A Stochastic Intra-Ring Synchronous Optical Network Design Problem

⋆ J. Cole Smith

⋆⋆ Andrew J. Schaefer

⋆⋆⋆ Joyce W. Yen

⋆ Department of Systems and Industrial Engineering

University of Arizona, Tucson, AZ 85721

cole@sie.arizona.edu

⋆⋆ Department of Industrial Engineering

University of Pittsburgh, Pittsburgh, PA 15261

schaefer@engrng.pitt.edu

⋆⋆⋆ Industrial Engineering

University of Washington, Seattle, WA 98195-2650

joyceyen@u.washington.edu

Subject Classifications: Programming: stochastic, integer (applications, Benders/decomposition).

Area of Review: TELECOMMUNICATIONS.

Abstract

We develop a stochastic programming approach to solving an intra-ring Synchronous Optical Network (SONET) design problem. This research differs from pioneering SONET design studies in two fundamental ways. First, while traditional approaches to solving this problem assume that all data are deterministic, we observe that for practical planning situations, network demand levels are stochastic. Second, while most models disallow demand shortages and focus only on the minimization of capital Add-Drop Multiplexer (ADM) equipment expenditure, our model minimizes a mix of ADM installations and expected penalties arising from the failure to satisfy some or all of the actual telecommunication demand. We propose an L-shaped algorithm to solve this design problem, and demonstrate how a nonlinear reformulation of the problem may improve the strength of the generated optimality cuts. We next enhance the basic algorithm by implementing powerful lower and upper bounding techniques via an assortment of modeling, valid inequality, and heuristic strategies. Our computational results conclusively demonstrate the efficacy of our proposed algorithm as opposed to standard L-shaped and extensive form approaches to solving the problem.

1 Introduction

The last decade has witnessed a proliferation in the research and development of optical telecommunication fiber. Modern optical fibers typically permit the simultaneous transmission of 2.4 Gbps (gigabits per second) of data over each strand of fiber. Additionally, optical fibers are also preferred over copper wires due to their increased security and maintainability. Specific advantages and technical details of fiber-optic technology may be found in Wu (1992).

With the advent of this technology, the failure of a single optical fiber may result in a substantial loss of customer service. Several network topologies, including point-to-point diverse protection (DP) and self-healing ring (SHR) architectures, have been proposed to sustain the survivability of optical networks in the event of equipment failure. These topologies consider a set of nodes, which usually represent voice/data traffic origins and destinations in the network, and a set of arcs that represent connecting fibers. DP topologies enhance network reliability by including protective lines within the network, which can be utilized in the event of a facility failure (either a node or an arc) to re-route traffic. DP networks may be cost-prohibitive if the number of protection lines that must be installed to ensure a desired level of network survivability is too large. By contrast, in SHR architectures, the nodes are connected by a ring of fibers, and traffic routed along the ring is automatically re-routed in the case of equipment failure.

The Synchronous Optical Network (SONET) is a standard of transmission technology used in optical fiber networks designed under SHR topologies (Wu (1992) and Wu and Burrowes (1990) provide an in-depth description of the SONET ring architecture). The costliness of constructing new SONET-based networks has spawned much interest in the optimization of ring design problems. Prior approaches regarding the design of SONET systems have been limited to deterministic settings with fully satisfied network demand. However, forecast telecommunication demands among client nodes are actually quite stochastic in practice. Moreover, it is often preferable for a network provider to partially satisfy certain demands at the cost of a given shortage penalty, rather than expanding the telecommunication infrastructure and incurring additional capital equipment expenditures to satisfy some marginal extra demand. We accommodate both the stochastic demand and shortage penalty features within our design model by incorporating new modeling and algorithmic strategies within

contemporary stochastic integer programming methods (Birge and Louveaux, 1997).

Within a SONET ring, voice and data streams may be *multiplexed* onto a fiber-optic strand in order to more fully utilize its available bandwidth. The equipment responsible for identifying traffic that should be added and dropped from the flow of ring traffic is called an *add-drop multiplexer* (ADM). Thus, in order to route traffic between two nodes assigned to a SONET ring, we must have an ADM installed at both nodes. Our study is motivated by the relatively low cost of ADMs as compared with digital cross-connect systems (DCS), which may transmit data between two nodes not placed on the same ring by acting as a hub, or central connection, among the rings. As a result, we forbid inter-ring traffic, and focus instead on networks in which all demands are satisfied via intra-ring routing. Each SONET ring is capacitated by both the number of ADMs and the total amount of demand it can support. The (deterministic) optimization problem seeks to construct a set of SONET rings that can satisfy all demands subject to the foregoing restrictions. We choose to minimize the total number of ADMs that must be installed in the network, as this specialized equipment is the driving cost in the system. Note that we do not address node arrangement within a particular ring, but rather focus on the assignment of a set of nodes to each ring.

Several researchers have investigated the use of mathematical modeling to optimize deterministic aspects of SONET systems. Laguna (1994) develops a mixed-integer programming model to solve a SONET design problem that permits inter-ring traffic (for example, by DCS). Goldschmidt, Laugier, and Olinick (1998), who also consider inter-ring traffic, minimize the number of rings necessary to satisfy a given set of demands. Two deterministic variants of the intra-ring problem considered in this paper have been investigated. When demands between nodes must be fully placed on one ring and not split between two or

more rings (non-split demand), Lee at al. (2000) present a branch-and-cut algorithm, and Sutter, Vanderbeck, and Wolsey (1998) utilize a column generation methodology to minimize the required number of ADMs that must be installed. When demands between nodes may be placed across several rings (split demand), Sherali, Smith, and Lee (2000) provide a preprocessing/cutting-plane technique for obtaining optimal solutions within reasonable computational limits.

The remainder of this paper is organized as follows. In Section 2, we provide a mixed-integer programming formulation for the deterministic variant of our problem. In Section 3, we introduce and formulate the stochastic version of this problem, develop a decomposition approach for its solution, and provide various modeling and algorithmic procedures to improve the efficiency of our algorithm. In Section 4, we develop a heuristic for quickly obtaining tight upper bounds to the problem, and show how it may be integrated as an upper bounding procedure within the exact solution algorithm. In Section 5, we demonstrate the efficacy of the proposed approach on a set of realistic test instances. Finally, we conclude our discussion in Section 6 by summarizing our research and providing directions for future studies.

2 Deterministic Problem Statement

Consider a set N of client nodes, indexed by $i \in N \equiv \{1, ..., n\}$. Let d_{ij} be the number of channels required to carry the traffic between node $i \in N$ and node $j\ (> i) \in N$. Accordingly, define an (undirected) edge set $A = \{(i, j) : i < j, d_{ij} > 0\}$ comprised of such demand pairs. A set of m rings exists on which this demand may be routed. No more than $R\ (\geq 2)$ ADMs may be installed on each ring, and each ring may carry no more than a total of b channels of demand. A demand shortage penalty ϕ_{ij} is assessed for every unsatisfied unit of demand

between nodes i and j, $\forall (i,j) \in A$. These penalties are chosen based on their cost relative to the cost of an ADM.

To model the deterministic version of our SONET ring design optimization problem, we define a set of binary decision variables

$$x_{ik} = \begin{cases} 1 & \text{if node } i \text{ is assigned to ring } k \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in N,\ k \in M \equiv \{1, ..., m\}.$$

We also define a set of continuous decision variables f_{ijk} $\forall (i,j) \in A$ and $\forall k \in M$, representing the fraction of demand between the node pair (i,j) that will be satisfied by ring k, and w_{ij} $\forall (i,j) \in A$, representing the fraction of demand between nodes i and j that is not satisfied by the network. For notation convenience, we define the set of arcs incident to node i as

$$S_i = \{\rho \in A : \rho = (i,j) \text{ or } \rho = (j,i) \text{ for some } j\}.$$

The deterministic ring design problem (DRD) can now be stated as follows.

$$\textbf{DRD}\text{: Minimize} \sum_{i \in N} \sum_{k \in M} x_{ik} + \sum_{\rho \in A} \phi_\rho d_\rho w_\rho \tag{1a}$$

subject to

$$\sum_{k \in M} f_{\rho k} + w_\rho = 1 \quad \forall \rho \in A \tag{1b}$$

$$\sum_{\rho \in A} d_\rho f_{\rho k} \leq b \quad \forall k \in M \tag{1c}$$

$$\sum_{i \in N} x_{ik} \leq R \quad \forall k \in M \tag{1d}$$

$$0 \leq f_{\rho k} \leq x_{ik} \quad \forall i \in N,\ k \in M,\ \rho \in S_i \tag{1e}$$

$$x_{ik} \in \{0,1\} \quad \forall i \in N,\ k \in M \tag{1f}$$

$$w_\rho \geq 0,\ \forall \rho \in A. \tag{1g}$$

The objective function (1a) minimizes the sum of the total number of node-to-ring assignments (ADM installations), plus the total demand shortages weighted by their corresponding penalties. Constraints (1b) define the fraction of demand between each node pair that is not satisfied in the network, and (1c,d) impose the demand and ADM capacity restrictions on each ring. Finally, (1e) states that demand between nodes i and j, $(i, j) \in A$, may be satisfied on ring $k \in M$ only if both nodes i and j are placed on ring k, while (1f,g) represent logical variable restrictions.

