \widehat{\mathbf{x}}_{01})^2 = (1,\ 0,\ 0,\ 0,\ 0,\ 1\ )$$
$$(\widehat{\mathbf{x}}_{11}\ \widehat{\mathbf{x}}_{01})^3 = (0,\ 1,\ 0,\ 0,\ 0,\ 1\ )$$

corresponding to expansion of edges 1 and 2 in stage 1; expansion of edge 1 (by 2 units) in stage 2 and edge 3 in stage 1; and expansion of edge 2 in stage 2 and edge 3 in stage 1. Similarly for scenario-node 2 we obtain

$$(\widehat{\mathbf{x}}_{12}\ \widehat{\mathbf{x}}_{02})^1 = (0,\ 0,\ 0,\ 0,\ 1,\ 1\ )$$
$$(\widehat{\mathbf{x}}_{12}\ \widehat{\mathbf{x}}_{02})^2 = (0,\ 1,\ 0,\ 1,\ 0,\ 0\ )$$
$$(\widehat{\mathbf{x}}_{12}\ \widehat{\mathbf{x}}_{02})^3 = (0,\ 0,\ 1,\ 1,\ 0,\ 0\ )$$

and for scenario-node 3 we obtain

$$(\widehat{\mathbf{x}}_{13}\ \widehat{\mathbf{x}}_{03})^1 = (0,\ 0,\ 0,\ 1,\ 0,\ 1\ )$$
$$(\widehat{\mathbf{x}}_{13}\ \widehat{\mathbf{x}}_{03})^2 = (1,\ 0,\ 0,\ 0,\ 1,\ 0\ )$$
$$(\widehat{\mathbf{x}}_{13}\ \widehat{\mathbf{x}}_{03})^3 = (0,\ 0,\ 1,\ 0,\ 1,\ 0\ )$$

The optimal fractional solution chooses $w_1^1 = w_3^1 = 0.5$, $w_1^2 = w_3^2 = 0.5$, $w_1^3 = w_2^3 = 0.5$, which is feasible for SV-MP-LP when considered along with $x'_{110} = x'_{210} = x'_{310} = 0.5$, and

$$x'_{221} = 0.5 \qquad \text{(in scenario 1 where } a \text{ is the supply node);}$$
$$x'_{322} = 0.5 \qquad \text{(in scenario 2 where } b \text{ is the supply node);}$$
$$x'_{123} = 0.5 \qquad \text{(in scenario 3 where } c \text{ is the supply node).}$$

The optimal integer solution for this example expands an (arbitrary) edge in the first stage by one unit. If this edge happens to be incident on the supply vertex in the second stage then the other incident edge is expanded by two units. Otherwise an edge that is incident on the supply vertex is (arbitrarily) chosen and expanded by two units. It is easily verified that this has optimal (expected) cost 3.

In addition to computational results for Dantzig-Wolfe (nodal) decomposition, we have tested a scenario-decomposition approach applied to CF. To define this, we let $\mathcal{T}$ denote the set of decision stages, $\mathcal{T}_t = \{1, ..., t\}$, and $s \in \mathcal{S}$ the set of scenarios, each of which occurs with probability ϕ_s. We now define ω_{ts} to be a realization of a random variable in scenario s at stage t, and $\mathcal{W}_{st}$ $= \{\omega_{s1}, \omega_{s2}, ..., \omega_{st}\}$ to be the information available in scenario s at stage t. This gives the following scenario-based formulation (which is also a type of split-variable formulation):

$$\textbf{CF-SD:} \quad \min \quad \sum_{s \in \mathcal{S}} \phi_s \sum_{t \in \mathcal{T}} \left(\mathbf{c}_{st}^\top \mathbf{x}'_{st} + \mathbf{q}_{st}^\top \mathbf{y}_{st} \right) \tag{52}$$

$$\text{s.t.} \quad V_{st}\mathbf{y}_{st} \leq \mathbf{u}_0 + \sum_{\tau \in \mathcal{T}_t} U_{s\tau}\mathbf{x}'_{s\tau} \ \ \forall\, s \in \mathcal{S},\ t \in \mathcal{T}, \tag{53}$$

