TECHNICAL RESEARCH REPORT

Strong Formulations for Network Design Problems with Connectivity Requirements

by Thomas L. Magnanti, S. Raghavan

T.R. 99-28

Strong Formulations for Network Design Problems with Connectivity Requirements

Thomas L. Magnanti* and S. Raghavan†

April 1999

Abstract

The network design problem with connectivity requirements (NDC) models a wide variety of celebrated combinatorial optimization problems including the minimum spanning tree, Steiner tree, and survivable network design problems. We develop strong formulations for two versions of the edge-connectivity NDC problem: unitary problems requiring connected network designs, and nonunitary problems permitting non-connected networks as solutions. We (i) present a new directed formulation for the unitary NDC problem that is stronger than a natural undirected formulation, (ii) project out several classes of valid inequalities—partition inequalities, odd-hole inequalities, and combinatorial design inequalities—that generalize known classes of valid inequalities for the Steiner tree problem to the unitary NDC problem, and (iii) show how to strengthen and direct nonunitary problems.

Our results provide a unifying framework for strengthening formulations for NDC problems, and demonstrate the strength and power of flow-based formulations for network design problems with connectivity requirements.

1 Introduction

Network Design Problems with Connectivity Requirements (NDC) arise in a wide variety of application domains including VLSI design and telecommunication network design. The increasing reliance on communication networks (and expectations of a digital future) places an enormous importance on the reliability of such networks. To stay apace of the explosive growth of data, traffic telecommunication companies (telcos) are adding new fiber as well as deploying fiber capacity enhancing technologies (like Dense Wave Division Multiplexing) to increase the capacity of their backbone networks. Telcos are also actively deploying

*Sloan School of Management and Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139

†The Robert H. Smith School of Business and the Institute for Systems Research, Van Munching Hall, University of Maryland, College Park, MD 20742

(competing) technologies in the local loop (the portion of the network that serves the customers premises) to provide greater access bandwidth to customers. Given the enormous bandwidth capabilities of these networks, and the increasing array of services provided over them, the failure of any link in such a network can have significant, perhaps even catastrophic consequences.

In this paper, we consider network design problems with edge connectivity requirements. Informally given requirements for the number of edge-disjoint paths between every pair of nodes , we wish to design a minimum cost network that satisfies these requirements. To set notation, and define the class of problems we consider, we formally state the NDC problem as follows:

Network Design Problem with Connectivity Requirements (NDC): We are given an *undirected* graph $G = (N, E)$, with node set N and edge set E, and a cost vector $\mathbf{c} \in \mathcal{R}_+^{|E|}$ on the edges E. We are also given a symmetric $|N| \times |N|$ requirement matrix $\mathbf{R} = [r_{ij}]$. The entry r_{ij} prescribes the number of edge-disjoint paths needed between nodes i and j. We wish to select a set of edges that satisfy these requirements at minimum cost, as measured by the sum of costs of edges we choose.

The NDC problem models a wide variety of combinatorial optimization problems including the classical minimum spanning tree and Steiner tree problems. One important specialization of the NDC problem that arises in the design of telecommunications networks (see [CMW89]) is the *Survivable Network Design Problem (SND).* In this application each node v in the graph has a connectivity requirement r_v and the connectivity requirements between nodes s and t are given by $r_{st} = \min\{r_s, r_t\}$. Table 1 shows several other noteworthy cases of the NDC problem.

A few observations concerning the entries in Table 1 are worth making. The k-edge disjoint path problem seeks, at minimum cost, k-edge disjoint paths between specified nodes s and t. Whitney [Whi32] showed that a graph is k-edge connected (i.e., remains connected after elimination of any $k-1$ edges) if and only if it contains k-edge disjoint paths between every pair of nodes. As a result, the minimum cost k-edge connected spanning subgraph problem is an SND problem with $r_v = k$ for all nodes. The network design problem with low connectivity requirements (NDLC) is of particular interest to local telephone companies (see [CMW89]). In this special case of the SND problem, the connectivity requirements are restricted to $\{0, 1, 2\}$. (Since most local telephone companies believe it is sufficient to protect against single link failures in the local loop, this problem is of significant importance to them.) In the Steiner forest problem, we are given a graph $G = (N, E)$ and node sets $T_1, T_2, \ldots, T_P$ with $T_i \cap T_j = \phi$ for all node set pairs i, j. We wish to design a graph at minimum cost that connects all the nodes in each node set. The point to point connection problem is a *special case* of the Steiner forest problem with $T_i = \{s_i, t_i\}$ for $i = 1, \ldots, P$.

NDC problems can be classified in two ways. If the connectivity requirements imply that

Problem Type	**SND or NDC**	**Connectivity Requirements**
Minimum Spanning Tree Problem	SND	$r_v = 1$ for all nodes v.
Steiner Tree Problem	SND	$r_v = 1$ for all nodes required in the tree; $r_v = 0$ for all other nodes.
k-Edge Disjoint Path Problem	SND	$r_s = r_t = k$ requires k edge-disjoint paths between nodes s and t.
Minimum Cost k-Edge-Connected Spanning Subgraph Problem	SND	$r_v = k$ for all nodes v.
Minimum Cost Steiner k-Edge-Connected Spanning Subgraph Problem	SND	$r_v = k$ for all required nodes; $r_v = 0$ for all other nodes.
Network Design with Low Connectivity Requirements (NDLC)	SND	$r_v \in \{0, 1, 2\}$ for all nodes v.
Point to Point Connection Problems	NDC	$r_{s_i t_i} = 1$ for given source sets $\{s_1, s_2, \ldots, s_p\}$ and terminal sets $\{t_1, t_2, \ldots, t_p\}$. $r_{ij} = 0$ otherwise.
Steiner Forest Problem	NDC	$r_{ij} = 1$ if $i \in T_q$ and $j \in T_q$ for some pairwise disjoint node set $T_1, T_2, \ldots, T_p$. $r_{ij} = 0$ otherwise.

Table 1: Specializations of Network Design Problems with Connectivity Constraints

all nodes with a (positive) connectivity requirement must be connected, we say the problem is a *unitary NDC problem*. Otherwise, it is a *nonunitary NDC problem*. For example, the SND problem is a unitary NDC problem, while the general Steiner forest problem is a nonunitary NDC problem.

The examples in Table 1 show that the NDC problem models a very wide variety of connectivity problems on graphs. These problems appear both as stand alone problems and as subproblems in more complex network design applications (like VLSI design and telecommunications network design and management). Consequently, techniques for modeling and solving NDC problems have widespread applicability.

Considerable accumulated experience in the optimization literature has demonstrated the value of developing good linear programming relaxations (strong formulations) of combinatorial optimization problems. Strong formulations are very useful in developing exact algorithms solution methods (branch and bound, branch and cut, column generation) since their use rapidly accelerates these solution techniques. Strong formulations can also provide good bounds on the optimal solution and so are useful in assessing heuristic solution methods. In particular, dual-ascent heuristic techniques (that generate both lower bounds on the optimal solution value and feasible solutions to the combinatorial optimization problem) based upon strong formulations typically provide better solutions than those based upon weaker linear programming relaxations. The development in this paper is motivated by a desire to develop *better linear programming relaxations* for NDC problems, and to provide a *unifying strengthening approach applicable to all NDC problems*.

Since the NDC problem models a wide variety of combinatorial optimization problems, the polyhedral structure of many special cases of the NDC problem have been well studied. Over the past twenty years, researchers have proposed a large number of formulations (and solution methods based on them) for the Steiner tree problem. Most noteworthy among these are the papers by Wong [Won84] proposing a (bi)directed model for the undirected Steiner tree problem; Chopra and Rao [CR94a, CR94b] examining the facial structure of the undirected Steiner tree polyhedron and its relationship to a directed formulation for the Steiner tree problem; Goemans [Goe94] investigating extended formulations[1] with node and edge variables for the Steiner tree problem and introducing combinatorial design inequalities for the Steiner tree problem; and Goemans and Myung [GM93] establishing the relationship between several formulations for the Steiner tree problem.

Several researchers have examined special cases of unitary NDC problems with higher connectivity requirements (i.e., greater than 1). For series-parallel graphs, Mahjoub [Mah94] and Baïou and Mahjoub [BM93] provide complete descriptions of the 2-edge-connected

[1] A *natural* formulation has one variable for each member of its "ground" set. For example, a natural formulation for the NDC problem would contain one binary (or integer) decision variable for each edge in the graph. An extended formulation contains additional variables—integer or continuous. Strong models for combinatorial optimization problems are often developed by using extended formulations.

spanning subgraph polytope and the Steiner 2-edge connected spanning subgraph polytope respectively. Boyd and Hao [BH93] introduce a class of valid inequalities for the 2-edge-connected spanning subgraph polytope and describe necessary and sufficient conditions for these valid inequalities to define facets. Based on a result by Robbins, Chopra [Cho92] describes a directed formulation for the NDLC problem in a model that permits unlimited edge replication. Using a result due to Nash-Williams, a generalization of Robbins theorem, Goemans [Goe90] shows how to strengthen a well-known cutset formulation for the SND problem with connectivities $r_v \in \{0, 1, \text{even}\}$ in a model that permits unlimited edge replication. Grötschel, Monma, and Stoer [GMS92b, GMS95b, GMS95a, Sto92] investigate the polyhedral structure of both the edge- and node-connectivity versions of the SND problem. One of these papers [GMS92b]investigates the polyhedral structure of the NDLC problem, while another in [GMS95b] examines the polyhedral structure of SND problems whose highest connectivity requirements are three or more. [GMS95a] and [Sto92] contain comprehensive summaries of polyhedral results for the SND problem.

Researchers have proposed many solution methods (both exact and approximate) for the NDC problem and its specializations. Our discussion has focused on polyhedral research in this area. Survey papers by Grötschel, Monma, and Stoer [GMS95a], Raghavan and Magnanti [RM97], and Frank [Fra94] provide more comprehensive reviews of research on the NDC and its specializations.

In this paper, we develop strong formulations for both unitary and nonunitary NDC problems. Our work differs from earlier research in several ways. Goemans [Goe90] and Grötschel, Monma, and Stoer [GMS95a] have shown in various forms how to use a result due to Nash-Williams to obtain stronger models for the SND problem with connectivities $r_v \in \{0, 1, \text{even}\}$. We show that although the Nash-Williams theorem is useful to motivate the directing procedure, it does not play a role in strengthening the formulation (i.e., it is not necessary)! Consequently, we are able to generalize the directing procedure to strengthen formulations for *all unitary NDC problems*.

Next, we project out three classes of valid inequalities from the strengthened (extended) formulation for the unitary NDC problem that are generalizations of facet-defining valid inequalities for the Steiner tree problem. For special cases of the unitary NDC problems several researchers have shown how to project from extended formulations that are equivalent to the flow-based strengthened formulation for the NDC problem. For example, Goemans [Goe94] describes a node weighted formulation for the Steiner tree problem and obtains the three classes of valid inequalities by projection. Chopra and Rao [CR94a, CR94b] describe a directed arc formulation for the Steiner tree problem and show how to project out two classes of valid inequalities—partition and odd-hole—from it. Grötschel, Monma, and Stoer [GMS95a] describe a directed formulation for the SND problem with connectivities $r_v \in \{0, 1, \text{even}\}$ and show how to project out partition inequalities from it. We illustrate the projection from the flow-based formulation for three reasons. First, several extended formulations that are equivalent to the flow formulation for the Steiner tree problem do not generalize to the NDC problem, for example, node weighted extended formulations for the Steiner tree problem do not generalize to the NDC problem. Second, for nonunitary problems we describe a directing and strengthening technique that requires flow variables. Consequently, a deeper understanding of the flow-based formulation and its relationship to the cutset formulation is important. Third, we believe this paper is the first to explicitly show how to project from the flow-based formulation (even for special cases of the unitary NDC problem like the Steiner tree problem).

Finally, we show how to direct nonunitary NDC problems which appear to have received significantly less attention in the literature. We implement our directing procedure by using flow variables to obtain strengthened (flow-based) formulations for nonunitary NDC problems, but it is not obvious how to implement the directing procedure without using flow variables.

A companion paper [MR99] empirically confirms the theoretical results presented in this paper. It shows that a solution procedure for the NDLC problem using the models developed in this investigation can be effective in solving large scale problems with up to 300 nodes and 3000 edges to within a (guaranteed) few percent of optimality.

The rest of this paper is organized as follows. In Section 2 we review two well-known formulations for the NDC problem, a natural formulation with edge variables, and an extended formulation containing both flow and edge variables. Next in Section 3, we first motivate the directing procedure applying a result by Nash-Williams that applies to unitary NDC problems with restricted connectivities. We then obviate the need for Nash-Williams result to strengthen the formulation of all unitary NDC problems. Sections 4, 5, 6, and 7 deal with the strength of the improved formulation. Section 4 provides some preliminary results regarding the projection of improved flow formulation onto the space of the edge variables. Sections 5, 6, and 7 show how to project partition inequalities, odd-hole inequalities, and combinatorial design inequalities respectively from the improved flow formulation. In Section 8 we examine nonunitary NDC problems. We first show how to strengthen a formulation of the Steiner forest problem by applying a new directing technique. In Section 9 we use this technique to strengthen formulations for all NDC problems. Finally, in Section 10 we provide some concluding remarks.