3 SONET Ring Design with Stochastic Demands

In practice, customer demand may depend on several unknown factors, such as the state of competing technology or economic conditions, and is not known with certainty when the network is designed. In such cases a network design that considers this uncertainty may perform better than one that does not. We present a stochastic programming model designed to accommodate uncertain demands in designing the network, and then explore several techniques for improving the computational efficiency of the algorithm.

3.1 An L-Shaped Approach

We will formulate this problem as a two-stage stochastic program. Such programs are characterized by an initial *first-stage* decision, after which the true values of the random events are realized, upon which a *second-stage* or *recourse* decision is made. The network design problem serves as the first-stage problem, after which the second-stage problem determines the optimal allocation of unmet demand. The first-stage decision is an integer program, while the recourse decision can be solved as a network flow problem.

Let $\tilde{\xi}$ be a discretely distributed random vector with finite support Ξ. In this problem, a scenario ξ represents a realization of the demand requests across the arcs in the network. Index the scenarios by $\kappa = 1, ..., r$, where $r = |\Xi|$. For all $\kappa = 1, ..., r$, define the parameters p^κ as the probability of realizing the κth scenario, and d_ρ^κ as the number of channels requested for demand pair ρ in scenario κ, $\forall \rho \in A$. The decision variables w_ρ^κ are now defined to be the fraction of demand pair ρ left unsatisfied in scenario ξ_κ, $\forall \rho \in A$ and $\forall \kappa = 1, ..., r$.

The *extensive form* of the resulting two-stage stochastic program can be formulated as

$$\textbf{SEF}: \text{Minimize} \sum_{i \in N} \sum_{k \in M} x_{ik} + \sum_{\kappa=1}^{r} \sum_{\rho \in A} \phi_\rho d_\rho^\kappa w_\rho^\kappa \tag{2a}$$

subject to

$$\sum_{k \in M} f_{\rho k}^\kappa + w_\rho^\kappa = 1 \quad \forall \rho \in A, 1 \leq \kappa \leq r \tag{2b}$$

$$\sum_{\rho \in A} d_\rho^\kappa f_{\rho k}^\kappa \leq b \quad \forall k \in M, 1 \leq \kappa \leq r \tag{2c}$$

$$\sum_{i \in N} x_{ik} \leq R \quad \forall k \in M \tag{2d}$$

$$0 \leq f_{\rho k}^\kappa \leq x_{ik} \quad \forall i \in N,\ k \in M,\ \rho \in S_i, 1 \leq \kappa \leq r \tag{2e}$$

$$x_{ik} \in \{0, 1\} \quad \forall i \in N,\ k \in M \tag{2f}$$

$$w_\rho^\kappa \ \geq \ 0,\ \forall \rho \in A, 1 \leq \kappa \leq r. \tag{2g}$$

The *deterministic equivalent* formulation is given by

$$\textbf{SRD}: \text{Minimize} \sum_{i \in N} \sum_{k \in M} x_{ik} \ + \ \mathcal{Q}(x) \tag{3a}$$

subject to

$$\sum_{i \in N} x_{ik} \leq R \quad \forall k \in M \tag{3b}$$

$$x \text{ binary}, \tag{3c}$$

where $\mathcal{Q}(x)$, the *expected recourse function*, is given by $\mathcal{Q}(x) = \sum_{\kappa=1}^{r} p^{\kappa} Q(x, \xi_{\kappa})$, and

$$\mathcal{Q}(x, \xi_{\kappa}) \;=\; \text{Minimize} \sum_{\rho \in A} \phi_{\rho} d_{\rho}^{\kappa} w_{\rho}^{\kappa} \tag{4a}$$

subject to

$$\sum_{k \in M} f_{\rho k}^{\kappa} + w_{\rho}^{\kappa} = 1 \quad \forall \rho \in A \tag{4b}$$

$$\sum_{\rho \in A} d_{\rho}^{\kappa} f_{\rho k}^{\kappa} \leq b \quad \forall k \in M \tag{4c}$$

$$0 \leq f_{\rho k}^{\kappa} \leq x_{ik} \quad \forall i \in N,\ k \in M,\ \rho \in S_i \tag{4d}$$

$$w_{\rho}^{\kappa} \;\geq\; 0,\ \forall \rho \in A. \tag{4e}$$

Note that this problem is said to have relatively complete recourse, that is, there exists a feasible solution to the second-stage problem for every feasible x to the first-stage problem.

Associate dual variables λ with equations (4b), $-\mu$ with equations (4c), and $-\pi$ with the upper bounding equations of (4d). For a given realization ξ_{κ} and a first-stage solution $\hat{x}$, the dual problem is given by

$$D(\hat{x}, \xi_{\kappa}) :\ \text{Maximize} \sum_{\rho \in A} \lambda_{\rho} - b \sum_{k \in M} \mu_k \;-\; \sum_{i \in N} \sum_{k \in M} \sum_{\rho \in S_i} \hat{x}_{ik} \pi_{ik\rho} \tag{5a}$$

subject to

$$\lambda_{\rho} - d_{\rho}^{\kappa} \mu_k - \pi_{ik\rho} - \pi_{jk\rho} \leq 0 \quad \forall k \in M, \rho = (i, j) \in A \tag{5b}$$

$$\lambda_{\rho} \leq d_{\rho}^{\kappa} \phi_{\rho} \quad \forall \rho \in A \tag{5c}$$

$$\mu, \pi \geq 0. \tag{5d}$$

We will decompose this extensive form (given by (2)) using a multicut version of the L-shaped method with integer first-stage variables. Additional information on the multicut L-shaped method for such stochastic programs can be found in Birge and Louveaux (1988),

Van Slyke and Wets (1969), and Wollmer (1980). At any iteration, let $L(\kappa)$ be the set of all optimality cuts involving scenario ξ_κ, where the constraint coefficients of the ℓth optimality cut are E^ℓ and the right-hand side is e^ℓ. Let Θ_κ represent the recourse cost under scenario ξ_κ. Since this problem has relatively complete recourse, no feasibility cuts are necessary. The multicut L-shaped reformulation is given by

$$\text{Minimize} \sum_{i \in N} \sum_{k \in M} x_{ik} + \sum_{\kappa=1}^{r} p^\kappa \Theta_\kappa \tag{6a}$$

subject to

$$\sum_{i \in N} x_{ik} \leq R \quad \forall k \in M \tag{6b}$$

$$\Theta_\kappa + \sum_{i \in N} \sum_{k \in M} E^\ell_{ik} x_{ik} \ \geq e^\ell \quad \forall \ell \in L(\kappa), \ \kappa = 1, ..., r \tag{6c}$$

$$x \text{ binary.} \tag{6d}$$

Rather than generating and adding all optimality cuts, we will add them as necessary. The restricted master problem is given by considering only a subset $L'(\kappa)$ of the set $L(\kappa)$ of all optimality cuts for scenario ξ_κ. At an arbitrary iteration the restricted master problem is given by

$$\text{Minimize} \sum_{i \in N} \sum_{k \in M} x_{ik} + \sum_{\kappa=1}^{r} p^\kappa \Theta_\kappa \tag{7a}$$

subject to

$$\sum_{i \in N} x_{ik} \leq R \quad \forall k \in M \tag{7b}$$

$$\Theta_\kappa + \sum_{i \in N} \sum_{k \in M} E^\ell_{ik} x_{ik} \ \geq e^\ell \quad \forall \ell \in L'(\kappa), \ \kappa = 1, ..., r \tag{7c}$$

$$x \text{ binary.} \tag{7d}$$

Let $\hat{x}$ be a solution to the restricted master problem (7). For each scenario $\xi_\kappa \in \Xi$ we solve the subproblem $D(\hat{x}, \xi_\kappa)$ and obtain the optimal dual variables $\hat{\lambda}, \hat{\mu}$ and $\hat{\pi}$. The optimality cut is then given by

$$\Theta_\kappa \; + \; \sum_{i \in N} \sum_{k \in M} \left(\sum_{\rho \in S_i} \hat{\pi}_{ik\rho} \right) x_{ik} \geq \; \sum_{\rho \in A} \hat{\lambda}_\rho \; - \; b \sum_{k \in M} \hat{\mu}_k, \tag{8}$$

so that $E^\ell_{ik} \; = \; \sum_{\rho \in S_i} \hat{\pi}_{ik\rho} \; \forall i \in N$ and $\forall k \in M$, and $e^\ell = \sum_{\rho \in A} \hat{\lambda}_\rho - b \sum_{k \in M} \hat{\mu}_k$. If (8) is violated by the current solution $(\Theta_\kappa, \hat{x})$, we add this optimality cut to $L'(\kappa)$. If $(\Theta_\kappa, \hat{x})$ satisfy the optimality cuts for each $\kappa = 1, ..., r$, then $\hat{x}$ is an optimal solution. Otherwise, the master problem is resolved, and another iteration is performed.

3.2 Subproblem Solution Procedure

We may garner some nominal computational improvements along with an insight into the nature of the optimality cuts (8) by converting (4) to a network flow problem and utilizing specialized network flow algorithms to obtain its solution. The solution of these subproblems is the bottleneck operation in an upper bounding heuristic developed in Section 4, and thus improving the speed of solving the subproblems provides an improvement in the efficiency of that algorithm. For the following discussion, suppose we are examining (4) given some binary network design solution vector $\hat{x}$, under scenario $\kappa \in \{1, ..., r\}$.