$$\mathbf{x}'_{st} = \mathbf{x}'_{\lambda t} \ \ \forall\, s \in \mathcal{S},\ \lambda \in \mathcal{S},\ t \in \mathcal{T} : \mathcal{W}_{st} = \mathcal{W}_{\lambda t}, \tag{54}$$

$$\mathbf{y}_{st} = \mathbf{y}_{\lambda t} \ \ \forall\, s \in \mathcal{S},\ \lambda \in \mathcal{S},\ t \in \mathcal{T} : \mathcal{W}_{st} = \mathcal{W}_{\lambda t}, \tag{55}$$

$$\mathbf{y}_{st} \in \mathcal{Y}_{st} \ \ \forall\, s \in \mathcal{S},\ t \in \mathcal{T}, \tag{56}$$

$$\mathbf{x}'_{st} \in \{0,1\}^F \ \ \forall\, s \in \mathcal{S},\ t \in \mathcal{T}. \tag{57}$$

With the restriction of "at most one capacity-expansion," this model becomes

$$\textbf{CF1-SD}: \quad \min \quad \sum_{s \in \mathcal{S}} \phi_s \sum_{t \in \mathcal{T}} \left(\mathbf{c}_{st}^\top \mathbf{x}'_{st} + \mathbf{q}_{st}^\top \mathbf{y}_{st} \right)$$

$$\text{s.t.} \quad V_{st}\mathbf{y}_{st} \leq \mathbf{u}_0 + U \sum_{\tau \in \mathcal{T}_t} \mathbf{x}'_{s\tau} \ \ \forall\, s \in \mathcal{S},\ t \in \mathcal{T}, \tag{58}$$

$$\mathbf{x}'_{st} = \mathbf{x}'_{\lambda t} \ \ \forall\, s \in \mathcal{S},\ \lambda \in \mathcal{S},\ t \in \mathcal{T} : \mathcal{W}_{st} = \mathcal{W}_{\lambda t}, \tag{59}$$

$$\mathbf{y}_{st} = \mathbf{y}_{\lambda t} \ \ \forall\, s \in \mathcal{S},\ \lambda \in \mathcal{S},\ t \in \mathcal{T} : \mathcal{W}_{st} = \mathcal{W}_{\lambda t}, \tag{60}$$

$$\sum_{t \in \mathcal{T}} \mathbf{x}'_{st} \leq \mathbf{1} \ \ \forall\, s \in \mathcal{S}, \tag{61}$$

$$\mathbf{y}_{st} \in \mathcal{Y}_{st} \ \ \forall\, s \in \mathcal{S},\ t \in \mathcal{T}, \tag{62}$$

$$\mathbf{x}'_{st} \in \{0,1\}^F \ \ \forall\, s \in \mathcal{S},\ t \in \mathcal{T}. \tag{63}$$

We obtain scenario decompositions of CF-SD and CF1-SD by applying Lagrangian relaxation to the nonanticipativity constraints (54-55, 59-60 respectively) (e.g., Carøe and Schultz 1999), although we only attempt to solve the LP relaxation of the Lagrangian master problem.

We use the following abbreviations to denote the various formulations tested here.

Abbreviation	**Formulation and Solution Procedure**
CF-DE	Compact formulation CF, solved as a deterministic equivalent
SV-DE	General split-variable formulation SV, solved as a deterministic equivalent
SV1-DE	Specialized split-variable formulation SV1 that allows the expansion of a facility at most once in a scenario, solved as a deterministic equivalent
SV-DW-M	Dantzig-Wolfe decomposition of SV with du Merle duals stabilization
SV-DW-I	Dantzig-Wolfe decomposition of SV with interior-point duals stabilization
SV1-DW-M	Dantzig-Wolfe decomposition of SV1 with du Merle duals stabilization
SV1-DW-I	Dantzig-Wolfe decomposition of SV1 with interior-point duals stabilization
CF-SD	Scenario decomposition of CF-SD
CF1-SD	Scenario decomposition of CF1-SD

Table 1 displays the scenario-tree statistics for the seven problem instances, along with their solution times as deterministic equivalents, or using Dantzig-Wolfe decomposition. These results illustrate the power of decomposition in solving the larger problem instances.

