A Directed Cycle-Based Column-and-Cut Generation Method for Capacitated Survivable Network Design

Deepak Rajan and Alper Atamtürk
Department of Industrial Engineering and Operations Research, University of California, Berkeley, California 94720-1777

A network is said to be survivable if it has sufficient capacity for rerouting all of its flow under the failure of any one of its edges. Here, we present a polyhedral approach for designing survivable networks. We describe a mixed-integer programming model, in which sufficient slack is explicitly introduced on the directed cycles of the network while flow routing decisions are made. In case of a failure, flow is rerouted along the slacks reserved on directed cycles. We give strong valid inequalities that use the survivability requirements. We present a computational study with a column-and-cut generation algorithm for designing capacitated survivable networks. © 2004 Wiley Periodicals, Inc.

Keywords: survivable network design; models; valid inequalities; column generation

1. INTRODUCTION

Given a graph, demands between pairs of nodes (commodities) and cost of installing capacity on the edges of the graph, the capacitated network design problem (NDP) is to install integer multiples of a capacity unit and route the flow so that the total cost is minimized. A network is said to be survivable if it admits a feasible flow under the failure (removal) of any one of its edges. Existence of two edge-disjoint paths between every pair of source and destination nodes is a necessary condition for survivability of the network, but this is clearly not sufficient. To ensure that the flow on the network can be rerouted in the case of a failure, sufficient spare (excess) capacity must be available on the remaining edges of the network. Because overprovisioning of capacity is a major concern due to the high investment cost associated with installing capacity, designing capacity-efficient survivable networks is a highly critical problem in the telecommunications industry.

The capacitated survivable network design problem can be formulated as a multicommodity network flow problem for each failure scenario, linked by integral capacity variables [2]. However, there are at least two reasons as to why such a model is not used in practice. The first one is that it has a cubic number of variables and constraints in the number of edges for a complete network and is, therefore, impractical for designing networks except for very small instances. The second reason is that its solution involves globally rerouting flow on the network whether the flow of a commodity is disrupted by the failed edge or not. Solutions with minimal changes to no-failure flow are preferred because it is highly undesirable in practice to manipulate unaffected flow while restoring affected flows. Therefore, a number of practical models and strategies have been developed for designing survivable networks that admit *local* rerouting of flow on a failed edge; see for instance [3, 7, 14, 27]. The reader is referred to [25] for a survey on the survivable network design problem.

Traditionally, to design and implement survivable networks one uses some variant of the following two different strategies: *protection* or *restoration. Protection* techniques completely identify ahead of time the routes that disrupted flows will take and the capacities that will be used. *Restoration* techniques determine which available capacity will be used for a specific failure (and the routes that will be used for each affected demand) when the failure occurs. We do not discuss any dynamic restoration schemes, and focus on protection.

Dedicated protection techniques [3, 12, 14] install and assign spare capacity specifically for each commodity to protect it against the different possible failures, i.e., spare capacity is dedicated to a particular commodity. A significant reduction in the amount of spare capacity can be achieved by using a *shared* protection strategy [2, 7, 27] when dealing with failures. In a shared protection scheme, instead of preassigning spare capacity to protect each commodity of the network independently, spare capacity is shared by more than one commodity, and used as required to restore the disrupted flow.

Survivable network design strategies can also be broadly

Received October 2002; accepted November 2003
Correspondence to: D. Rajan; e-mail: deepak@ieor.berkeley.edu
Contract grant sponsor: National Science Foundation; contract grant numbers: 0070127 and 0218265.

classified into two frameworks: *hierarchical* (nonjoint) and *integrated* (joint). The hierarchical one (i.e., [17, 24]) involves a two-stage approach; first, no-failure routing and working capacities are determined, followed by rerouting of disrupted flows and spare capacities. Solving these two problems simultaneously in an integrated framework provides significant savings in installed capacity [23].

The advent of add/drop multiplexers introduced a technique known as *Self-Healing Rings* (SHRs) [3, 14, 18, 19, 25]. The topology of SHR networks is a set of rings (undirected cycles) covering the nodes of the network. SHR networks are inherently survivable, because the flow on a failed edge on a ring can be sent along the ring in the reverse direction. Even though SHRs provide good survivability characteristics and extremely fast reconfiguration of flow, imposing a ring topology on the telecommunication network leads to inefficient capacity utilization, and therefore high capacity cost.

Recently, there has been an increased effort in designing *hybrid* networks that achieve reconfiguration times comparable to the regime of rings, but without giving up the desired capacity efficiency of a general/mesh network, i.e., a network with no restriction on its topology. There are several ways in which hybrid networks are already being implemented in telecommunication networks. The first one is referred as "ring access and mesh transport." In this widely used form of hybrid networks, shared protection is applied for backbone networks, and SHRs are used for Local Area Networks (LANs). The second way of building a hybrid network is through "meshed rings," which is essentially ring-based networking with inter-ring transitions being managed by shared protection. In the third type of hybrid networks, capacity is allocated on various cycles to act as rings. Any fraction of a commodity may be routed on these rings, and is inherently survivable (see [15] for details); the rest of the commodity is routed as in a mesh network.

In this article, we focus on a fourth approach for designing hybrid networks, which involves the use of specific failure-flow patterns for shared protection of disrupted flow. However, no particular topology is imposed on the network. Thus, this approach uses a general/mesh network for no-failure routing, but utilizes specified patterns for routing flow in case of disruption. In particular, we use directed cycles as our failure-flow patterns. We explicitly reserve sufficient slack between flow on an arc and the capacity on the arc using directed cycles of the network. We ensure that the total slack on the cycles in the reverse direction is at least equal to the flow on an arc. So when an edge fails, the flow in either direction can be rerouted from between the two ends of the edge along the slacks of the cycles in the reverse direction. We first present the hierarchical optimization problem, and then extend it to the integrated optimization framework.

A related combinatorial survivable graph problem is studied in [13], where the edges of a graph are covered with directed cycles. The authors show that a bridgeless undirected planar graph* can be decomposed into directed cycles where each edge is used exactly twice (once in each direction). They conjecture that this result also applies to any bridgeless graph and give a heuristic for finding directed cycle covers of undirected graphs. In [22], the authors cover the arcs of a directed graph using an overlay graph.† These articles do not take into consideration demands on the nodes, flows, capacities, or costs.

Another related work is [16], in which the authors first determine edge capacities for the no-failure scenario, and then add sufficient capacity on undirected cycles (p-cycles) to protect working edge capacities. Their approach differs from ours in the following aspects. First, they protect working capacities and not flows; hence, their model does not utilize existing slack on the edges of the network. Second, their model allows the p-cycles to reroute disrupted flow on the chords.

The proposed models are presented in Section 2. In Section 3, valid inequalities that explicitly consider the survivability requirements are described. These inequalities are used in a column-and-cut generation algorithm to strengthen the linear programming relaxations of the formulations. In Section 4, we present computational results with the column-and-cut generation algorithm. We compare capacity-efficiency of the models and the impact of the valid inequalities in reducing the computation time when used as cutting planes. Finally, in Section 5, we conclude with a summary and directions for future research.

2. MODELS

In this section we present two mixed-integer programming models for designing capacitated survivable networks. We consider a common variant of the problem as it appears in telecommunication networks, where capacity installed on an edge can be used to send flow up to capacity in both directions. Thus, the capacity is undirected even though flow and cycles are directed. The results in this article can be easily adopted to the case where capacity is directed. We introduce sufficient slack on directed cycles of the network, so that the flow on each arc can be rerouted along these cycles in case of an edge failure. For ease of exposure and the purpose of comparison, we first discuss a hierarchical approach.