Smith (2002) addresses the transformation of (5) to a minimum cost flow problem. Define $\mathcal{P}_\rho$ as the set of all rings on which demand pair ρ may be satisfied, that is, for $\rho = (i, j) \in A$, $\mathcal{P}_\rho = \{k \in M : \hat{x}_{ik} = \hat{x}_{jk} = 1\}$. The minimum cost flow problem is then constructed with $m + |A| + 1$ nodes. Nodes q_ρ correspond to each demand pair $\rho \in A$ and have a surplus of d^κ_ρ units, nodes t_k correspond to each ring $k \in M$ and have a demand of b units, and node u is an intermediate node with a surplus/demand of $mb - \sum_{\rho \in A} d^\kappa_\rho$ units. For each $\rho \in A$, an arc with cost 0 exists from q_ρ to all nodes t_k such that $k \in \mathcal{P}_\rho$. An arc also exists from

q_ρ to u with a cost of ϕ_ρ for each $\rho \in A$. Finally, an arc with zero cost exists from u to node t_k $\forall k \in M$. Solving this network flow problem, we obtain the optimal f values on the arcs from the q nodes to the t nodes, and optimal w values on the arcs from the q nodes to node u. Consider the demand network depicted by Figure 1, and suppose that nodes 1, 2, 3, and 4 have been assigned to ring 1 and nodes 3, 4, 5, and 6 have been assigned to ring 2. The minimum cost flow network is shown in Figure 2, where the flow costs are displayed alongside the arcs and all demand and surplus values are displayed alongside their respective nodes.

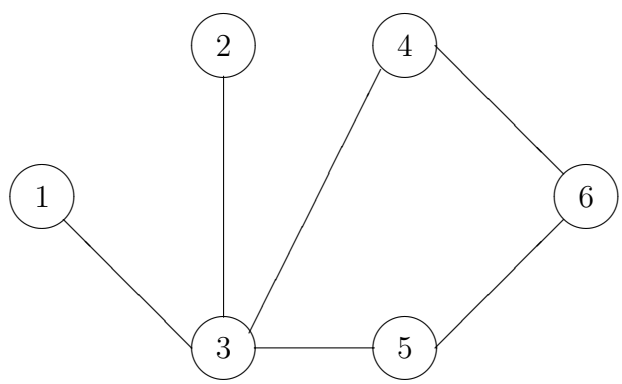

Figure 1: Minimum Cost Flow Demand Graph.

The dual formulation to this network flow problem can be stated as follows, where dual variables α_ρ are associated with node q_ρ $\forall \rho \in A$, and variables β_k are associated with t_k $\forall k \in M$. (The dual variable associated with u is arbitrarily set equal to 0.)

$$\text{Maximize} \sum_{\rho \in A} d_\rho^\kappa \alpha_\rho - b \sum_{k \in M} \beta_k \tag{9a}$$

subject to

$$\alpha_\rho - \beta_k \leq 0 \quad \forall \rho \in A, k \in \mathcal{P}_\rho \tag{9b}$$

$$\alpha_\rho \leq \phi_\rho \quad \forall \rho \in A \tag{9c}$$

$$\beta_k \geq 0 \quad \forall k \in M. \tag{9d}$$

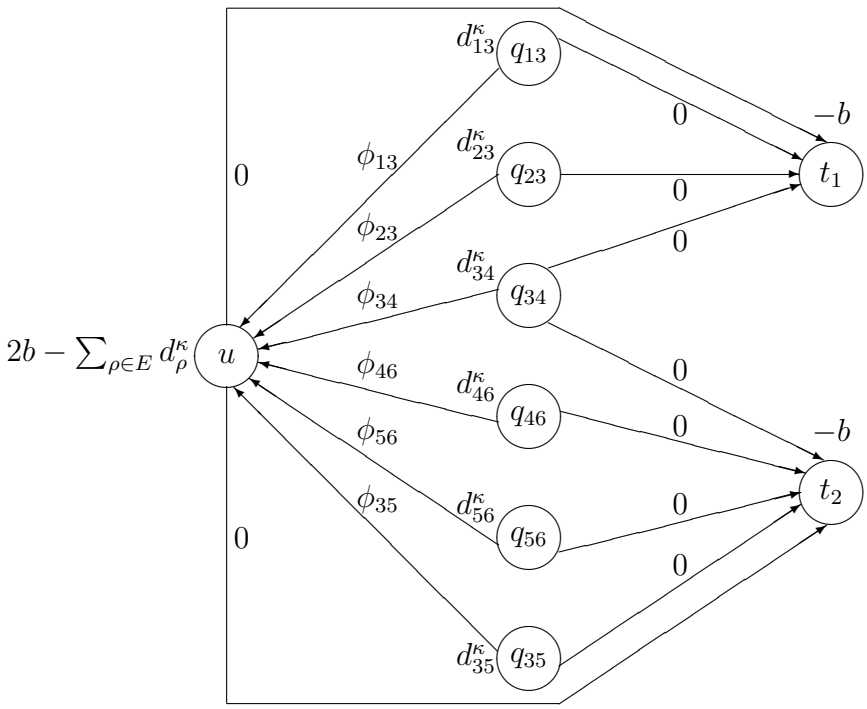

Figure 2: Minimum Cost Flow Graph for the Demand Network Depicted in Figure 1.

It is not readily obvious how one can recover the optimal π variables to (5), since primal constraints (4d) have been implicitly defined by the construction of the minimum cost flow problem. The following proposition prescribes a fast method for determining the optimal solution to (5) given an optimal dual basic feasible solution to (9).

Proposition 1. Given a binary network design vector $\hat{x}$ and a corresponding optimal basic feasible solution (α^*, β^*) to (9), there exists an optimal solution to (5) in which $\lambda^*_\rho = d_\rho \alpha^*_\rho \ \forall \rho \in A$ and $\mu^*_k = \beta^*_k \ \forall k \in M$. The optimal values for $\left(\pi^*_{ik\rho}, \pi^*_{jk\rho}\right)$ for $\rho = (i, j) \in A$ are

determined as

$$\left(\pi^*_{ik\rho}, \pi^*_{jk\rho}\right) = \begin{cases} (0,0) & \text{if } k \in \mathcal{P}_\rho \text{ or } \lambda^*_\rho \leq d^\kappa_\rho \mu^*_k \\ (\lambda^*_\rho - d^\kappa_\rho \mu^*_k, 0) & \text{if } \hat{x}_{ik} = 0 \text{ and } \hat{x}_{jk} = 1 \\ (0, \lambda^*_\rho - d^\kappa_\rho \mu^*_k) & \text{if } \hat{x}_{jk} = 1 \text{ and } \hat{x}_{jk} = 0 \\ \text{Any solution to } \pi^*_{ik\rho} + \pi^*_{jk\rho} = \lambda^*_\rho - d^\kappa_\rho \mu^*_k,\ \pi^*_{ik\rho}, \pi^*_{jk\rho} \geq 0 & \\ & \text{if } \hat{x}_{ik} = 0 \text{ and } \hat{x}_{jk} = 0 \end{cases} \tag{10}$$

Proof. Since models (5) and (9) are equivalent, we need to show that the objective function value given by (5a) for $(\lambda^*, \mu^*, \pi^*)$ is the same as that given by (9a) for (α^*, β^*), and that $(\lambda^*, \mu^*, \pi^*)$ is feasible to constraints (5b-d). The equivalence of the objective function values is clear, noting that

$$\sum_{i \in N} \sum_{k \in M} \sum_{\rho \in S_i} \hat{x}_{ik} \pi^*_{ik\rho} = 0$$

due to the construction of π^* values according to (10). For Constraints (5b) corresponding to $\rho \in A$ and $k \in \mathcal{P}_\rho$, feasibility of $(\lambda^*, \mu^*, \pi^*)$ is assured due to (9b). For (5b) corresponding to $\rho = (i,j) \in A$ and $k \notin \mathcal{P}_\rho$, we must have by definition of $\mathcal{P}_\rho$ that at least one of $\hat{x}_{ik} = 0$ or $\hat{x}_{jk} = 0$. Hence, $(\pi^*_{ik\rho}, \pi^*_{jk\rho})$ is defined according to one of the latter three cases of (10) in order to ensure feasibility to (5b). Finally, Constraints (5c) and (5d) hold true due to (9c), (9d), and (10). This completes the proof. □

The dual value recovery process reveals that when $\lambda_\rho - d^\kappa_\rho \mu_k > 0$ for some demand pair $\rho = (i,j) \in A$ and $k \notin \mathcal{P}_\rho$ such that $\hat{x}_{ik} = \hat{x}_{jk} = 0$, we must choose between setting $\pi_{ik\rho}$ or $\pi_{jk\rho}$ to equal this difference. The outcome of this decision has a direct impact on the optimality cut being passed back to the first-stage problem: if $\pi_{ik\rho}$ is set equal to $\lambda_\rho - d^\kappa_\rho \mu_k$, it will encourage node i to be added to ring k in the next master problem iteration, and vice versa if $\pi_{jk\rho}$ is chosen to be positive. Magnanti and Wong (1981) underscore the

importance of properly selecting dual solutions by investigating methods by which dual solutions leading to nondominated Benders' cuts may be identified. Unfortunately, each choice of dual values gives rise to a nondominated cut in the context of our problem. Since there are 2^h nondominated optimality cuts that can be formulated, where h is the number of "choices" as mentioned above, enumerating each possible cuts is computationally prohibitive. We conducted a brief computational experiment to investigate the effectiveness of various rules for choosing among setting $\pi_{ik\rho}$ and $\pi_{jk\rho}$ such that $\pi_{ik\rho} + \pi_{jk\rho} = \lambda_\rho - d_\rho^\kappa \mu_k$. The results of this experiment indicate that setting either $\pi_{ik\rho}$ or $\pi_{jk\rho}$ equal to $\lambda_\rho - d_\rho^\kappa \mu_k$ when given a choice is the best strategy. We arbitrarily select $\pi_{ik\rho} = \lambda_\rho - d_\rho^\kappa \mu_k$ for the last case of (10) in our computational study in Section 5.