Notation: We assume familiarity with standard graph theory terminology. We work with undirected graphs and directed graphs which we refer to as *graphs* and *digraphs*. To distinguish between directed and undirected graphs, we refer to undirected graphs as *graphs*, undirected edges as *edges*, directed graphs as *digraphs*, and directed edges as *arcs*. We use braces to denote an edge between nodes i and j, i.e., $\{i, j\}$, and parentheses to denote a directed arc from node i to node j, i.e., (i, j). $Econ(T) := \max\{r_{ij} | j \in T; i \in N \backslash T\}$ denotes the connectivity requirements of a set of nodes T. It is the maximum *edge-connectivity* requirement between any node in T and its complement. For NDC problems we refer to $\text{econ}(i)$ as the *maximum connectivity* requirement of node i. If $\text{econ}(i) > 0$, we say node i is a *required* node. In models that permit parallel edges, we let $G = (N, E)$ represent the underlying graph and b_{ij} represent the number of parallel edges allowed between nodes i and j. For example, if a model permits two edges between nodes i and j, then the graph G contains the edge $\{i, j\}$ and $b_{ij} = 2$. In an undirected graph, any set of nodes T defines a *cut* $\delta(T) = \{\{i, j\} : i \in N \backslash T, j \in T\}$. Similarly, any set of nodes T in a directed graph defines a *dicut* $\delta^-(T) = \{(i, j) : i \in N \backslash T, j \in T\}$ of arcs directed into the node set T and a set of arcs $\delta^+(T) = \{(i, j) : i \in T, j \in N \backslash T\}$ directed out of T. An *s–t cut* is a cut $\delta(T)$ with $s \notin T$ and $t \in T$. Similarly, an *s–t dicut* is a dicut, say $\delta^-(T)$, with $s \notin T$ and $t \in T$. The *capacity of a cut* $\delta(W)$ is defined as the sum of the capacity of the edges in the cut, and the *capacity of a dicut* $\delta^-(W)$ is defined as the sum of the capacity of the arcs in the dicut.

Sometimes we will want to eliminate the variables from an "extended" formulation of a problem. Let $\mathbf{A}$ and $\mathbf{B}$ be two given matrices and $\mathbf{d}$ be a column vector, all with the same number of rows. Consider the polyhedron $P = \{(\mathbf{x}, \mathbf{f}) : \mathbf{Ax} + \mathbf{Bf} \geq \mathbf{d}\}$. The polyhedron $Q = \{\mathbf{x} : \mathbf{Ax} + \mathbf{Bf} \geq \mathbf{d} \text{ for some vector } \mathbf{f}\}$ obtained by eliminating the $\mathbf{f}$ variables is called the *projection* of the polyhedron P. If two formulations for a problem provide, for all objective function coefficients, the same optimal objective value when solved as linear programs, we say the two formulations are *equivalent*. When we compare two extended formulations, we say the two formulations are *equivalent* if they provide the same objective value for all objective function coefficients on the natural variables (the objective function coefficients for the additional variables are zero). In other words two extended formulations are equivalent if their projection onto the space of the natural variables is identical. We say that adding an inequality $\mathcal{I}$ *strengthens* a formulation of a (mixed) integer programming problem if it is valid and adding it to the formulation improves the objective value of the linear programming relaxation of the formulation for some choice of the objective function coefficients. We say that a formulation $\mathcal{P}_1$ is *stronger* than a formulation $\mathcal{P}_2$ if, when solved as linear programs, the objective value of $\mathcal{P}_1$ is always better than the objective value of $\mathcal{P}_2$, and sometimes is strictly better than the objective value of $\mathcal{P}_2$.

2 Formulations for the NDC Problem

In this section we describe two well-known models for the NDC problem, one a cutset model, and the other a multicommodity flow-based model. For the flow-based model we also show how to minimize the number of commodities, a method that proves invaluable in our subsequent discussions.

Menger's theorem [Men27] states that the number of edge-disjoint paths between a pair of nodes, say s and t, is equal to the minimum number of edges across any cut between them, i.e., any s–t cut. Consequently, the following well-known "cutset" formulation, with x_{ij} representing the number of copies of edge $\{i,j\}$ in the network, is a valid representation of the NDC problem.

Cutset formulation for the NDC problem:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{\{i,j\}\in E} c_{ij}x_{ij} & & \text{(1a)}\\
\text{subject to:} \quad & \sum_{\{i,j\}\in\delta(S)} x_{ij} \ge \operatorname{econ}(S) & \text{for all node sets } S \text{ with } \phi \ne S \ne N; & \quad \text{(1b)}\\
& x_{ij} \le b_{ij} & \text{for all } \{i,j\}\in E; & \quad \text{(1c)}\\
& x_{ij} \ge 0 & \text{and integer, for all } \{i,j\}\in E. & \quad \text{(1d)}
\end{aligned}
$$

An alternative way to formulate the problem is to enforce the connectivity requirements of the matrix $\mathbf{R}$ using commodity flows. For each pair $\{s,t\}$ of nodes, with $r_{st} \ge 1$, create a commodity, arbitrarily choosing one of the nodes as the origin of the commodity and the other node as the destination. Let K denote the set of commodities and let q_k, for each $k \in K$, denote the edge-connectivity requirement between the origin and destination of commodity k: if $r_{st} = 3$, then $q_k = 3$ for the commodity k corresponding to the node pair $\{s,t\}$. The following mixed integer program, with x_{ij} representing the number of copies of edge $\{i,j\}$ in the network and f_{ij}^k flows, is a valid formulation for the NDC problem.

Undirected flow formulation for the NDC problem:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{\{i,j\}\in E} c_{ij}x_{ij} & & \text{(2a)}\\
\text{subject to:} \quad & \sum_{j\in N} f_{ji}^k - \sum_{l\in N} f_{il}^k = \begin{cases} -q_k & \text{if } i = O(k);\\ q_k & \text{if } i = D(k);\\ 0 & \text{otherwise;} \end{cases} & \text{for all } i\in N \text{ and } k\in K & \quad \text{(2b)}\\
& \left.\begin{matrix} f_{ij}^k \\ f_{ji}^k \end{matrix}\right\} \le x_{ij} & \text{for all } \{i,j\}\in E \text{ and } k\in K & \quad \text{(2c)}\\
& f_{ij}^k, f_{ji}^k \ge 0 & \text{for all } \{i,j\}\in E \text{ and } k\in K & \quad \text{(2d)}\\
& x_{ij} \le b_{ij} & \text{for all } \{i,j\}\in E; & \quad \text{(2e)}\\
& x_{ij} \ge 0 & \text{and integer, for all } \{i,j\}\in E. & \quad \text{(2f)}
\end{aligned}
$$

The max-flow min-cut theorem implies that the cutset and flow formulations are equivalent in the following sense.

Lemma 2.1 *The projection of the feasible space of the linear programming relaxation of the undirected flow formulation for the NDC problem (2) onto the space of the edge variables* *is the feasible space of the linear programming relaxation of the cutset formulation for the NDC problem (1).*

Notice that the cutset formulation is of exponential size, while the flow formulation is compact: it has $\mathcal{O}(|K|(|E|+|N|))$ constraints and $\mathcal{O}(|K||E|)$ variables.

A simple, and naive, way to determine the number of commodities in the flow formulation is to create a commodity for every pair of nodes with a connectivity requirement. For an underlying graph with 100 nodes and 1000 edges and positive connectivity requirements between all nodes this approach would create a commodity for every node pair and so 4,950 commodities. Consequently, the model would contain $495,000$ flow balance constraints, $9,900,000$ constraints of type (2c), 9,900,000 nonnegativity constraints for the flow variables, 1,000 constraints of type (2e), and 1,000 nonnegativity and integrality constraints for the edge variables. As this example shows, the flow formulation can be very large.

By using fewer commodities, if possible, we could reduce the size of the formulation. To accomplish this objective, we can use an idea that Gomory and Hu [GH61] used when solving the classical network synthesis problem. Given the connectivity requirements matrix $\mathbf{R}$, create a "requirement" graph G^R on the node set N, giving edge $\{i,j\}$ between nodes i and j in G^R a weight r_{ij}. Gomory and Hu showed that it is sufficient to consider the connectivity requirements only for the edges on a maximum spanning tree of this graph. It is easy to verify this result using the max-flow min-cut theorem and the maximum spanning tree optimality conditions. As is well known, a spanning tree is a maximum spanning tree if and only if it satisfies the following optimality condition: For every nontree edge $\{k,l\}$ of G^R, $r_{ij} \geq r_{kl}$ for every edge $\{i,j\}$ contained in the (tree) path P on the maximum spanning tree connecting nodes k and l. As a result, any network that satisfies the requirements of the maximum spanning tree has sufficient capacity to satisfy the requirements of nontree edges. (By the max-flow min-cut theorem, the network has sufficient capacity to connect nodes k and l if every cut in the network design separating these nodes has capacity at least r_{kl}. Since some pairs i,j of nodes on the path P must lie on opposite sides of this cut, the cut must have capacity at least $r_{ij} \geq r_{kl}$.)

Gomory and Hu's result permits us to model the edge-connectivity requirements in any NDC problem with $|N|-1$ or fewer commodities. We simply compute the maximum spanning tree of the requirement graph, which we now refer to as the requirement spanning tree, and create commodities only for those edges of the maximum spanning tree with nonzero weight. Since the requirement spanning tree has $|N|-1$ edges and finding it requires $\mathcal{O}(|E|+|N|\log|N|)$ time, this procedure creates at most $|N|-1$ commodities (we will not create commodities for zero weight edges of the requirement spanning tree) and requires $\mathcal{O}(|E|+|N|\log|N|)$ time.

The two formulations 1 and 2 for the NDC problem are known to be weak. That is, the objective value of the linear programming relaxation are typically significantly less than the optimal objective value of the integer program. Computational experiments reported by Grötschel, Monma, and Stoer [GMS92a, GMS95b] confirm this result, particularly when the requirement spanning tree has edges with connectivity requirement 1. (In a remarkable paper, Jain (see [Jai98]) provides some theoretical evidence to support these results. He shows that the worst case ratio of the optimal value of the integer program to the optimal value of the linear programming relaxation of the cut formulation is 2, *independent* of the problems's size.)

3 Stronger Formulations for Unitary NDC Problems

In this section we first show how to direct unitary NDC problems for situations when the connectivity requirements are all even or one, and obtain a stronger formulation for this special case of the unitary NDC problem. We then generalize this result, developing an improved model for any unitary NDC problem (i.e., even those with odd connectivity requirements). For ease of exposition, for the rest of this paper we assume that the model *does not permit edge replication.* It is straightforward to verify that the results apply to models that permit edge replication.

3.1 Directing the Unitary NDC Problem

The following result due to Nash-Williams [NW60] provides a key ingredient for transforming the undirected formulation to a directed one.

Theorem 3.1 (Nash-Williams) *Suppose G is an undirected graph with r_{xy} edge-disjoint paths connecting each pair x and y of its nodes. Then it is possible to direct the graph (i.e., orient its edges) so that the resulting digraph contains $\lfloor r_{xy}/2 \rfloor$ arc-disjoint paths from node x to node y and $\lfloor r_{xy}/2 \rfloor$ arc-disjoint paths from node y to node x.*

Consider any unitary NDC problem whose connectivity requirements r_{st} are even or one. We can view any feasible integer solution to this problem as follows: it is connected and contains several 2-edge-connected components. If we contract the 2-edge-connected components, the solution becomes a tree. The edges on the tree are the *bridge* edges in the feasible solution before we contracted the 2-edge-connected components; that is, removing these edges disconnects the graph defined by that solution.

The Nash-Williams theorem permits us to direct the edges of each 2-edge-connected component so that for any pair of nodes i and j with $r_{ij} \geq 2$ (by assumption these requirements must be even), the network contains $r_{ij}/2$ directed arc-disjoint paths from node i to node j, and $r_{ij}/2$ directed arc-disjoint paths from node j to node i. Once oriented, each 2-edge-connected component contains a directed path between every pair of its nodes. Therefore, if nodes i and j belong to the same 2-edge-connected component and $r_{ij} = 2$, the oriented network contains a directed path from node i to node j and a directed path from node j to node i. To direct the bridges, consider the tree obtained by contracting each 2-edge-connected component of the solution. Select any one of the nodes created by the contraction as a root node and direct the tree away from this node.

Figure 1 illustrates this directing procedure. In this example a, b, c and g are 2-edge-connected components. Between every pair of nodes s and t in these components $r_{st} = 2$. We orient the edges of each component (see Figure 1b) so that it contains a directed path between every pair of nodes in each 2-edge-connected component. Next, we select the node created by contracting component b as the root node and direct the tree edges (i.e., the bridges of the solution) away from node b. Figure 1c shows the graph at the conclusion of the directing procedure.

These observations permit us to formulate the unitary NDC problem as follows. Let y_{ij} be 1 if edge $\{i, j\}$ is oriented from node i to node j in the directing procedure applied to the optimal solution (i.e., the oriented network contains arc (i, j)) and be 0 otherwise.