2.1. Hierarchical Approach

Here we present a hierarchical scheme for designing capacitated survivable networks. In the first stage, the capacitated network design problem without survivability requirements, NDP, is solved. In the second stage, an optimal solution to NDP (a vector of flows and edge capacities) is

* A graph is bridgeless if its nodes cannot be partitioned into two groups such that only one edge connects them.

† A graph superimposed on the original graph.

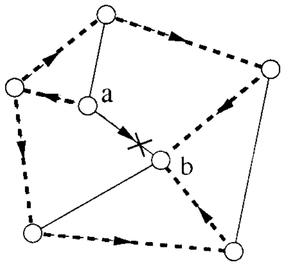


FIG. 1. Two directed cycles protecting the flow on (ab).

used as input to a spare capacity installation model, where sufficient slack is reserved on directed cycles so that the flow on each arc can be safely rerouted along these cycles.

Formally, let $G = (N, E)$ be an undirected graph with node set N and edge set E. Let F be the set of all ordered pairs (arcs) from E, i.e., $F = \{(ij), (ji) : [ij] \in E\}$. Let g_{ij} be the flow quantity through arc $(ij) \in F$, and $w_{[ij]}$ be the capacity on edge $[ij] \in E$ in the solution to NDP. Let $\mathscr{C}$ be the set of *directed* cycles of $G' = (N, F)$. Define the cycle variable z_c to denote the amount of slack reserved on cycle $C \in \mathscr{C}$. Define the capacity variable $y_{[ij]}$ as the amount of spare capacity installed on edge $[ij]$, and let $h_{[ij]}$ be the cost of installing unit capacity on edge $[ij] \in E$. Then the *Spare Capacity Installation* (SCI) problem is formulated as

$$\min \sum_{[ij]\in E} h_{[ij]}y_{[ij]}$$

$$\text{s.t.:} \quad \sum_{C\in\mathscr{C}:(ji)\in C} z_c \geq g_{ij} \quad \forall\,(ij)\in F \tag{1}$$

$$g_{ij} + \sum_{C\in\mathscr{C}:(ij)\in C} z_c \leq w_{[ij]} + y_{[ij]} \quad \forall\,(ij)\in F$$

$$y_{[ij]} \in \mathbb{Z}_+ \;\; \forall\,[ij]\in E,\; z_c \in \mathbb{R}_+ \quad \forall\, C\in\mathscr{C}. \tag{2}$$

In SCI, spare capacity on a directed cycle provides coverage for flows in the reverse direction to the arcs on the cycle (see Fig. 1). Constraints (1) ensure that for each arc (ij) the total slack reserved on the directed cycles using the reverse arc (ji) is at least the total flow on (ij). Constraints (2) ensure that total capacity installed on edge $[ij]$ is large enough to accommodate the flow routed on arc (ij) as well as the slack reserved for cycles containing the arc. Notice that by scaling the data without loss of generality (*wlog*) we may assume that the capacity is installed in batches of one unit.

In Figure 1 we show two directed cycles, drawn in dashed arcs, that cover the disrupted flow on arc (ab) in the reverse direction. If edge $[ab]$ fails, then the flow along arc (ab) can be rerouted from node a to node b along these two directed cycles if the sum of reserved slack on the cycles is at least the flow quantity on (ab). Note that these cycles are used to reroute not only the flow on arc (ab), but also of the other arcs they contain.

2.2. Integrated Approach

Next we present the integrated optimization model that makes flow routing and capacity installation decisions for no-failure and failure cases simultaneously. Because routing and capacity decisions are made together, the integrated model gives a network with a cost that is at most the optimal cost of the hierarchical scheme.

Let z, y, and h be defined as in Section 2.1. Let $\{(s_k, t_k)\}_{k\in K}$ be the commodity pairs of source and destination nodes and d_k be the supply at s_k for t_k, $k \in K$. Let b_i^k be the supply of commodity k at node i, i.e., $b_{s_k}^k = d_k$, $b_{t_k}^k = -d_k$, and $b_i^k = 0$ for $i \in N\backslash\{s_k, t_k\}$. Also define variable x_{ij}^k as the flow of commodity k through arc $(ij) \in F$ in the no-failure case, and let e_{ij}^k be the cost associated with routing each unit of commodity $k \in K$. Then the problem of *Routing of Flows and Slacks on Cycles* (RFC) is formulated as

$$\min \sum_{(ij)\in F}\sum_{k\in K} e_{ij}^k x_{ij}^k + \sum_{[ij]\in E} h_{[ij]}y_{[ij]}$$

$$\text{s.t.:} \quad \sum_{(ij)\in F} x_{ij}^k - \sum_{(ji)\in F} x_{ji}^k = b_i^k, \quad \forall\, i\in N,\ \forall\, k\in K \tag{3}$$

$$\sum_{k\in K} x_{ij}^k - \sum_{C\in\mathscr{C}:(ji)\in C} z_c \leq 0 \quad \forall\,(ij)\in F \tag{4}$$

$$\sum_{k\in K} x_{ij}^k + \sum_{C\in\mathscr{C}:(ij)\in C} z_c \leq y_{[ij]} \quad \forall\,(ij)\in F \tag{5}$$

$$y_{[ij]} \in \mathbb{Z}_+ \;\; \forall\,[ij]\in E,\; z_c\in\mathbb{R}_+ \;\; \forall\, C\in\mathscr{C},$$

$$x_{ij}^k \in \mathbb{R}_+ \;\; \forall\,(ij)\in F,\;\; \forall\, k\in K.$$

Constraints (4) ensure that for each arc (ij) the total slack reserved on the directed cycles using the reverse arc (ji) is at least the total flow on (ij). Constraints (5) ensure that capacity installed on edge $[ij]$ is large enough to accommodate the flow routed on arc (ij) as well as the slack reserved for cycles using the arc. Observe that if constraints (4) and the cycle variables z are dropped from RFC, we obtain the network design problem without survivability restrictions, NDP.

For all practical purposes, the cost of sending flow on arc (ij) is insignificant when compared to the cost of installing capacity on edge $[ij]$ ($e_{ij} \ll h_{[ij]}$); we keep e_{ij} in the formulation for generality. No cost is associated with the cycle variables, because (a) we wish to compare the cost of the survivable network with that of NDP, and (b) flow is routed using cycle variables only if there is a failure.

2.3. Pricing Cycle Variables

RFC has only one more constraint (4) for each arc than the network design problem without survivability requirements. This is a big advantage of the model in being able solve large instances. However, the number of cycle variables is exponential in the number of arcs, and all of the variables cannot be included in the formulation when solving large instances. Therefore, we develop a column generation method to include the cycle variables into the formulation as they are needed when solving its linear programming (LP) relaxation.

Given an LP relaxation solution to RFC, we search for a directed cycle C in G' that has at least three arcs such that z_c has a negative reduced cost. Let (u, v) be the dual variables corresponding to constraints (4) and (5), respectively. For each arc (ij) on cycle C, the variable z_c appears twice in the formulation: in constraint (4) for arc (ji) with coefficient -1 and in constraint (5) for arc (ij) with coefficient $+1$. Hence, the reduced cost of cycle C is $\Sigma_{ij \in C} (u_{ji} - v_{ij})$. Let $q_{ij} = u_{ji} - v_{ij}$ be the cost of arc $(ij) \in F$. Notice that because $u, v \leq 0$, q is unrestricted in sign.

Accordingly, we define the Cycle Pricing Problem (CPP): given a directed graph $G' = (N, F)$ and a cost function $q : F \mapsto \mathbb{R}$, either find a negative-cost directed cycle in G' with at least three arcs, or conclude that no such cycle exists. It is easy to show that CPP can solved with a variant of the Bellman-Ford algorithm [1] that ignores cycles with two arcs in $O(|F||N|^2)$. A delayed column generation approach that finds negative cycles with at least three arcs is developed and used to add the cycle variables into the formulation as needed.

