Design of Survivable Networks Using Three- and Four-Partition Facets

Yogesh Agarwal
Indian Institute of Management, Prabandh Nagar, Lucknow 226 013, India, yka@iiml.ac.in

This paper considers the problem of designing a multicommodity network with single facility type subject to the requirement that under failure of any single edge, the network should permit a feasible flow of all traffic. We study the polyhedral structure of the problem by considering the multigraph obtained by shrinking the nodes, but not the edges, in a k-partition of the original graph. A key theorem is proved according to which a facet of the k-node problem defined on the multigraph resulting from a k-partition is also facet defining for the larger problem under a mild condition. After reviewing the prior work on two-partition inequalities, we develop two classes of three-partition inequalities and a large number of inequality classes based on four-partitions. Proofs of facet-defining status for some of these are provided, while the rest are stated without proof. Computational results show that the addition of three- and four-partition inequalities results in substantial increase in the bound values compared to those possible with two-partition inequalities alone. Problems of 35 nodes and 80 edges with fully dense traffic matrices have been solved optimally within a few minutes of computer time.

Subject classifications: multicommodity flow; network design; survivability; integer programming; polyhedral structure; facet inequalities; k-partition.
Area of review: Optimization.
History: Received August 2010; revisions received November 2011, April 2012, October 2012; accepted October 2012.

1. Introduction and Problem Formulation

The standard network design problem (NDP) is to find the minimum cost installation of capacities on the edges of a graph that will permit a feasible multicommodity flow of a given set of traffic demands. In the survivable network design problem, the capacities must be installed so that a feasible flow of all traffic will be possible under certain prespecified failure scenarios or faults. The problem may have several versions depending on the permissible rerouting (restoration) options and the types of faults (edge or node) considered. In this paper, we consider only single edge failure scenarios. The problem is of considerable applied importance to telecommunications industry and has been extensively studied during the last two decades. However, the study of the polyhedral structure of the problem has received rather limited attention so far (Bienstock and Muratore 2000; Atamtürk and Rajan 2008; Balakrishnan et al. 2001, 2002). Moreover, the specific version we study here, seems to have received little attention as far as the study of polyhedral results is concerned.

There are several alternative restoration schemes to achieve survivability in telecom networks. The simplest but most expensive option is to provide for each demand, a working path, that is used when there is no failure (referred to as no-fault routing), and a dedicated edge-disjoint protection path, which is used in the event of a fault that affects the working path. This scheme is too expensive because separate protection capacity is dedicated for each demand that remains completely unused unless the demand in question is affected by a fault. A more efficient scheme is the so-called shared-protection routing, where the protection path capacity is shared among several demands that cannot simultaneously fail because their working paths are fault disjoint.

Alternatively, the most flexible scheme is one in which, for any given fault, all demands, irrespective of whether they are affected by the fault or not, are allowed to be rerouted along any arbitrary route. All we require is that under any fault a feasible routing of all demands should exist. Such a scheme has been called global rerouting by Atamtürk and Rajan (2008). Even though it may be difficult to implement such a scheme in practice, it is important to study this model because it provides a lower bound on the capacity requirement for all other restoration schemes. More importantly, the solution sets for the models of all other restoration schemes are contained within the solution set of this model. Therefore, any valid inequalities developed for this model are also valid for all other models. This is the model we study in the present work.

We consider the single-facility version of the problem, i.e., capacity on any edge can be installed only in integer multiples of a single fixed capacity, F. The traffic demands as well as the edges are assumed to be undirected. We are given an undirected graph \( G = (V, E) \) with \( n = |V| \) and \( m = |E| \), called the supply graph, and a matrix of traffic demands \( d_{ij} \). Using the aggregate commodity definition (Bienstock et al. 1998), we define a set \( K \) of \( (n - 1) \) commodities as follows. Commodity \( \kappa \in K \) represents traffic from node \( \kappa \) to all nodes \( j > \kappa \). Let \( \delta^\kappa_j \) be the demand
of commodity $\kappa$ at node $i$. Then for $1 \leq \kappa \leq n - 1$, $\delta^\kappa_i$ is defined as follows:

$$
\delta^\kappa_i = \begin{cases} 
0 & \text{for } i < \kappa \\
- \sum_{j > \kappa} d_{kj} & \text{for } i = \kappa \\
d_{ki} & \text{for } i > \kappa.
\end{cases}
$$

By rescaling the demands by a factor of $F$, we assume that each facility has unit capacity, and $c_i$ is the cost of installing one facility on edge $e \in E$. Let $x_e$ be the number of facilities installed on edge $e \in E$, and $f^\kappa_{ij}$ the (directed) flow of commodity $\kappa$ on edge $e = (i, j)$ under fault $t$, i.e., when edge $t \in E$ has failed. Then the problem can be formulated as the following mixed-integer program:

Minimize $\sum_{e \in E} c_e x_e$ \hspace{1cm} (SNDP-F)

$$
\sum_{i} f^\kappa_{ij} - \sum_{i} f^\kappa_{ji} = \delta^\kappa_j \quad \forall j \in V, \forall \kappa \in K, \forall t \in E \hspace{1cm} (1)
$$

$$
\sum_{\kappa \leq k} (f^\kappa_{ij} + f^\kappa_{ji}) \leq x_e \quad \forall e = (i, j) \in E \setminus \{t\}, \forall t \in E \hspace{1cm} (2)
$$

$$
f^\kappa_{ij} = f^\kappa_{ji} = 0 \quad \forall \kappa \in K, \forall t = (i, j) \in E \hspace{1cm} (3)
$$

$$
x_e \geq 0, \text{ integer}.
$$

We call the above formulation SNDP-F, because this formulation is known as the flow formulation in the literature. Constraints (1) are the flow conservation constraints at each node for each commodity under each fault. Constraints (2) ensure that under any fault $t$, the flow on any edge in $E \setminus \{t\}$ does not exceed the capacity of that edge. Constraints (3) ensure that there is no flow on the failed edge $t$. The number of constraints is $nm^2$ in set (1) and $(m - 1)^2$ in set (2). The problem has $m$ capacity variables, and $(n - 1)m^2$ flow variables, i.e., $(n - 1)m$ variables for each of the $m$ faults. Clearly, the size of this formulation is much larger than the standard NDP because the flow variables and all constraints are replicated for each fault. For this reason, some authors (see Atamtürk and Rajan 2008) have found this model intractable beyond 10-node problems under this formulation. In this paper, we show that it is possible to solve problems as large as 35 nodes and 80 edges with a fully dense demand matrix using an alternative formulation.

In §3, using the well-known Japanese Theorem (see Onaga and Kakusho 1971 and Iri 1971) on multicommodity flows, we reformulate the problem as a pure integer program SNDP, which is also called the capacity formulation in the literature. In this paper we study the polyhedral structure of the capacity formulation. We derive two families of three-partition-based valid inequalities, and prove the facet-defining status for one of them. It is possible to similarly derive a large number of four-partition-based inequalities. The details of some of these are provided in §6, and for the rest they are provided in the electronic companion of this paper (available at http://dx.doi.org/10.1287/opre.1120.1147).

Our approach for investigating the polyhedral structure of SNDP is based upon the previous work by this author (see Agarwal 2006, 2009) on the standard network design problem (NDP). Given a NDP defined on graph $G = (V, E)$, partitioning $V$ into $k$ subsets, and shrinking the nodes of each subset leads to a smaller $k$-node NDP. Agarwal (2006) presented a key theorem, according to which the facets of this $k$-node problem translate into facets of the original problem under mild conditions. Using this theorem, the author showed that two- and three-partition inequalities presented by other researchers were facet defining for NDP, and then derived three classes of four-partition facets by studying the four-node NDP. An additional class of four-partition facets was presented in Agarwal (2009), and it was proved that these four classes provide a complete polyhedral description of the four-node NDP. In this paper, we present and prove a similar key theorem for SNDP. By virtue of this theorem, the two-partition or cutset inequalities studied by Bienstock and Muratore (2000) are also shown to be facet defining for SNDP.

The computational results in §8 demonstrate the effectiveness of these inequalities on a number of benchmark problems. We report optimal solutions for a set of randomly generated problems with 35 nodes and 80 edges within a few minutes of central processing unit (CPU) time. We also report solutions for several problems in the SNDLib database (Orlowski et al. 2009) for which no solutions have been reported in the past.

The rest of this paper is organized as follows. After reviewing the past work in §2, the projection polyhedron of the SNDP is discussed in §3. Then, the key theorem mentioned earlier is presented in §4. In §5, after reviewing the two-partition inequalities, we present two new classes of three-partition inequalities, and develop the framework to prove their facet-defining status. Section 6 is devoted to inequalities based on four-partitions. Out of 25 such classes of inequalities, three are described in §6, and the derivations for the rest are given in the electronic companion. After a brief description of the overall implementation and separation heuristics in §7, we describe our computational experiments and their results in §8. Section 9 provides some concluding remarks with possible directions for future research on this problem.

2. Literature Review

The problem of designing survivable networks has received considerable research attention during the last two decades. The problem has been addressed in primarily three broad versions. First, there is the problem of finding a minimum cost two-connected network without any traffic or capacity considerations. Some significant contributions on this version are Monma and Shallcross (1989), Monma et al. (1990), Grötschel et al. (1992), and Grötschel and
A very recent significant contribution on a similar problem is by Balakrishnan et al. (2009).