Table 1: Solution times, in CPU seconds, for each procedure. The problems are solved to optimality, or until a 7,200-second limit is reached. Values in parentheses give the optimality gap at 7,200 seconds for those problems that do not solve, except that a dash indicates “greater than 100%.”

Scenario-tree Statistics			Deterministic Equivalent			Dantzig-Wolfe Decomposition			
Stages (num.)	Scenars. (num.)	Nodes (num.)	CF-DE (sec.)	SV-DE (sec.)	SV1-DE (sec.)	SV-DW-M (sec.)	SV-DW-I (sec.)	SV1-DW-M (sec.)	SV1-DW-I (sec.)
2	2	3	9.1	7.5	2.5	20.4	55.9	4.3	17.7
3	4	7	640.5	(1.8%)	1457.6	-	203.5	95.0	55.8
4	8	15	(42.4%)	(42.2%)	(34.9%)	-	2852.3	638.1	284.5
5	16	31	(68.7%)	(69.3%)	(65.6%)	-	(85.1%)	3624.2	1212.8
6	32	63	-	-	-	-	-	-	4301.4
5	81	121	-	-	-	-	(26.9%)	7043.5	2812.3
6	243	364	-	-	-	-	-	-	(7.6%)

The test problems are quite large. The largest problem instance we can solve by decomposition, with the 7,200-second limit imposed, has 5 stages and 81 scenarios. For this instance, CF-DE has 59,048 binary variables and 95,310 constraints, SV-DE has 158,602 binary variables and 194,864 constraints, and and SV1-DE has 81,070 binary variables and 117,332 constraints. All models have 15,004 continuous variables. Neither CPLEX 9.0 nor Xpress Optimizer 14.24 can solve any of these models in one day of computing time.

For this same largest instance, the largest subproblems for SV-DW have only 1,216 binary variables, while the SV1-DW subproblems have just 488 binary variables. The subproblems share the same 124 continuous variables and 800 constraints, and each solves in under 3 seconds on average. (Recall that the number of binary variables in the SV-DW subproblem for node n increases with its depth in the scenario tree, so that the subproblems for leaf nodes are the largest.)

The restricted master problems for SV-DW and SV1-DW are also of modest size. The master problem for SV1-DW-I, in the 5-stage-81-scenario problem, has only 23,161 variables in its last iteration, iteration 18 (see Table 3), and requires only 8.5 seconds to solve. In all iterations it has 44,165 constraints. The SV-DW master problem always has more constraints (see section 2.2), but its LP relaxation usually solves quickly, too. The SV-DW master problem has 99,675 constraints for the 5-stage-81-scenario problem instance. Although SV-DW-I cannot solve this problem in under 7,200 seconds, at iteration 18 its LP master problem has 24,181 variables and solves in 7.3 seconds, while at iteration 92 the number of variables grows to 27,808, but still requires only 9.9 seconds to solve.

We also need to discuss results, not shown in the table, regarding the quality of the LP bound obtained from Dantzig-Wolfe decomposition. In all instances from Table 1 that can be solved to optimality, SV1-MP-LP has an integral solution and is therefore tight, i.e., $z^*_{\text{SV1-MP-LP}} = z^*$. A similar statement holds for SV-MP-LP. The improvement over the LP bound can be large, also. For example, (a) in the smallest problem instance, $z^*_{\text{CF-DE-LP}} = 201,017$, whereas $z^*_{\text{SV-MP-LP}} = z^*_{\text{SV1-MP-LP}} =$ 444,149, and (b) in the largest problem instance, $z^*_{\text{CF-DE-LP}} = 123,388$ while $z^*_{\text{SV1-MP-LP}} = 960,881$. (Table 1 does not show that we can, in fact, solve CF-DE-LP and SV1-MP-LP for the largest problem instance; Table 2 gives the solution time for SV1-MP-LP.) Case (a) demonstrates Dantzig-Wolfe decomposition of SV1 can improve upon the standard LP lower bound by 779%.