An alternative for resolving the choice in π-values is to consider the following problem reformulation. Define variable $v_{\rho k} = x_{ik} x_{jk}$ for $\rho = (i, j) \in A$ and $k \in M$. We may resolve this nonlinearity by enforcing the restrictions

$$v_{\rho k} \leq x_{ik}, \; v_{\rho k} \leq x_{jk}, \text{ and } v_{\rho k} \geq 0 \quad \forall \rho \in A, \; \forall k \in M \tag{11}$$

in (3) (and thus in (6) as well). Note that this technique is a subset of the linearization strategy employed in the Reformulation-Linearization Technique of Sherali and Adams (1990, 1994). We omit the lower bounding constraint $v_{\rho k} \geq x_{ik} + x_{jk} - 1$ since it will be implied at optimality. The second-stage problem may now be modified by replacing (4d) with

$$0 \leq f_{\rho k}^\kappa \leq v_{\rho k} \quad \forall \rho \in A, \; \forall k \in M, \text{ given scenario } \kappa \in \{1, ..., r\}. \tag{12}$$

The dual problem would now be modified as follows, noting that only $\hat{v}$ values need to be passed to the subproblem:

$$D(\hat{v}, \xi_\kappa): \text{ Maximize} \sum_{\rho \in A} \lambda_\rho - b \sum_{k \in M} \mu_k \; - \; \sum_{\rho \in A} \sum_{k \in M} \hat{v}_{\rho k} \pi_{\rho k} \tag{13a}$$

subject to

$$\lambda_\rho - d_\rho^\kappa \mu_k - \pi_{\rho k} \leq 0 \quad \forall k \in M, \rho \in A \tag{13b}$$

$$\lambda_\rho \leq d_\rho^\kappa \phi_\rho \quad \forall \rho \in A \tag{13c}$$

$$\mu, \pi \geq 0. \tag{13d}$$

Using this higher-dimensional formulation, we would compute λ and μ values as before. The π values will now be set according to

$$\pi_{\rho k} = \begin{cases} \lambda_\rho - d_\rho^\kappa \mu_k & \text{if } \lambda_\rho - d_\rho^\kappa \mu_k > 0 \\ 0 & \text{otherwise} \end{cases} \quad \forall \rho \in A, \; \forall k \in M. \tag{14}$$

Since no choice of optimality cuts are given, a unique dual solution exists to every nondegenerate primal problem. Moreover, it can be shown that this higher-dimensional representation captures *all* of the exponentially many cuts that may be identified from the original model. The proof, omitted here for brevity, constructs any dual solution that could be obtained in the original model by surrogating (14) with negative multiples of the variable upper bounding constraints in (11).

3.3 Valid Inequality Generation

An important consideration in constructing a Benders-type decomposition algorithm with integer first-stage variables is the tightness of the master problem. If the master problem formulation is weak, excessive computational effort must be expended in finding an optimal solution to be passed to the second-stage subproblems. We thus develop classes of cutting

planes to reduce both the effort required to solve the master problem and the number of master problems that must be solved before the algorithm terminates.

Define a *component* $\hat{G}(\hat{N}, \hat{A})$ of $G(N, A)$ to be a connected subgraph such that if $i \in \hat{N}$ then neither edge (i, j) nor (j, i) belongs to A $\forall j \in N \setminus \hat{N}$, that is, $\hat{G}$ is disconnected from the rest of the graph. We may then execute the following procedure, which enumerates all subsets of nodes in $\hat{N}$ which may be placed on a ring, and evaluates whether the weighted amount of demand that could be satisfied on a ring containing these nodes would justify the requisite expenditure on ADM equipment.

Procedure 1. Consider the component $\hat{G}(\hat{N}, \hat{A})$, and initially set $v = \min(R, |\hat{N}|)$.

Step 1. If $v \leq 1$, go to Step 4. Else, select an ungenerated subset $\tilde{N}$ of $\hat{N}$ such that $|\tilde{N}| = v$. Create the graph $\tilde{G}(\tilde{N}, \tilde{A})$ by defining arc set $\tilde{A} = \{(i, j) \in \hat{A} : i \text{ and } j \in \tilde{N}\}$, i.e., $\tilde{A}$ is the set of arcs induced by $\tilde{N}$ on the graph $\hat{G}$. Set $max_c = 0$.

Step 2. For each scenario ξ_κ, $\kappa = 1, ..., r$, solve the linear knapsack problem

$$c^\kappa = \max \sum_{\rho \in \tilde{A}} \phi_\rho d^\kappa_\rho y_\rho \tag{15a}$$

$$\text{subject to} \sum_{\rho \in \tilde{A}} d^\kappa_\rho y_\rho \leq b \tag{15b}$$

$$0 \leq y_\rho \leq 1 \quad \forall \rho \in \tilde{A}, \tag{15c}$$

where y_ρ represents the fraction of demand satisfied for demand pair ρ. If $\sum_{\kappa=1}^{r} p^\kappa c^\kappa > max_c$, set $max_c = \sum_{\kappa=1}^{r} p^\kappa c^\kappa$. If $max_c > v$, go to Step 3. Otherwise, if not all node subsets of size v have been generated, go to Step 1. Else set $v = \lceil (max_c - 1) \rceil$ and go to Step 1.

Step 3. If $v < \min\{R, |\hat{N}|\}$, generate the constraint $\sum_{i \in \hat{N}} x_{ik} \leq v$ $\forall k \in M$. In either case, terminate the procedure.

Step 4. Set $x_{ik} = 0$ $\forall i \in \hat{N}, k \in M$ and terminate.

The constraint generated in Step 3 is justified by observing that any solution containing a subgraph having at least $v+1$ of the nodes in $\hat{G}$ on one of its rings may be improved or kept the same by removing those nodes from the ring. Hence, an optimal solution exists in which no ring contains $v+1$ nodes of $\hat{G}$. The variable fixing in Step 4 is a special case of the constraint in Step 3 in which $v=0$.

Unfortunately, the algorithm must enumerate a exponential number of subgraphs, and is not useful if R or $|\hat{N}|$ is large. We thus limit the use of this preprocessing algorithm to components having $\min(R, |\hat{N}|) \leq 4$, ensuring the polynomiality of our preprocessing routine. (Note that Step 2 can be executed in $O(r(n \ \log \ n))$ time, since we must solve r linear knapsack problems, each requiring $O(n \ \log \ n)$ computations.) A further rationale for limiting the size of subgraphs that may be enumerated is that the average number of arcs that exist in a subgraph grows quadratically in terms of the size of the subgraph. It thus becomes less likely to generate valid inequalities in Steps 3 or 4 as the size of the component increases.

Next, recall that no node will be assigned to a ring in an optimal solution unless an adjacent node in A is also assigned to the ring. (Otherwise, the node could be deleted without affecting the amount of demand that could be satisfied.) We thus incorporate the following *connectivity* constraints within the master problem.

$$x_{ik} \leq \sum_{j \in S_i} x_{jk} \quad \forall i \in N, \ \forall k \in M. \tag{16}$$

Note that these constraints also implicitly require the valid restriction that each ring either contains zero or at least two ADMs.

Now, observe that since each SONET ring is identical, any given solution in which some $\hat{m}$ rings are utilized to route demand has as many as $(\hat{m}! - 1)$ alternatively optimal solutions

that may be obtained by simply reindexing the rings. Sherali and Smith (2000) and Sherali et al. (2000) have shown such symmetry unduly burdens a branch-and-bound approach to solving these problems by requiring the enumeration of several symmetrical branches. We state here two effective hierarchies for defeating this symmetry. The first requires that the number of nodes placed on ring k is no less than the number of nodes placed on ring $k+1$, for $k = 1, ..., m-1$, that is,

$$\sum_{i=1}^{n} x_{ik} \geq \sum_{i=1}^{n} x_{i,(k+1)} \qquad \forall k = 1, ..., m-1. \tag{17}$$

Note that although the hierarchy embodied by (17) is unlikely to completely break the symmetry within the model, it contains a constraint structure with nonzeros equal to 1 or -1, which tends not to create additional fractional solutions in a branch-and-bound framework. Alternatively, we may impose the following hierarchy, which requires that the sum of the node indices placed on ring k is no less than the sum of node indices placed on ring $k+1$, for $k = 1, ..., m-1$:

$$\sum_{i=1}^{n} i x_{ik} \geq \sum_{i=1}^{n} i x_{i,(k+1)} \qquad \forall k = 1, ..., m-1. \tag{18}$$

This hierarchy is more likely to eliminate all symmetric solutions, but is more likely to cause fractional solutions in addition to those that would normally be encountered in a branch-and-bound algorithm. Note that we may replace the sum of the indices in (18) with the sum squared of the indices, or any other reasonable exponent to further discourage symmetrical solutions, at the possible expense of creating additional fractional solutions.