Directed cut formulation for the unitary NDC problem ($r_{st} \in \{$0,1,even$\}$):

$$\text{Minimize} \quad \sum_{\{i,j\}\in E} c_{ij}x_{ij} \tag{3a}$$

$$\text{subject to:} \quad \sum_{(i,j)\in\delta^-(S)} y_{ij} \geq \frac{\text{econ}(S)}{2} \quad \text{if econ}(S) \geq 2;\ \text{for all } S \subseteq N, \tag{3b}$$

$$\sum_{(i,j)\in\delta^-(S)} y_{ij} \geq 1 \quad \text{if econ}(S) = 1;\ \text{for all } S,\ \text{root} \notin S \tag{3c}$$

$$y_{ij} + y_{ji} \leq x_{ij} \quad \text{for all } \{i, j\} \in E \tag{3d}$$

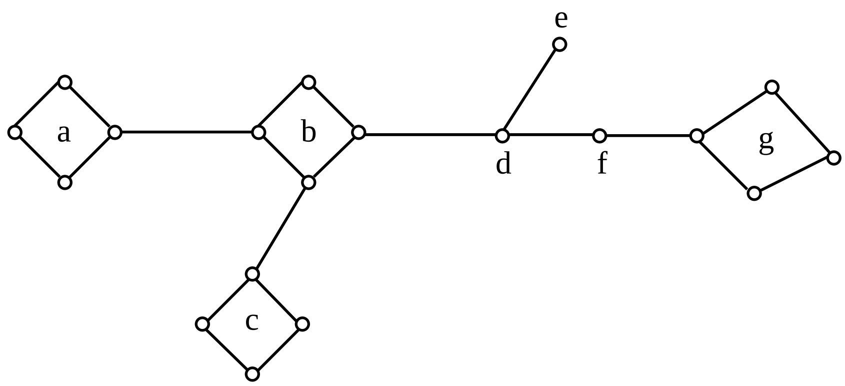
$$x_{ij} \leq 1 \quad \text{for all } \{i, j\} \in E \tag{3e}$$

$$y_{ij}, y_{ji}, x_{ij} \geq 0 \quad \text{and integer, for all } \{i, j\} \in E. \tag{3f}$$

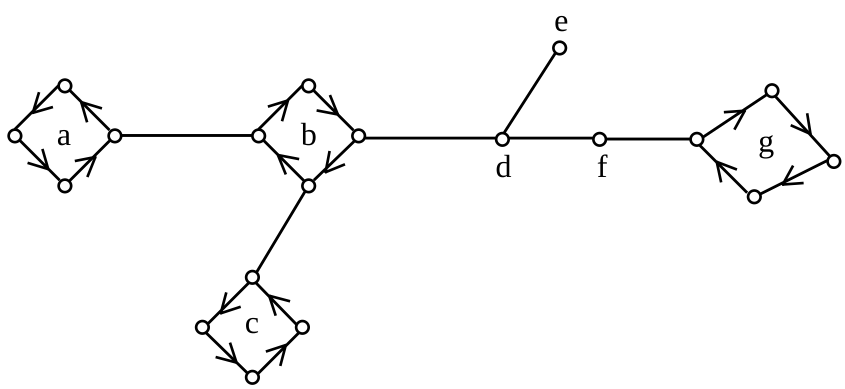
Since $\text{econ}(S) \equiv \max\{r_{ij} | j \in S; i \in N \backslash S\}$, constraint (3b) ensures that for every pair of nodes s and t with $r_{st} \geq 2$, every s–t dicut contains at least $r_{st}/2$ arcs and every t–s dicut contains at least $r_{st}/2$ arcs. Menger's theorem ensures that the oriented network contains at least $r_{st}/2$ arc-disjoint paths from node s to node t and $r_{st}/2$ arc-disjoint paths from node t to node s. Similarly, constraints (3b) and (3c) ensure that the oriented network contains a directed path from the root to every required node. Constraint (3d) ensures that the oriented network contains at most one of the arcs (i, j) and (j, i).

Proposition 3.2 *The directed cut model is a valid formulation for the unitary NDC problem in the sense that* $(\mathbf{x}, \mathbf{y})$ *is a feasible solution to this model if and only if* $\mathbf{x}$ *is an incidence vector of a feasible NDC design (that is,* $\mathbf{x}$ *is a feasible solution to the cutset model (1)).*

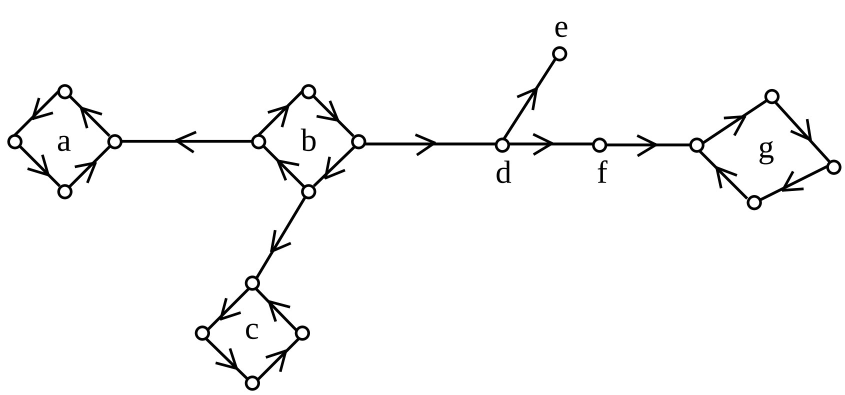
Proof: Suppose $\mathbf{x}$ is an incidence vector of a feasible NDC design. The argument preceding



(a)



(b)



(c)

Figure 1: Directing the bridges of a feasible solution to the unitary NDC problem. (a) Feasible solution. (b) Direct the edges of each 2-edge-connected component. (c) Direct the bridges away from component b.

Formulation (3) shows how to construct an integer vector $\mathbf{y}$ so that $(\mathbf{x}, \mathbf{y})$ is a valid solution for the directed cut model.

To establish the converse, suppose $(\mathbf{x}, \mathbf{y})$ is a feasible solution to the directed cut model. Let Q be any node set with $\text{econ}(Q) = \text{econ}(N \backslash Q) \geq 2$. Combining inequality (3b) for $S = Q$ and $S = N \backslash Q$ and the inequalities (3d) summed over all $\{i, j\} \in \delta(Q)$ gives

$$\sum_{\{i,j\} \in \delta(Q)} x_{ij} \geq \sum_{(i,j) \in \delta^+(Q)} y_{ij} + \sum_{(j,i) \in \delta^-(Q)} y_{ji} \geq \text{econ}(Q).$$

If Q is any node set with $\text{econ}(Q) = \text{econ}(N \backslash Q) = 1$, assume without loss of generality that the root node is not in Q. Then the inequality (3c) with $S = Q$ and the inequality (3d) summed over $\{i, j\} \in \delta(Q)$ implies that

$$\sum_{\{i,j\} \in \delta(Q)} x_{ij} \geq 1.$$

Therefore, $\mathbf{x}$ is a feasible solution to the cutset formulation (1). ■

To see that the linear programming relaxation of the directed cut formulation is stronger than linear programming relaxation of the cutset formulation, consider the NDLC example shown in Figure 2. In this example, each edge has unit cost. Nodes a, b and c have a connectivity requirement of 2. Nodes d, e and f have a connectivity requirement of 1. The optimal solution to the linear programming relaxation of the cutset formulation is $x_{ab} = x_{ac} = 1$ and $x_{bc} = x_{bd} = x_{cd} = x_{de} = x_{df} = x_{ef} = 0.5$, with objective value 5. The optimal solution to the linear programming relaxation of the directed cut model has integer values for the edge variables. The solution is $y_{ab} = y_{bd} = y_{dc} = y_{ca} = y_{de} = y_{df} = 1$, $x_{ab} = x_{ac} = x_{bd} = x_{cd} = x_{de} = x_{df} = 1$, with objective value 6. By reformulating the problem and solving the linear programming relaxation, we have obtained the optimal solution.

In the next section we show how that the Nash-Williams procedure is not necessary for this directing procedure. Consequently, we generalize the results in this section to all unitary NDC problems.

3.2 Generalizing the Directing Procedure

The directed cut formulation (3) is not valid for unitary NDC problems with odd connectivity requirements. As an example, consider an SND problem defined on K_4, the complete graph on four nodes, assuming each node has a connectivity requirement of 3. The optimal solution for this problem is K_4. For any node i in K_4, there is no way to direct the edges so that both $\delta^+(i)$ and $\delta^-(i)$ are at least 1.5.

Suppose, however, that in the directed cut formulation we relax the integrality constraints imposed upon the y_{ij} variables, and interpret y_{ij} as the capacity on the flow from node i to node j. We will show that this formulation is a valid mixed integer program for

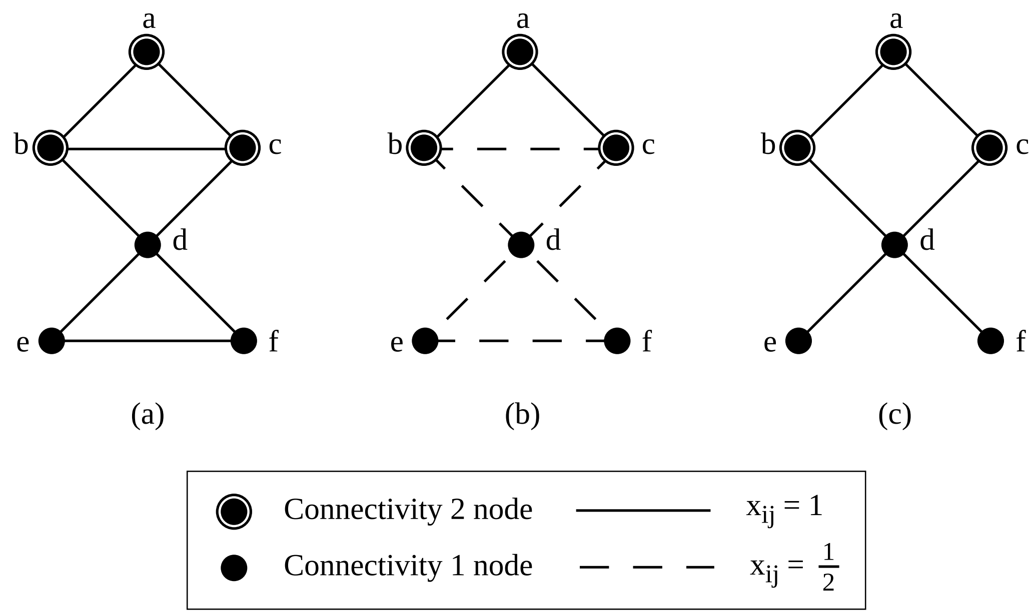


Figure 2: The directed cut formulation is stronger than the cutset formulation. (a) Underlying graph with all edge costs equal to 1. (b) Optimal LP solution to the cutset formulation. (c) Optimal LP solution to the directed cut formulation.

any unitary NDC problem. Consider any feasible solution to an unitary NDC problem. As we noted previously, the solution is a connected graph consisting of 2-edge-connected components and bridges. Suppose (i) we select $y_{ij} = y_{ji} = 1/2$ for each edge on the 2-edge-connected components, and (ii) direct the bridges away from the component that contains the root node, setting y_{ij} to 1 if edge $\{i, j\}$ is oriented from node i to node j, and 0 otherwise. (If we contract the 2-edge-connected components the directing procedure for the bridges is similar to the directing procedure for the Steiner tree.) The resulting solution $(\mathbf{x}, \mathbf{y})$ is feasible in the directed cut formulation if we relax the integrality condition on $\mathbf{y}$. Therefore, the following directed cut formulation is valid for all unitary NDC problems.

Directed cut formulation for the unitary NDC problem:

$$\text{Minimize} \quad \sum_{\{i,j\} \in E} c_{ij} x_{ij} \tag{4a}$$

$$\text{subject to:} \quad \sum_{(i,j) \in \delta^-(S)} y_{ij} \geq \frac{\text{econ}(S)}{2} \quad \text{if econ}(S) \geq 2; \text{ for all } S \subseteq N, \tag{4b}$$

$$\sum_{(i,j) \in \delta^-(S)} y_{ij} \geq 1 \quad \text{if econ}(S) = 1; \text{ for all } S, \text{ root} \notin S \tag{4c}$$

$$y_{ij} + y_{ji} \leq x_{ij} \quad \text{for all } \{i, j\} \in E \tag{4d}$$

$$x_{ij} \leq 1 \quad \text{for all } \{i, j\} \in E \tag{4e}$$

$$y_{ij}, y_{ji} \geq 0 \qquad \text{for all } \{i,j\} \in E \tag{4f}$$

$$x_{ij} \geq 0 \qquad \text{and integer, for all } \{i,j\} \in E. \tag{4g}$$

Proposition 3.3 *The directed cut model (4) is a valid formulation for the unitary NDC problem in the sense that* $(\mathbf{x}, \mathbf{y})$ *is a feasible solution to this model if and only if* $\mathbf{x}$ *is an incidence vector of a feasible NDC design.*

Proof: Similar to the proof of Proposition 3.2. ■

3.2.1 Flow Formulation

The max-flow min-cut theorem permits us to formulate an improved flow model, with multiple commodities, that is equivalent to the directed cut model (4).

Improved undirected flow formulation for the unitary NDC problem:

$$\text{Minimize} \quad \sum_{\{i,j\} \in E} c_{ij} x_{ij} \tag{5a}$$

$$\text{subject to:} \quad \sum_{j \in N} f_{ji}^k - \sum_{l \in N} f_{il}^k = \begin{cases} -q_k & \text{if } i = O(k); \\ q_k & \text{if } i = D(k); \\ 0 & \text{otherwise;} \end{cases} \quad \text{for all } i \in N \text{ and } k \in K \tag{5b}$$

$$f_{ij}^k + f_{ji}^h \leq x_{ij} \qquad \text{for all } \{i,j\} \in E \text{ and } k, h \in K \tag{5c}$$

$$f_{ij}^k, f_{ji}^k \geq 0 \qquad \text{for all } \{i,j\} \in E \text{ and } k \in K \tag{5d}$$

$$x_{ij} \leq 1 \qquad \text{for all } \{i,j\} \in E \tag{5e}$$

$$x_{ij} \geq 0 \qquad \text{and integer, for all } \{i,j\} \in E. \tag{5f}$$

Using the procedure described in Section 2, we can create the commodities as follows. Select any node i whose maximum connectivity requirement is greater than or equal to 2 as the root node. (If the maximum connectivity requirement of all nodes is 1, and so the problem is a Steiner tree problem, we arbitrarily select any one of the nodes as the root node.)

Commodity selection procedure for Formulation (5)

1. Find the requirement spanning tree.
2. Delete all edges with $r_{st} = 0$ from the requirement spanning tree. The resulting tree is connected because, by assumption, the NDC problem is unitary.
3. For each edge $\{s,t\}$ of the requirement spanning tree with $r_{st} \geq 2$, create two commodities: one with origin node s and destination node t, and the other with origin node t and destination node s; each of these commodities has a flow requirement of $r_{st}/2$.