In a second version of the problem, the no-fault routing of demands and the associated capacity network is assumed to be already given, and the problem is to find a minimum cost installation of additional spare capacity that will enable the restoration of traffic in case of any fault. Two types of restoration schemes are considered: line restoration and path restoration. In line restoration, the affected traffic is routed around the failed edge without changing the path of the traffic before or after the failed edge. On the other hand, in path restoration, the traffic is assigned a fresh end-to-end route that bypasses the failed edge. Sakacchi et al. (1990) use a cut-set-based formulation to address the line-restoration version of the problem, and Kennington and Lewis (2001) have addressed the path-restoration version. Balakrishnan et al. (2002) have used cutting planes and polyhedral approaches to solve the line-restoration version.

In the third, and perhaps most comprehensive and challenging version of the problem, which is also the focus of the present work, the network is to be designed from scratch, and no-fault routing is also a part of the design problem. This model has been called the joint capacity planning model in Kennington et al. (2007), where the authors have given mathematical programming models for various survivable network design problems. They refer to the work of Murakami and Kim (1995), Saito et al. (1998) and Irschik et al. (1998) on this problem. A rather early work on this problem is by Agarwal (1989) where three priority levels of traffic were considered, and a Lagrangian approach was proposed. Dahl and Stoer (1998) have used cutting plane approaches to solve the multifacility version of this problem. Rajan and Atamtürk (2004) and Atamtürk and Rajan (2008) have considered the design of survivable networks using directed cycles and the so-called p-cycles, respectively. In the latter case, they have used partition-based facets along with a column generation scheme to generate the p-cycles for their design.

Only a very limited investigation of the polyhedral structure of this problem has been carried out so far. The main contribution on this front has come from Bienstock and Muratore (2000), who have characterized all extreme point of the cut-set polyhedron of this problem, and thereby given all facets of the two-node multigraph. We note that the cut-set inequalities proposed in Agarwal (1989) are very similar to these.

By contrast, a considerable amount of literature exists on the polyhedral structure of the standard network design problem. Magnanti et al. (1993) introduced the cut-set inequalities and three-partition inequalities for this problem under the name of the network loading problem and proved the facet-defining status of these inequalities. Magnanti et al. (1995) have derived cut-set-based facets and some three-partition inequalities for the two-facility version of the problem. Atamtürk (2002) has studied the cut-set polyhedra of several different types of network design problems. Agarwal (2006) presented a theorem that translates the facets of the k-partition problem to the original problem and used this theorem to derive several four-partition facets. Agarwal (2009) gave a complete polyhedral description of the four-node network design problem. These same ideas have been extended to the survivable network design problem in the present work.

3. The Projection Polyhedron

According to a well-known theorem on multicommodity flows by Iri (1971) and Onaga and Kakusho (1971) (also see Lomonosov 1985), the flow conservation constraints in the formulation SNDP-F can be replaced by a set of constraints involving only the capacity variables. A vector \( \mu \in \mathbb{R}^{q} \), where \( q = \binom{n}{2} \), is said to be a metric on \( G \), if for any three distinct \( x, y, z \in V \) it satisfies \( \mu_{xy} + \mu_{yz} \geq \mu_{xz} \). Let \( c \in \mathbb{R}^{q} \) and \( d \in \mathbb{R}^{q} \), respectively, be the vectors of capacities and demands between various node pairs. This theorem, popularly referred to as the Japanese Theorem, asserts that for a multicommodity flow problem, a capacity vector \( c \) is feasible for a demand vector \( d \) if and only if the inequality \( \mu(c - d) \geq 0 \), known as a metric inequality, holds for every metric \( \mu \). One of the capacity vector is represented by variables \( x_{ij} \), and \( d_{ij} \) is the traffic demand between \( i \) and \( j \), this translates into the constraint \( \sum_{x_{ij} \geq \mu} \mu_{ij} d_{ij} \), if \( G \) is not complete, then \( x_{ij} \) for nonexistent edges is taken as zero.

The set of all metrics is infinitely large, and it forms a convex cone in \( \mathbb{R}^{q} \). A metric corresponding to an extreme ray of this cone is called a primitive metric. Because any metric can be expressed as a convex combination of primitive metrics, it is enough to ensure that the metric inequalities corresponding to all primitive metrics are satisfied, which are finite in number. Let \( G_{\mu} = (V, E) \) be the complete graph corresponding to \( G \). If \( \mathcal{M} \) is the set of all primitive metrics, then using the Japanese Theorem, SNDP-F can be reformulated as the following pure integer program:

\[
\text{Minimize } \sum_{e \in E} c_{e} x_{e} \\
\sum_{e \in E \setminus \{e\}} \mu_{e} x_{e} \geq \mu_{t} d_{e} \quad \forall \mu \in \mathcal{M}, \forall t \in E \\
x_{e} \geq 0, \quad \text{integer}.
\] (SNDP)

In this paper we study the polyhedral structure of the above program, which is also known as the capacity formulation, because it involves only the capacity variables. We note that this formulation is noncompact, because there are an exponential number of primitive metrics in \( \mathcal{M} \). However, given a solution \( \bar{x} \), a metric inequality violated by this solution can be found by solving an LP (see Avella et al. 2007). The standard NDP has been solved successfully using the capacity formulation. In our experience too, using formulation SNDP for solving the problem is a much more efficient alternative than SNDP-F, even though the problem of finding a violated metric inequality for SNDP
requires solving \( m \) LPs, one for each fault, rather just than a single LP in case of standard NDP. We describe an efficient computational shortcut pertaining to this issue in §7.1.

Although an explicit characterization of \( \mathcal{M} \) is not available, it is well known that indicator vectors of all cut sets form a very important subclass of primitive metrics (Lomonosov 1985). Consider a cut set \((S, S')\), where \( S \subseteq V \) and \( S' = V \setminus S \), and define \( \mu_{ij} = 1 \) if \( i \in S \) and \( j \in S' \), and \( \mu_{ij} = 0 \) otherwise. It is easy to verify that \( \mu \), as defined above for the cut set \((S, S')\), is a metric. The metric inequality then reduces to the form \( \sum_{i \in S} \sum_{j \in S'} x_{ij} \geq \sum_{i \in S} \sum_{j \in S'} d_{ij} \). If the right-hand side (RHS) is nonintegral, it can be rounded up, leading to the familiar rounded cut inequality that has been extensively used in the network-design literature, e.g., Barahona (1996) and Raack et al. (2011).

Let \( \mathcal{M}' \) be the subset of primitive metrics corresponding to all cut sets. It is well known that undirected graphs of up to four nodes are cut dependent, i.e., \( \mathcal{M} = \mathcal{M}' \) for problems defined on such graphs (Lomonosov 1985). In other words, for a problem with up to four nodes, if a capacity assignment satisfies all cut inequalities, then, according to the Japanese Theorem, a feasible multicommodity flow is guaranteed to exist for this capacity assignment.

In the rest of this article, we make use of these observations to explore the polyhedral structure of SNDPs of up to four nodes obtained by shrinking the nodes of the original graph. A key theorem presented next translates the facets of these smaller problems into the facets of the original problem.

4. The \( k \)-Partition Subproblem and a Key Theorem

In Agarwal (2006), the facets of the standard NDP were derived by partitioning the node-set \( V \) into \( k \) subsets: \( V_1, V_2, \ldots, V_k \), and then shrinking the nodes within each subset, as also the edges between the subsets, leading to a smaller \( k \)-node NDP. According to Theorem 1 of Agarwal (2006), the facets of such a \( k \)-node subproblems yield facets of the original problem if certain rather mild conditions are satisfied.

Here we extend the same approach for deriving the facets of SNDP and present a similar theorem applicable to the subproblems resulting from \( k \)-partitions of the SNDP graph. However, an important difference between the two problems is the following. Whereas in the case of NDP, all the edges between subsets \( V_i \) and \( V_j \) of the partition were shrunk into a single edge of the subproblem, this cannot be done in the case of SNDP. The SNDP solution must permit a feasible routing of all traffic in the event of each edge failure. Therefore, a separate identity of each edge must be maintained when defining the subproblem resulting from the \( k \)-partition of \( G \). In other words, while defining the subproblem, we shrink the nodes within each subset \( V_i \) into a single node, but the edges across the subsets are not shrunk, resulting in a \( k \)-node multigraph, which may have multiple edges between each pair of nodes. This shrinking process and the resulting \( k \)-node multigraph is illustrated with an example in Figure 1 for the case of \( k = 3 \).

Consider the partition \((V_1, V_2, \ldots, V_k)\) of \( V \) such that \( V_i \cap V_j = \emptyset \), \( \forall i, j \), and \( \bigcup_{i=1}^k V_i = V \). Define a multigraph \( G' = (V', E') \) based on this \( k \)-partition, i.e., \(|V'| = k\), \( E' = \bigcup_{i,j \in V} E_{ij} \), where \( E_{ij} = \{(p, q) \in E : p \in V_i, q \in V_j\} \). Let \( m' = |E'| \). Let \( P \) be the polyhedron defining the convex hull of solutions of SNDP. Let \( G_i = (V_i, E_i) \) be the subgraph of \( G \) induced by nodes in \( V_i \), which we also call a *component*. We assume that the edges of \( E' \) are numbered from one to \( m' \) and the remaining edges in \( E \setminus E' \) are numbered from \( m' + 1 \) to \( m \).