Our results also show that interior-point duals stabilization is an important adjunct to the decomposition methodology, and that it is clearly superior to du Merle stabilization. For the 2-stage-2-scenario problem, the du Merle stabilization requires extensive tuning of its parameters to get SV-DW-M to converge. We also spent considerable effort tuning parameters for the 3-stage-4-scenario problem instance, but without success (as indicated by the dash). In contrast, the interior-point duals stabilization requires no tuning, and it significantly outperforms the du Merle alternative. Nonetheless, the results of both duals-stabilization schemes exhibit the well-known tailing-off effect. Thus, terminating the Dantzig-Wolfe decomposition early by setting an acceptable optimality tolerance for MP-Gap may be worthwhile. Table 2 reports the time it takes SV-DW-I and SV1-DW-I to satisfy tolerances of 5%, 1% and 0%.

Table 3 reports the number of major iterations corresponding to the times reported in Table 2. Here, we see that SV-DW requires many more iterations to converge than SV1-DW. As observed above, the differences in the average solution times between the restricted master problems and

Table 2: Computation times for SV-DW-I and SV1-DW-I to reach relative optimality gaps of 5%, 1% and 0%.

Scenario-tree Statistics			Dantzig-Wolfe Decomposition					
			SV-DW-I soln. time			SV1-DW-I soln. time		
Stages (num.)	Scenars. (num.)	Nodes (num.)	5% (sec.)	1% (sec.)	0% (sec.)	5% (sec.)	1% (sec.)	0% (sec.)
2	2	3	30.9	35.6	55.9	14.1	14.1	17.7
3	4	7	151.4	174.0	203.5	33.9	52.0	55.8
4	8	15	1886.3	2088.7	2852.3	188.1	188.1	284.5
5	16	31	16908.2	21355.2	24620.0	838.8	935.5	1212.8
6	32	63	-	-	-	2303.4	3286.7	4301.4
5	81	121	18005.3	21875.7	29820.7	1171.2	1370.4	2812.3
6	243	364	-	-	-	7407.1	11146.2	23637.9

Table 3: Number of major iterations for SV-DW-I and SV1-DW-I to reach relative optimality gaps of 5%, 1% and 0%.

Scenario-tree Statistics			Dantzig-Wolfe Decomposition					
			SV-DW-I iterations			SV1-DW-I iterations		
Stages (num.)	Scenarios (num.)	Nodes (num.)	5% (num.)	1% (num.)	0% (num.)	5% (num.)	1% (num.)	0% (num.)
2	2	3	17	19	26	11	11	13
3	4	7	27	30	35	10	14	15
4	8	15	67	72	88	10	10	13
5	16	31	183	221	245	12	13	16
6	32	63	-	-	-	11	14	17
5	81	121	63	73	92	10	11	18
6	243	364	-	-	-	11	14	23

subproblems for SV and SV1 are relatively small. So the large differences seen in overall solution times clearly result from SV-DW-I requiring many more iterations than SV1-DW-I. (It is interesting to see that the number of iterations for SV1-DW-I does not increase commensurately with problem size, at least for this application. This bodes well for solving even larger problems.)

It is important to note that the subproblems for this particular application are difficult, deterministic network-design problems (Johnson, Lenstra and Rinnooy Kan 1978). For this reason, and because we solve one subproblem for each scenario-tree node in each major iteration, the total time spent solving subproblems is substantial. SV-DW-I spends 93.7% of its time solving subproblems while SV1-DW-I spends 98.2%, averaged over the problems both methods can solve. Clearly, then, any improvement in solution time for the subproblems will improve overall solution time almost as much. All of the technology that has proved useful for solving deterministic network-design problems is worth evaluating for this purpose (e.g., Bienstock and Muratore 2000, Magnanti and Raghavan 2005).

Models that fit the paradigm of SV or SV1, but which have simpler subproblems, may solve very quickly. For instance, the multi-stage stochastic model of Riis and Anderson (2004) does fit the paradigm of SV, and its subproblems are easily solved knapsack problems.