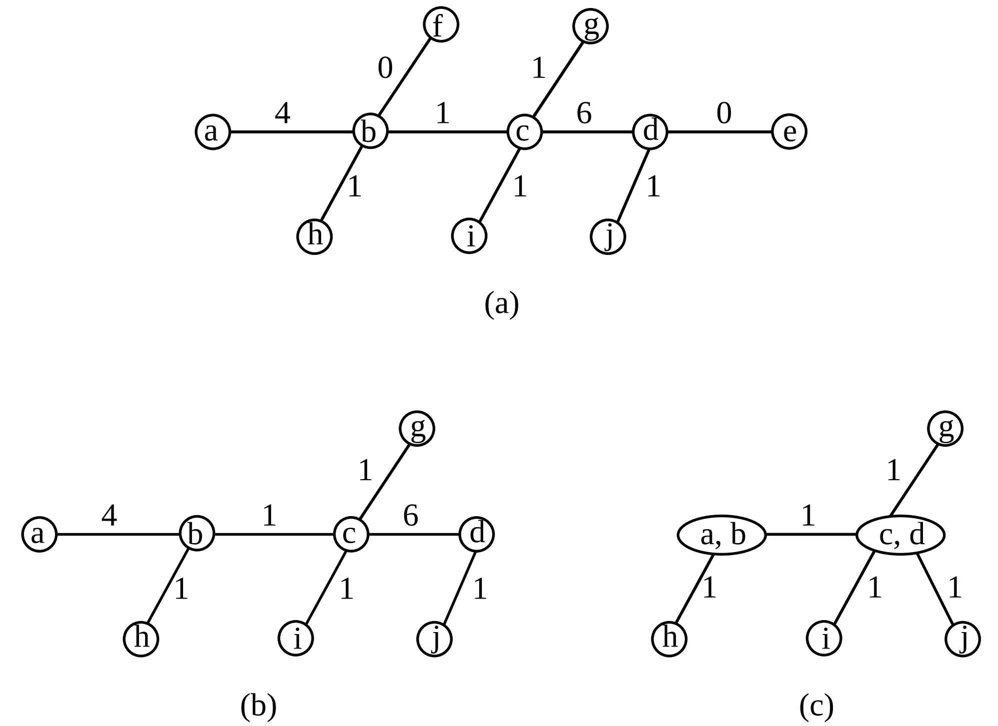


Figure 3: Commodity selection procedure for the unitary NDC problem. (a) Requirement spanning tree. (b) Tree obtained by deleting edges $\{s,t\}$ with $r_{st} = 0$. (c) Tree obtained by contracting edges $\{s,t\}$ with $r_{st} \geq 2$.

4. Contract each edge $\{s,t\}$ with $r_{st} \geq 2$ in the requirement spanning tree, creating a contracted requirement spanning tree T with $r_{ij} = 1$ for all edges $\{i,j\}$. We distinguish nodes created by the contraction from the original nodes, by calling them *components*. We denote a component by any one of the nodes it contains in the original requirement spanning tree (e.g., if we create a component by contracting nodes s and t, then we denote the component s). Select a component i in T as the root node (if T does not contain any components, then select any node as the root node arbitrarily). Create a commodity for every node j in T other than the root node, with node i as its origin (in the original graph), and node/component j as its destination (in the original graph), with a requirement of 1.

Figure 3 illustrates this procedure. Figure 3a shows the requirement spanning tree of a unitary NDC problem. Figure 3b shows the requirement spanning tree after we have deleted edges $\{s,t\}$ with $r_{st} = 0$. Notice that because the problem is a unitary NDC problem, the graph in Figure 3b is connected. Otherwise, it would be a forest.

Table 2 identifies the commodities obtained by applying the directing procedure to the example in Figure 3. Edges $\{a,b\}$ and $\{c,d\}$ are the only edges with a requirement of at least 2 on this tree. Therefore, we create four commodities—those shown in the first

Commodity Origin	Commodity Destination	Commodity Requirement
a	b	2
b	a	2
c	d	3
d	c	3
a	c	1
a	g	1
a	h	1
a	i	1
a	j	1

Table 2: Commodities in Formulation (5) for example in Figure 3.

four rows of Table 2. Figure 3c shows the contracted requirement spanning tree (i.e., after contracting edges with $r_{st} \geq 2$). Let node a denote the component $\{a, b\}$, and node c denote component $\{c, d\}$. We select node a as the root. The tree in Figure 3c contains 6 nodes. Therefore, we create 5 commodities, each with origin node a, and destination nodes c, g, h, i and j.

The following useful property is a consequence of the commodity selection procedure.

Property 3.4 *For any node set S,*

1. *If $econ(S) \geq 2$, then the improved flow formulation contains a commodity k whose flow requirement is $econ(S)/2$, origin is in $N \backslash S$, and destination is in S.*

2. *If $econ(S) = 1$ and $root \notin S$, then the improved flow model contains a commodity whose flow requirement is 1, origin is the root node, and destination is in S.*

3. *If $econ(S) = 1$ and $root \in S$, then no commodity in the improved flow model has its origin in $N \backslash S$, and destination in S.*

Proof: This result follows from the commodity selection procedure and the fact that $r_{st} = \max\{r_{ij} | j \in S; i \in N \backslash S\} \equiv \mathrm{econ}(S)$ for any edge $\{s, t\}$ in the requirement spanning tree.

■

We now establish the validity of the improved undirected flow formulation for the unitary NDC problem by showing that the improved undirected flow formulation and the directed cut formulation are equivalent.

Lemma 3.5 *The improved undirected flow formulation (5) and the directed cut formulation (4) are equivalent.*

Proof: We assume that we select the same root node in both formulations. First, consider any feasible solution $(\mathbf{x}^*, \mathbf{y}^*)$ to the directed cut formulation. If we interpret y^*_{ij} as a capacity imposed upon the flow from node i to node j, the max-flow min-cut theorem implies that we can (i) send $r_{st}/2$ units of flow between any pair of nodes s and t in the requirement spanning tree with $r_{st} \geq 2$, and (ii) send one unit of flow from the root component in the contracted requirement spanning tree to any other node/component in the contracted requirement spanning tree. Furthermore, the constraint $y^*_{ij} + y^*_{ji} \leq x^*_{ij}$ implies that we can fulfill conditions (i) and (ii) while ensuring that for each edge $\{i, j\}$, the sum of the maximum flow sent (on the edge $\{i, j\}$) from node i to node j, and the maximum flow sent from node j to node i does not exceed x_{ij}. These arguments show that we can find flow variables $\mathbf{f}^*$ so that $(\mathbf{x}^*, \mathbf{f}^*)$ is feasible in the flow formulation.

Suppose $(\bar{\mathbf{x}}, \bar{\mathbf{f}})$ is a feasible solution to the improved flow formulation. For each edge $\{i, j\}$, set $\bar{y}_{ij} = \max_{k \in K} \bar{f}^k_{ij}$ and $\bar{y}_{ji} = \max_{k \in K} \bar{f}^k_{ji}$. We claim the solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ is feasible in the directed cut formulation. Whenever edge $\{s, t\}$ is in the requirement spanning tree and $r_{st} \geq 2$, the improved undirected flow formulation sends $r_{st}/2$ units of flow from node s to node t and $r_{st}/2$ units of flow from node t to node s. Consequently, if edge $\{s, t\}$ is in the requirement spanning tree and $r_{st} \geq 2$, the capacity of every s–t dicut and every t–s dicut is at least $r_{st}/2$ (for the solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$). For any node set S, the requirement spanning tree contains an edge $\{s, t\}$ in $\delta(S)$ with $\text{econ}(S) = r_{st}$. Therefore, whenever $\text{econ}(S) \geq 2$, the capacity of the dicut $\delta^-(S)$ is at least $\text{econ}(S)/2$. The improved undirected flow formulation sends 1 unit of flow from the root component to every node/component in the contracted requirement spanning tree. Therefore, the capacity of every dicut $\delta^-(S)$ with root $\notin S$ and $\text{econ}(S) = 1$ is at least 1. The constraint $\bar{f}^k_{ij} + \bar{f}^h_{ji} \leq \bar{x}_{ij}$ implies that for every edge, $\bar{y}_{ij} + \bar{y}_{ji} \leq \bar{x}_{ij}$. Consequently, $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ is feasible for the directed cut model, and thus the improved undirected flow formulation and the directed cut formulation are equivalent. ■

Before concluding this section, we note the improved models (4) and (5) are stronger, as linear programs, than the cutset (1) and the undirected flow (2) model only if the requirement spanning tree contains an edge $\{s, t\}$ with $r_{st} = 1$. To see this result, observe that if no pair of nodes i and j has a connectivity requirement of 1, then for all node sets S, $\text{econ}(S) \neq 1$. But then, if $\bar{\mathbf{x}}$ is any feasible solution to the cutset formulation, the vector $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$, with $\bar{y}_{ij} = \bar{y}_{ji} = \bar{x}_{ij}/2$, is feasible in the directed cut formulation. As we have shown before, if $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ is any feasible solution to the directed cut formulation, then $\bar{\mathbf{x}}$ is feasible in the cutset formulation. Therefore, in this case, the two models are equivalent.

Finally we note that a simple modification of the formulations we have considered permits us to model situations that allow edge replication: we just replace the constraint $x_{ij} \leq 1$ by the constraint $x_{ij} \leq b_{ij}$ throughout our discussion.

4 Projecting from the Improved Flow Formulation

To compare the improved flow formulation and the cutset formulation, we would like to project out the flow variables from the improved flow formulation so that the resulting models have the same set of variables. An elegant method for projection, proposed by Balas and Pulleyblank [BP83], and implicit in the work of Benders [Ben62], is based upon a theorem of the alternatives.

Theorem 4.1 (Projection Theorem) *The projection of the set* $P = \{(\mathbf{x}, \mathbf{f}) \in \mathcal{R}^{n+m} | \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{f} \geq \mathbf{d}\}$ *onto the space of the x variables is*

$$Proj_{\mathbf{x}}(P) = \{\mathbf{x} \in \mathcal{R}^n | (\mathbf{g}^j)^T \mathbf{A}\mathbf{x} \geq (\mathbf{g}^j)^T \mathbf{d}, \quad for\, j = 1, 2, \ldots, J\},$$

which is defined by a finite set of generators $\{\mathbf{g}^j | j = 1, \ldots, J\}$ *of the cone* $C = \{\mathbf{g} | \mathbf{B}^T \mathbf{g} = \mathbf{0}; \mathbf{g} \geq 0\}$.

The cone C in the statement of Theorem 4.1 is just the linear programming dual to the feasibility problem obtained by deleting the $\mathbf{x}$ variables and setting the righthand side to zero in the inequality $Ax + Bf \geq d$. When the polyhedron P is defined by equality as well as inequality constraints, as in the improved undirected formulation Theorem 4.1 assumes the following form.

Corollary 4.2 *The projection of the set* $P = \{(\mathbf{x}, \mathbf{f}) \in \mathcal{R}^{n+m} | \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{f} \geq \mathbf{d}; \mathbf{A}'\mathbf{x} + \mathbf{B}'\mathbf{f} = \mathbf{d}'; \mathbf{x} \in \mathbf{X}; \mathbf{f} \geq \mathbf{0}\}$ *onto the space of the x variables is*

$$Proj_{\mathbf{x}}(P) = \{\mathbf{x} \in \mathbf{X} | (\mathbf{g}_u^j)^T \mathbf{A}\mathbf{x} + (\mathbf{g}_v^j)^T \mathbf{A}'\mathbf{x} \geq (\mathbf{g}_u^j)^T \mathbf{d} + (\mathbf{g}_v^j)^T \mathbf{d}', \quad for\, j = 1, 2, \ldots, J\},$$

which is defined by a finite set of generators $\{(\mathbf{g}_u^j, \mathbf{g}_v^j) | j = 1, \ldots, J\}$ *of the cone* $C = \{(\mathbf{u}, \mathbf{v}) | \mathbf{B}^T \mathbf{u} + \mathbf{B}'^T \mathbf{v} \geq \mathbf{0}; \mathbf{u} \geq 0; \mathbf{v}\ unrestricted\}$.

If we can identify a set of finite generators of the cone C, then we obtain the projection of the set P. The Projection Theorem has the additional advantage that every member $(\mathbf{u}, \mathbf{v})$ of the cone C defines a valid inequality $(\mathbf{u}^T A + \mathbf{v}^T A')\mathbf{x} \geq \mathbf{u}^T \mathbf{d} + \mathbf{v}^T \mathbf{d}')$ for $\text{Proj}_{\mathbf{x}}(P)$. As a consequence, even if we cannot characterize the generators of the cone, we can still use the cone to obtain valid inequalities for $\text{Proj}_{\mathbf{x}}(P)$.

4.1 Projection Cone of the Improved Undirected Flow Formulation (5)

In Sections 5, 6, and 7, we will use the Projection Theorem to show that the improved flow formulation (5) implies three classes of valid inequalities for the cutset formulation and, therefore, that the flow formulation is stronger than the cutset formulation. To develop these results, we need to find generators for the cone C in the statement of Theorem 4.1, which is the dual linear program to the feasibility problem obtained by deleting the $\mathbf{x}$ variables and setting the righthand side to zero. For the improved flow formulation (5), we need to consider the following projection cone:

$$\left.\begin{array}{l}\sum_{h\in K} u_{ij}^{kh} + v_i^k - v_j^k \\ \sum_{h\in K} u_{ij}^{hk} + v_j^k - v_i^k\end{array}\right\} \geq 0 \quad \forall\, \{i,j\} \in E,\ \forall\, k \in K \tag{6a}$$

$$u_{ij}^{kh} \geq 0 \quad \forall\, \{i,j\} \in E,\ \forall k,h \in K. \tag{6b}$$

In these inequalities, v_i^k is the dual variable corresponding to the flow balance equation at node i for commodity k, and u_{ij}^{kh} is the dual variable corresponding to the forcing constraint $f_{ij}^k + f_{ji}^h \leq x_{ij}$. Note that for any edge $\{i,j\}$ and any pair of commodities k and h, the model contains two forcing constraints $f_{ij}^k + f_{ji}^h \leq x_{ij}$ and $f_{ij}^h + f_{ji}^k \leq x_{ij}$. We identify the dual variable u_{ij}^{kh} with the constraint $f_{ij}^k + f_{ji}^h \leq x_{ij}$ and the dual variable u_{ij}^{hk} with the constraint $f_{ij}^h + f_{ji}^k \leq x_{ij}$. By convention, the dual variables obtained by reversing the indices, i.e., u_{ij}^{kh} and u_{ji}^{hk}, are the same.