Let \( (SNDP') \) be the survivable network design problem defined on graph \( G' \), where the traffic demand between nodes \( i, j \in V' \) is given by \( d'_{ij} = \sum_{p \in V_i} \sum_{q \in V_j} d_{pq} \). Let \( P' \) be the polyhedron defining the convex hull of solution of \( (SNDP') \).

Let \( x \in \mathbb{R}^m \) be a capacity vector for SNDP and \( x' \in \mathbb{R}^{m'} \) a capacity vector for \( (SNDP') \). We assume that in vector \( x \), the first \( m' \) elements correspond to edges in \( E' \) and thus have a one-to-one correspondence to elements in \( x' \).

Let \( \alpha' x' \geq \beta \) be a facet-defining inequality of \( P' \). Given \( \alpha' \), let \( \alpha \) be an \( m \)-vector such that the first \( m' \)
elements of $\alpha$ are identical to those in $\alpha'$, and the remaining elements are zero. Then, Theorem 1 asserts that $\alpha x \geq \beta$ is a facet-defining inequality for SNDP if the condition of the theorem is satisfied.

**Theorem 1.** Given a facet-defining inequality $\alpha'x' \geq \beta$ of polyhedron $P'$, inequality $\alpha x \geq \beta$, with $\alpha$ constructed from $\alpha'$ as described above, is facet defining for the polyhedron $P$ if and only if the subgraph $G_i$ of $G$ induced by each $V_i$ is two-(edge)connected.

**Proof.** Because $\alpha'x' \geq \beta$ is facet defining for $P'$, there exist $m'$ affinely independent integer vectors in $\mathbb{R}^{m'}$ that satisfy $\alpha'x' = \beta$ and are feasible solutions of (SNDP'). Let these vectors be $y_1', y_2', \ldots, y_{m'}$. Using these vectors, we shall construct $m$ integer vectors in $\mathbb{R}^m$, which are affinely independent, satisfy $\alpha x = \beta$, and are feasible solutions of SNDP, thus proving that $\alpha x \geq \beta$ is facet defining for $P$.

First $m'$ vectors are constructed as follows. Given vector $y_i'$, construct $m$-vector $y_i$ so that the subvector of $y_i$ corresponding to edges in $E'$ is identical to vector $y_i'$, and each of the remaining elements that correspond to edges in $E \setminus E'$ has value $\theta$, where $\theta$ is a suitably large integer. The remaining $m - m'$ vectors are constructed as follows. Each of these vectors $y_{m+i}$, for $1 \leq i \leq m - m'$ is identical to vector $y_i$, except that the $(m'+i)$th element of the vector has value $\theta + 1$ instead of $\theta$. The resulting $m$ by $m$ matrix of these vectors is shown in Table 1. Each column of the matrix is a solution vector and each row represents an edge. First $m'$ rows correspond to edges in $E'$ and the rest to edges in $E \setminus E'$.

Next we show that the vectors of this matrix (i) satisfy $\alpha x = \beta$, (ii) are feasible solutions of SNDP, and (iii) are affinely independent.

It is easy to see from the construction of these vectors that they satisfy $\alpha y_i = \beta$ for any value of $\theta$ because $y_i'$ satisfies $\alpha'y_i' = \beta$ and the coefficient of each edge in $E \setminus E'$ is zero in $\alpha$.

To demonstrate the feasibility of vector $y_i$, we need to show that under any single edge failure, there exists a feasible routing of all traffic demands on graph $G$ for this solution. To show this, we partition the set of traffic demands into two subsets: intracomponent demands and intercomponent demands. If both end points of a traffic demand belong to the same component $G_i$, it is called an intracomponent demand. Otherwise, it is an intercomponent demand. Let $D_1$ and $D_2$, respectively, be the sets of intercomponent and intracomponent demands. Each demand in $D_1$ translates into a demand of (SNDP'), while demands in $D_2$ are absent in (SNDP').

Because each component $G_i$ is two-connected (a graph is two-connected if the removal of any single edge leaves it connected; we note that a graph with a single node trivially satisfies the definition of two-connectivity), under any single edge failure, there exists at least one path for each pair of nodes within a component that uses only the edges in $E \setminus E'$ and does not contain the failed edge. Thus, each demand in $D_2$ can be routed under any single edge failure, because there is a large capacity $\theta$ available on the edges within a component.

For solution $y_i'$, under any fault $t \in E'$, let $P'_t$ be the path of demand $k \in D_1$ on $G$. We translate this path into a path $P_t$ on $G$ as follows. Consider two consecutive edges $e_1$ and $e_2$ in the path $P'_t$. Although these edges are incident on a common node in $G'$, they need not be incident on a common node in $G$. Let the end points of $e_1$ in $G$ be $(r, q)$, and those of $e_2$ be $(r, s)$. In case $q \neq r$, there exists a path from $q$ to $r$ that uses only the edges within the component containing $q$ and $r$. We connect $q$ with $r$ using such a path, and thus obtain a complete end-to-end path for demand $k$ over graph $G$.

For solution $y_i$, under fault $t \in E'$, if all demands are routed over $G$ along the paths as constructed above, it is easy to see that such a routing is feasible for capacity vector $y_i$, if the corresponding routing of demands in $D_1$ was feasible for capacity vector $y_i'$ for (SNDP'). This is because each edge $e \in E'$ has exactly the same capacity as well as the same flow under both solutions under each fault. As for edges in $E \setminus E'$, the feasibility of the solution is ensured by choosing a sufficiently large value of $\theta$.

Under any fault $t \in E \setminus E'$, a feasible flow is clearly assured because of two-connectivity of each component $G_i$, and the fact that $\theta$ is sufficiently large.

It is easy to see that given the affine independence of $y_i'$ through $y_{m'}$, the vectors $y_1$ through $y_m$ are also affinely independent. □

The importance of this theorem lies in the fact that it opens the possibility of studying the polyhedral structure of SNDP by examining smaller instances of the problem obtained from the $k$-partitions of $G$. In the extreme case of $k = 2$, the edges in the multigraph correspond to a cut set or two-partition of $G$. Bienstock and Muratore (2000) have studied valid inequalities that are based on such two-partitions. They have enumerated all facet-defining inequalities of the two-node SNDP, which are clearly valid inequalities for the larger SNDP. However, the issue whether these inequalities are also facet defining for the larger SNDP has yet been unaddressed. The above theorem
confirms that these inequalities are indeed facet defining for the larger problem if the condition of the theorem is satisfied. More importantly, it paves the way to explore the facial structure of SNDP by studying the k-partitions for larger values of k. In the next two sections we explore such inequalities for k = 3 and k = 4.

5. Three-Partition Inequalities

Before introducing new valid inequalities based on three- and four-partitions, we briefly review the two-partition inequalities from which the new inequalities introduced in this paper are derived.

5.1. Review of Two-Partition Inequalities

These inequalities have been reported by Bienstock and Muratore (2000), Balakrishnan et al. (2002), and Magnanti and Wang (1997). Consider an SNDP defined on a multigraph \( G = (V, E) \) with \( |V| = 2 \) and \( |E| = m \). Let \( b \) be the traffic demand (rounded up if nonintegral) between the two nodes. Then, according to the Japanese Theorem, the following set of \( m \) inequalities defines the projection of this problem:

\[
\sum_{e \in t(\{1\})} x_e \geq b \quad \forall t \in E.
\]

(B1)

Here, \( t \) in the inequality label \((B1)\) denotes the index of the failed edge. We refer to these inequalities as \( B1 \)-inequalities. These inequalities have been shown to be facet defining by Bienstock and Muratore (2000) for the polyhedron of two-node SNDP with multiple edges, and hence they are also facet defining for the original SNDP by virtue of Theorem 1, if each of the two components are two-(edge)-connected.

Now consider adding all \( m \) \( B1 \)-inequalities, divide both sides by \((m – 1)\), and round up the RHS. This yields the following valid inequality:

\[
\sum_{e \in E} x_e \geq b',
\]

(B2)

where \( b' = \left\lceil \frac{bm}{m – 1} \right\rceil \). This inequality has also been shown to be facet defining for two-node SNDP under certain conditions (see Bienstock and Muratore 2000), and is therefore facet defining for the original problem by virtue of Theorem 1 if the condition of the theorem is satisfied.

In the rest of this section and the next, we derive several classes of three-partition and four-partition inequalities by combining \( B1 \)- and \( B2 \)-inequalities described above, and prove the facet-defining status for some of them. For convenience, we refer to \( B1 \)- and \( B2 \)-inequalities together as \( B \)-inequalities. The letter “\( B \)” in this labelling denotes a bi-partition. Similarly, inequalities derived from a three-partition (or tri-partition) are referred to as \( T \)-inequalities, and those from a four-partition (tet\( \hat{K} \)-partition) as \( R \)-inequalities.

5.2. The Three-Node SNDP

Consider a three-node SNDP defined on a multigraph \( G \). A three-node SNDP has three two-partitions: 1:23, 2:13, and 3:12. Note that for the sake of brevity, we use 1:23 to represent the partition \((\{1\}, \{2, 3\})\). Let the traffic demands (rounded upward) across these partitions be \( b_1, b_2, \) and \( b_3 \), respectively. Let \( E_{ij} \) represent the index set of edges between nodes \( i \) and \( j \), and \( m_{ij} = |E_{ij}| \).