As a final test, we compare our approach with scenario decomposition. The work by Dentcheva and Römisch (2004) implies that the optimal duality gap from a scenario decomposition must be at least as tight as the gap from our nodal decomposition. This would make scenario decomposition a preferred approach if its master problem and subproblems can be solved efficiently. Table 4 shows the solution times of the problem instances using both Dantzig-Wolfe (nodal) decomposition and scenario decomposition. Only times for SV-DW-I and SV1-DW-I are shown, as they are the most efficient of the approaches in Table 1.

Table 4: Solution times, in CPU seconds, for the problem instances using Dantzig-Wolfe decomposition and scenario decomposition. We attempt to solve all problems to optimality. A dash in the middle set of columns indicates "greater than 100%," while a dash in the last set indicates "greater than 7,200 seconds."

Scenario-tree Statistics			Dantzig-Wolfe Decomp.		Scenario Decomp.	
Stages (num.)	Scenarios (num.)	Nodes (num.)	SV-DW-I (sec.)	SV1-DW-I (sec.)	CF-SD (sec.)	CF1-SD (sec.)
2	2	3	55.9	17.7	7.0	3.7
3	4	7	203.5	55.8	(1.1%)	(1.1%)
4	8	15	2852.3	284.5	-	-
5	16	31	(85.1%)	1212.8	-	-
6	32	63	-	4301.4	-	-
5	81	121	(26.9%)	2812.3	-	-
6	243	364	-	(7.6%)	-	-

The results illustrate that the scenario decomposition is significantly more efficient than our nodal decomposition for the smallest 2-stage-2-scenario problem. The scenario decomposition, although potentially stronger, becomes intractable for larger problem instances. This happens because the size of each scenario subproblem increases in proportion to the number of stages, and the larger subproblems become impossible to solve. This is indeed a core limitation of the scenario-decomposition approach. On the other hand, Dantzig-Wolfe decomposition of the split-variable formulations leads to subproblems for scenario-tree nodes which increase in size and difficulty only marginally with the number of stages. Moreover, in the instances we have tested, the duality gap of zero from the Dantzig-Wolfe decomposition cannot be improved upon.

6 Conclusions

We have described a general, compact ("deterministic-equivalent") formulation of a multi-stage stochastic integer-programming model for planning the capacity expansion of a production system with one or more production facilities. Capacity-expansions are discrete, and a scenario tree represents uncertainty.

We reformulate the compact formulation using a variable-splitting technique to give a general, split-variable model (SV) that allows multiple capacity expansions of a facility over the planning horizon. We also devise SV1, which is a special case of SV that restricts each facility to at most one capacity expansion over the planning horizon. A Dantzig-Wolfe reformulation of either model results in a master problem having a substantially stronger LP relaxation than the compact formulation.

For each node n in the scenario tree, we define $\mathcal{P}_n$ to be the set of all predecessors of n, including n itself. Apart from variables $\mathbf{x}_{hn}$, which may be viewed as requests for capacity to be installed in nodes $h \in \mathcal{P}_n$, the variables in a subproblem SP(n) for the Dantzig-Wolfe reformulation of SV pertain only to node n. Indeed the variables $\mathbf{x}_{hn}$ can be viewed in the subproblem simply as alternative capacity-expansion options at node n of the scenario tree. As a result, the subproblems increase in difficulty only slightly with an increasing number of stages in a scenario tree. In SV1, the situation is even better, because the column-generation subproblems involve no variables (such as $\mathbf{x}_{hn}$) from predecessor nodes in the scenario tree. Thus, these subproblems do not become larger as the number of stages increases. This situation contrasts with scenario-decomposition methods in which the subproblems must cover the entire planning horizon, and so increase in size as more stages are added.

We have applied our methods to solve a capacity-planning problem for an electricity-distribution network, which requires the use of mixed-integer subproblems. However, the algorithm described is quite general. As long as good algorithms exist to solve them, the subproblems can incorporate arbitrary non-linearities or other complexities that the relevant application requires.