3. STRONG VALID INEQUALITIES

Cutting plane methods have been successfully used in solving network design problems; see [5], [6], [8], [9], [10], [11], [20], [21]. In this section, we describe polyhedral inequalities that explicitly utilize the cycle variables for ensuring survivability of the network. Because the integrated model RFC produces solutions with lower cost than the hierarchical model SCI, we develop inequalities for RFC.

3.1. Residual Capacity Inequalities

The first class of valid inequalities we describe are derived from the capacity constraints (5). Inequalities that completely describe the related set $conv\{ x_{ij} \in \mathbb{R}^K_+, y_{[ij]} \in \mathbb{Z}_+ : \Sigma_{k \in K} x^k_{ij} \leq y_{[ij]}, x^k_{ij} \leq d_k \ \forall k \in K\}$ are given in [20] and shown how to separate in polynomial time in [6]. These inequalities are referred to as the residual capacity inequalities, and are of the form

$$\sum_{k \in S} (d_k - x^k_{ij}) \geq f(\lceil D \rceil - y_{[ij]}), \tag{6}$$

where $S \subseteq K$, $f = D - \lfloor D \rfloor$, and $D = \Sigma_{k \in S} d_k$.

It can be shown that all nontrivial facets of $conv\{ x_{ij} \in \mathbb{R}^K_+, z \in \mathbb{R}^{\mathscr{C}}_+, y_{[ij]} \in \mathbb{Z}_+ : \Sigma_{k \in K} x^k_{ij} + \Sigma_{c \in \mathscr{C}} z_c \leq y_{[ij]}, x^k_{ij} \leq d_k \ \forall k \in K\}$ have zero coefficient for the unbounded continuous variables z_c [4]. Therefore, residual capacity inequalities (6) are the only class of cutting planes for RFC that can be derived from the capacity constraints (5).

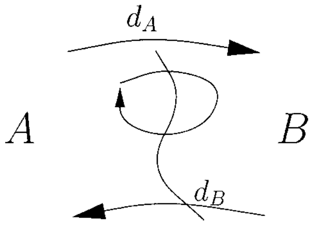


FIG. 2. Two-partition.

3.2. Two-Partition Inequalities

We derive valid inequalities for RFC from its 2-commodity 2-partition relaxations. Let (A, B) be a nonempty partitioning of the nodes of G. Let $[AB]$ be the edges with one end in A, the other in B; corresponding to these edges, let AB be the arcs directed from A to B, and BA be the arcs directed from B to A. Let d_A denote the total supply in A for nodes $N\backslash A$ (referred to as commodity A), and d_B denote the total supply in B for nodes in $N\backslash B$ (referred to as commodity B). We assume *wlog* that $d_A \geq d_B$. Consider the following 2-commodity 2-partition relaxation of RFC

$$x^A(AB) - x^A(BA) = d_A \tag{7}$$

$$x^B(BA) - x^B(AB) = d_B \tag{8}$$

$$x^A_{ij} + x^B_{ij} + z_{ij} \leq y_{[ij]} \quad \forall\ (ij) \in AB \cup BA \tag{9}$$

$$0 \leq x^A_{ij} + x^B_{ij} \leq z_{ji} \quad \forall\ (ij) \in AB \cup BA \tag{10}$$

$$z(AB) = z(BA), \tag{11}$$

where x^A_{ij} and x^B_{ij} denote the flow on arc (ij) for commodities A and B, respectively, z_{ij} the slack reserved for cycle variables on arc (ij) and $y_{[ij]}$ the capacity installed on edge $[ij]$. Constraints (7) and (8) are obtained by aggregating the flow balance constraints across the partition. Constraints (9) and (10) are the capacity and survivability constraints for the arcs in the partition. Constraint (11) states that the total slack reserved for cycles across the partition is the same in either direction, because any directed cycle that goes across the partition (using arcs in AB) has to come back across the partition (using arcs in BA) (see Fig. 2). We denote the convex hull of all points satisfying (7), (8), (9), (10), and (11) by $\mathcal{F}_2$. Clearly, all feasible points of RFC are contained in $\mathcal{F}_2$.

Before presenting the general form of 2-partition inequalities, we motivate and explain the simplest version of the inequalities. The total flow of commodity A on arcs in

AB must be at least d_A. Furthermore, total slack reserved for cycles on the arcs AB (going from A to B and back) must be sufficient to cover this flow. Consequently, the net capacity across this partition must be at least the sum of these two values, each of which must be at least d_A. Finally, because the capacity variables are integral, capacity across the partition is lower bounded by $\lceil 2d_A \rceil$, i.e.,

$$y([AB]) \geq \lceil 2d_A \rceil. \qquad (12)$$

The following theorem states that this lower bound is not only tight, but also that inequality (12) is indeed strong. Recall that by assumption $d_A \geq d_B$ and define $r_A = 2d_A - \lfloor 2d_A \rfloor$.

Theorem 3.1. *For any nonempty* 2*-partition* (A, B) *of* G *with* $|[AB]| \geq 3$, *the* 2*-partition inequality* (12) *is facet-defining for the convex hull of feasible solutions of RFC if the two subgraphs formed by the removal of the edges* $[AB]$ *are* 2*-connected, and either* $r_A > \frac{1}{2}$ *or* $d_A > \max\{d_B, 2\}$.

Proof. Consider an arbitrary equation

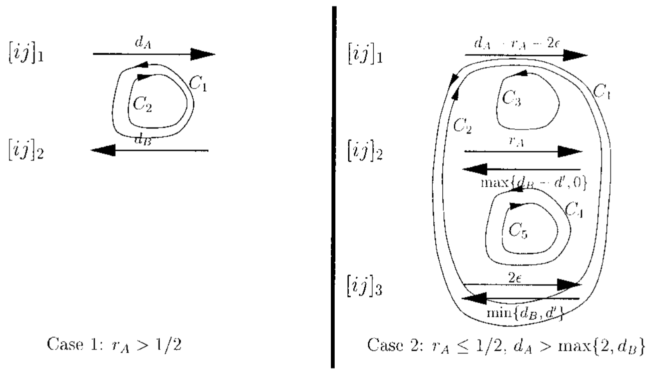
$$\sum_{(ij)\in F} \sum_{k\in K} \pi_{ij}^k x_{ij}^k + \sum_{[ij]\in E} \beta_{[ij]} y_{[ij]} + \sum_{C\in\mathscr{C}} \alpha_c z_c = \gamma \qquad (13)$$

on the variables (x, y, z). For each commodity k, consider an arbitrary arborescence (T_k) rooted at s_k (source of commodity k). Because by adding appropriate multiples of (3) (for commodity k) for all nodes in the depth-first sequence of T_k, we can eliminate the coefficients of the flow variables corresponding to arcs in T_k in (13); *wlog* we assume $\pi_{ij}^k = 0$, $\forall (ij) \in T_k$ $\forall k$. Let y' be an arbitrary feasible solution that satisfies (12) at equality. Later in the proof, we show the existence of such y' under the assumptions of the theorem. We will use y' to derive other feasible points that satisfy (12) at equality, and prove that they define (13) up to a scalar multiple, and a multiple of equalities (3). Let ϵ be an infinitesimally small positive constant, and $d' = d_A + r_A - 1$.

We first show that $\beta_{[ij]} = 0$, $\forall [ij] \in E\backslash[AB]$. For any edge $[ij] \in E\backslash[AB]$, we increase the capacity by one unit to get a new feasible solution y'' that still satisfies (12) at equality. Substituting y' and y'' into (13), we get $\beta_{[ij]} = 0$ for all $[ij] \in E\backslash[AB]$. Let $\mathscr{C}'$ be the set of all cycles that do not contain any edges in $[AB]$.