As per the Japanese Theorem, the projection of this three-node SNDP has the following constraint sets comprising of \( B1 \)-inequalities:

\[
\sum_{e \in \{E_{12} \cup E_{13}\}} x_e \geq b_1 \quad \forall t \in E_{12} \cup E_{13} \tag{B1_1}
\]

\[
\sum_{e \in \{E_{12} \cup E_{23}\}} x_e \geq b_2 \quad \forall t \in E_{12} \cup E_{23} \tag{B1_2}
\]

\[
\sum_{e \in \{E_{13} \cup E_{23}\}} x_e \geq b_3 \quad \forall t \in E_{13} \cup E_{23} . \tag{B1_3}
\]

Note that the superscript in the label of each inequality denotes the specific two-partition of the three-node problem for which it is applicable. This additional notation that precisely labels each inequality is necessary for further discussion, where we derive new valid inequalities using specific combinations of some of these \( B \)-inequalities. There are a total of \( 2(m_{12} + m_{13} + m_{23}) \) of these \( B1 \)-inequalities in a three-node problem. In addition, we have the following three \( B2 \)-inequalities, one for each two-partition:

\[
\sum_{e \in \{E_{12} \cup E_{13}\}} x_e \geq b_1' \tag{B2_1}
\]

\[
\sum_{e \in \{E_{12} \cup E_{23}\}} x_e \geq b_2' \tag{B2_2}
\]

\[
\sum_{e \in \{E_{13} \cup E_{23}\}} x_e \geq b_3' \tag{B2_3}
\]

where \( b_i' = \left\lceil \frac{(m_{ij} + m_{ik})b_j}{(m_{ij} + m_{ik}) - 1} \right\rceil \), where \((i, j, k)\) is any permutation of \((1, 2, 3)\).

5.3. Inequality T1

For a given \( t \in E_{12} \), consider adding the inequalities \( B1_1, B1_2, \) and \( B2_1 \), and dividing the resulting inequality by 2. If \( b_1 + b_2 + b_3' \) is odd, then the resulting inequality can be strengthened by rounding up the RHS. This yields the following valid inequality:

\[
\sum_{e \in t(\{1\})} x_e \geq \alpha_1',
\]

(T1)

where \( \alpha_1' = \left\lceil \frac{b_1 + b_2 + b_3'}{2} \right\rceil \). If \( b_1 + b_2 + b_3' \) is odd, the resulting inequality is nonredundant. In this inequality, all edges have unit coefficients except for edge \( t \in E_{12} \), which has coefficient zero. Edge \( t \) is called the excluded edge, and we note that several inequalities derived in the rest of this paper have either one or two such excluded edges. The specific edge \( t \) that is excluded is indicated in the superscript
of the label of the generated inequality. Moreover, we use the notation \( \tilde{E}_{ij} \) for \( E_{ij} \setminus \{t\} \), when \( t \in E_{ij} \).

The derivation is further clarified by looking at the coefficients of different edge sets in the \( B \)-inequalities from which this inequality is derived, as shown in Table 2 and the adjoining figure.

The table shows the coefficients of various edges in each of the inequalities used in the derivation. The last row shows the left-hand side coefficients and the RHS of the derived inequality. In the adjoining figure (Figure 2), the thin line labeled \( t \) represents a single edge, and the thick lines represent the aggregation of all remaining edges on the multieedge. Note that edge \( t \) is the excluded edge of the generated inequality. The dotted line curves depict the two-partitions corresponding to the \( B1 \)-inequalities being combined to derive the new inequality, and the dashed-line curve represents the \( B2 \)-inequality. A dot placed at the intersection of the dotted line curve and the thin line means that the edge represented by the thin line is excluded from the \( B1 \)-inequality represented by the dotted line curve. In the table, under the column labeled \( t \), although all coefficients are zero, some of these coefficients are enclosed in parentheses to indicate the fact that the coefficient would have been 1, had it not been an excluded edge. We depict the derivations of all inequalities derived in the rest of this paper and the electronic companion with a similar table and an adjoining diagram.

We note that it is possible to generate similar inequalities for any \( t \in E_{13} \) or \( t \in E_{23} \) as well by appropriately selecting the \( B \)-inequalities to combine. Inequality \( T1 \) represents a very large class of inequalities as there are as many as \( m_{13} + m_{12} + m_{23} \) inequalities of this type for each three-partition of the original graph.

### Table 2. Derivation of inequality \( T1 \).

<table>
<thead>
<tr>
<th>Inequality</th>
<th>( t \in E_{12} )</th>
<th>( \tilde{E}_{12} )</th>
<th>( E_{13} )</th>
<th>( E_{23} )</th>
<th>( \geq )</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B1^1 )</td>
<td>(0)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>( \geq )</td>
<td>( b_1 )</td>
</tr>
<tr>
<td>( B1^2 )</td>
<td>(0)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>( \geq )</td>
<td>( b_2 )</td>
</tr>
<tr>
<td>( B2^3 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( \geq )</td>
<td>( b_3' )</td>
</tr>
<tr>
<td>( T1' )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( \geq )</td>
<td>( \alpha_i' )</td>
</tr>
</tbody>
</table>

\[ \alpha_i' = [(b_1 + b_2 + b_3')/2] \]

#### Figure 2. Derivation of \( T1 \).
3. \[ \left[ \frac{X_{ij} / m_{ij}}{m_{ij}} \right] \leq b_i' + Y_k - b_k \quad \forall (i, j) \in E^k \forall k \in \Omega_o, \text{ where} \]
\[ Y_k = \sum_{(i, j) \in E^k} X_{ij} - b_i' \] is the slack of condition 2 above.

The reader is referred to Appendix A in the electronic companion of this paper for the proof.

For a full-dimensional polyhedron \( P \in \mathbb{R}^m \), the facet-defining status of an inequality is usually proved by demonstrating the existence of \( m \) affinely independent integer feasible solutions that satisfy the inequality with equality. For a valid inequality \( \pi x \geq \pi_0 \) derived for a three- or four-node SNDP, we propose to prove its facet-defining status by first finding a set of \( m \) aggregate solutions, and then showing by virtue of the next lemma, again stated without the proof, that if these solutions satisfy certain conditions, we can construct, using Lemma 1, the \( m \) solutions needed for the facet proof.

**Lemma 2.** Given \( m \) affinely independent aggregate integer solutions \( X^1, X^2, \ldots, X^m \), each satisfying all the conditions of Lemma 1, and satisfying \( \Pi X = \pi_0 \), it is possible to construct \( m \) affinely independent integer feasible solutions \( x^1, x^2, \ldots, x^m \) that satisfy \( \pi x = \pi_0 \). If for each \( (i, j) \in E \), there exists \( k \) between 1 and \( m \), such that \( \text{mod}(X_{ij}^k, m_{ij}) \neq 0 \).

Again, the reader is referred to Appendix B in the electronic companion for the proof.

Given these lemmas, our proof approach works by finding a set of \( m \) affinely independent aggregate solution vectors that are integral, satisfy the inequality with equality, and also satisfy the conditions of both the above lemmas. Then the facet proof follows by virtue of these lemmas.

### 5.5. Facet-Defining Status of \( T^1 \)

In light of the above discussion, note that in the context of inequality \( T^1 \), \( \Omega_o \) contains cut set 12:3 and \( \Omega_1 \) contains cut sets 1:23 and 2:13. Then, according to conditions 1 and 2 of Lemma 1, each aggregate solution \( X \) needed for the facet proof must satisfy the following:

\[ X_{12} + X_{13} = b_1 + Y_1 \quad (4) \]
\[ X_{12} + X_{23} = b_2 + Y_2 \quad (5) \]
\[ X_{13} + X_{23} = b_3 + Y_3 \quad (6) \]

Note that we have turned the three \( B \)-inequalities into equations by introducing aggregate surplus variables \( Y_i \). Clearly, for a given aggregate solution vector \( X \), there is unique surplus vector \( Y \), and vice versa. A solution \( X \) satisfies (4)–(6), if its corresponding \( Y \) is nonnegative. From the derivation of \( T^1 \) it is clear that a solution \( X \) will satisfy \( T^1 \) with equality, if and only if \( Y_1 + Y_2 + Y_3 = 1 \). We call this equation the characteristic equation of inequality \( T^1 \).

To show the facet-defining status of \( T^1 \), we need to produce three nonnegative aggregate solutions \( X \) whose corresponding \( Y \)-vectors are also nonnegative, and these solutions should satisfy the conditions of Lemmas 1 and 2. Rather than producing these solutions directly, we produce their corresponding \( Y \)-vectors, and demonstrate that the solutions corresponding to these \( Y \)-vectors meet the conditions of the two Lemmas, if certain conditions are satisfied. According to a theorem given in Agarwal (1998), if the \( Y \)-vectors are affinely independent, so are their corresponding solution vectors.

**Theorem 2.** Inequality \( T^1 \) for \( t \in E_{12} \) defines a facet of the three-node SNDP if and only if the following conditions are satisfied:

1. \((b_1 + b_2 + b_3) \) is odd
2. \( m_{12} \geq 2 \)
3. \((b_1 + b_2 - b_3) \geq 1 \)
4. \([(b_1 + b_3 - b_2 + 1)/2m_{13}] \leq (b_2 - b_3) \)
5. \([(b_2 + b_3 - b_1 + 1)/2m_{23}] \leq (b_2 - b_3) \).