3.4 Incorporating Subproblem Data Within the Master Problem

If demand shortage penalties are large relative to the cost of installing ADMs, several master problems must be solved before optimality cuts (8) relate ample information about demand shortage penalties under various proposed node to ring assignments. In this case, the fore-

going implementation of the decomposition algorithm will fail to converge in a reasonable amount of time. We offer a modification to the master problem that captures a minimum shortage penalty to be incurred for each demand pair given a proposed network design. First, let us define *baseline* demands, $\underline{d}$, as follows.

$$\underline{d}_\rho = \min_{\kappa \in \{1,\ldots,r\}} d_\rho^\kappa \qquad \forall \rho \in A.$$

Proposition 2. Given some binary vector x^*, suppose (f^*, w^*) is an optimal solution to (3) with $x \equiv x^*$ and demands given by $\underline{d}$. Then for all demand pairs $\rho \in A : w_\rho^* > 0$, an optimal solution to each subproblem $\mathcal{Q}(x^*, \xi_\kappa)$ $\forall \kappa = 1, ..., r$ exists in which the shortage for pair ρ is at least $\underline{d}_\rho w_\rho^* + \left(d_\rho^\kappa - \underline{d}_\rho\right)$.

Proof. We prove this proposition by induction, given any scenario $\kappa \in \{1, ..., r\}$ and binary vector x^*. Recall from the discussion of Section 3.2 that given the node-to-ring assignments of a network, we may compute the optimal values for f and w by solving a minimum cost flow problem. This proposition trivially holds true if $d^\kappa = \underline{d}$. Now, suppose that the result is true for a set of demands $\overline{d}$, where $\underline{d} \leq \overline{d} \leq d^\kappa$, and define (f^*, w^*) to be the optimal solution to the minimum cost flow subproblem given x^* and $\overline{d}$. Recall that the arcs corresponding to basic solution variables will form a tree with respect to the minimum cost flow graph.

Suppose that we seek a solution to the minimum cost flow problem in which the demands have changed to $\hat{d} = \overline{d} + e_{\hat{\rho}}$, where $e_{\hat{\rho}}$ is a vector of length $|A|$ containing a 1 in the element corresponding to some $\hat{\rho} \in A$ and zeros elsewhere, for some $\hat{\rho} \in A$ such that $\overline{d}_\rho < d_\rho^\kappa$. With respect to the minimum cost flow subproblem, this change is akin to increasing the surplus of $q_{\hat{\rho}}$ by 1 and decreasing the surplus/demand of u by 1. Note that since the basic arcs of (f^*, w^*) form a tree, exactly one path (in the undirected sense) exists from $q_{\hat{\rho}}$ to u. This path consists of a (possibly null) series of arcs between the q-nodes and t-nodes, followed

either by an arc from some q-node to u (call this a path of Type I), or an arc from some t-node to u (path of Type II).

Let us first consider the case in which all of the prior flow values along the arcs of the path from $q_{\hat{\rho}}$ to u oriented in the opposite direction of the path are positive. We retain an optimal basic feasible solution with respect to demands $\hat{d}$ by increasing the flows on arcs oriented in the same direction of the path by one, and decreasing the flows of arcs oriented in the opposite direction of the path by one. In particular, if $w^*_{\hat{\rho}} > 0$, then an arc exists directly from $q_{\hat{\rho}}$ to u, and flow is increased by one along this arc. However, if the path from $q_{\hat{\rho}}$ to u is another path of Type I, the f-values in the new solution are modified, and the value of some other w-variable increases by one. If a Type II path exists, we will modify the values of the f-variables, and will decrease the amount of unused capacity for some ring by decreasing the flow from node u to some t-node. In any of these cases, the previous solution continues to give rise to the same complementary slack primal and dual feasible bases. Since the w-values are nondecreasing, and since $w_{\hat{\rho}}$ increases by one if $w^*_{\hat{\rho}}$ was a positive value, we have that Proposition 2 holds true for this case.

We next discuss the case in which some arc oriented in the opposite direction of the path from $q_{\hat{\rho}}$ to u has a flow of zero. Note that since $w^*_{\hat{\rho}} = 0$ for this case, we must simply show that no w-values will decrease in the new solution. Suppose we follow the path from $q_{\hat{\rho}}$ to u, and identify the first degenerate basic arc oriented in the opposite direction of the path. Removing this arc from the basis, we now have two components of the minimum cost flow graph with respect to the basic arcs. Note that the component containing node $q_{\hat{\rho}}$ does not contain node u, and thus a basic set of arcs is reconstituted by adding an arc from the component containing $q_{\hat{\rho}}$ to the component containing u. This connecting arc must either be

from a q-node to a t-node, or from a q-node to u. A finite number of such degenerate pivots may need to be performed before a nondegenerate iteration takes place, at which time the situation reduces to the one addressed previously.

Note that we may iteratively construct any set of demands d^κ for $\kappa \in \{1, ..., r\}$ by independently incrementing the baseline demands, with Proposition 2 holding true after each such incremental step. This completes the proof. $\square$

Suppose the deterministic ring design problem (1) is solved using the baseline demands, with one optimal solution having a set of shortages given by $\hat{w}$. The result of Proposition 2 implies that if $\hat{w}_\rho > 0$ for some $\rho \in A$, then there will exist an optimal recourse decision in which the shortage for demand pair ρ in scenario $\kappa \in \{1, ..., r\}$ is at least $\underline{d}_\rho \hat{w}_\rho + \left(d^\kappa_\rho - \underline{d}_\rho\right)$ units. The following *baseline strategy* explicitly solves the baseline demand problem in the first stage, and relates a minimum penalty measure that will be incurred in the second-stage problems to the first-stage variables. First, we include the following constraints within (6):

$$\sum_{k \in M} f^0_{\rho k} + w^0_\rho = 1 \quad \forall \rho \in A \tag{19a}$$

$$\sum_{\rho \in A} \underline{d}_\rho f^0_{\rho k} \leq b \quad \forall k \in M \tag{19b}$$

$$0 \leq f^0_{\rho k} \leq x_{ik} \quad \forall i \in N,\ \forall k \in M,\ \forall \rho \in S_i,\ \text{and } w^0_\rho \geq 0 \quad \forall \rho \in A. \tag{19c}$$

To fully exploit the strength of Proposition 2, we introduce a new set of binary variables to the problem. Define y_ρ to equal 1 if a shortage for demand pair $\rho \in A$ exists in the solution of the baseline demand, and 0 otherwise. We impose the following additional constraints in (6):

$$y_\rho \geq w^0_\rho \text{ and } y_\rho \in \{0, 1\} \quad \forall \rho \in A \tag{19d}$$

$$\Theta_\kappa \geq \sum_{\rho \in A} \phi_\rho \left[y_\rho \left(d_\rho^\kappa - \underline{d}_\rho \right) + \underline{d}_\rho w_\rho^0 \right] \quad \forall \kappa = 1, ..., r. \tag{19e}$$

Constraints (19d) state the lower bounds on the binary variables y, while constraints (19e) state the lower limits for shortage penalties given the baseline scenario solution. We will empirically examine the tradeoff between increasing the difficulty of the master problem by employing the baseline strategy embodied by (19) and the decrease in the number of problem iterations that must be executed.

4 Heuristic Procedure

Since finding an optimal solution to Problem SRD may be too difficult to achieve within imposed computational limits, we may wish to quickly determine a heuristic solution to SRD. This solution would also benefit the multicut L-shaped algorithm of Section 3 by providing an upper bound to both the overall stochastic program and to each master integer program, and by prescribing a solution from which an initial set of optimality cuts may be obtained. We describe in this section the Greedy Augmenting Heuristic (**GAH**), which operates by selecting initial node-to-ring assignments for each ring, and then identifying promising augmentations of the proposed network design.

The logic of GAH is depicted as a flowchart in Figure 3. We first introduce a metric that measures the benefit of adding a node to a particular ring. Assume we are given a set of node-to-ring assignments previously prescribed by the heuristic. We define the *utility* of a node i on a ring k as the additional amount of weighted demand that could be satisfied on k with the addition of node i, minus the unit cost for including an extra ADM in the network. GAH constructs one ring at a time by assigning to the ring currently under consideration a node having the highest utility, until the ring cardinality restriction is met.