We now show that $\alpha_c = 0$, $\forall C \in \mathscr{C}'$. Consider again the feasible solution y'' obtained by adding one unit of capacity to y' on all edges $[ij] \in E\backslash[AB]$. This new solution also satisfies (12) at equality. For any cycle C, we increase the slack reserved for the cycle by one unit to get a new feasible solution y''' that also satisfies (12) at equality. Substituting y'' and y''' into (13), we get $\alpha_c = 0$, $C \in \mathscr{C}'$.

Next we prove that $\pi_{ij}^k = 0$, $\forall (ij) \in F\backslash(AB \cup BA)$, $k \in K$. Consider the feasible solution y'' obtained by adding one unit of capacity to y' on all edges $[ij] \in E\backslash[AB]$; redirecting the flow such that there exists some positive flow on all arcs in $T_k\backslash(AB \cup BA)$, for each commodity k (note that this can be done without increasing the flow on the arcs in $AB \cup BA$); and increasing the slack reserved on cycles in $\mathscr{C}'$ such that the new flow is covered (because the subgraphs obtained by removing $[AB]$ are 2-connected). Because there exists unused capacity on all edges in $E\backslash[AB]$, y'' is feasible, and also satisfies (12) at equality. For commodity k, consider any arc $(ij) \in F\backslash(AB \cup BA \cup T_k)$. We can redirect ϵ additional units of flow through this arc by changing flow by ϵ units only among the arcs in T_k (using the fundamental circuit). We satisfy (4) by reserving additional slack on cycles in $\mathscr{C}'$ that cover arcs on which flow is increased. This new solution (y''') is feasible, because unused capacity exists on these edges by definition of y'', and also satisfies (12) at equality. Substituting y'' and y''' into (13), we get $\pi_{ij}^k = 0$, $(ij) \in F\backslash(AB \cup BA)$, $k \in K$.



Case 1: $r_A > 1/2$

Case 2: $r_A \leq 1/2$, $d_A > \max\{2, d_B\}$

FIG. 3. Feasible solution y'.

For the coefficients $\beta_{[ij]}$, $[ij] \in [AB]$, we treat the case $r_A > \frac{1}{2}$ separately from $d_A > \max\{2, d_B\}$. For both cases, we only need to route commodities that have source and destination in opposite sides of the 2-partition (commodities A and B) using the arcs in $AB \cup BA$. Because the two subgraphs formed by removing $[AB]$ are 2-connected, all other commodities can be routed using arcs in $F\backslash(AB \cup BA)$, and can be covered using cycles that do not cross the 2-partition ($\mathscr{C}'$). For all feasible points considered in the rest of the proof, we install sufficiently large capacity on edges in $E\backslash[AB]$ and slack on cycles in $\mathscr{C}'$ while still satisfying (12) at equality.

Case 1. $r_A > \frac{1}{2}$. Consider the feasible solution y' shown in Figure 3. Here, flow on arcs are $x_{(ij)_1}^A = d_A$ and $x_{(ji)_2}^B = d_B$. Slack reserved on cycles are $z_{c_1} = d_A + \epsilon$ and $z_{c_2} = \epsilon$. Installed capacity on edges are $y_{[ij]_1} = \lceil d_A \rceil$ and $y_{[ij]_2} = \lceil d_A \rceil$. Because $r_A > \frac{1}{2}$, $2\lceil d_A \rceil = \lceil 2d_A \rceil$; thus y' satisfies (12) at equality. Furthermore, we can obtain some other feasible solution y'' by choosing another edge $[ij]_3$ instead of either $[ij]_1$ or $[ij]_2$ because the subgraphs obtained by removing $[AB]$ are connected, and there exists sufficient spare capacity on all the arcs. This new solution still satisfies (12) at equality. Substituting such pairs of solutions into (13), we get $\beta_{[ij]} = \beta$, $\forall [ij] \in [AB]$.

Case 2. $r_A \le \frac{1}{2}$, $d_A > \max\{2, d_B\}$. Consider the solution y' shown in Figure 3. Here, flow on arcs are $x^A_{(ij)_1} = d_A - r_A - 2\epsilon$, $x^A_{(ij)_2} = r_A$, $x^A_{(ij)_3} = 2\epsilon$, $x^B_{(ji)_2} = \max\{d_B - d', 0\}$ and $x^B_{(ji)_3} = \min\{d_B, d'\}$. Slack reserved on cycles are $z_{c_1} = d_A - 1 + \epsilon$, $z_{c_2} = \epsilon$, $z_{c_3} = 1 - r_A - 2\epsilon$, $z_{c_4} = r_A + \epsilon$, and $z_{c_5} = 3\epsilon/2$. Installed capacity on edges are $y_{[ij]_1} = \lceil d_A \rceil - 1$, $y_{[ij]_2} = 1$ and $y_{[ij]_3} = \lfloor d_A \rfloor$. Because $r_A \le \frac{1}{2}$, $\lceil d_A \rceil + \lfloor d_A \rfloor = 2\lceil d_A \rceil - 1 = \lceil 2d_A \rceil$; thus, y' satisfies (12) at equality. Furthermore, we can obtain another feasible solution y'', that satisfies (12) at equality, by interchanging the values (flow, cycle, capacity) on edges $[ij]_2$ and $[ij]_3$, because the subgraphs obtained by removing $[AB]$ are connected, and there exists sufficient spare capacity on all the arcs. Substituting such pairs of solutions y' and y'' into (13), we get $\beta_{[ij]} = \beta$, $\forall [ij] \in [AB]$.

For the rest of the coefficients, we define X_{ij} and Z_{ij} as the total flow and slack reserved for cycles on arc (ij), respectively. We note that for the solution y' (for both cases), we have $X_{ij} < Z_{ji}$ and $X_{ij} + Z_{ij} < y_{[ij]}$, $(ij) \in AB \cup BA$, whenever $y_{[ij]} > 0$. For instance, consider edge $[ij]_2$ in *Case 2*. Here, $X_{(ij)_2} = r_A$, $Z_{(ji)_2} = z_{c_4} = r_A + \epsilon > X_{(ij)_2}$ and $X_{(ji)_2} = \max\{d_B - d', 0\} < 1 - r_A$, $Z_{(ij)_2} = z_{c_3} + z_{c_5} = 1 - r_A - \epsilon/2 > X_{(ji)_2}$. Furthermore, $X_{(ij)_2} + Z_{(ij)_2} = 1 - \epsilon/2 < y_{[ij]_2}$ and $X_{(ji)_2} + Z_{(ji)_2} < 1 = y_{[ij]_2}$. Hence, we can get a new feasible solution (y'') by reserving ϵ additional units of slack on any cycle $C \in \mathscr{C} \backslash \mathscr{C}'$. This new solution ($y''$) satisfies (12) at equality. Substituting y' and y'' into (13), we get $\alpha_c = 0$, $c \in \mathscr{C} \backslash \mathscr{C}'$.

Now for commodity k, consider any arc $(ij) \in (AB \cup BA) \backslash T_k$. Starting with solution y', we can redirect ϵ additional units of flow through this arc by changing flow by ϵ units only among the arcs in T_k (using the fundamental circuit). Furthermore, we satisfy (4) by reserving additional slack on some cycle in $\mathscr{C}$ that covers arc (ij). This new solution (y''') is feasible because unused capacity exists on all edges by definition of y', and also satisfies (12) at equality. Substituting y' and y''' into (13), we get $\pi^k_{ij} = 0$, $(ij) \in (AB \cup BA) \backslash T_k$. Finally, plugging y' into (13), we get $\gamma = \lceil 2d_A \rceil \beta$. Dividing (13) by β, we arrive at (12). ■

Defining $\eta_i = \lceil 2d_i \rceil$ and $r_i = 2d_i - \lfloor 2d_i \rfloor$ for $i \in \{A, B\}$, we can generalize inequality (12) to include flow and cycle variables in both directions of the 2-partition.