In order to enable a fair comparison between formulations SV and SV1, all computational tests carried out in this paper assume (as is the case in our application) that capacity increments U are independent of scenario and time. When capacity increments vary, SV1 is no longer valid, and we describe the model variant, SV1$'$, that must be applied. Further testing is needed to determine the computational implications of relaxing this assumption.

Much of the benefit to our approach will derive from situations (as with the electricity network) in which the subproblems are difficult mixed-integer programs (MIPs). In such a setting, it may be impossible to solve a single large-scale MIP or even the MIP subproblems generated by a scenario decomposition. On the other hand, in models with easier subproblems, our approach might be improved by amalgamating subproblems to obtain tighter relaxations and faster convergence. (The "DQA algorithm," described by Mulvey and Ruszczyński 1995, makes use of this technique.)

The efficiency of column generation hinges on the use of a good duals-stabilization scheme for the master problem. For our application, the "interior-point duals stabilization" scheme, which obtains dual variables from an interior-point algorithm, greatly outperforms the well-known scheme of du Merle et al. (1999). Note that we re-solve the master problems using an interior-point algorithm, from a cold-start, after adding a new set of columns. There is some potential to increase the speed of our algorithm by re-solving the master problems faster, using a suitable hot-start procedure for interior-point methods (e.g., Gondzio and Grothey 2003). Most of the computational time for our application accrues from subproblem solutions, however, so we can expect only minor speed-ups. Nonetheless, hot starts might be worthwhile in an application with simpler subproblems, or more difficult master problems.

Our split-variable formulation uses non-anticipativity constraints (6) that are inequalities. The validity of these constraints depends on the assumption that capacity expansions are non-negative quantities. With this assumption removed (for example, to admit facility closures), the inequalities must be replaced by equalities, and the master problem becomes equality-constrained. Based on this observation, it is tempting to suppose that more general multi-stage stochastic integer-programming problems might be attacked profitably using our decomposition approach. Our experiments show that this approach does work for small problems, with only modest increases in computational effort. Larger problems can take ten times longer to run, however, so more research is needed on this topic.

Acknowledgments

The authors thank Vector Electricity for providing the data used in the computational experiments. Andy Philpott acknowledges support of the New Zealand Foundation of Research, Science and Technology under grant UOAX0203. Kevin Wood thanks the Office of Naval Research, the Air Force Office of Scientific Research, the Naval Postgraduate School, and the University of Auckland for their support. Kavinesh Singh acknowledges support of the New Zealand Tertiary Education Commission and Vector Electricity.

References

[1] Ahmed, S., A.J. King, G. Parija. 2003. A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. *J. Global Opt.* **26** 3–24.

[2] Ahmed, S., N.V. Sahinidis. 2003. An approximation scheme for stochastic integer programs arising in capacity expansion. *Oper. Res.* **51** 461–471.

[3] Alonso-Ayuso, A., L.F. Escudero, M.T. Ortuño. 2003. BFC, a branch-and-fix coordination algorithmic framework for solving some types of stochastic pure and mixed 0-1 programs. *Eur. J. Oper. Res.* **151** 503–519.

[4] Appelgren, L.H. 1969. Column generation algorithm for a ship scheduling problem. *Transportation Sci.* **3** 53–68.

[5] Balas, E., S. Ceria, G. Cornuejols. 1996. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Mgmt. Sci.* **42** 1229–1246.

[6] Barahona, F., S. Bermon, O. Gunluk, S. Hood. 2005. Robust capacity planning in semiconductor manufacturing. *Naval Res. Logist.* **52** 459–468.

[7] Barnhart, C., E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance. 1998. Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* **46** 316–329.

[8] Berman, O., Z. Ganz, J.M. Wagner. 1994. A stochastic optimization model for planning capacity expansion in a service industry under uncertain demand. *Naval Res. Logist.* **41** 545–564.

[9] Bienstock, D., G. Muratore. 2000. Strong inequalities for capacitated survivable network design problems. *Math. Prog.* **89** 127–147.

[10] Birge, J.R., F. Louveaux. 1997. *Introduction to Stochastic Programming.* Springer Series in Operations Research. Springer-Verlag, New York.