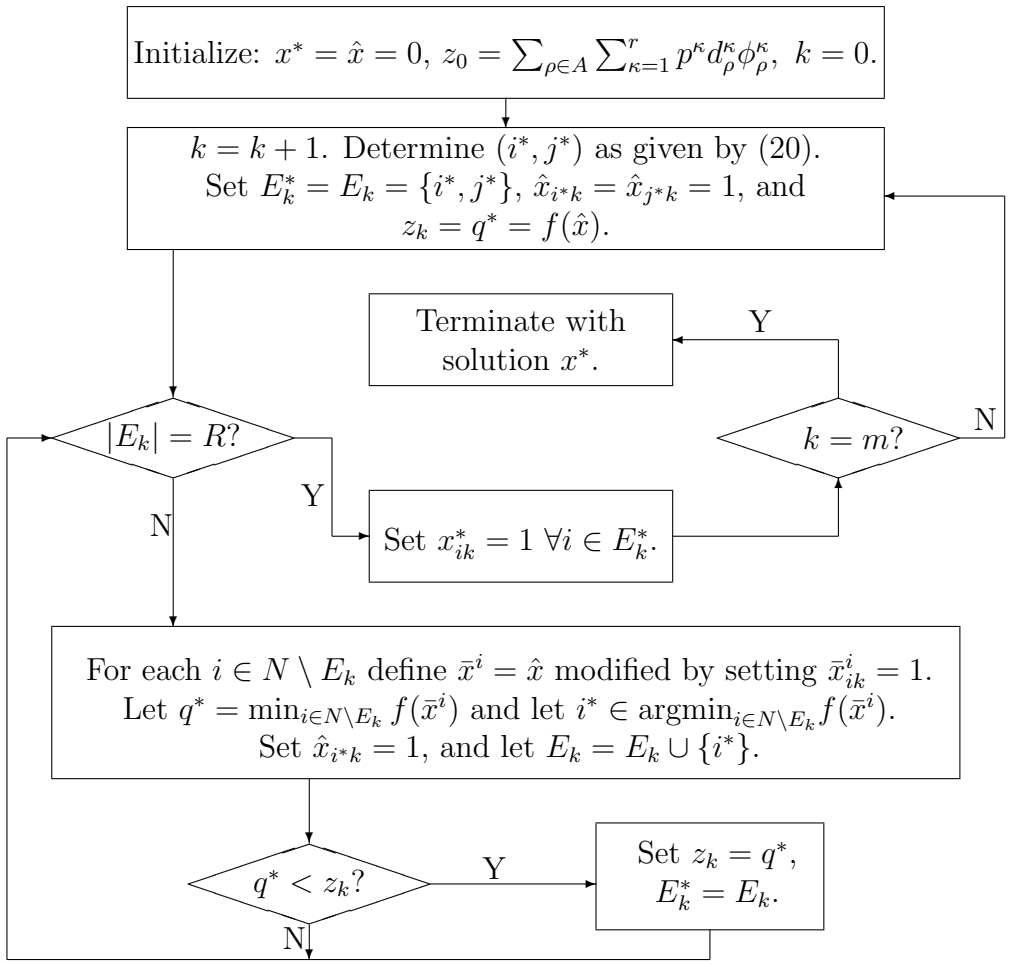

Figure 3: Heuristic Procedure for Generating Node-To-Ring Assignment.

We initialize the algorithm by setting the solution vector $\hat{x}_{ik} = 0$ $\forall i \in N$ and $k \in M$. For some assignment vector x, define $f(x) = \sum_{i \in N} \sum_{k \in M} x_{ik} + \mathcal{Q}(x)$, that is, $f(x)$ is the optimal solution to SRD given x. We will denote by z_k the best objective function value found after constructing (nonempty) rings $1, ..., k$.

Given an empty ring currently under consideration, the utility of each node must equal zero. To prevent an arbitrary assignment of the initial node for an empty ring, we determine two initial nodes by identifying the "best" demand pair that could be added to a new ring. Define $\overline{x}^{ij}$ as $\hat{x}$ modified by setting $\bar{x}_{ik}^{ij} = \bar{x}_{jk}^{ij} = 1$, where k is the new ring being constructed.

We initialize ring k with nodes i^* and j^* satisfying

$$\rho^* = (i^*, j^*) \in \text{argmin}_{\rho \in A} \left(f(\overline{x}^{ij}) \right). \tag{20}$$

We may now meaningfully compute the utility of each node $i \in \overline{N} = N \setminus \{i^*, j^*\}$. Let $\bar{x}^i$ be the current vector, $\hat{x}$, modified by setting $\bar{x}^i_{ik} = 1$, i.e., adding node i to the current ring k. We compute the utility of node i on ring k as $f(\hat{x}) - f(\bar{x}^i)$, and identify the node with the best utility as

$$i^* \in \text{argmin}_{i \in \overline{N}} \left(f(\overline{x}^i) \right). \tag{21}$$

Three properties of the behavior of GAH are discussed next that justify the remaining steps of the algorithm.

Property 1. Consider the construction of ring k by GAH, and let $f(x^\alpha_k)$ be the objective value obtained by the heuristic in constructing ring k, when ring k contained α nodes. Then $f(x^\alpha_k)$ is not necessarily quasiconvex in terms of α.

Example. Consider the demand graph given by Figure 4 as a one-scenario (deterministic) example whose penalties are given by the values alongside the arcs, and whose demands are given by $d^1_\rho = 1$ for all $\rho \in A$. Suppose that the demand capacity restriction is set to $b = 5$ channels and that the node cardinality restriction is set to $R = 4$ nodes. Table 1 describes the construction of the first ring by GAH for this example. In this example, a two-node ring is more expensive than either a zero-node ring or a three-node ring. However, assigning all four nodes to the ring results in the least expensive design. □

As a result of the foregoing property, GAH continues to add nodes to the current ring until the node cardinality limit is reached in hopes of identifying improved solutions. It is thus necessary to track the best objective value z_k achieved in the construction of the ring,

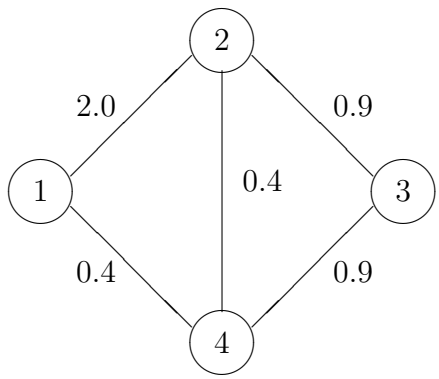

Figure 4: Illustration of Property 1.

Iteration	Nodes Assigned	Shortage Penalty	Number of ADMs	Ring Cost (q^*)
Initialize	1,2	2.6	2	4.6
1	1,2,3	1.7	3	4.7
2	1,2,3,4	0	4	4

Table 1: GAH Solution of Figure 4 Problem.

the set of nodes E_k^* assigned to ring k leading to that solution, and the current set of nodes E_k assigned to ring k. At the end of the construction of this ring, we set $x_{ik}^* = 1 \; \forall i \in E_k^*$.

Property 2 illustrates a similar behavior in the overall construction of the rings, wherein a ring may be constructed that temporarily worsens the objective function value, but ultimately leads to the construction of another ring that results in an improved solution.

Property 2. The sequence of objective function values z_k, $k = 1, ..., m$ is not quasiconvex in terms of k.

Example. We again consider a one-scenario instance whose demand network is depicted in Figure 5, where penalties are displayed alongside the arcs and $d_\rho^1 = 1$ for all $\rho \in A$, and instance parameters are defined as $b = 6$, $R = 4$, and $m = 3$. Table 2 summarizes the GAH steps for this example. Note that $z_2 > z_1$ and $z_2 > z_3$ for this example, justifying the construction of new rings even if previously constructed rings are not cost-efficient. □

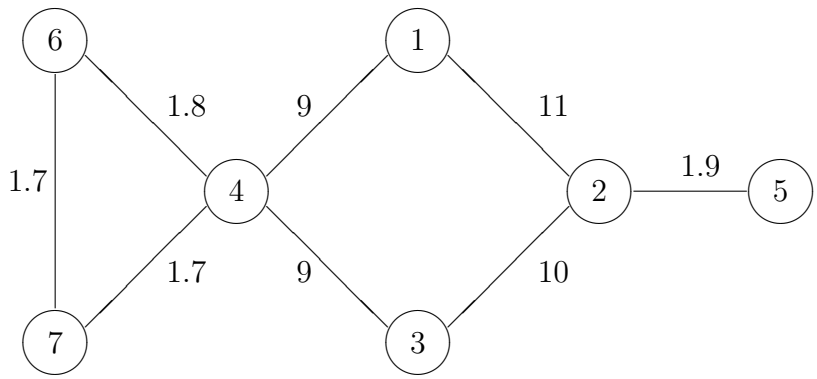

Figure 5: Illustration of Property 2.

Number of Rings (k)	E_k^*	Shortage	Total Number of ADMs	Total Cost (z_k)
0		46.1	0	46.1
1	1,2,3,4	7.1	4	11.1
2	2,5	5.2	6	11.2
3	4,6,7	0	9	9

Table 2: GAH Solution of Figure 5 Problem.

Property 2 begs the question of whether or not a ring that leads to a temporary objective function increase may be deleted from the final solution, improving the objective function value. For instance, deleting ring 2 in the previous example would result in a solution with objective function 8.9 (7 ADMs plus a penalty of 1.9 for not satisfying demand between nodes 2 and 5). Proposition 3 answers this question affirmatively.

Proposition 3. Let $z^* = z_m$ be the objective function value corresponding to the solution $\hat{x}$, let $\mathcal{Q}_k(\hat{x})$ be the expected weighted amount of demand satisfied on ring k for $k = 1, ..., m$, and let $|E_k^*|$ be the number of nodes attached to ring k for $k = 1, ..., m$. If there exists a $\overline{k}$ such that $|E_{\overline{k}}^*| - \mathcal{Q}_{\overline{k}}(\hat{x}) > 0$, then removing ring $\overline{k}$ from the current solution will decrease z^* by at least $|E_{\overline{k}}^*| - \mathcal{Q}_{\overline{k}}(\hat{x})$.