Theorem 3.2. *For $S_1 \subseteq AB$, $S_2 \subseteq BA$, the 2-partition inequalities*

$$r_A y([S_1]) + (1 - r_A) y([S_2]) + x^A(AB \backslash S_1) + z(AB \backslash S_1) - x^A(S_2) - x^A(BA \backslash [S_1]) \ge r_A \eta_A \quad (14)$$

$$r_B y([S_2]) + (1 - r_B) y([S_1]) + x^B(BA \backslash S_2) + z(BA \backslash S_2) - x^B(S_1) - x^B(AB \backslash [S_2]) \ge r_B \eta_B \quad (15)$$

are valid for $\mathscr{F}_2$.

Proof. We show the validity of inequality (14). Relaxing the flow balance constraint for commodity A, we get

$$x^A(AB) \ge d_A + x^A(BA \backslash [S_1]).$$

Using (11) and the fact that the slack reserved for cycles containing arcs AB is greater than the flow in the reverse direction, we have

$$z(AB) \ge d_A + x^A(S_2).$$

Because the capacity installed on a set of arcs is greater than the net flow and the slack reserved on cycles containing these arcs, we get

$$y([S_1]) + z(AB \backslash S_1) + x^A(AB \backslash S_1) \ge x^A(S_1) + z(S_1) + z(AB \backslash S_1) + x^A(AB \backslash S_1).$$

Finally, adding these three inequalities, we have

$$y([S_1]) + x^A(AB \backslash S_1) + z(AB \backslash S_1) - x^A(S_2) - x^A(BA \backslash [S_1]) \ge 2d_A.$$

Subtracting and adding $y([S_2])$ from/to the left-hand side and applying mixed-integer rounding [26] to the resulting inequality, we obtain the inequality (14). Validity of (15) can be shown in the same way by considering commodity B instead of commodity A. ■

Note that when $S_1 = AB$ and $S_2 = \emptyset$, inequality (14) reduces to (12). For a fixed 2-partition (A, B), the separation problem of (14) is easy. Given $(\bar{y}, \bar{x}, \bar{z})$, if $r_A \bar{y}_{[ij]} < \bar{x}^A_{ij} + \bar{z}_{ij} - \bar{x}^A_{ji}$ for $(ij) \in AB$, then we include (ij) in S_1; if $(1 - r_A)\bar{y}_{[ij]} < \bar{x}^A_{ij}$ for $(ij) \in BA$, then we include (ij) in S_2. The separation for inequality (15) is similar.

Inequalities (14) are not always facet-defining for $\mathscr{F}_2$; however, they are facet-defining for the convex hull of the 1-commodity 2-partition relaxation of RFC, i.e.,

$$x^A(AB) - x^A(BA) = d_A \quad (16)$$

$$x^A_{ij} + z_{ij} \le y_{[ij]} \quad \forall\, (ij) \in AB \cup BA \quad (17)$$

$$0 \le x^A_{ij} \le z_{ji} \quad \forall\, (ij) \in AB \cup BA \quad (18)$$

$$z(AB) = z(BA) \quad (19)$$

under mild conditions.

Theorem 3.3. *Let $\mathscr{F}_1$ denote the convex hull of points satisfying the 1-commodity 2-partition relaxation (16)–(19). The inequality*

$$r_A y([S_1]) + x^A(AB\backslash S_1) + z(AB\backslash S_1) - x^A(BA\backslash[S_1]) \geq r_A \eta_A \quad (20)$$

is facet-defining for $\mathcal{F}_1$ *if and only if* $r_A > 0$ *and* $S_1 \neq \emptyset$.

Proof. First, we prove that (20) is not facet-defining for $\mathcal{F}_1$ if $r_A = 0$ or $S_1 = \emptyset$. If $r_A = 0$, (20) reduces to $z(AB\backslash S_1) + x(AB\backslash S_1) - x(BA\backslash[S_1]) \geq 0$, which is dominated by the sum of the nonnegativity constraints of $AB\backslash S_1$ and the survivability constraints of $(BA\backslash[S_1])$. On the other hand, if $S_1 = \emptyset$, then the inequality reduces to $z(AB) + x^A(AB) - x^A(BA) \geq r_A\eta_A$, which is dominated by $z(AB) = z(BA) \geq x^A(AB) \geq d_A$ (because $2d_A \geq r_A\eta_A$).

Next, we prove that (20) is facet-defining if $r_A > 0$ and $S_1 \neq \emptyset$. Let f_{ij} and g_{ij} denote the unit vectors of flow and cycle variables, respectively, for $(ij) \in AB \cup BA$, and $h_{[ij]}$ denote the unit vector of the capacity variables for $[ij] \in [AB]$.

Let $\Sigma_{(ij)\in AB\cup BA}\ (\pi_{ij}x^A_{ij} + \alpha_{ij}z_{ij}) + \Sigma_{[ij]\in[AB]}\ \beta_{[ij]}y_{[ij]} = \beta_0$ define an arbitrary hyperplane that contains the face induced by (20) and $(st) \in S_1$. Because all points of $\mathcal{F}_1$ satisfy $x^A(AB) - x^A(BA) = d_A$ and $z(AB) = z(BA)$, we may add multiples of these equalities to a valid inequality without changing it; therefore *wlog* we assume $\pi_{st} = \alpha_{st} = 0$. Consider the following points of the face. Let $u^0 = \eta_A h_{[st]} + d_A g_{st} + d_A g_{ts} + d_A f_{st}$. From points $u^0 + h_{[ij]}$, we see that $\beta_{[ij]} = 0$, $\forall [ij] \in [AB]\backslash[S_1]$. From points $u^0 + \epsilon g_{st} + h_{[ij]} + \epsilon g_{ji}$, we get $\alpha_{ij} = 0$, $\forall (ij) \in BA\backslash[S_1]$. Similarly, from the point $u^0 + \epsilon g_{st} + \epsilon g_{ts}$, we get $\alpha_{ts} = 0$. From points $u^0 + \epsilon g_{ts} + \epsilon f_{st} + h_{[ij]} + \epsilon g_{ij} + \epsilon f_{ji}$, we get $\alpha_{ij} = -\pi_{ji}$, $\forall (ij) \in AB\backslash S_1$. For the rest of the coefficients, consider points $v^{ij} = u^0 - h_{[st]} - r_A g_{ts} - r_A f_{st} + h_{[ij]} + r_A g_{ji} + r_A f_{ij}$ and $w^{ij} = u^0 - h_{[st]} - r_A g_{st} + h_{[ij]} + r_A g_{ij}$, for $(ij) \in AB\backslash\{(st)\}$. Comparing u^0 and v^{ij}, we have $\beta_{[st]} = r_A \pi_{ij}$, $\forall (ij) \in AB\backslash S_1$. On the other hand, comparing u^0 and w^{ij}, we have $\beta_{[st]} = r_A\alpha_{ij}$, $\forall (ij) \in AB\backslash S_1$. From w^{ij} and $w^{ij} + \epsilon g_{ij} + \epsilon g_{ji}$, we get $\alpha_{ij} = \alpha_{ji}$, $\forall (ij) \in S_1\backslash\{(st)\}$. Further, comparing v^{ij} and w^{ij}, we get $\pi_{ij} = 0$, $\forall (ij) \in S_1\backslash\{(st)\}$. From v^{ij} and $v^{ij} - \epsilon f_{st} + \epsilon f_{ij} + \epsilon g_{ij} - \epsilon g_{ji}$, we get $\alpha_{ij} = -\alpha_{ji}$, $\forall (ij) \in S_1\backslash\{(st)\}$ and $\alpha_{ij} = \alpha_{ji} = 0$, $\forall (ij) \in S_1\backslash\{(st)\}$. Comparing u^0 and w^{ij}, we also get $\beta_{[st]} = \beta_{[ij]}$, $\forall [ij] \in [S_1]$. Also, from points w^{ij} and $w^{ij} - \epsilon g_{st} + \epsilon g_{ts} + \epsilon f_{st} + 2\epsilon g_{ij} + \epsilon f_{ji}$, we get $\pi_{ji} = 0$, $\forall (ij) \in S_1\backslash\{(st)\}$. From the point $u^0 + \epsilon g_{st} + \epsilon g_{ts} + \epsilon f_{st} + \epsilon f_{ts}$, we get $\pi_{ts} = 0$. Finally, plugging in these coefficients for u^0, we find that $\eta_A\beta_{[st]} = \beta_0$. Therefore, the points described above define the hyperplane up to a scalar multiple and a multiple of the two equalities. Dividing all coefficients by $\beta_{[st]}/r_A$, we arrive at (20). Hence, we have shown that the face of $\mathcal{F}_1$ induced by (20) has $5|[AB]| - 2$ affinely independent points. ■