[11] Carøe, C.C., R. Schultz. 1999. Dual decomposition in stochastic integer programming. *Oper. Res. Let.* **24** 37–45.

[12] Chen, Z., S. Li, D. Tirupati. 2002. A scenario-based stochastic programming approach for technology and capacity planning. *Comput. and Oper. Res.* **29** 781–806.

[13] Damodaran, P., W. Wilhelm. 2004. Branch-and-price methods for prescribing profitable upgrades of high-technology products with stochastic demands. *Dec. Sci.* **35** 55–81.

[14] Dantzig, G.B., P. Wolfe. 1960. Decomposition principle for linear programs. *Oper. Res.* **8** 101–111.

[15] Dentcheva, D., W. Römisch. 2004. Duality gaps in nonconvex stochastic optimization. *Math. Prog.* **101** 515–535.

[16] Desrosiers, J., H. Ben Amor, F. Soumis, D. Villeneuve. 2002. Stabilized column generation. *IMA Workshop: Computational Methods for Large Scale Integer Programs.* University of Minnesota, Minneapolis, MN. October 14-19.

[17] du Merle, O., D. Villeneuve, J. Desrosiers, P. Hansen. 1999. Stabilized column generation. *Discrete Math.* **194** 229–237.

[18] Eppen, G.D., R.K. Martin, L. Schrage. 1989. A scenario approach to capacity planning. *Oper. Res.* **37** 517–527.

[19] Erlenkotter, D. 1967. Optimal plant size with time-phased imports. In *Investments for Capacity Expansion: Size, Location, and Time-Phasing*, A.S. Manne, editor, pages 156–177. George Allen & Unwin, Ltd., London.

[20] Freidenfelds, J. 1980. Capacity expansion when demand is a birth-death random process. *Oper. Res.* **28** 712–721.

[21] Giglio, R.J. 1970. Stochastic capacity models. *Mgmt. Sci.* **17** 174–184.

[22] Gondzio, J., A. Grothey. 2003. Reoptimization with the primal-dual interior point method. *SIAM J. Opt.* **13** 842–864.

[23] Higle, J.L., S. Sen. 1996. Stochastic decomposition: A statistical method for large scale stochastic linear programming. Kluwer Academic Publishers, Dordrecht.

[24] Huang, K., S. Ahmed. The value of multi-stage stochastic programming in capacity planning under uncertainty. 2005. *Stochastic Programming E-Print Series.* http://dochost.rz.hu-berlin.de/speps.

[25] Johnson, D.S., J.K. Lenstra, A.H.G. Rinnooy Kan. 1978. The complexity of the network design problem. *Networks.* **8** 279–285.

[26] Krarup, J., O. Bilde. 1977. Plant location, set covering and economic lot size: An $O(mn)$-algorithm for structured problems. In *Numerische Methoden bei Optimierungsaufgaben, Band 3 (Tagung, Oberwolfach, 1976)*, pages 155–180. Internat. Ser. Numer. Math., Vol. 36. Birkhäuser, Basel.

[27] Laguna, M. 1998. Applying robust optimization to capacity expansion of one location in telecommunications with demand uncertainty. *Mgmt. Sci.* **44** S101–S110.

[28] Laporte, G., F.V. Louveaux. 1993. The integer L-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Let.* **13** 133–142.

[29] Lasserre, J.B. 2001. Global optimization with polynomials and the problem of moments. *SIAM J. Opt.* **11** 796–817.

[30] Lovasz, L., A. Schrijver, 1991. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Opt.* **1** 166–190.

[31] Lübbecke, M.E., J. Desrosiers. 2005. Selected topics in column generation. *Oper. Res.* **53** 1007–1023

[32] Lulli, G., S. Sen. 2004. A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. *Mgmt. Sci.* **50** 786–796.

[33] Luss, H. 1982. Operations research and capacity expansion problems: A survey. *Oper. Res.* **30** 907–947.

[34] I.J. Lustig, J.M. Mulvey, T.J. Carpenter. 1991. Formulating two-stage stochastic programs for interior point methods. *Oper. Res.* **39** 757–770.