Since one flow balance equation for each commodity is redundant, we can set, for each commodity k, $v_{O(k)}^k$ to value zero. Using Corollary 4.2 for any member of this cone, we obtain a valid inequality of the form

$$\sum_{\{i,j\}\in E}(\sum_{k\in K}\sum_{h\in K} u_{ij}^{kh})x_{ij} \geq \sum_{k\in K} q_k v_{D(k)}^k. \tag{7}$$

In this expression, q_k is the number of units of commodity k sent from commodity k's origin to its destination. We refer to the coefficient of x_{ij} in this inequality as π_{ij} and the righthand side coefficient as π_0.

Given some choice of the variables v_i^k for all the nodes i and commodities k, there are a number of choices for u_{ij}^{hk}. How do we determine the best such choice? Since the coefficient π_{ij} of x_{ij} in the inequality $\pi\mathbf{x} \geq \pi_0$ is $\sum_{k\in K}\sum_{h\in K} u_{ij}^{kh}$, we would like $\sum_{k\in K}\sum_{h\in K} u_{ij}^{kh}$ to be as small as possible for each edge. The following theorem describes the choice of u_{ij}^{hk} that minimizes $\sum_{k\in K}\sum_{h\in K} u_{ij}^{kh}$. We give a constructive proof that also shows how to determine the u_{ij}^{hk} values.

Theorem 4.3 *Suppose we are given values for v_i^k for all nodes i and all commodities k. For any edge $\{i,j\}$, let $t_{ij}^k = \max(0, v_j^k - v_i^k)$ and $t_{ji}^k = \max(0, v_i^k - v_j^k)$. Define $t_{ij} = \sum_{k\in K} t_{ij}^k$ and $t_{ji} = \sum_{k\in K} t_{ji}^k$. Then $\max(t_{ij}, t_{ji})$ is the minimum feasible value of $\sum_{k\in K}\sum_{h\in k} u_{ij}^{kh}$ in the inequalities (6a).*

Proof: We will establish this result for each edge $\{i,j\}$. Let us first show that $\max(t_{ij}, t_{ji})$ is a lower bound on the value of $\sum_{k\in K}\sum_{h\in K} u_{ij}^{kh}$ in any feasible solution to the inequalities (6a). Let I be the set of all commodities with $t_{ij}^k > 0$ and J be the set of commodities with $t_{ji}^k > 0$. For any edge $\{i,j\}$ with $v_j^k - v_i^k > 0$, equation (6a) implies $\sum_{h\in K} u_{ij}^{kh} \geq v_j^k - v_i^k$. Summing over all commodities in the set I gives $\sum_{k\in I}\sum_{h\in K} u_{ij}^{kh} \geq \sum_{k\in I}(v_j^k - v_i^k) = t_{ij}$. Similarly, by considering the commodities in the set J, we obtain $\sum_{k\in J}\sum_{h\in K}u_{ij}^{hk} \geq \sum_{k\in J}(v_i^k - v_j^k) = t_{ji}$. But then the inequalities $\sum_{k\in K}\sum_{h\in K}u_{ij}^{kh} \geq \sum_{k\in I}\sum_{h\in K}u_{ij}^{kh}$ and $\sum_{k\in K}\sum_{h\in K}u_{ij}^{kh} \geq \sum_{k\in J}\sum_{h\in K}u_{ij}^{hk}$ imply that $\sum_{k\in K}\sum_{h\in K}u_{ij}^{kh} \geq \max(t_{ij}, t_{ji})$.

We next prescribe values for the variables u_{ij}^{kh} that achieve the lower bound $\max(t_{ij}, t_{ji})$. Initially, each $u_{ij}^{kh} = 0$ and $\sum_{k\in K}\sum_{h\in K}u_{ij}^{kh} = 0$. Select a commodity l from I and a commodity m from J. Set $u_{ij}^{lm} = \min\{t_{ij}^l, t_{ji}^m\}$. If $t_{ij}^l \geq t_{ji}^m$, delete m from J, and if $t_{ij}^l \leq t_{ji}^m$, delete l from I. Set $t_{ij}^l = t_{ij}^k - u_{ij}^{lm}$ and $t_{ji}^m = t_{ji}^m - u_{ij}^{lm}$. Repeat this procedure until one of the two sets I and J, say J, is empty. Note that at this point $\sum_{k\in K}\sum_{h\in K}u_{ij}^{kh} = \min(t_{ij}, t_{ji})$ and the $\mathbf{u}$ and $\mathbf{v}$ variables satisfy inequality (6a) for every commodity we have deleted from I and J. For the remaining commodities $l \in I$, let m be any commodity in K and set $u_{ij}^{lm} = t_{ij}^l$. Thus, $\sum_{k\in K}\sum_{h\in K}u_{ij}^{kh} = \min(t_{ij}, t_{ji}) + (\max(t_{ij}, t_{ji}) - \min(t_{ij}, t_{ji})) = \max(t_{ij}, t_{ji})$. By construction, this choice of u_{ij}^{hk} satisfies the equations of the cone. ■

With the aid of Theorem 4.3, we will now derive three classes of valid inequalities—partition inequalities, odd-hole inequalities, and combinatorial design inequalities—that generalize known classes of valid inequalities for the Steiner tree problem to the unitary NDC problem. For ease of exposition we will first consider the NDLC problem, and then generalize the results to the unitary NDC problem.

5 Partition Inequalities

For the NDLC problem, partition inequalities (also called multicut inequalities by some authors) have the following form. Partition the node set N into disjoint node sets $N_0, N_1, \ldots, N_p$ satisfying the property that each node set has at least one node i with a connectivity requirement $r_i > 0$. A partition inequality is an inequality of the form

$$\frac{1}{2}\sum_{k=0}^{k=p}\sum_{\delta(N_k)} x_{ij} \geq \begin{cases} p+1 & \text{if at least two sets have a node with} \\ & \text{a connectivity requirement of two;} \\ p & \text{otherwise.} \end{cases}$$

This inequality implies that (i) if no set or exactly one set in the partition has a node with a connectivity requirement of two, then the network contains at least p edges between the $p+1$ sets, and (ii) if two or more sets have a node with a connectivity requirement of two, then the network contains at least $p+1$ edges between these sets. Chopra [Cho89] and Magnanti and Wolsey [MW95] show that the partition inequalities describe the dominant of the spanning tree polytope. Chopra and Rao [CR94a] and Grötschel and Monma [GM90] show that under appropriate conditions, partition inequalities are facet defining for the Steiner tree problem and for the NDLC problem respectively.

Consider the five node example shown in Figure 4. Nodes 0 and 1 have a connectivity requirement of two; nodes 2, 3 and 4 have a connectivity requirement of 1. The improved undirected formulation (with node 0 as the root node) for this network has four commodities

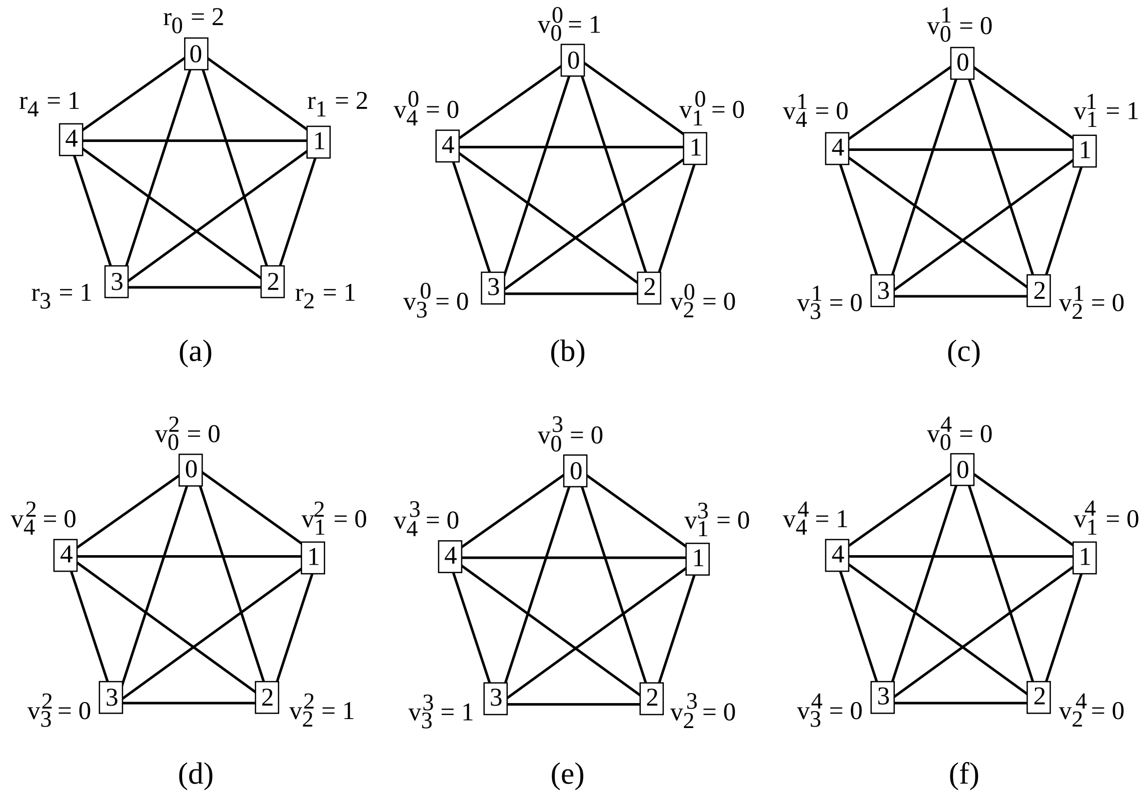


Figure 4: Projecting partition inequalities for a five node example. (a) Connectivity requirements (b) Value of the v_i^0 variables (c) Value of the v_i^1 variables (d) Value of the v_i^2 variables (e) Value of the v_i^3 variables (f) Value of the v_i^4 variables.

(1, 2, 3 and 4) with the origin as the root node and node 1, 2, 3 and 4, respectively, as the destination node. It also has a commodity (commodity 5) with node 1 as the origin and node 0 as the destination. We will refer to commodity 5 as commodity 0 to simplify our discussion. Therefore, for each commodity i, node i is the destination. Consider the partition inequality with each node defining a set in the partition (i.e., $\frac{1}{2}\sum_i \sum_{j\neq i} x_{ij} \geq 5$). Each edge variable in the partition inequality has a coefficient of 1 (note that the inequality counts each edge twice in the double summation). Consequently, in projecting out these variables to obtain this inequality, we need $\sum_{k\in K}\sum_{h\in K} u_{ij}^{kh} = 1$. Since the righthand side has value 5, we also need $\sum_{k\in K} v_{D(k)}^k = 5$ ($q_k = 1$ for all commodities in the NDLC problem).

In the example of Figure 4, for each commodity $k = 0, \dots, 4$, let $v_i^k = 0$ if $i \neq k$ and $v_k^k = 1$. Consider any edge $\{i, j\}$. Note that the v_i^k and v_j^k values are the same for all commodities, except for (i) commodity i, with $v_i^i = 1$ and $v_j^i = 0$, and (ii) commodity j, with $v_i^j = 0$ and $v_j^j = 1$. Thus, for each edge $\{i, j\}$, $\sum_{k\in K} \max(0, v_j^k - v_i^k) = \sum_{k\in K} \max(0, v_i^k - v_j^k) = 1$. Therefore, in accordance with Theorem 4.3 we can set $u_{ij}^{hk} = 1$ if $h = j$ and $k = i$, and $u_{ij}^{hk} = 0$ otherwise. This choice of variables in the projection cone imply the partition

$\frac{1}{2}\sum_i \sum_{j \neq i} x_{ij} \geq 5$.

In general, consider a partition $N_0, \ldots, N_p$. Without loss of generality assume that the root node is a node in N_0. Suppose $\text{econ}(N_i) = 2$. Recall, from Property 3.4, if $\text{econ}(N_i) = 2$ the improved flow formulation must contain two commodities: one with origin $n_i \in N_i$ and destination some node $m_i \notin N_i$, and one with destination $n_i \in N_i$ and origin $m_i \notin N_i$, both with a flow requirement of $\text{econ}(N_i)/2 = 1$. Let i denote the commodity with destination node $n_i \in N_i$ (and origin $m_i \notin N_i$) and a requirement of 1. Similarly, if $\text{econ}(N_i) = 1$ and $i \neq 0$, the improved flow formulation must contain a commodity i with destination a node $n_i \in N_i$ (the origin would be the root node) and a requirement of 1.

For commodities $0, \ldots, p$, we set

$$v_i^k = \begin{cases} 1 & \text{if } i \in N_k, \\ 0 & \text{otherwise,} \end{cases}$$

and set $v_i^k = 0$ otherwise.