EXAMPLE 1. Here we illustrate the 2-partition inequalities on a small example. Consider the graph with four nodes in Figure 4. In this example, there are 3 commodities: $s_1 = i$, $t_1 = l$, $d_1 = 1$; $s_2 = j$, $t_2 = l$, $d_2 = 0.4$; $s_3 = k$, $t_3 = j$, $d_3 = 1.4$. Suppose in the LP solution all demands are satisfied using the arcs directly connecting the source nodes to the destination nodes, i.e., $x^1_{il} = 1$, $x^2_{jl} = 0.4$, $x^3_{kj} = 1.4$ and survivability is ensured using cycle *ijkli* with slack 1 and cycle *jklj* with slack 0.4. Thus, $z_{ij} = 1$, $z_{jk} = 1.4$, $z_{kl} = 1.4$, $z_{li} = 1$, $z_{lj} = 0.4$, and zero for all other arcs. Installing capacities fractionally gives $y_{[ij]} = 1$, $y_{[il]} = 1$, $y_{[jk]} = 1.4$, $y_{[jl]} = 0.4$, $y_{[kl]} = 1.4$. Observe that this solution satisfies all of the constraints (3)–(5).

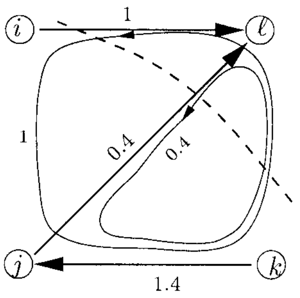


FIG. 4. Example: Two-partition inequality.

Now consider the 2-partition defined by $A = \{i, j, k\}$, $B = \{l\}$. For this partition, $d_A = 1.4$ and $d_B = 0$. Thus, we have $\eta_A = 3$, $r_A = 0.8$ and the corresponding inequality (12) is

$$y_{[il]} + y_{[jl]} + y_{[kl]} \geq 3, \qquad (21)$$

which is violated by the given fractional solution. Suppose we increase $y_{[jl]}$ until inequality (21) is no longer violated. Now, the allocation to capacity variables is $y_{[ij]} = 1$, $y_{[il]} = 1$, $y_{[jk]} = 1.4$, $y_{[jl]} = 0.6$, $y_{[kl]} = 1.4$.

For the same 2-partition $A = \{i, j, k\}$, $B = \{l\}$, inequality (14) with $S_1 = \{(il), (kl)\}$, $S_2 = \emptyset$ is

$$0.8y_{[il]} + 0.8y_{[kl]} + x^1_{jl} + x^2_{jl} - x^1_{lj} - x^2_{lj} + z_{jl} \geq 2.4. \qquad (22)$$

This inequality with continuous variables is violated by the new fractional point. Again, suppose we increase $y_{[kl]}$ until inequality (22) is no longer violated. Now, the capacity variables take the values $y_{[ij]} = 1$, $y_{[il]} = 1$, $y_{[jk]} = 1.4$, $y_{[jl]} = 0.6$, $y_{[kl]} = 1.5$. We resume this example in Section 3.3 after introducing a more general class of inequalities.

3.3. Three-Partition Inequalities

In this section we show how to generalize the 2-partition inequalities for 3-partitions of the graph. The ideas presented here can be extended to k-partitions for $k > 3$ as well.

Mixed-Integer Inequalities. Consider a nonempty 3-partition (A, B, C) of the nodes of G. As before, AB is defined as the arcs directed from A to B, and the other arc sets are defined in the same way. For each proper subset U of $\{A, B, C\}$, we again let d_U denote the total supply of U. More precisely, $d_U = \{\sum_k d_k : s_k \in U,\ t_k \in N\backslash U\}$. As before, $\eta_U = \lceil 2d_U \rceil$ and $r_U = 2d_U - \lfloor 2d_U \rfloor$. Further, we divide each set of edges into two groups ($[AB]$ into $[AB_1]$ and $[AB_2]$, etc.).

Now, each of the six proper subsets of $\{A, B, C\}$ results in a nonempty 2-partition $(U, N\backslash U)$ of the nodes of G. For each of them, we obtain a subclass of intermediate 2-partition inequalities as follows: choose $S_1 = \{(ij) : i \in U, j \in N\backslash U, [ij] \in [AB_1] \cup [BC_1] \cup [AC_1]\}$, and $S_2 = \emptyset$. Below, we present two such inequalities, corresponding to $U = \{A\}$ ($S_1 = AB_1 \cup AC_1$) and $U = \{B\}$ ($S_1 = BA_1 \cup BC_1$), respectively.

$$r_A y([AB_1] \cup [AC_1]) + x^A(AB_2 \cup AC_2) + z(AB_2 \cup AC_2) - x^A(BA_2 \cup CA_2) \geq r_A \eta_A \quad (23)$$

$$r_B y([BA_1] \cup [BC_1]) + x^B(BA_2 \cup BC_2) + z(BA_2 \cup BC_2) - x^B(AB_2 \cup CB_2) \geq r_B \eta_B \quad (24)$$

Adding (23) and (24), and mixed-integer rounding [26] the resulting inequality, we get the 3-partition inequality

$$\begin{aligned} r(r_A + r_B) y([AB_1]) &+ r_A y([AC_1]) + r_B y([BC_1]) \\ &+ x^A(AB_2 \cup AC_2) + x^B(BA_2 \cup BC_2) - x^A(BA_2 \cup CA_2) \\ &- x^B(AB_2 \cup CB_2) + z(AB_2 \cup BA_2 \cup AC_2 \cup BC_2) \\ &\geq r\eta(r_A + r_B) \quad (25) \end{aligned}$$

where

$$\eta = \left\lceil \frac{r_A\eta_A + r_B\eta_B}{r_A + r_B} \right\rceil \text{ and } r = \frac{r_A\eta_A + r_B\eta_B}{r_A + r_B} - \left\lfloor \frac{r_A\eta_A + r_B\eta_B}{r_A + r_B} \right\rfloor.$$

There are exponentially many inequalities of the form (25) depending on how $[AB]$, $[BC]$, and $[AC]$ are partitioned into groups. Nevertheless, for a fixed 3-partition (A, B, C), the separation problem for (25) is simple. Given $(\bar{y}, \bar{x}, \bar{z})$, $(ij) \in AB$ is included in AB_1 if $r(r_A + r_B)\bar{y}_{[ij]} < \bar{x}^A_{ij} - \bar{x}^A_{ji} - \bar{x}^B_{ij} + \bar{x}^B_{ji} + \bar{z}_{ij} + \bar{z}_{ji}$, in AB_2 otherwise; whereas $(ij) \in BC$ is included in BC_1 if $r_B\bar{y}_{[ij]} < \bar{x}^B_{ij} - \bar{x}^B_{ji} + \bar{z}_{ij}$, in BC_2 otherwise. Finally, $(ij) \in AC$ is included in AC_1 if $r_A\bar{y}_{[ij]} < \bar{x}^A_{ij} - \bar{x}^A_{ji} + \bar{z}_{ij}$, in AC_2 otherwise.