[35] Magnanti, T.L., S. Raghavan. 2005. Strong formulations for network design problems with connectivity requirements. *Networks.* **45** 61–79.

[36] Magnanti, T.L., L.A. Wolsey. 1995. Design of survivable networks. In *Handbook in Operations Research and Management Science, Volume 7: Network Models,* M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser, editors, pages 503-616. Elsevier Publishing Company, Amsterdam.

[37] Manne, A.S. 1961. Capacity expansion and probabilistic growth. *Econometrica.* **29** 632–649.

[38] Manne, A.S. 1967. Two producing areas constant cycle time policies. In *Investments for Capacity Expansion: Size, Location, and Time-Phasing*, A.S. Manne, editor, pages 193-209. George Allen & Unwin, Ltd., London. *Discrete Appl. Math.* **123** 397–446.

[39] Masse, P., R. Gibrat. 1957. Application of linear programming to investments in the electric power industry. *Mgmt. Sci.* **3** 149–166.

[40] Mulvey, J., A. Ruszczyński. 1995. A new scenario decomposition method for large-scale stochastic optimization. *Oper. Res.* **43** 477–490 .

[41] Padberg, M.W. 1974. Perfect zero-one matrices. *Math. Prog.* **6** 180–196.

[42] Rajagopalan, S., M.R. Singh, T.E. Morton. 1998. Capacity expansion and replacement in growing markets with uncertain technological breakthroughs. *Mgmt. Sci.* **44** 12–30.

[43] Riis, M., K.A. Andersen. 2002. Capacitated network design with uncertain demand. *INFORMS J. Comput.* **14** 247–260.

[44] Riis, M., K.A. Andersen. Multiperiod capacity expansion of a telecommunications connection with uncertain demand. 2004. *Comput. and Oper. Res.* **31** 1427–1436.

[45] Riis, M., J. Lodahl. 2002. A bicriteria stochastic programming model for capacity expansion in telecommunications. *Math. Methods Oper. Res.* **56** 83–100.

[46] Ruszczyński, A., A. Shapiro. 2003. Stochastic programming models. In *Stochastic programming*, Vol. 10 of *Handbook in Operations Research and Management Science*, pages 1–64. Elsevier, Amsterdam.

[47] Ryan, D.M., J.C. Falkner. 1988. On the integer properties of scheduling set partitioning models. *Eur. J. Oper. Res.* **35** 442–456.

[48] Savelsbergh, M. 1997. A branch-and-price algorithm for the generalized assignment problem. *Operations Research.* **45** 831–841.

[49] Sen, S., R.D. Doverspike, S. Cosares. 1994. Network planning with random demand. *Telecomm. Sys.* **3** 11–30.

[50] Sherali, H.D., W.P. Adams. 1990. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.* **3** 411–430.

[51] Sherali, H.D., W.P. Adams, P.J. Driscoll. 1998. Exploiting special structures in constructing a hierarchy of relaxations for 0-1 mixed integer problems. *Oper. Res.* **46** 396–405.

[52] Shiina, T., J.R. Birge. 2004. Stochastic unit commitment problem. *International Trans. Oper. Res.* **11** 19–32.

[53] Silva, E.F., R.K. Wood. 2006. Solving a class of stochastic mixed integer programs with branch and price. *Math. Prog.* **108** 395–418.

[54] Singh, K.J., 2004. Column-generation for capacity-expansion planning of electricity distribution networks. In *Proceedings of the 39th Annual OR Society of New Zealand Conference*, A. Mason, editor, pages 76–85. Operational Research Society of New Zealand.

[55] Singh, K.J., A.B. Philpott, R.K. Wood. 2007. Column-generation for survivable design of electricity distribution networks. In review.

[56] Vanderbeck, F., L.A. Wolsey. 1996. An exact algorithm for IP column generation. *Oper. Res. Let.* **19** 151–159.

[57] Van Mieghem, J.A. 2003. Capacity management, investment, and hedging: Review and recent developments. *Manuf. and Service Oper. Mgmt.* **5** 269–302.

[58] Wolsey, L.A. *Integer Programming.* 1998. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York.