**Proof.** Given a surplus vector \( Y^k \), the unique aggregate solution \( X^k \) of Equations (4)–(6) corresponding to \( Y^k \)

\[ X_{12}^k = 0.5[(b_1 + b_2 - b_3) + (Y_1^k + Y_2^k - Y_3^k)] \]
\[ X_{13}^k = 0.5[(b_1 + b_3 - b_2) + (Y_1^k + Y_3^k - Y_2^k)] \]
\[ X_{23}^k = 0.5[(b_2 + b_3 - b_1) + (Y_2^k + Y_3^k - Y_1^k)]. \]

Consider three surplus vectors: \( Y^1 = (1, 0, 0) \), \( Y^2 = (0, 1, 0) \), and \( Y^3 = (0, 0, 1) \) and their corresponding aggregate solutions. We show that these solutions are feasible, integral, affinely independent, and satisfy the conditions of Lemmas 1 and 2, if the conditions of the theorem are satisfied. The proof then follows from Lemmas 1 and 2.

First we show the necessity of the conditions of the theorem. If \((b_1 + b_2 + b_3) \) is even, the new inequality can be expressed as a linear combination of inequalities it is derived from, and cannot be facet defining. If \( m_{12} = 1 \), \( E_{12} \) has only one edge that is excluded from the inequality, and \( X_{12} \) must necessarily be zero in each feasible solution. Thus, these solutions cannot be affinely independent. If condition 3 is not satisfied, \( X_{12} \) will be negative in the solution corresponding to \( Y = (0, 0, 1) \), and it is not possible to construct three distinct feasible solutions that satisfy the inequality with equality. Conditions 4 and 5 only ensure that condition 3 of Lemma 1 is satisfied by the \( X \)-vectors generated as above.

Because the three \( Y \)-vectors satisfy the characteristic equation of \( T^1 \), their corresponding solution vectors satisfy \( T^1 \) with equality. It is easy to show that if \((b_1 + b_2 + b_3) \) is odd, so is \((\pm b_1 \pm b_2 \pm b_3) \). This, together with the fact that \( Y_1 + Y_2 + Y_3 = 1 \), ensures that each \( X \)-solution is integral. The three \( Y \)-vectors are clearly affinely independent, implying the affine independence of their corresponding \( X \)-vectors. Next we show that these solutions are nonnegative. Condition 3 together with the observation that \((Y_1 + Y_2 - Y_3) \geq -1 \) ensures the nonnegativity of \( X_{12} \) in each solution. It can be shown that for a three-node multicommodity flow problem, \((b_i + b_j - b_k) \) is always
nonnegative (see Agarwal 2006), where \((i, j, k)\) is any permutation of \((1, 2, 3)\). This, together with the fact that \(b'_i \geq b_i\) for \(i \in \{1, 2, 3\}\), implies the nonnegativity of \(X_{13}\), and \(X_{23}\).

The following table shows various cases of condition 3 of Lemma 1 applied to multiedge \((1, 3)\) for different \(Y\)-vectors:

\[
Y^1: \frac{(b_i + b_j - b_2 + 1)}{2m_{12}} \leq b'_j - b_j \\
Y^2: \frac{(b_i + b_j - b_2 - 1)}{2m_{12}} \leq b'_j - b_j \\
Y^3: \frac{(b_i + b_j - b_2 + 1)}{2m_{12}} \leq b'_j - b_j + 1.
\]

It is clear that the first condition, which is the same as condition 4 of the present theorem, implies the remaining two. Thus, condition 4 ensures that condition 3 of Lemma 1 is satisfied for multiedge \((1, 3)\). Similarly, condition 5 ensures the same for edge \((2, 3)\).

Thus, we have shown that the aggregate solution vectors corresponding to the three surplus vectors are integral, feasible, affinely independent, and satisfy \(T1\) with equality.

To complete the proof, we also need to show that condition 3 of Lemma 2 is satisfied, i.e., for each \((i, j)\) one of the three solutions satisfies \(\text{mod}(x_{ij}^k, m_{ij}) \neq 0\) for some \(k \in \{1, 2, 3\}\). Note that each variable has the same value in two out of three solutions, but in the third solution the value differs by one. Therefore, one of these two values must satisfy \(\text{mod}(x_{ij}^k, m_{ij}) \neq 0\). \(\square\)

### 5.6. Inequality T2

Consider adding the inequalities \(B2^1, B2^2,\) and \(B2^3\), and dividing the resulting inequality by 2. If \((b'_i + b'_j + b'_i)\) is odd, then the inequality can be strengthened by rounding up the RHS. This yields the following valid inequality:

\[
\sum_{e \in k} x_e \geq \alpha_2, \tag{T2}
\]

where \(\alpha_2 = \frac{(b'_i + b'_j + b'_i)}{2}\). If \((b'_i + b'_j + b'_i)\) is odd, the resulting inequality is nonredundant. This inequality is similar to the “\(p\)-partition” inequality mentioned by Balakrishnan et al. (2002) in the context of the spare-capacity version of the problem addressed by them. The theorem below states that this inequality is facet defining for the three-node SNDP (joint-capacity version considered here) if the conditions of the theorem are satisfied. Hence, by virtue of Theorem 1, it defines a facet of the original SNDP whose three-partition led to the three-node problem.

**Theorem 3.** Inequality \(T2\) defines a facet of the three-node SNDP if \((b'_i + b'_j + b'_i)\) is odd, and the following conditions are satisfied:

1. \[\frac{(b'_i + b'_j - b'_2 + 1)}{2m_{12}} \leq \min\{b'_i - b_i, (b'_j - b_j)\}\]
2. \[\frac{(b'_i + b'_j - b'_2 + 1)}{2m_{12}} \leq \min\{b'_i - b_i, (b'_j - b_j)\}\]
3. \[\frac{(b'_i + b'_j - b'_2 + 1)}{2m_{12}} \leq \min\{b'_i - b_i, (b'_j - b_j)\}\]
4. \[b'_i + b'_j - b'_2 \geq 1\] for all permutations \((i, j, k)\) of \((1, 2, 3)\).

We omit the proof, as it is quite similar to the proof of Theorem 2.

### 6. Four-Partition Inequalities

In this section we explore the four-node SNDP and discuss four broad classes of valid inequalities: \(R_1, R_2, R_3,\) and \(R_4\), which are motivated by similarly named four-partition inequalities of the standard NDP analyzed in Agarwal (2006, 2009). Although we prove the facet-defining status for only one of these inequalities (see Appendix C of the electronic companion), it may be possible to prove the facet-defining status of other inequalities using similar approaches. We note that the counterparts of these inequalities for the standard NDP have been shown to be facet-defining in Agarwal (2006, 2009). We plan to undertake the task of exploring the facet-defining status of these inequalities separately in a future work. Irrespective of their facet-defining status, our computational results in §8 demonstrate that these inequalities lead to a significant reduction in the integrality gap over and above that obtained from two- and three-partition inequalities. Correspondingly, there is a substantial reduction in the computer time and the branch-and-bound tree size, if the problem is solved after adding these inequalities at the root node.

Consider a four-node SNDP defined on a multigraph \(G = (V, E)\), with \(E = \cup_{i,j \in V} E_{ij}\), with \(m_{ij} = |E_{ij}|\), and \(m = |E|\). The four-node problem is cut dependent, and therefore, if the capacity across each cut is not less than the demand across the cut, the Japanese Theorem guarantees a feasible multicommodity flow of all traffic. A four-node problem has exactly seven cuts or two-partitions. Four of these are \(1:3\) partitions and the remaining three are \(2:2\) partitions. Specifically, these partitions are \(1:234, 2:134, 3:124, 4:123,\) and \(12:34, 13:24, 14:23\). Given a partition \((S, V\backslash S)\) of nodes in \(V\), let \(\delta(S) = \{(i, j): i \in S, j \in V\backslash S\}\), and \(m(\delta(S)) = |\delta(S)|\). For convenience of notation, we use \(\delta_i\) in place of \(\delta(i)\) and \(\delta_{ij}\) in place of \(\delta((i, j))\).

The constraint set of the projection of a four-node SNDP has the following \(B1\)-inequalities:

\[
\sum_{e \in k_i \backslash \{i\}} x_e \geq b_i \quad \forall t \in \delta_i, \forall 1 \leq i \leq 4 \quad (B1_i) \tag{7}
\]

\[
\sum_{e \in k_j \backslash \{j\}} x_e \geq b_j \quad \forall t \in \delta_j, \forall 2 \leq j \leq 4. \quad (B1_{1,j}) \tag{8}
\]

Note that (7) represents four groups of constraints, one for each of the four \(1:3\) partitions, and (8) represents three groups, one for each of the three \(2:2\) partitions. The superscript in the label of each inequality specifies the nodes in the left subset of the partition in each case. Also note that
node 1 is always assumed to be in the left subset in case of a 2:2 partition.

In addition, we have the following seven B2-inequalities, one for each two-partition:

\[ \sum_{e \in E_i} x_e \geq b'_i \quad \forall 1 \leq i \leq 4 \quad (B2') \]  
\[ \sum_{e \in E_{ij}} x_e \geq b'_{ij} \quad \forall 2 \leq j \leq 4. \quad (B2^{ij}) \]

where \( b'_i = [b_i \cdot m(\delta_i)/(m(\delta_i) - 1)] \), and \( b'_{ij} \) is similarly defined.