Proof. Consider the solution $\overline{x}$, given by $\hat{x}$ modified by setting $\overline{x}_{i\overline{k}} = 0$ for all $i = 1, ..., n$. Utilizing the same demand-to-ring assignments given by the f variables as before (except with $f_{\rho\overline{k}} = 0$ for all $\rho \in A$), we obtain a feasible solution with objective function value

$z^* - |E^*_{\overline{k}}| + \mathcal{Q}_{\overline{k}}(\hat{x})$. As such, this solution is an upper bound to the problem of minimizing the expected shortage penalty arising from the new network design given by $\overline{x}$, and thus the objective function decrease must be at least $|E^*_{\overline{k}}| - \mathcal{Q}_{\overline{k}}(\hat{x})$. This completes the proof. $\square$

Based on Proposition 3, we develop the following postprocessing procedure. Given a current solution $\hat{x}$, identify ring $\overline{k} \in \text{argmax}_{k \in M}(|E^*_k| - \mathcal{Q}_k(\hat{x}))$. If $|E^*_{\overline{k}}| - \mathcal{Q}_{\overline{k}}(\hat{x}) > 0$, delete ring $\overline{k}$, modify $\hat{x}$ by setting $\hat{x}_{i\overline{k}} = 0 \; \forall i \in E^*_{\overline{k}}$, determine the optimal f-variables given the remaining node-to-ring assignments, and repeat. The heuristic is complete when $|E^*_k| - \mathcal{Q}_k(\hat{x}) \leq 0$ $\forall k \in M$. Note that we do not simultaneously delete all rings meeting this criteria, as the deletion of one ring may change the demand-to-ring assignments to make the inclusion of each other ring cost-effective.

5 Computational Results

To test the computational efficacy of the procedures developed in this paper, we generated three test sets, each containing 10 stochastic SONET instances. Each instance in Set 1 contains 10 scenarios, while Set 2 instances contain 20 scenarios, and Set 3 instances contain 30 scenarios. For each instance, b is set to 48, R is set to 5, and $m = 4$ SONET rings may be constructed. Each demand graph consists of 10 nodes and has a 40% density, leading to the random construction of 18 edges. For each demand pair $\rho \in A$, the shortage penalty ϕ_ρ is randomly generated according to a uniform distribution on the interval (0,1], and an expected demand d^e_ρ is randomly generated as an integer according to a uniform distribution on the interval [4,10]. Finally, the demands for each scenario $\kappa = 1, ..., r$ are set equal to $d^\kappa_\rho = d^e_\rho$ with probability 0.3, $d^\kappa_\rho = d^e_\rho + 1$ or $d^e_\rho - 1$, each with probability 0.2, $d^\kappa_\rho = d^e_\rho \pm 2$ with probability 0.1, and $d^\kappa_\rho = d^e_\rho \pm 3$ with probability 0.05. The demand graphs turn out to be connected for each of these instances, and hence Procedure 1 of Section 3.3 is omitted in

Instance Set	Heur Soln	Optimum	# Opt	Opt Gap	CPU (secs)
Set 1	17.49	15.24	0	14.87%	1.56
Set 2	16.60	15.47	2	7.27%	3.12
Set 3	17.20	15.31	1	12.47%	4.82

Opt: Number of times the heuristic identifies the optimal solution.
Opt Gap: Average (UB - Optimal Value)/(Optimal Value)*100%.

Table 3: Average Performance of the Greedy Augmenting Heuristic.

our implementations. All computations were executed on a SUN-Ultra 10 Workstation with 256 MB of RAM using the CPLEX 7.0 callable library to solve integer programming and network flow problems.

We begin by testing the effectiveness of the Greedy Augmenting Heuristic. Table 3 compares the average heuristic objective function value ("Heur Soln") with the average optimal solution over 10 instances in each of the three test sets, and provides the average CPU time in seconds required to complete the heuristic. While GAH executes quickly and does manage to identify the optimal solution in three of the thirty instances, the average optimality gap is still large enough to warrant the use of an exact algorithm. For the remaining implementations of the L-shaped algorithm, we use GAH to obtain an initial upper bound as well as an initial set of optimality cuts based on this heuristic solution.

Next, we analyze the effectiveness of incorporating the connectivity valid inequalities given by (16) and the symmetry-breaking inequalities given by (17) within the master problem. (A preliminary set of tests demonstrated that the node-based hierarchy (17) is more effective for this class of problems than is the demand-based hierarchy (18).) Table 4 illustrates the average running times in terms of CPU seconds required by our algorithm using combinations of the connectivity and symmetry-breaking constraints. All runs were performed with the baseline demand scenario incorporated in the master problem. The symmetry-breaking con-

	(16) and (17)		(17) Only		(16) Only		No VI's	
Instance Set	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter
Set 1	34.18	2.5	33.44	2.1	136.61	10.0	147.42	9.6
Set 2*	36.61	2	45.94	2.22	482.03	16.11	483.45	15.11
Set 3*	68.25	2.89	73.87	3	834.71	30.22	1080.06	30.56

Iter: Average number of iterations required.

*One instance could not be solved within 3 hours for combinations not using (17).

Table 4: Effectiveness of Connectivity (16) and Symmetry-Breaking (17) Valid Inequalities.

straints are clearly vital to the efficiency of this algorithm, while the connectivity constraints provide a slight improvement in the average running time. Without the symmetry-breaking constraints (17), one instance in each of Set 2 and Set 3 could not be solved within the imposed time limit of three hours. Hence, the averages listed for Sets 2 and 3 are only over the 9 instances that all implementations could solve within 3 hours. The true benefit of incorporating (17) within the master problem is dampened somewhat by disregarding the two instances in which these cuts are most vital. The average CPU time over *all* Set 2 and Set 3 instances using both (16) and (17) is 90.61 seconds and 129.30 seconds, respectively. When only inequalities (17) are used, the CPU average for all Set 2 and Set 3 instances is 146.89 seconds and 131.97 seconds, respectively. We thus incorporate both of these sets of valid inequalities within the master problem in all future computational tests.

Our next set of computational tests conclusively demonstrate the ineffectiveness of the L-shaped method when the baseline demand scenario is *not* incorporated within the master problem. Using only (16) and (17) in the master problem along with the initial optimality cuts provided by the heuristic solution obtained from GAH, none of the thirty test instances could be solved to optimality within three hours. In fact, Table 5 demonstrates that the average final gap between the lower and upper bounds identified by the algorithm is consid-

Instance Set	Iter	LB	UB	Relative Gap	Opt. Gap
Set 1	43.6	12.73	17.46	26.96%	14.65%
Set 2	34.6	12.28	16.60	25.48%	7.27%
Set 3	31.1	12.09	16.97	28.44%	10.94%

Iter: Average number of iterations performed within the time limit.
LB: Average lower bound obtained after 3 CPU hours.
UB: Average best solution found within 3 CPU hours.
Relative Gap: Average (UB - LB)/UB*100%.
Opt. Gap: Average (UB - Optimal Value)/(Optimal Value)*100%.

Table 5: Algorithmic Performance without Baseline Demands After 3 CPU Hours.

erable. The strategy of using the best known solution after the three hour mark would be only slightly better than using the heuristic solution; in just three out of the thirty instances is the solution found at this point better than the heuristic solution.

Our last set of computational tests compares in detail four implementation options of our overall L-shaped algorithm versus simply solving the extensive form of the stochastic program. We denote the large integer programming approach (as formulated in (2)) as "Extensive Form" in Table 6. The "Multicut" approach uses the multicut L-shaped algorithm augmented by the heuristic, valid inequality, and baseline enhancements mentioned thus far. The "Single Cut" option is performed in the same manner, with the following modification to the multicut L-shaped approach. After the solving of each master problem, instead of adding an optimality cut to the master problem for each subproblem violating (8), we aggregate all such inequalities into one optimality cut. This single cut is formed by multiplying each generated optimality cut by the probability of realizing its corresponding scenario and summing them together. Note that whereas this strategy would normally allow Θ to be considered as one variable instead of r separate variables, we retain the individual Θ_κ variables for $\kappa = 1, ..., r$ to garner the effectiveness of the baseline demand subproblem. The

"Quadratic" option refers to the use of the reformulated dual subproblem (13) requiring the addition of quadratic variables $v_{\rho k}$ for $\rho \in E$ and $k \in M$, developed in Section 3.2. Finally, "Quad + Single" denotes the implementation of the quadratic model in conjunction with the single cut strategy.