Similar to (25), we can get fourteen more 3-partition inequalities for the 3-partition (A, B, C) by choosing other pairs of proper subsets of $\{A, B, C\}$ to form the intermediate 2-partition inequalities ($\binom{6}{2}$ in total).

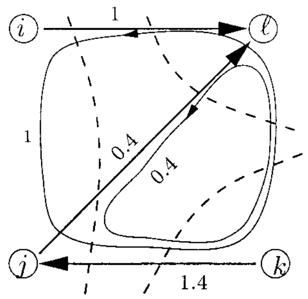


FIG. 5. Example: Three-partition inequality.

Integer Inequalities. We present another class of 3-partition inequalities written in terms of the integral capacity variables; that is, for the nonempty partition (A, B, C), we let $[AB_1] = [AB]$, $[AC_1] = [AC]$ and $[BC_1] = [BC]$. Defining S_1 and S_2 as before, we have the three 2-partition inequalities with only capacity variables

$$y([AB]) + y([AC]) \geq \lceil 2 \max\{d_A, d_{BC}\} \rceil,$$

$$y([AB]) + y([BC]) \geq \lceil 2 \max\{d_B, d_{AC}\} \rceil,$$

$$y([AC]) + y([BC]) \geq \lceil 2 \max\{d_C, d_{AB}\} \rceil.$$

By adding these three inequalities, dividing by 2 and rounding up the right-hand side, we obtain the 3-partition inequality

$$y([AB]) + y([AC]) + y([BC]) \geq \left\lceil \frac{\theta}{2} \right\rceil, \quad (26)$$

where $\theta = \lceil 2 \max\{d_A, d_{BC}\} \rceil + \lceil 2 \max\{d_B, d_{AC}\} \rceil + \lceil 2 \max\{d_C, d_{AB}\} \rceil$.

Example 1 (cont.). Consider the fractional solution $x^1_{il} = 1$, $x^2_{jl} = 0.4$, $x^3_{kj} = 1.4$; $z_{ij} = 1$, $z_{jk} = 1.4$, $z_{kl} = 1.4$, $z_{li} = 1$, $z_{lj} = 0.4$; and $y_{[ij]} = 1$, $y_{[il]} = 1$, $y_{[jk]} = 1.8$, $y_{[jl]} = 0.6$, $y_{[kl]} = 1.5$, illustrated in Figure 5.

By enumerating among all subsets S_1, S_2 for all the 2-partitions, it can again be seen that no 2-partition inequality violates this solution. However, for 3-partition $A = \{i, j\}$, $B = \{k\}$, $C = \{l\}$, we have $d_A = 1.4$, $d_B = 1.4$, $d_C = 0$, $d_{AB} = 1.4$, $d_{AC} = 0$, $d_{BC} = 1.4$. Correspondingly, $\theta = 9$ and the 3-partition inequality (26)

$$y_{[il]} + y_{[jl]} + y_{[jk]} + y_{[kl]} \geq \lceil 9/2 \rceil$$

is violated by the fractional point.

4. COMPUTATIONAL RESULTS

In this section we present computational results with a column-and-cut generation algorithm for solving the models introduced in Section 2. We compare the capacity requirements of the models and test the effectiveness of the valid inequalities given in Section 3 in reducing the solution times when used as cutting planes.

The column-and-cut generation algorithm is imple-

TABLE 1. Comparison of the models.

Size	Time/(endgap)			zub		
$\|N\|$	NDP	SCI	RFC	NDP	SCI	RFC
5	0.03	0.01	0.06	50.5	110.3	103.5
6	0.16	0.02	0.23	129.5	251.7	235.6
7	0.27	0.03	2.29	103.4	222.3	189.4
8	2.88	0.04	14.0	146.6	286.2	259.6
9	517.1	0.15	20.9	172.7	364.9	311.6
10	(1.6)	0.21	(0.3)	235.8	457.6	432.4
11	1216	0.16	(0.8)	289.3	562.7	527.1
12	(3.1)	0.30	(2.7)	326.0	641.2	592.1

mented using the callable library of CPLEX[‡] Version 8.1 Beta. All experiments are done on a 2GHz Intel Pentium4 Linux workstation with 1GB memory. The data set consists of random instances of networks with number of nodes ($|N|$) ranging from 5 to 12. The instances have 75% edge density and 50% demand density. More precisely, this means that each of the $|N|(|N| - 1)/2$ edges (and both the directed arcs corresponding to that edge) exists with probability 0.75, and that each of the $|N|(|N| - 1)$ demand pairs exists with probability 0.5. The demand values are chosen from $0.1 \times \mathit{IntUni}$ [1, 20].

The cycle variables with negative reduced cost are generated as explained in Section 2.3. Because CPLEX does not allow addition of variables to the formulation in the branch-and-bound tree, cycle variables are generated only at the root node of the tree. It must, therefore, be noted that the solutions and gap reported at termination are for the formulation with cycles generated at the root node of the tree.

In the first experiment we compare the network design model without survivability requirements, NDP, and the hierarchical and integrated models of survivable network design models using directed cycles, SCI and RFC, respectively. In particular, in Table 1 we present the time taken to solve the three models and the total capacity installed with each model. If a problem is solved within 1 hour, we report the objective value of the solution and the elapsed CPU time in seconds (`time`); otherwise, we report the objective value of the best-known feasible solution (`zub`), and the gap (`endgap`) between this solution and the best lower bound at termination (`zlb`) as a percentage of the best lower bound, i.e., endgap = 100 × (zub − zlb)/zlb.

Comparing the capacity requirements of the models, we observe that SCI needs about 100% more capacity than the no-survivability model NDP, whereas RFC requires on the average 80% more capacity. Comparing the models in terms of ease of solvability, we see that SCI takes the least amount of time. However, one must keep in mind that we first need to solve NDP before we can use its solution as an input to SCI. Interestingly, RFC is not any harder to solve than NDP. This is important, because RFC incorporates survivability. Later in this section, we will also see that RFC scales well with increasing network size when solved in a column-and-cut framework using the cutting planes proposed in Section 3.

Next, we investigate the effect of the valid inequalities described in Section 3 in reducing the number of branch-and-bound nodes (`nodes`) and solution times (`time`), when used as cutting planes to improve the linear programming relaxations. Residual capacity inequalities are added using the linear-time separation method given in [6] for each arc at the root node of the branch-and-bound tree. We enumerate all two partitions with at most three nodes in one partition and all three partitions with at most two nodes in two of the partitions, and add the corresponding inequalities with only capacity variables whenever they are violated in the tree.

In Table 2 we report the number of cuts added (`cuts`), improvement of the integrality gap at the root node (`rootimp`), the number of branch-and-bound nodes (`nodes`), and the solution times or gap at termination, with and without the polyhedral cuts. The default CPLEX cuts are added in both runs. The results for experiments using only the CPLEX cuts are reported under heading (1) and results for experiments using both CPLEX and polyhedral cuts are reported under heading (2). All runs have a time limit of 10 hours. We see in Table 2 that more than twice as much integrality gap improvement is observed at the root node when polyhedral cuts are added. This leads to significant reduction in the number of nodes and solution time. In

[‡] CPLEX is a trademark of ILOG, Inc.