In addition to the B-inequalities above, a four-node SNDP has six three-partitions, and the associated three-partition inequalities. A three-partition of the four-node problem is obtained by shrinking any two nodes, say \( i \) and \( j \). Thus, we can associate each three-partition with the multiedge \( (i, j) \) that has been shrunk to obtain it. Accordingly, inequalities labeled \( T1'_{ij} \) and \( T2'_{ij} \) refer to the inequalities associated with the three-partition obtained by shrinking the multiedge \((i, j)\). The RHS of these inequalities are denoted by \( \alpha'_{ij} \) and \( \alpha''_{ij} \), respectively. Recall that in case of \( T1' \)-inequality, the value of \( \alpha'_{ij} \) will depend on the specific multiedge to which \( t \) belongs, as that will determine the specific B-inequalities used in the derivation of \( T1' \)-inequality. For example, when deriving \( T1'_{12} \) when \( t \in E_{13} \cup E_{23} \), we have \( \alpha'_{12} = \lfloor (b_{12} + b_{13} + b_{23})/2 \rfloor \). However, if \( t \in E_{24}, \) we have \( \alpha'_{12} = \lfloor (b_{12} + b_3)/2 \rfloor. \) We emphasize this point in order to underline the fact that while referring to \( \alpha'_{ij} \), one should be conscious of the multiedge to which \( t \) belongs. This point is particularly relevant because the derivations of some inequalities in classes R3 and R4 involve B- as well as T-inequalities.

In the rest of this section we describe the details of class R1 and its subclasses, while the derivations of the remaining three classes and their subclasses are provided in Appendix D of the electronic companion. Each of these classes is divided into three distinct subclasses, which are labeled A, B, and C, respectively. Furthermore, in many instances, a subclass would have more than one distinct inequalities, which are labeled with lowercase letters: \( a, b, \) etc. We have derived a total of 25 distinct four-partition inequalities. Thus, R2Ab denotes the second inequality in class R2 and subclass A (or type A). Whereas R1 and R2 inequalities are obtained by combining B-inequalities alone, R3 and R4 are obtained by combining both B- and T-inequalities.

### 6.1. Inequality R1A

For \( t_1 \in E_{12} \) and \( t_2 \in E_{34} \), consider adding the inequalities: \( B1^{1}_{t_1}, B1^{2}_{t_1}, B1^{3}_{t_1}, \) and \( B1^{4}_{t_1} \), and dividing the resulting inequality by 2. If \( (b_1 + b_2 + b_3 + b_4) \) is odd, then the inequality can be strengthened by rounding up the RHS, yielding the following valid inequality:

\[ \sum_{e \in E \setminus \{t_1, t_2\}} x_e \geq \beta_{1A}. \quad (R1A^{t_1-t_2}) \]

where \( \beta_{1A} = \lfloor (b_1 + b_2 + b_3 + b_4)/2 \rfloor \). Table 3 and the adjoining Figure 3, which follow the same convention as Table 2, further clarify the derivation. It is important to note that there are several choices available for the edge-pair \((t_1, t_2)\) for this inequality. The inequality is valid as long as \( t_1 \) and \( t_2 \) belong to nonadjacent multiedges by appropriate selection of B1-inequalities. For example, we may have \( t_1 \in E_{13} \) and \( t_2 \in E_{23} \), or \( t_1 \in E_{14} \) and \( t_2 \in E_{23} \). However, all these cases are isomorphic to each other under reindexing of nodes. Thus, R1A represents a large class of inequalities for the four-node SNDP. Ideally, the indices \( t_1 \) and \( t_2 \) should be mentioned in the label of the inequality, but we avoid it for the sake of simplicity of notation, as it would usually be clear from the context.

Inequality R1A is facet defining under some conditions. The derivation and proof of these facet-defining condition-sis given in Appendix C of the electronic companion.

### 6.2. Inequality R1B

In inequality R1A, two nonadjacent edges \( t_1 \in E_{12} \) and \( t_2 \in E_{34} \) were excluded from the inequality. By replacing \( B1^{4}_{t_1} \) and \( B1^{4}_{t_2} \) with \( B2^{3} \) and \( B2^{4} \), respectively, in the derivation of R1A, we obtain a valid inequality in which only one edge, i.e., \( t_1 \) is excluded. This inequality is

\[ \sum_{e \in E \setminus \{t_1, t_2\}} x_e \geq \beta_{1B}. \quad (R1B^{t_1-t_2}) \]

where \( \beta_{1B} = \lceil (b_1 + b_2 + b_3 + b_4)/2 \rceil \). As before, the derivation of the inequality is further illustrated by Table 4 and the adjoining Figure 4. In the figure, B1 and B2 inequalities are distinctly shown with dotted and dashed line curves, respectively.

### Table 3. Derivation of inequality R1A.

<table>
<thead>
<tr>
<th>Inequality</th>
<th>( t_1 )</th>
<th>( E_{12} )</th>
<th>( E_{13} )</th>
<th>( E_{14} )</th>
<th>( E_{23} )</th>
<th>( E_{24} )</th>
<th>( E_{34} )</th>
<th>( t_2 )</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1_{12}^{1}</td>
<td>(0)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( b_{1} )</td>
</tr>
<tr>
<td>B1_{12}^{2}</td>
<td>(0)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( b_{2} )</td>
</tr>
<tr>
<td>B1_{12}^{3}</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(0)</td>
<td>( b_{3} )</td>
</tr>
<tr>
<td>B1_{12}^{4}</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>(0)</td>
<td>( b_{4} )</td>
</tr>
<tr>
<td>R1A</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>( \beta_{1A} )</td>
</tr>
</tbody>
</table>

\( \beta_{1A} = \lceil (b_1 + b_2 + b_3 + b_4)/2 \rceil \).
Table 4. Derivation of inequality $R1B$.

<table>
<thead>
<tr>
<th>Inequality</th>
<th>$E_{12}$</th>
<th>$E_{13}$</th>
<th>$E_{14}$</th>
<th>$E_{23}$</th>
<th>$E_{24}$</th>
<th>$E_{34}$</th>
<th>$\geq$ RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B1^1_l$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$b_1^l$</td>
</tr>
<tr>
<td>$B1^2_l$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$b_2^l$</td>
</tr>
<tr>
<td>$B2^3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$b_3^l$</td>
</tr>
<tr>
<td>$B2^4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$b_4^l$</td>
</tr>
<tr>
<td>$R1B^l$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$\beta_{1B}$</td>
</tr>
</tbody>
</table>

$\beta_{1B} = [(b_1^l + b_2^l + b_3^l + b_4^l)/2], t_1 \in E_{12}$

Figure 4. Derivation of $R1B$.

6.3. Inequality $R1C$

In this case, all four inequalities used in the derivation are $B2$ inequalities rather than $B1$ inequalities. Accordingly, none of the edges is excluded from the inequality. The derivation is clear from Table 5 and the adjoining Figure 5.

Table 5. Derivation of inequality $R1C$.

<table>
<thead>
<tr>
<th>Inequality</th>
<th>$E_{12}$</th>
<th>$E_{13}$</th>
<th>$E_{14}$</th>
<th>$E_{23}$</th>
<th>$E_{24}$</th>
<th>$E_{34}$</th>
<th>$\geq$ RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B2^1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$b_1^l$</td>
</tr>
<tr>
<td>$B2^2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$b_2^l$</td>
</tr>
<tr>
<td>$B2^3$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$b_3^l$</td>
</tr>
<tr>
<td>$B2^4$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$b_4^l$</td>
</tr>
<tr>
<td>$R1C$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$\beta_{1C}$</td>
</tr>
</tbody>
</table>

$\beta_{1C} = [(b_1^l + b_2^l + b_3^l + b_4^l)/2]$.

Figure 5. Derivation of $R1C$.

6.4. Inequality Classes $R2$, $R3$, $R4$, and Subclasses $A$, $B$, $C$

Having seen an example of subclasses $A$, $B$, and $C$ for inequality $R1$, we now formally define these subclasses. Inequality $R1$ is derived by adding four $B$-inequalities. Each $B$-inequality has two versions, namely, $B1$ and $B2$, where $B2$ contains all edges in the cut set, while one of the edges is excluded in version $B1$. If all four inequalities used in the derivation are of type $B1$, two edges get excluded from the resulting inequality, resulting in subclass $A$. When two inequalities are of type $B1$ and the remaining two of type $B2$, only one edge gets excluded, resulting in subclass $B$. Finally, when all four inequalities are of type $B2$, no edge gets excluded, resulting in subclass $C$.

Along the same lines, it is possible to extend inequalities $R2$, $R3$, and $R4$ of standard NDP (Agarwal 2006, 2009) to SNDP, with each one having subclasses $A$, $B$, and $C$, as in the case of $R1$. Moreover, in the case of $R2$, $R3$, and $R4$, these subclasses also contain several distinct versions of valid inequalities that are not isomorphic to each other, leading to a total of 25 distinct inequalities based on four-partitions. For details of these inequalities, please refer to Appendix D in the electronic companion. The numbers of these inequalities for each class and subclass are provided in Table 6.

Table 6. The number of four-partition inequalities by class and type.