With this choice of values for the v_i^k variables, we ensure that for all edges $\{i, j\}$ across the partition $\sum_{k \in K} \max(0, v_j^k - v_i^k) = \sum_{k \in K} \max(0, v_i^k - v_j^k) = 1$ and for edges $\{i, j\}$ not in the partition $\sum_{k \in K} \max(0, v_j^k - v_i^k) = \sum_{k \in K} \max(0, v_i^k - v_j^k) = 0$. Thus, by choosing the values of u_{ij}^{kh} as indicated by Theorem 4.3, we find that $\pi_{ij} = \sum_{k \in K} \sum_{h \in K} u_{ij}^{kh}$ is 1 if i and j are in different sets of the partition and $\pi_{ij} = 0$ otherwise. Also, $\pi_0 = \sum_{k \in K} v_{D(k)}^k = p + 1$, since $v_{D(k)}^k = 1$ for commodities $0, \ldots, p$ and $v_{D(k)}^k = 0$ otherwise.

If all the nodes with a connectivity requirement of two lie in one of the sets (i.e., N_0) of the partition or the problem has no node with a connectivity requirement of two (in this case the problem is a Steiner tree problem), by Property 3.4, there is no commodity in the improved flow formulation with destination in N_0. Consequently, we select p commodities with destinations in $N_1, \ldots, N_p$ and origin in N_0 (i.e., the root node). With the same choice of v_i^k for $k = 1, \ldots, p$ (i.e., $v_i^k = 1$ if $i \in N_k$ and 0 otherwise), for all edges $\{i, j\}$ across the partition, $\max(\sum_{k \in K} \max(0, v_j^k - v_i^k), \sum_{k \in K} \max(0, v_i^k - v_j^k)) = 1$; and for edges $\{i, j\}$ not in the partition, $\sum_{k \in K} \max(0, v_j^k - v_i^k) = \sum_{k \in K} \max(0, v_i^k - v_j^k) = 0$. Thus, in accordance with Theorem 4.3 the edges across the partition have coefficient 1; all others have coefficient 0. Since we have only p commodities, the resulting righthand side is $\pi_0 = \sum_k v_{D(k)}^k = p$.

5.1 Partition Inequalities for the Unitary NDC Problem

Since the structure of the projection cone obtained from the improved flow model for the NDLC problem and the improved flow model for the unitary NDC problem are identical, any member of the cone that provides a valid inequality for the NDLC problem also provides a valid inequality for the unitary NDC problem.

Consider a partition $N_0, \ldots, N_p$. Without loss of generality assume that the root node is a node in N_0. Consider any set N_i of the partition. If $\text{econ}(N_i) \geq 2$, Property 3.4

implies the improved flow formulation must contain two commodities: one with origin $n_i \in N_i$ and destination some node $m_i \notin N_i$, and one with destination $n_i \in N_i$ and origin $m_i \notin N_i$, both with a flow requirement of $\text{econ}(N_i)/2$. Let i denote the commodity with destination node $n_i \in N_i$ (and origin $m_i \notin N_i$) and a requirement of $\text{econ}(N_i)/2$. Similarly, if $\text{econ}(N_i) = 1$ and $i \neq 0$, the improved flow formulation must contain a commodity i with destination a node $n_i \in N_i$ (the origin would be the root node) and a requirement of 1. For commodities $0, \ldots, p$, we set

$$v_i^k = \begin{cases} 1 & \text{if } i \in N_k, \\ 0 & \text{otherwise,} \end{cases}$$

and set $v_i^k = 0$ otherwise.

We have not changed anything so far. These values for v_i^k variables are the same as for the NDLC problem. Therefore, for all edges in the partition, $\pi_{ij} = 1$ if i and j are in different sets of the partition and $\pi_{ij} = 0$ otherwise. For commodities $k = 0, \ldots, p$, $q_k = 1$ if $\text{econ}(N_k) = 1$, and $q_k = \text{econ}(N_k)/2$ if $\text{econ}(N_k) \geq 2$. Substituting these values, we obtain the following valid inequality:

$$\frac{1}{2}\sum_{k=0}^{k=p}\sum_{\delta(N_k)} x_{ij} \geq \begin{cases} p & \text{if } I_2 = \phi; \\ \frac{1}{2}\sum_{i \in I_2} \text{econ}(N_i) + |I_1| & \text{otherwise.} \end{cases} \tag{8}$$

In this inequality, $I_1 = \{i : \text{econ}(N_i) = 1\}$ and $I_2 = \{i : i \in \text{econ}(N_i) \geq 2\}$.

But since, in the mixed integer program, the variables x_{ij} are either 0 or 1 we can round up the righthand side in this inequality and still maintain feasibility. Thus, the following inequalities are valid:

$$\frac{1}{2}\sum_{k=0}^{k=p}\sum_{\delta(N_k)} x_{ij} \geq \begin{cases} p & \text{if } I_2 = \phi \\ \lceil \frac{1}{2}\sum_{i \in I_2} \text{econ}(N_i) \rceil + |I_1| & \text{otherwise.} \end{cases} \tag{9}$$

Grötschel, Monma and Stoer [GMS95b, Sto92] call inequalities (9) partition inequalities and show that under appropriate conditions they are facet defining for the SND problem. Our derivation shows that these inequalities are valid for the more general unitary NDC problem. If the number of sets in the partition inequality with odd connectivity requirement greater than one is odd, then the improved undirected formulation implies a weaker form of the partition inequality that we refer to as weak partition inequalities (i.e., inequalities (8)). Otherwise, the formulation implies the partition inequality. Note that the weak partition inequalities are stronger than the cutset formulation (as long as $I_1 \neq \phi$).

Since the flow model implies the weak partition inequalities, and does not always imply the partition inequalities, we might like to characterize, in a certain sense, how much stronger a model containing the partition inequalities would be compared to a model containing the weak partition inequalities.

To compare two classes of valid inequalities, we use the following notion previously introduced by Goemans [Goe95]. Let $\boldsymbol{\mathcal{X}_1}$ and $\boldsymbol{\mathcal{X}_2}$ be two classes of valid inequalities. Then, the *relative strength* of the class of the valid inequalities $\boldsymbol{\mathcal{X}_1}$ to the class of the valid inequalities $\boldsymbol{\mathcal{X}_2}$ is defined as

$$\max \left\{ \frac{\{\min \mathbf{cx} : \mathbf{x} \in \boldsymbol{\mathcal{R}}_+^{|E|}; \mathbf{x} \in \boldsymbol{\mathcal{X}_1}; \mathbf{x} \in \boldsymbol{\mathcal{X}_2}\}}{\{\min \mathbf{cx} : \mathbf{x} \in \boldsymbol{\mathcal{R}}_+^{|E|}; \mathbf{x} \in \boldsymbol{\mathcal{X}_2}\}} \right\}.$$

The relative strength measures how much, in the best case, the objective function of a linear program that contains the class of valid inequalities $\boldsymbol{\mathcal{X}_2}$ improves by adding to it the class of valid inequalities $\boldsymbol{\mathcal{X}_1}$.

The following result characterizes the relative strength of the partition inequalities with respect to the weak partition inequalities when (unlimited) edge replication is permitted.

Theorem 5.1 *The relative strength of the class of partition inequalities $\boldsymbol{\mathcal{X}_1}$ with respect to the class of weak partition inequalities $\boldsymbol{\mathcal{X}_2}$ is $\frac{10}{9}$.*

Proof: We will show that by multiplying any feasible solution (including any optimal solution) to the linear program $\{\min \mathbf{cx} : \mathbf{x} \in \boldsymbol{\mathcal{R}}_+^{|E|}; \mathbf{x} \in \boldsymbol{\mathcal{X}_2}\}$ (we refer to this linear program as LP2) by $\frac{10}{9}$ gives a feasible solution to the linear program $\{\min \mathbf{cx} : \mathbf{x} \in \boldsymbol{\mathcal{R}}_+^{|E|}; \mathbf{x} \in \boldsymbol{\mathcal{X}_1}; \mathbf{x} \in \boldsymbol{\mathcal{X}_2}\}$ (we refer to this linear program as LP1). This result implies that the optimal value to LP1 is at most $\frac{10}{9}$ times the optimal value to LP2. Note that the weak partition inequalities are implied by the partition inequalities. Consequently, we can delete $\mathbf{x} \in \boldsymbol{\mathcal{X}_2}$ from LP1.

LP1 and LP2 differ when the righthand side of the weak partition inequalities is fractional. If we show the maximum ratio between the righthand side of any partition inequality and its corresponding weak partition inequality is $\frac{10}{9}$, we have shown that the relative strength of the partition inequalities with respect to the weak partition inequalities is $\frac{10}{9}$. (Since multiplying any solution that satisfies the weak partition inequalities by $\frac{10}{9}$ gives a solution that satisfies the partition inequalities.)

The righthand sides of the weak partition inequalities and the partition inequalities differ by at most 0.5. This happens when the cardinality of the set $\{i : \text{econ}(N_i) \text{ odd; and } \text{econ}(N_i) \geq 3\}$ is odd (i.e., for an odd number of sets, $\text{econ}(N_i)$ is (i) odd, and (ii) greater than or equal to 3). Noting that the partition contains at least two sets with the highest value of $\text{econ}(N_i)$, we find that the maximum ratio is obtained by considering a partition with three sets, each with connectivity 3. The righthand side for the weak partition inequality is 4.5 and the righthand side for the partition inequality is 5. The ratio is $\frac{10}{9}$. ■

6 Odd-Hole Inequalities

Chopra and Rao [CR94a] introduced odd-hole inequalities for the Steiner tree problem. These inequalities have the following form. Consider a graph $G_p = (N_p, E_p)$ with $N_p =$

$(T_p \cup Z_p)$, $T_p = \{a_0, a_1, \ldots, a_p\}$ the set of nodes with nonzero connectivity requirements, and $Z_p = \{b_0, b_1, \ldots, b_p\}$ the set of nodes with zero connectivity requirements. The edge set E_p consists of edges of the form $\{a_i, b_i\}$, $\{a_i, b_{i-1}\}$ and $\{b_i, b_{i-1}\}$ for $i = 0, \ldots, p$. Note that we define $b_{-1} = b_p$, $a_{-1} = a_p$, $b_{p+1} = b_0$ and $a_{p+1} = a_0$. An odd-hole inequality is an inequality of the form

$$\sum_{\{i,j\} \in E_p} x_{ij} \geq 2p.$$

When p is even, the graph G_p is called an odd-hole.[2] Chopra and Rao [CR94a] show that if p is even and $p \geq 2$, then an odd-hole inequality defines a facet for the Steiner tree polytope on the graph G_p.

We extend the definition of odd-hole inequalities to obtain the following valid inequalities for the NDLC problem.

$$\sum_{\{i,j\} \in E_p} x_{ij} \geq \begin{cases} 2(p+1) & \text{if at least two nodes have a} \\ & \text{connectivity requirement of two;} \\ 2p & \text{otherwise.} \end{cases}$$

To prove that these inequalities are valid, we now show how to project these inequalities from the improved undirected formulation.

Consider the odd-hole shown in Figure 5. Nodes a_0 and a_1 have a connectivity requirement of 2, and a_2, a_3 and a_4 have a connectivity requirement of 1. The improved undirected formulation, with root node a_0, has five commodities. Commodities 1, 2, 3 and 4 have a_0 as the origin and a_1, a_2, a_3 and a_4 as the destination nodes. Commodity 5, which we will also refer to as commodity 0, has a_1 as the origin and a_0 as the destination. Note, with this notation, each commodity i has destination a_i.

In the example of Figure 5, for all commodities $k = 0, \ldots, 4$, let $v_i^k = 0$ if $i \neq a_k, b_k$ or b_{k-1}. Let $v_{a_k}^k = 2$ and $v_{b_k}^k = v_{b_{k-1}}^k = 1$. Notice that with this choice of values

$$\begin{aligned} v_{a_i}^i - v_{b_i}^i &= v_{b_i}^{i+1} - v_{a_i}^{i+1} &= 1 \\ v_{a_i}^i - v_{b_{i-1}}^i &= v_{b_{i-1}}^{i-1} - v_{a_i}^{i-1} &= 1 \\ v_{b_{i-1}}^{i-1} - v_{b_i}^{i-1} &= v_{b_i}^{i+1} - v_{b_{i-1}}^{i+1} &= 1. \end{aligned}$$

All other $v_i^k - v_j^k = 0$. Thus, for each edge $\{i,j\}$, $\sum_{k \in K} \max(0, v_j^k - v_i^k) = \sum_{k \in K} \max(0, v_i^k - v_j^k) = 1$. Therefore in accordance with Theorem 4.3 we can set $u_{a_k b_k}^{(k+1)k}$, $u_{a_k b_{k-1}}^{(k-1)k}$ and $u_{b_k b_{k-1}}^{(k+1)(k-1)}$ to 1 for $k = 0, \ldots, 4$, and $u_{ij}^{hk} = 0$ otherwise. Since $\sum_{k \in K} v_{a_k}^k = 10$, this choice of variables defines the odd-hole inequality $\sum_{\{i,j\} \in E_4} x_{ij} \geq 10$.

In general, let node a_0 be the root node. For each node a_l, select a commodity l with $D(l) = a_l$. For commodities $k \in \{0, \ldots, p\}$, set

[2] The odd in odd-hole refers to the number of nodes with nonzero connectivity requirements.