The results in Table 6 demonstrate that the single cut implementations outperform their disaggregated counterparts for these runs. Quite surprisingly, even though the disaggregated (multicut) version provides stronger cuts to the master problem, our aggregation strategy resulted in slightly *fewer* overall iterations for the L-shaped algorithm. The aggregation of optimality cuts thus becomes a very attractive strategy computationally, as it reduces the size and difficulty of each master problem while requiring roughly the same number of iterations to solve each instance. As expected, the quadratic model option was effective in reducing the number of overall iterations. Unfortunately, this reduction in iterations was not always sufficient to make up for the increase in difficulty of solving the master problem. Due to the inconsistency of the running times when using the quadratic model (even when used in conjunction with the single cut strategy), we recommend the use of the original dual subproblem within the L-shaped algorithm. However, the effectiveness of this option on some of the test instances (e.g., on Set 2 instances) implies that it might indeed be useful on certain classes of problems for which the quadratic representation provides a substantial reduction in the number of iterations executed. Finally, these results demonstrate that the extensive form option is clearly inferior to any of the four L-shaped implementations. One interesting result is that the extensive form approach seems to suffer the same difficulties in solving the hardest instances as do the L-shaped approaches, implying that the slow running times for those instances are not due to weaknesses in the concept of the L-shaped algorithm, but rather to the inherent difficulty of those instances.

	Multicut		Quadratic		Single Cut		Quad + Single		Extensive Form
Instance	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
1	14.1	1	20.77	1	14.45	1	15.8	1	439.48
2	8.73	1	7.56	1	5.63	1	7.92	1	158.62
3	48.46	6	37.1	2	60.15	4	60.87	5	223.07
4	7.46	1	10.23	1	7.06	1	6.69	1	116.6
5	7.48	1	9.46	1	7.32	1	9.71	1	110.56
6	5.69	1	9.88	1	7.35	1	6.19	1	125.87
7	21.42	2	8.22	1	20.38	2	5.5	1	103.55
8	103.33	4	138.14	4	62.9	4	101.42	4	207.03
9	37.07	4	74.59	5	29.42	4	34.16	4	158.18
10	88.05	4	51.5	2	58.47	3	44.03	3	285.97
Avg	34.18	2.5	36.75	1.9	27.31	2.2	29.23	2.2	192.89

(a) Set 1 Instances

	Multicut		Quadratic		Single Cut		Quad + Single		Extensive Form
Instance	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
1	8.45	1	10.81	1	9.09	1	10.15	1	355.2
2	15.01	1	17.48	1	12.63	1	13	1	1153.8
3	576.59	7	510.98	4	399.86	7	302.15	5	3094.88
4	105.45	3	256.18	3	147.69	3	146.3	3	1504.42
5	8.35	1	8.85	1	8.86	1	7.52	1	380.47
6	32.12	2	65.07	2	14.97	1	11.56	1	510.75
7	71.8	3	129.44	3	55.08	3	94.41	4	696.38
8	30.88	2	55.64	2	38.28	2	40.49	2	499.45
9	25.31	2	36.77	2	26.2	2	37.3	2	400.85
10	32.09	3	44.33	3	29.92	3	45.79	3	526.94
Avg	90.61	2.5	113.56	2.2	74.26	2.4	70.87	2.3	912.31

(b) Set 2 Instances

	Multicut		Quadratic		Single Cut		Quad + Single		Extensive Form
Instance	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
1	678.72	11	543.53	5	423.01	11	621.45	9	3176.91
2	17.25	1	41.56	2	12.71	1	13.95	1	1253.81
3	28.33	2	34.54	2	24.44	2	26.16	2	708.19
4	81.07	4	155.06	4	71.16	4	72.99	4	750.70
5	96.04	4	116.76	4	71.4	4	86.14	4	894.49
6	50.77	2	101.65	2	33.73	2	46.99	2	1110.11
7	95.27	2	102.86	2	81.35	2	69.42	2	2336.42
8	110.32	5	186.72	5	112.15	6	99.91	5	831.83
9	78.26	2	101.37	2	67.7	2	53.55	2	1321.32
10	56.93	4	43.83	2	38.53	3	34.91	3	667.71
Avg	129.30	3.7	142.79	3.0	93.62	3.7	112.55	3.4	1305.15

(c) Set 3 Instances

Table 6: Effectiveness of Various L-Shaped Implementations and the Extensive Form.

Another interesting question relates to the single-cut versus multicut approaches for the L-shaped method. Works by Birge and Louveaux (1988) and Gassmann (1990) conduct experiments on stochastic *linear* programs, suggesting that the multicut approach is usually preferable when the number of realizations r is not significantly larger than the number of first-stage constraints. We are unaware of similar experiments on stochastic programs with first-stage integer variables. Our findings indicate that the single-cut version is preferable even though the number of first-stage constraints greatly exceeds the number of scenarios. One reason for this behavior might be that continuous relaxations of the master problem must be solved many times within a branch-and-bound framework, making additional constraints more burdensome than when the master problem is a linear program. Further theoretical and empirical investigation is warranted.

6 Summary and Conclusions

We consider a stochastic intra-ring SONET design problem that generalizes the deterministic SONET ring design problem. We begin by providing a basic decomposition framework based on the L-shaped method, whose subproblems may be solved by a transformation to a minimum cost flow problem. Using concepts derived from a simple dual recovery algorithm for these subproblems, we derive an alternative formulation capable of generating strong optimality cuts to the master program. Following this, we identify several classes of valid inequalities along with a novel technique for including subproblem data into the master problem. These techniques for obtaining strong lower bounds to the problem are used in conjunction with an effective upper bounding heuristic executed after the solution of each master problem. We then examine the effectiveness of our methodologies on a suite of SONET test instances having varying numbers of scenarios to demonstrate the benefit

of our overall prescribed algorithm, versus simply executing a rudimentary L-shaped implementation or solving the extensive form program.

Several opportunities for future research exist. For instance, this paper only presents the case in which demand amounts are stochastic. Our basic L-shaped approach is easily extendable to the case in which the shortage penalties are stochastic as well, permitting the ring capacity b to vary with each scenario as well by scaling the demand and penalty parameters accordingly. Hence, we would implicitly absorb the uncertainty of the ring demand capacity within the problem parameters. However, this generality would make the baseline scenario addition to our problem invalid, and would thus appreciably increase the difficulty of solving the problem.

Another interesting extension of this problem might consider the addition of ADMs to the network in a future decision-making stage. Such an extension would require the repeated solution of mixed-integer second-stage problems. While the current state-of-the-art in stochastic integer programming may not permit the direct solution of such a problem, the heuristic developed in this paper could readily be modified to account for these second-stage network design decisions.

7 Acknowledgements

Cole Smith was supported by Defense Advanced Research Projects Agency Grant #N66001-01-1-8925. The authors wish to acknowledge Julie Higle for her insightful comments.

REFERENCES

BIRGE, J. R., AND F. V. LOUVEAUX. 1997. Introduction to Stochastic Programming. Springer, New York, New York.

BIRGE, J. R., AND F. V. LOUVEAUX. 1988. A Multicut Algorithm for Two-Stage Stochastic Linear Programs. European Journal of Operational Research 34(3) 384-392.

GASSMANN, H. I. 1990. MSLiP: A Computer Code for the Multistage Stochastic Linear Programming Problem. Mathematical Programming 47 237-254.

GOLDSCHMIDT, O., A. LAUGIER, AND E. V. OLINICK. 1999. SONET/SDH Ring Assignment with Capacity Constraints. Technical Report, Department of Industrial Engineering and Operations Research, University of California, Berkeley. To appear in Discrete Applied Mathematics.

LAGUNA, M. 1994. Clustering for the Design of SONET Rings in Interoffice Telecommunications. Management Science 40(11) 1533-1541.

LEE, Y., H. D. SHERALI, J. HAN, AND S. KIM. 2000. A Branch-and-Cut Algorithm for Solving an Intra-Ring Synchronous Optical Network Design Problem. Networks 35 223-232.

MAGNANTI, T. L., AND R. T. WONG. 1981. Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. Operations Research 29(3) 464-484.

SHERALI, H. D., AND W. P. ADAMS. 1990. A Hierarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems. SIAM Journal on Discrete Mathematics 3(3) 411-430.

SHERALI, H. D., AND W.P. ADAMS. 1994. A Hierarchy of Relaxations and Convex Hull Characterizations for Mixed-Integer Zero-One Programming Problems. Discrete Applied Mathematics 52 83-106.

SHERALI, H. D., AND J. C. SMITH. 2001. Improving Discrete Model Representations Via

Symmetry Considerations. Management Science 47(10) 1396-1407.

SHERALI, H. D., J. C. SMITH, AND Y. LEE. 2000. Enhanced Model Representations for an Intra-Ring Synchronous Optical Network Design Problem Allowing Demand Splitting. INFORMS Journal on Computing 12(4) 284-298.

SMITH, J. C. 2002. Algorithms for Distributing Telecommunication Traffic on a Multiple-Ring SONET-based Network. Working Paper, Department of Systems and Industrial Engineering, University of Arizona, Tucson.

SUTTER, A., F. VANDERBECK, AND L. A. WOLSEY. 1998. Optimal Placement of Add/Drop Multiplexers: Heuristic and Exact Algorithms Operations Research 46(5) 719-728.

VAN SLYKE, R., AND R.J.B. WETS. 1969. L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. SIAM Journal on Applied Mathematics 17 638-663.

WOLLMER, R. 1980. Two-Stage Linear Programming under Uncertainty with 0-1 Integer First-Stage Variables. Mathematical Programming 19 279-288.

WU, T. 1992. Fiber Network Service Survivability. Artech House, Boston, Massachusetts.

WU, T., AND M.E. BURROWES. 1990. Feasibility Study of a High-Speed SONET Self-Healing Ring Architecture in Future Interoffice Networks. IEEE Communications Magazine 28 33-43.