<table>
<thead>
<tr>
<th>Inequality class</th>
<th>Subclass $A$</th>
<th>Subclass $B$</th>
<th>Subclass $C$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$R2$</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>$R3$</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>$R4$</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>Total</td>
<td>12</td>
<td>9</td>
<td>4</td>
<td>25</td>
</tr>
</tbody>
</table>

7. Implementation and Separation Heuristics

7.1. The Capacity Formulation

We solve the problem using the capacity formulation, because it has a much smaller problem size because of elimination of flow variables and the flow-conservation constraints, which are replicated for each fault in the flow formulation. The price of this reduction is that the formulation has an exponential number of valid inequalities. However, these inequalities can be handled implicitly via a constraint generation scheme. It is well known that the separation problem of finding a violated metric inequality for a given capacity solution is a linear program (see Avella et al. 2007). Several authors (see Fortz and Poss 2009 and Mattia 2012) have solved different versions of the network design problems with capacity formulation using this approach.

Whereas in the case of standard NDP, solving a single metric separation LP can determine if all metric inequalities are satisfied or not, in the case of SNDP such an LP
must be solved \( m \) times, i.e., once for each fault. Therefore, it would be prohibitively expensive to solve the metric separation problem at each node of the branch-and-bound tree. We have used the following approach to overcome this difficulty. Rather than solving the metric separation LPs at each node of the branch-and-bound tree, we do so only when an integer solution is obtained at a node. Normally, if an integer solution is found that is better than the current incumbent, the incumbent should be updated. However, before saving this solution as an incumbent, we solve the metric separation LPs for all faults to determine if the new solution satisfies all metric inequalities or not. It is saved as an incumbent only if it does. Otherwise, the violated metric inequality found is added to the LP at the current node, and branch-and-bound process is continued. Very likely, addition of this inequality will make the solution nonintegral, and further branching will resume, until another integer solution is found, which will again be tested for feasibility by solving the metric separation LPs.

The above approach was found to be computationally very efficient. As a result, we are able to report optimal solutions of problems as large as 35 nodes, 80 edges, and a fully dense traffic matrix. We note that the flow formulation for a problem of this size would have 435,200 flow variables, and 95,200 flow conservation constraints, making it virtually impossible to solve the problem optimally in a branch-and-bound framework. The capacity formulation has only 80 capacity variables, and the problem could be solved optimally within a few minutes of CPU time with the help of two-, three- and four-partition inequalities.

### 7.1.1. Overall Implementation

Before describing the separation heuristics, we describe the overall implementation of our approach. At the root node, we start the capacity formulation with no constraints, i.e., \( x = 0 \) being the obvious LP optimal solution. Given the current LP solution, we first look for any violated two-partition inequalities, and keep adding them to the LP, resolving it each time. When no further violated two-partition inequalities can be found, we look for a violated three-partition inequality and add it. But after adding a three-partition inequality, we switch back to the addition of two-partition inequalities. Only if we cannot find a violated two-partition inequality, we look for a three-partition inequality. Similarly, when neither a two-partition, nor a three-partition violated inequality can be found, only then we look for a violated four-partition inequality. This approach is summarized in the following algorithmic description.

**Algorithm 1** (Overall Implementation)

1. **Step 0.** Initialize the capacity formulation LP with \( m \) capacity variables and no constraints. Let \( LB2 = LB3 = LB4 = 0.0 \).
2. **Step 1.** Solve the LP, and let \( \bar{x} \) be the optimal solution with \( Z \) being its objective function value.
3. **Step 2.** Given \( \bar{x} \), solve the separation problem for two-partition inequalities. If a violated two-partition inequality is found, add it to the current LP, and go back to Step 1. Otherwise, if \( LB2 = 0.0 \), let \( LB2 = Z \). Go to Step 3.
4. **Step 3.** Solve the separation problem for three-partition inequalities. If a violated inequality is found, add it and go back to Step 1. Otherwise, if \( LB3 = 0.0 \), let \( LB3 = Z \). Go to Step 4.
5. **Step 4.** Solve the separation problem for four-partition inequalities. If a violated inequality is found, add it and go back to Step 1. Otherwise, if \( LB4 = 0.0 \), let \( LB4 = Z \). Go to Step 5.
6. **Step 5.** Root node computation is complete. Start the branch-and-cut search.

It is clear from the algorithm that \( LB2 \) is the bound value with two-partition inequalities alone, \( LB3 \) with two- and three-partition inequalities, and \( LB4 \) with two-, three- and four-partition inequalities.

Note that we do not add metric inequalities at any stage at the root node, or at any other node, except when the LP solution is integral. Nevertheless, the bound at the root node is quite strong even though the LP solution is not guaranteed to permit a feasible flow. As mentioned earlier, the metric separation LP is solved, and a violated metric inequality, if found, is added, only when the current solution is integral. Because solving the separation problem for three- and four-partition inequalities is more expensive than for two-partition inequalities, we look for only two-partition inequalities in the branch-and-cut phase (Step 5). Moreover, this is done only up to a limited depth of the tree in the interest of computational efficiency.

### 7.2. The Separation Problem

Given a fractional LP-solution, the separation problem is to find a \( k \)-partition (for \( k = 2, 3, \) or \( 4 \) in our case), so that one of the valid inequalities generated by that \( k \)-partition is violated by the current solution. This problem, though NP-hard, easily lends itself to a neighborhood search heuristic. Given a \( k \)-partition, we can define the following perturbation operation to generate a neighboring \( k \)-partition. Pick any node from a subset with more than one node, and shift it to a different subset. A given \( k \)-partition will thus have \( n(k - 1) \) neighboring \( k \)-partitions.

Starting from a randomly generated initial \( k \)-partition, we examine all neighboring \( k \)-partitions. The violations of all valid inequalities for each neighboring \( k \)-partition are evaluated, and the \( k \)-partition with largest violation is selected. The process is repeated until there is no further increase in the violation. This basic idea of "shifting" a node from one subset to another, can be further extended to "exchanging" two nodes belonging to different subsets. When the shifting does not produce any further improvement, the heuristic switches to the "exchange" mode. A algorithmic description of the heuristic is given below.
Algorithm 2 (Separation Heuristic)

Step 0. Create a random \( k \)−partition \( P = (N_1, \ldots, N_k) \). Let \( \tau_i \) denote the index of the subset to which node \( i \) belongs in this partition.

Step 1. Generate all inequalities associated with \( k \)-partition \( P \), and let \( (\pi^j, \pi_0^j) \) represent the \( j \)th inequality: \( \pi^j x \geq \pi_0^j \). Define \( \psi = \pi_0^j - \pi^j \bar{x} \), as the violation, where \( \bar{x} \) is the fractional solution of the current LP. Note that \( \psi \) may be negative for some or all inequalities. Let \( j^* \) be the inequality with maximum violation. Let \( v^* = v^j^* \), and \( (\pi^*, \pi_0^*) = (\pi_0^j, \pi_0^j) \).

Step 2. For each node \( i \) with \( |N_{\tau_i}| > 1 \), and subset \( N_{\tau_i} \), where \( t \neq \tau_i \), let \( P_{i, t} \) be the partition obtained by shifting \( i \) from \( N_{\tau_i} \) to \( N_t \). Evaluate all inequalities and their violations for partition \( P_{i, t} \).

Step 3. Let \( v^*(i, t) \), be the maximum violation. Let \( v_{max}^* = max_i(v^*(i, t)) \), and let \( i^*, t^* \) be the indices for which the maximum occurs. If \( v_{max}^* > v^* \), let \( P = P_{i^*, t^*} \), \( v^* = v_{max}^* \), update \( (\pi^*, \pi_0^*) \) accordingly, and go back to Step 2. Otherwise \( \text{STOP} \) and report \( (\pi^*, \pi_0^*) \) as the most violated inequality.

The above algorithm is described for the “shifting” phase of the heuristic, and can be appropriately modified for the exchange phase.

The effectiveness of the heuristic can be increased by restarting it from several randomly generated starting solutions, and selecting the best among the resulting solutions. We use two different values of this restart parameter. Initially, this parameter is set to a high value, i.e., 100 for two-partition inequalities. With reference to Algorithm 1, when Step 2 fails to find a violated two-partition inequality, the value of this parameter is decreased to 10, and is not changed thereafter. Similarly, for three-partition inequalities, the initial parameter value is 100, which is decreased to 10 if Step 3 fails to find a violated inequality. The motivation is to try hard to find violated inequalities in the initial phase, but not waste too much time in future attempts when returning after solving higher level inequalities. Even in the initial phase, we do not continue for all 100 attempts. If a violated inequality is found sooner, further attempts are not made.

8. Computational Experiments

We report computational results on two sets of problem instances. The first set consists of randomly generated problems with 35 nodes, 80 edges, and fully dense traffic matrices. The second set of instances were taken from the SNDlib database (see Orlowski et al. 2009).

For the first set, we generate coordinates of nodes randomly on a grid of 100 by 100. The traffic demands among various node pair are generated from uniform(1, 10) distribution. The facility cost between two nodes has a fixed cost component of $100 and a distance cost of $10 per unit distance. To decide the edges of graph \( G \), we start with a complete graph, and gradually remove long edges provided that the shortest path distance between the end nodes of that edge would not increase too much by removal of the edge. We also ensure that the degree of any node does not fall below two in this edge removal process. The data for all test problems is available from the author. The first instance of 35-node problems is also available in SNDlib database (india35).