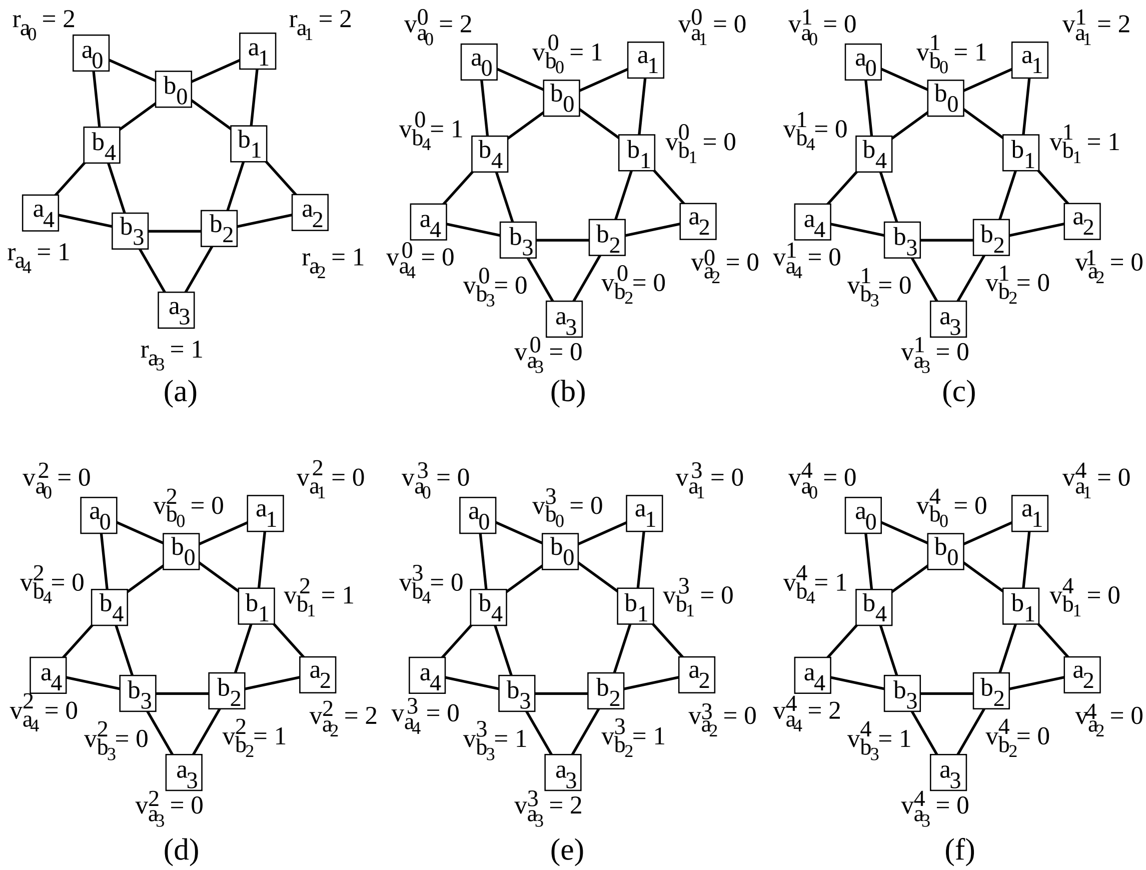


Figure 5: Projecting odd-hole inequalities for a ten node example. (a) Connectivity requirements (b) Value of the v_i^0 variables (c) Value of the v_i^1 variables (d) Value of the v_i^2 variables (e) Value of the v_i^3 variables (f) Value of the v_i^4 variables.

$$v_i^k = \begin{cases} 2 & \text{if } i = a_k; \\ 1 & \text{if } i = b_k \text{ or } i = b_{k-1}; \\ 0 & \text{otherwise,} \end{cases}$$

and set all other v_i^k values to zero.

With this choice of the v_i^k variables, we once again ensure that $\sum_{k \in K} \max(0, v_j^k - v_i^k) = \sum_{k \in K} \max(0, v_i^k - v_j^k) = 1$ for any edge $\{i, j\}$. Thus, by choosing the values of u_{ij}^{kh} as indicated in Theorem 4.3, we find that $\pi_{ij} = \sum_{k \in K} \sum_{h \in K} u_{ij}^{kh}$ is 1 for each edge $\{i, j\}$; and $\pi_0 = \sum_{k \in K} v_{D(k)}^k = 2(p+1)$, since $v_{D(k)}^k = 2$ for commodities $0, \ldots, p$, and $v_{D(k)}^k = 0$ otherwise.

Suppose the graph G_p has more edges than E_p. What inequality is implied? Retain the same choice of v_i^k variables and consider an edge $\{i, j\} \notin E_p$. This edge can be of three types—$\{a_i, b_j\}$, $\{b_i, b_j\}$ and $\{a_i, a_j\}$. For edges of type $\{a_i, a_j\}$,

$$v^j_{a_j} - v^j_{a_i} = v^i_{a_i} - v^i_{a_j} = 2.$$

For edges of type $\{a_i, b_j\} \notin E_p$, $j \neq i$ and $j \neq i-1$,

$$v^j_{b_j} - v^j_{a_i} = v^{j+1}_{b_j} - v^{j+1}_{a_i} = 1$$

and

$$v^i_{a_i} - v^i_{b_j} = 2.$$

Finally, for edges of type $\{b_i, b_j\} \notin E_p$, $j \neq i-1$ and $j \neq i+1$,

$$v^j_{b_j} - v^j_{b_i} = v^{j+1}_{b_j} - v^{j+1}_{b_i} = v^i_{b_i} - v^i_{b_j} = v^{i+1}_{b_i} - v^{i+1}_{b_j} = 1.$$

Thus, for any edge $\{i, j\} \notin E_p$,

$$\textstyle\sum_{k\in K} \max\left(0, v^k_i - v^k_j\right) = \sum_{k\in K} \max\left(0, v^k_j - v^k_i\right) = 2.$$

Selecting u^{kh}_{ij} values as prescribed by Theorem 4.3, shows that $\pi_{ij} = \sum_{k\in K}\sum_{h\in K} u^{kh}_{ij}$ is 2 for each edge $\{i, j\} \notin E_p$. The projected inequality

$$\sum_{\{i,j\}\in E_p} x_{ij} + \sum_{\{i,j\}\in E\setminus E_p} 2x_{ij} \geq 2(p+1)$$

is called a lifted odd-hole inequality and is also a facet-defining inequality for the Steiner tree problem (see Chopra and Rao [CR94a]).

If the problem has no node with a connectivity requirement of two, it does not contain any commodity with destination the root node. Therefore we select p commodities, one for each node $a_1, \ldots, a_p$, each with origin a_0 (the root node). With the same choice of v^k_i for commodities $1, \ldots, p$, we obtain $\max\left(\sum_{k\in K} \max\left(0, v^k_j - v^k_i\right), \sum_{k\in K} \max\left(0, v^k_i - v^k_j\right)\right) = 1$ for all edges $\{i, j\} \in E_p$ and $\max\left(\sum_{k\in K} \max\left(0, v^k_j - v^k_i\right), \sum_{k\in K} \max\left(0, v^k_i - v^k_j\right)\right) = 2$ for edges $\{i, j\} \in E\setminus E_p$. Since we have only p commodities, the resulting righthand side is $\pi_0 = \sum_{k\in K} v^k_{D(k)} = 2p$.

Both the odd-hole inequality and its lifted version are new valid inequalities for the NDLC problem. By examining the cone element that generates the odd-hole inequality, we can obtain valid odd-hole inequalities for the unitary NDC problem. In the next section, we study a generalization of odd-hole inequalities called combinatorial design inequalities. Therefore, to avoid repetition, we will not derive odd-hole inequalities for the unitary NDC problem.

7 Combinatorial Design Inequalities

Goemans [Goe94] introduced a new class of facet defining valid inequalities called *combinatorial design inequalities* for Steiner tree problems. He showed that under appropriate conditions combinatorial design inequalities are facet defining for the Steiner tree problem. He derives the combinatorial design inequalities by projecting from a node weighted (undirected) extended formulation for the Steiner tree problem. We show how to project out the combinatorial design inequalities from the improved undirected flow formulation (5), and as a result generalize the combinatorial design inequality to the unitary NDC problem, obtaining a new class of valid inequalities for this problem.

The description of the combinatorial design inequality is fairly involved. Let $T_p = \{a_0, \ldots, a_p\}$ be the set of nodes with nonzero connectivity requirements. $Z_q = N - T_p = \{b_0, \ldots, b_q\}$ is the set of nodes with zero connectivity requirements. Associate with each node a_i of T_p a subset T_{a_i} containing elements of Z_q. Based on these subsets, we also define sets Z_{b_i} associated with each node b_i in Z_q. Z_{b_i} contains those elements of T_p whose associated subset contains the node b_i. For example, consider the odd hole with $p = 4$ (see Figure 5). $T_4 = \{a_0, a_1, a_2, a_3, a_4\}$ and $Z_4 = \{b_0, b_1, b_2, b_3, b_4\}$. Here $T_{a_0} = \{b_0, b_4\}$, $T_{a_1} = \{b_0, b_1\}$, $T_{a_2} = \{b_1, b_2\}$, $T_{a_3} = \{b_2, b_3\}$, $T_{a_4} = \{b_3, b_4\}$. Thus, the sets associated with the elements in Z_4 are $Z_{b_0} = \{a_0, a_1\}$, $Z_{b_1} = \{a_1, a_2\}$, $Z_{b_2} = \{a_2, a_3\}$, $Z_{b_3} = \{a_3, a_4\}$, $Z_{b_4} = \{a_4, a_0\}$. (The odd-hole inequality is a special case of the combinatorial design inequality).

Define the $(q + 1) \times (p + 1)$ matrix $\mathbf{D} = [d_{ij}]$ with $d_{ij} = 1$ if $a_j \in Z_{b_i}$ and $d_{ij} = 0$ otherwise. The following two conditions are imposed on $\mathbf{D}$: (i) rank$(\mathbf{D}) = p + 1$, and (ii) the unit vector e belongs to the cone generated by the columns of $\mathbf{D}$; i.e., $\mathbf{Dy} = \mathbf{e}$ for some vector $\mathbf{y} \in \mathcal{R}_+^{(p+1)}$. For any fixed $d > 0$, if we set $\beta = d\mathbf{y}$, we see that $\mathbf{D}\beta = d\mathbf{e}$. Letting β_j denote the jth component of β (i.e., $\beta_j = dy_j$), we see that

$$\sum_{a_j \in Z_{b_i}} \beta_j \quad = \quad d \qquad \text{for all } i = 0, 1, \ldots, q. \tag{10}$$

If we select d so that the greatest common divisor of $\beta_0, \beta_1, \ldots, \beta_p$, and d is 1, the coefficients of x_{ij} in the following combinatorial design inequalities will be integer and as small as possible. For every edge $\{s, t\}$, define

$$d_{st} = \begin{cases} \sum_{k: a_k \in (Z_{b_i} \cap Z_{b_j})} \beta_k & \{s,t\} = \{b_i, b_j\} \text{ where } b_i, b_j \in Z_p; \\ \beta_j & \{s,t\} = \{a_j, b_i\} \text{ with } b_i \in T_{a_j}; \\ 0 & \text{otherwise.} \end{cases}$$

For the Steiner tree problem, the inequality

$$\sum_{\{i,j\} \in E} (d - d_{ij}) x_{ij} \geq dp,$$

is a combinatorial design inequality. The odd-hole inequality is a special case of the combinatorial design inequality when $d = 2$, $\beta_j = 1$ for $j = 0, \ldots, p$, and $|T_p| = |Z_q|$.

We extend the definition of the combinatorial design inequality to obtain the following valid inequalities for the NDLC problem.

$$\sum_{\{i,j\}\in E}(d-d_{ij})x_{ij} \geq \begin{cases} d(p+1) & \text{if at least two nodes have a} \\ & \text{connectivity requirement of two;} \\ dp & \text{otherwise.} \end{cases} \tag{11}$$

Let us now project the combinatorial design inequality from the improved undirected flow formulation. In the improved undirected flow formulation, let node a_0 be the root node. For each node a_l, select a commodity l with destination a_l and set

$$v_i^k = \begin{cases} d & \text{if } i = a_k; \\ \beta_k & \text{if } i \in T_{a_k}; \\ 0 & \text{otherwise.} \end{cases}$$

For all other commodities k set v_i^k to zero.

Consider an edge $\{s,t\} = \{a_j, b_i\}$ with $b_i \in T_{a_j}$. When a_j is the destination node, $v_{a_j}^j = d$ and $v_{b_i}^j = \beta_j$ if $i \in T_{a_j}$. Consider any node $a_k \in Z_{b_i}\backslash\{a_j\}$. $v_{a_k}^l \neq 0$ only if a_k is a destination node for commodity l, i.e., $l = k$ and, in this case, $v_{b_i}^k = \beta_k$ and $v_{a_j}^k = 0$. Note that, $v_{b_i}^k - v_{a_j}^k = \beta_k$. Thus,[3]

$$\sum_{a_k\in\{Z_{b_i}\backslash\{a_j\}\}} v_{b_i}^k - v_{a_j}^k = \sum_{a_k\in\{Z_{b_i}\backslash\{a_j\}\}} \beta_k \overset{(10)}{=} d - \beta_j = v_{a_j}^j - v_{b_i}^j.$$

If $a_k \notin Z_{b_i}$, then $v_{a_j}^k = v_{b_i}^k = 0$. Therefore, $\sum_{k\in K} \max(0, v_{a_j}^k - v_{b_i}^k) = \sum_{k\in K} \max(0, v_{b_i}^k - v_{a_j}^k) = d - \beta_j$. By Theorem 4.3, in the projected inequality $d - \beta_j$ is the coefficient of an edge $\{s,t\} = \{a_j, b_i\}$ with $b_i \in T_{a_j}$.

Consider an edge $\{s,t\} = \{b_i, b_j\}$. For any commodity k with destination a_k, $v_{b_i}^k$ and $v_{b_j}^k$ differ only if exactly one of b_i and b_j belongs to T_{a_k}. If b_i belongs to T_{a_k} then $v_{b_i}^k = \beta_k$ and $v_{b_j}^k = 0$. Thus $v_{b_i}^k - v_{b_j}^k = \beta_k$. Summing over all sets T_{a_k} that contain node b_i but not node b_j, we find that

$$\begin{aligned} \sum_{k:b_i\in T_{a_k};b_j\notin T_{a_k}} v_{b_i}^k - v_{b_j}^k &= \sum_{k:b_i\in T_{a_k};b_j\notin T_{a_k}} \beta_k \\ &\overset{(10)}{=} d - \sum_{k:b_i\in T_{a_k};b_j\in T_{a_k}} \beta_k. \end{aligned}$$

Similarly, summing up over all sets T_{a_k} that contain node b_j but not node b_i, we find that

$$\begin{aligned} \sum_{k:b_j\in T_{a_k};b_i\notin T_{a_k}} v_{b_j}^k - v_{b_i}^k &= \sum_{k:b_j\in T_{a_k};b_i\notin T_{a_k}} \beta_k \\ &\overset{(10)}{=} d - \sum_{k:b_j\in T_{a_k};b_i\in T_{a_k}} \beta_k. \end{aligned}$$

[3]The notation $\overset{(10)}{=}$ means that the equality follows from expression (10).