TABLE 2. Effect of cutting planes.

Size	Cuts		Rootimp		Nodes		Time/(Endgap)	
$\|N\|$	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
5	24	20	76.5	77.1	47	36	0.06	0.06
6	22	24	30.6	100	279	0	0.23	0.01
7	34	35	24.0	72.0	2298	203	2.29	0.35
8	46	58	34.6	78.7	7060	547	14.0	1.76
9	50	47	38.1	77.3	7122	196	20.9	1.33
10	67	90	31.4	74.6	2,009,857	88,145	4876	267
11	116	164	41.2	67.6	2,177,271	65,744	12,296	552
12	142	188	22.3	50.0	2,111,101	1,762,640	(2.2)	(1.2)

TABLE 3. Large graphs.

Size	Cycles	Cuts	Rootimp	Nodes	(Endgap)
20	458	225	42.9	498,668	(0.58)
30	614	248	33.3	315,777	(0.44)
40	842	219	34.3	123,080	(0.21)
50	1450	279	26.4	27,791	(0.26)
60	1248	276	25.9	13,328	(0.18)
70	1546	292	20.8	1100	(0.21)

general, the improvement in the solution time is more than an order of magnitude. Based on these computations, we conclude that the cutting planes developed in Section 3 improve the performance of the algorithm significantly.

In the final experiment we tested how well the column-and-cut generation algorithm scales for large instances. For this, we ran the algorithm for instances ranging from 20 to 70 nodes for ten hours. In Table 3 we report the number of cycles added (`cycles`), the number of cuts added (`cuts`), percentage improvement at the root node (`rootimp`), total number of nodes in the branch-and-bound tree (`nodes`), and the gap at termination (`endgap`). Although none of the instances could be solved to optimality, the gap at termination was less than 1% for all instances. To some extent, the drop in the gap at termination for larger instances can be attributed to the fact that the LP relaxations of the formulation seem to get stronger with increasing problem size. This fact reiterates the scalability of our model (and methodology). These experiments suggest that the proposed methodology is a computationally effective way for designing capacitated survivable networks.

5. CONCLUSIONS

We introduced a new methodology for designing capacitated survivable networks, that explicitly reserves slack on directed cycles. We first presented the hierarchical optimization model, and then extended it to an integrated model that makes all routing and capacity decisions simultaneously. Even though the number of variables in the models is exponential in the size of the input graph, they were generated in polynomial time in a column generation framework.

We also developed strong polyhedral cutting planes for the integrated capacitated survivable network design model. Finally, we compared the models and the effectiveness of the cutting planes computationally using a column-and-cut generation algorithm. The integrated approach provided savings of about 10% over the hierarchical scheme. The polyhedral cuts reduced the solution times by an order of magnitude.

Our experiments suggest that the proposed methodology is a computationally effective way for designing capacitated survivable networks. At the same time, we can reduce the capacity requirements further by considering other failure-flow patterns; for instance, *directed* p-cycles to route disrupted flows when their chords fail as well. The models and pricing subproblems change significantly in this case. We will study the polyhedral structure of this more complicated failure flow pattern in a subsequent article.

REFERENCES

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, Network flows: Theory, algorithms, and applications, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] D. Alevras, M. Grötschel, and R. Wessäly, Cost-efficient network synthesis from leased lines, Ann Operat Res 76 (1998), 1–20.

[3] K. Altinkemer, Topological design of ring networks, Comput Operat Res 21 (1994), 421–431.

[4] A. Atamtürk, On the facets of the mixed-integer knapsack polyhedron, Math Program 98 (2003), 145–175.

[5] A. Atamtürk, On capacitated network design cut-set polyhedra, Math Program 92 (2002), 425–437.

[6] A. Atamtürk and D. Rajan, On splittable and unsplittable flow capacitated network design arc-set polyhedra, Math Program 92 (2002), 315–333.

[7] A. Balakrishnan, T.L. Magnanti, J.S. Sokol, and Y. Wang, Spare-capacity assignment for line-restoration using a single-facility type, Operat Res 50 (2002), 617–635.

[8] F. Barahona, Network design using cut inequalities. SIAM J Optimizat 6 (1996), 823–837.

[9] D. Bienstock and O. Günlük, Capacitated network design—Polyhedral structure and computation, INFORMS J Comput 8 (1996), 243–259.

[10] D. Bienstock and G. Muratore, Strong inequalities for capacitated survivable network design problems, Math Program 89 (2000), 127–147.

[11] B. Brockmüller, O. Günlük, and L.A. Wolsey, Designing private line networks—Polyhedral analysis and computation, CORE Discussion Paper 9647, Université Catholique de Louvain, 1996.

[12] S.H. Chung, H.G. King, Y.S. Yoon, and D.W. Tcha, Cost-minimizing construction of a unidirectional SHR with diverse protection, IEEE Transact Network 4 (1996), 921–928.

[13] G. Ellinas, A.G. Hailemariam, and T.E. Stern, Protection cycles in mesh wdm networks, IEEE J Selected Areas Commun 18 (2000), 1924–1937.

[14] O. Goldschmidt, A. Laugier, and E.V. Olinick, SONET/SDH ring assignment with capacity constraints, Discrete Appl Math 129 (2003), 99–128.

[15] W.D. Grover and R.G. Martens, Optimized design of ring-mesh hybrid networks, Proceedings of IEEE/VDE Design of Reliable Communication Networks 2000, 2000, pp. 291–297.

[16] W.D. Grover and D. Stamatelakis, Cycle-oriented distributed pre-configuration: Ring-like speed with mesh-like capacity for self-planning network restoration, Proceedings of IEEE International Conference on Communications, 1998, pp. 537–543.

[17] M. Herzberg, S.J. Bye, and A. Utano, The hop-limit ap-

proach for spare capacity assignment in survivable networks, IEEE/ACM Transact Network 3 (1955), 775–784.

[18] D.S. Hochbaum and E. Olinick, The bounded cycle cover problem, INFORMS J Comput 13 (2001), 104–119.

[19] H. Luss, M.B. Rosenwein, and R.T. Wong, Topological network design for SONET ring architecture, IEEE Transact Systems Man Cybernet 28 (1998), 780–790.

[20] T.L. Magnanti, P. Mirchandani, and R. Vachani, The convex hull of two core capacitated network design problems, Math Program 60 (1993), 233–250.

[21] T.L. Magnanti, P. Mirchandani, and R. Vachani, Modeling and solving the two-facility capacitated network loading problem, Operat Res 43 (1995), 142–157.

[22] M. Médard, R.A. Barry, S.G. Finn, W. He, and S.S. Lumetta, Generalized loop-back recovery in optical mesh networks, IEEE Transact Network 10 (2002), 153–164.

[23] K. Murakami and H.S. Kim, Joint optimization of capacity and flow assignment for self-healing ATM networks, Proceedings of the IEEE International Conference on Communications, 1995, pp. 216–220.

[24] H. Sakauchi, Y. Nishimura, and S. Hasegawa, A self-healing network with an economical spare-channel assignment, Proceedings of IEEE GLOBECOM 1990, 1990, pp. 438–443.

[25] P. Soriano, C. Wynants, R. Séguin, M. Labbé, M. Gendreau, and B. Fortz, "Design and dimensioning of survivable SDH/SONET networks," Telecommunications network planning, B. Sansò and P. Soriano (Editors), Kluwer Academic Publishers, The Netherlands, 1998, pp. 147–168.

[26] L.A. Wolsey, Integer programming, John Wiley and Sons, New York, 1998.

[27] Y. Xiong and L.G. Mason, Restoration strategies and spare capacity requirements in self-healing ATM networks, IEEE/ACM Transact Network 7 (1999), 98–110.