All computational experiments were conducted on a Pentium Dual Core Processor with 3.9 MHz clock speed running Windows XP. Version 12.1 of CPLEX was used to solve the LPs and IPs.

8.1. Results on 35-Node Problems

For these instances we assume the facility capacity \( F = 600 \). Results on five randomly generated problems are reported in Table 7. All results are reported with and without CPLEX generated cuts, so as to enable comparison of three- and four-partition inequalities with CPLEX generated cuts. Columns 2–7 of the table report the values of LB2, LB3, and LB4 (expressed as percent of optimal solution) at the root node, and column 8 reports the value of optimal solution. A time limit of 300 seconds was imposed on the solution time. In columns 9–14 we report the CPU time if the problem was solved to optimality, otherwise the optimality gap at the end of time limit is reported. Note that with four-partition inequalities and CPLEX cuts, all five problems were solved to optimality within the time limit of five minutes, while only one problem out of five could be solved with two-partitions inequalities alone.

A comparison of columns 3 and 4 shows that, on the average, two-partition inequalities with CPLEX cuts produce a stronger root node bound than two- and three- partition inequalities without CPLEX cuts, but still the latter performs much better in solving the problem as shown by a comparison of columns 10 and 11. A possible reason for this is that the cuts generated by CPLEX are perhaps not very likely to be facet defining, whereas the three- and four-partition inequalities have a high chance of being facet defining. This suggests the conclusion that the CPLEX cuts are not a substitute for three- and four-partition cuts, though they can compliment them and further improve the performance of branch-and-bound process.

8.2. Results on SNDlib Problems

SNDlib 1.0 (Orlowski et al. 2009) is a database of network design problem instances maintained at the website http://sndlib.zib.de, which invites and publishes solutions and bound values on various versions of network design problems. We found nine problem instances on this database that match the version addressed in this paper. As of writing this paper, no solutions or bound values were reported for any of these instances on the website. Barring two rather large instances ("pi0ri40" and "germany50"), we were able to find optimal solutions of all the remaining instances within a few seconds of CPU time. Even
Table 7. Results on 35-node 80-edge problems for $F = 600$

<table>
<thead>
<tr>
<th>No</th>
<th>LB2 (No CPLEX cuts)</th>
<th>LB3 (No CPLEX cuts)</th>
<th>LB4 (No CPLEX cuts)</th>
<th>CPU (sec.)/optimality gap [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>1</td>
<td>91.45</td>
<td>94.32</td>
<td>93.21</td>
<td>95.67</td>
</tr>
<tr>
<td>2</td>
<td>91.33</td>
<td>95.20</td>
<td>93.35</td>
<td>95.84</td>
</tr>
<tr>
<td>3</td>
<td>93.43</td>
<td>96.38</td>
<td>94.92</td>
<td>96.49</td>
</tr>
<tr>
<td>4</td>
<td>91.83</td>
<td>94.31</td>
<td>93.58</td>
<td>95.71</td>
</tr>
<tr>
<td>5</td>
<td>89.90</td>
<td>92.71</td>
<td>92.94</td>
<td>95.57</td>
</tr>
<tr>
<td>Avg.</td>
<td>91.59</td>
<td>94.58</td>
<td>93.60</td>
<td>95.86</td>
</tr>
</tbody>
</table>

Table 8. Results on SNDlib problems.

<table>
<thead>
<tr>
<th>Prob</th>
<th>n</th>
<th>m</th>
<th>G.D.</th>
<th>$F$</th>
<th>$F/\bar{d}$</th>
<th>LB2</th>
<th>LB3</th>
<th>LB4</th>
<th>$Z_{ip}$</th>
<th>CPU/gap</th>
<th>BBTS</th>
<th>N2</th>
<th>N3</th>
<th>N4</th>
</tr>
</thead>
<tbody>
<tr>
<td>pioro40</td>
<td>40</td>
<td>89</td>
<td>4.45</td>
<td>622</td>
<td>4.20</td>
<td>99.42</td>
<td>99.42</td>
<td>99.42</td>
<td>551,868</td>
<td>[0.18%]</td>
<td>49,830</td>
<td>663</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>germany50</td>
<td>50</td>
<td>88</td>
<td>3.52</td>
<td>40</td>
<td>20.75</td>
<td>98.85</td>
<td>98.85</td>
<td>99.09</td>
<td>828,640</td>
<td>[0.01%]</td>
<td>8,726</td>
<td>937</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>france</td>
<td>25</td>
<td>45</td>
<td>3.60</td>
<td>2,500</td>
<td>7.50</td>
<td>99.11</td>
<td>99.11</td>
<td>99.70</td>
<td>33,600</td>
<td>13</td>
<td>1</td>
<td>250</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>norway</td>
<td>27</td>
<td>51</td>
<td>3.78</td>
<td>1,000</td>
<td>65.61</td>
<td>98.62</td>
<td>99.23</td>
<td>99.44</td>
<td>569,320</td>
<td>12</td>
<td>1</td>
<td>240</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>tal</td>
<td>24</td>
<td>51</td>
<td>4.25</td>
<td>100,800</td>
<td>27.48</td>
<td>97.93</td>
<td>98.54</td>
<td>99.00</td>
<td>29,483</td>
<td>6</td>
<td>1</td>
<td>223</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>atlanta</td>
<td>15</td>
<td>22</td>
<td>2.93</td>
<td>4,000</td>
<td>3.08</td>
<td>98.07</td>
<td>98.07</td>
<td>99.83</td>
<td>178,520</td>
<td>2</td>
<td>1</td>
<td>92</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>newyork</td>
<td>16</td>
<td>49</td>
<td>6.13</td>
<td>1,000</td>
<td>67.68</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>1,257,600</td>
<td>2</td>
<td>1</td>
<td>86</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pdh</td>
<td>11</td>
<td>34</td>
<td>6.18</td>
<td>480</td>
<td>22.83</td>
<td>93.21</td>
<td>96.65</td>
<td>97.86</td>
<td>20,755</td>
<td>1</td>
<td>2</td>
<td>57</td>
<td>25</td>
<td>20</td>
</tr>
<tr>
<td>polska</td>
<td>12</td>
<td>18</td>
<td>3.00</td>
<td>622</td>
<td>4.14</td>
<td>99.36</td>
<td>99.36</td>
<td>99.58</td>
<td>36,864</td>
<td>1</td>
<td>1</td>
<td>53</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Notes. G.D.: Graph density; $Z_{ip}$: Best-known integer solution; BBTS: Branch-and-bound tree size; N2, N3, N4: No. of two-, three- and four-partition inequalities generated.

for pioro40 and germany50 problems, we were able to find solutions with negligible optimality gaps of 0.2% and 0.01%, respectively, within the time limit of 600 seconds. The results are reported in Table 8. We note that some of these problem instances involve multiple facility capacities. However, because we consider only the single facility version of the problem, we have solved these instances with one of the capacity values, which is mentioned in column (5) of the table. All problems have been solved with facility costs only, and routing costs, if any, have not been included in the model.

Barring the “pdh” problem, the LB4 value was above 99% in all instances. After addition of CPLEX cuts, six problems were solved at the root node itself without requiring any branching. Both these observations suggest that these problems are relatively easy problems to solve.

It is noteworthy that for most of these problems, three- and four-partition inequalities make negligible contributions to the bound value. The reason is that the facility capacity relative to traffic demand (shown in column (6) of the table) for these problems is either too low (<10) or too high (>20). In our experience, in both cases, the two-partition inequalities produce very tight bounds, and three- and four-partitions have a limited role to play. It is interesting to note that all these problems are easily solvable under capacity formulation with two-partition inequalities alone, which have been known for quite some time. However, it seems that an attempt to do so was never made.

9. Conclusion

In this paper, we have presented a key theorem that permits the study of polyhedral structure of SNDP by considering smaller problems obtained from $k$-partitions of the original problem. After reviewing the past work on two-partition facets, we presented a number of three- and four-partition-based inequalities and proved the facet-defining status of some of them by virtue of the key theorem. The computational results amply demonstrate the effectiveness of these inequalities coupled with our implementation of the capacity formulation in solving fairly large instances of the problem optimally.

The inequalities presented in this paper are also valid for the survivable network design problems under alternative routing schemes such as shared protection disjoint path routing, and $p$-cycle protection scheme. An important area of future research would be to investigate the role of
these inequalities in solving these models. We are currently exploring this possibility for shared protection disjoint-path routing.

Another possible avenue of future research is to investigate the potential of using five-partition inequalities to obtain even better bound values.

Electronic Companion

An electronic companion to this paper is available as part of the online version at http://dx.doi.org/10.1287/opre.1120.1147.

Acknowledgments

The author would like to thank two anonymous referees for their valuable comments that have helped in substantially improving the content, structuring, and presentation of this paper. The author is also grateful to Vishnu Narayanan of the Indian Institute of Technology, Bombay, India, with whom he had some valuable discussions in the early stages of this work.

References


Yogesh Agarwal is a professor of decision sciences at the Indian Institute of Management (IIM), Lucknow, India. Prior to joining IIM Lucknow, he worked at AT&T Bell Laboratories from 1985–1990 and was a recipient of the Distinguished Member of Technical Staff Award. His research interests include telecommunications network design, distribution/logistics networks, integer programming, polyhedral theory, and combinatorial optimization.