Therefore, $\sum_{k \in K} \max(0, v^k_{b_j} - v^k_{b_i}) = \sum_{k \in K} \max(0, v^k_{b_i} - v^k_{b_j}) = d - \sum_{k: b_j \in T_{a_k}; b_i \in T_{a_k}} \beta_k$. By Theorem 4.3, $d - \sum_{k: b_j \in T_{a_k}; b_i \in T_{a_k}} \beta_k$ is the coefficient of an edge $\{s,t\} = \{b_i, b_j\}$ in the projected inequality.

All the other edges are either of the form $\{s,t\} = \{a_i, a_j\}$ or $\{s,t\} = \{a_j, b_i\}$ with $b_i \notin T_{a_j}$. For any edge of the form $\{a_i, a_j\}$, $v^j_{a_j} - v^j_{a_i} = v^i_{a_i} - v^i_{a_j} = d$ for commodities i and j and $v^k_{a_i} = v^k_{a_j} = 0$ otherwise. Consider an edge of the form $\{s,t\} = \{a_j, b_i\}$ with $b_i \notin T_{a_j}$. For all commodities k with $a_k \in Z_{b_i}$, i.e. a_k is the destination, $v^k_{b_i} - v^k_{a_i} = \beta_k$. For commodity j, a_j is the destination and $v^j_{a_j} - v^j_{b_i} = d$. For all other commodities k, $v^k_{a_j} = v^k_{b_i} = 0$. Therefore, for both edges of the form $\{s,t\} = \{a_i, a_j\}$ and $\{s,t\} = \{a_j, b_i\}$ with $b_i \notin T_{a_j}$, we find that $\sum_{k \in K} \max(0, v^k_s - v^k_t) = \sum_{k \in K} \max(0, v^k_s - v^k_t) = d$. By Theorem 4.3, d is the coefficient of these edges in the projected inequality.

The righthand side of the projected inequality is $\pi_0 = \sum_k v^k_{D(k)} = d(p+1)$ since $v^k_{D(k)} = d$ for commodities $0, \ldots, p$, and $v^k_{D(k)} = 0$ otherwise.

If the problem has exactly one node or no nodes with a connectivity requirement of 2, with the same choice of v^k_i variables we obtain the same coefficients for x_{ij} and a righthand side of dp (there will be no commodity with destination the root node a_0).

7.1 Combinatorial Design Inequalities for the Unitary NDC Problem

To conclude this discussion, we examine the valid inequality for the unitary NDC problem that is implied by the cone element that generates the combinatorial design inequality for the NDLC problem.

For each node a_k, $k = 0, \ldots, p$, we select a commodity k with destination a_k as follows. If the maximum connectivity requirement r_{a_k} of node a_k is greater than or equal to 2, then we select a commodity k with destination node a_k and flow requirement $r_{a_k}/2$. If the maximum connectivity requirement r_{a_k} of node a_k is equal to 1, then we select a commodity k with destination node a_k and flow requirement 1. Retain the same choice of variables v^k_i. The projected inequality is

$$\sum_{\{i,j\} \in E} (d - d_{ij}) x_{ij} \geq \begin{cases} dp & \text{if } L_2 = \phi, \\ d(\frac{1}{2} \sum_{l \in L_2} r_l + |L_1|) & \text{otherwise.} \end{cases} \tag{12}$$

In this inequality $L_1 = \{i : i \in T_p, r_i = 1\}$ and $L_2 = \{i : i \in T_p, r_i \geq 2\}$ (r_i denotes the maximum connectivity requirement of a node).

Once again, noting that the lefthand side should be integer if the x variables are integer, we can round up the righthand side, giving the inequality

$$\sum_{\{i,j\} \in E} (d - d_{ij}) x_{ij} \geq \begin{cases} dp & \text{if } L_2 = \phi, \\ (\lceil \frac{d}{2} \sum_{l \in L_2} r_l \rceil + d|L_1|) & \text{otherwise.} \end{cases} \tag{13}$$

We refer to inequalities (12) and (13) as weak combinatorial design and combinatorial design inequalities. Noting that $d \geq 1$, it is easy to prove a result similar to Theorem 5.1—

namely, if edge replication is permitted, the value of the linear program determined by the weak combinatorial design inequalities is at least $\frac{10}{9}$ths the value of the linear program determined by the combinatorial design inequalities.

8 Nonunitary Problems

So far we have restricted our attention to unitary problems. We now examine nonunitary NDC problems. Our starting point will be the special case of the Steiner forest problem. Recall that in the Steiner forest problem we are given a graph $G = (N, E)$ and node sets $T_1, T_2, \ldots, T_P$ with $T_i \cap T_j = \phi$ for all node set pairs $(i \neq j)$. We wish to design a graph at minimum cost that connects the nodes in each node set (possibly by including multiple node sets in any connected component of the graph).

For the unitary NDC problem, we derived a stronger formulation by generalizing a well-known directing procedure for the Steiner tree problem. The essential idea used was to direct the bridge edges of a solution to the unitary NDC problem in a manner akin to the directing procedure for the Steiner tree. Analogously, to strengthen the formulation for the nonunitary NDC problem we will first determine how to strengthen the Steiner forest problem—a nonunitary NDC problem with each $r_{st} \in \{0, 1\}$—by directing it.

To our knowledge, we are unaware of any models stronger than the cutset model for the Steiner forest problem. We believe the model we present is the first directed model in the literature for this problem.

8.1 Directing the Steiner Forest Problem

For convenience, we once again describe the cutset and undirected flow formulations for the NDC problem as applied to the Steiner forest problem.

Cutset formulation for the Steiner forest problem

$$\text{Minimize} \quad \sum_{\{i,j\} \in E} c_{ij} x_{ij} \tag{14a}$$

$$\text{subject to:} \quad \sum_{\{i,j\} \in \delta(S)} x_{ij} \geq 1 \quad \begin{array}{l} \text{for all } S,\ \phi \subset S \subset N, \\ S \cap T_i \neq \phi \text{ and } (N \backslash S) \cap T_i \neq \phi \text{ for some i,} \end{array} \tag{14b}$$

$$x_{ij} \leq 1 \quad \text{for all } \{i,j\} \in E \tag{14c}$$

$$x_{ij} \geq 0 \quad \text{and integer.} \tag{14d}$$

In the following undirected flow formulation for the Steiner forest problem, we select a root node for each node set and send one unit of flow from the root node of each node set to every node in that node set. Recall that K denotes the set of all commodities in this formulation.

Undirected flow formulation for Steiner forest problem:

$$\text{Minimize} \quad \sum_{\{i,j\}\in E} c_{ij}x_{ij} \tag{15a}$$

$$\text{subject to:} \quad \sum_{j\in N} f_{ji}^k - \sum_{l\in N} f_{il}^k = \begin{cases} -1 & \text{if } i = O(k); \\ 1 & \text{if } i = D(k); \\ 0 & \text{otherwise}; \end{cases} \quad \text{for all } i \in N \text{ and } k \in K \tag{15b}$$

$$\left.\begin{array}{c} f_{ij}^k \\ f_{ji}^k \end{array}\right\} \le x_{ij} \qquad \text{for all } \{i,j\} \in E \text{ and } k \in K \tag{15c}$$

$$f_{ij}^k, f_{ji}^k \ge 0 \qquad \text{for all } \{i,j\} \in E \text{ and } k \in K \tag{15d}$$

$$x_{ij} \le 1 \qquad \text{for all } \{i,j\} \in E \tag{15e}$$

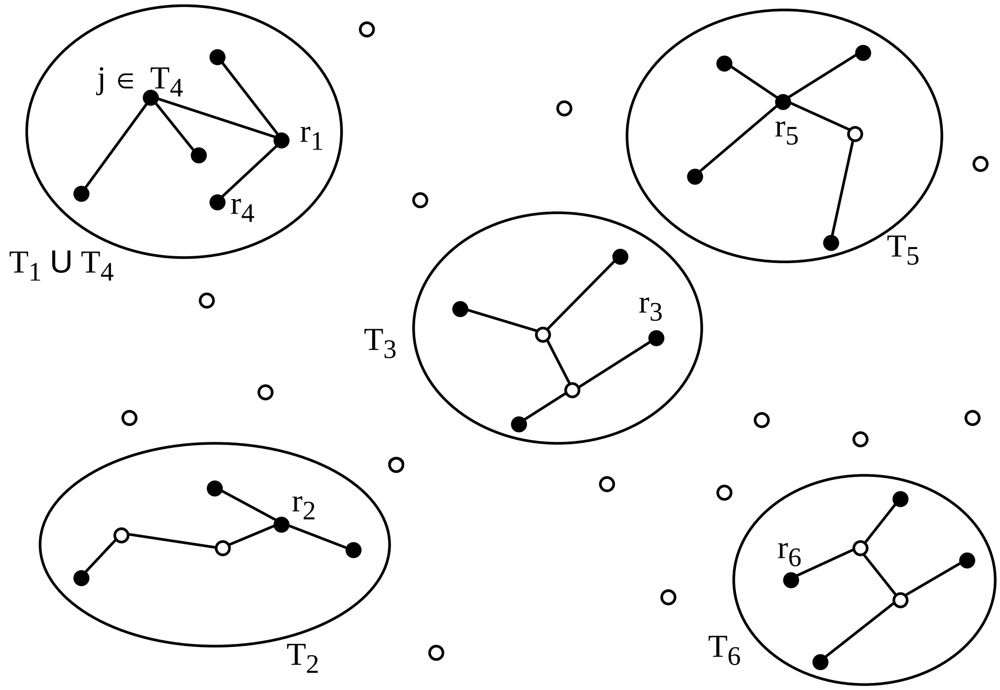
$$x_{ij} \ge 0 \qquad \text{and integer; for all } \{i,j\} \in E. \tag{15f}$$

If we assume each $c_{ij} \ge 0$, these formulations always have a Steiner forest as an optimal solution, and so each component of the forest is a tree. Nodes belonging to any node set T_i, for any i, lie in the same component. As an example, Figure 6a shows the optimal solution to a Steiner forest problem with five components. One component contains the two node sets T_1 and T_4. All the other components do not contain nodes from other node sets.

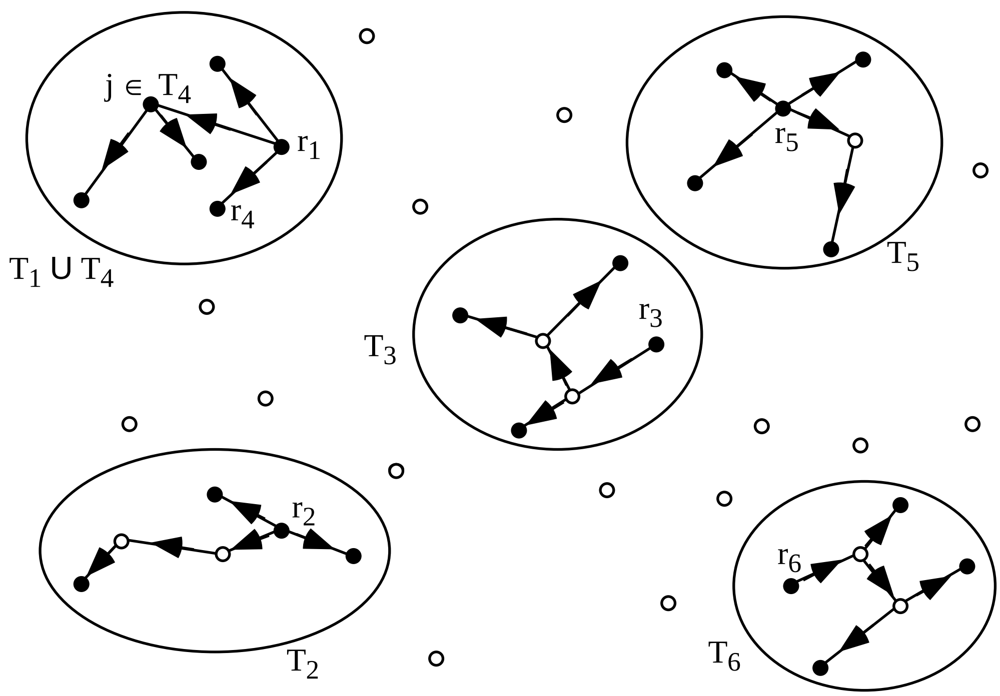
How might we direct the Steiner forest problem? Since each component in the optimal solution is a tree, we could arbitrarily choose a node in each component and direct each tree away from it. Unfortunately, before we solve the problem, although we know that nodes in each node set will lie in the same component, we do not know the number of components in the optimal solution and the node sets they contain. The problem is to determine, *a priori*, the root node for each component. For this reason, directing the Steiner forest problem raises difficulties not encountered in directing the Steiner tree (and the unitary NDC) problem.

To direct the Steiner forest, for each set T_i, we choose an arbitrary root node $r_i \in T_i$. We then direct each component (tree) away from the lowest indexed root node that it contains. In the example shown in Figure 6(a), one component contains two node sets T_1 and T_4. Since T_1 is the lowest indexed node set in this component, we have directed the component away from the root node r_1 of node set T_1. All the other components contain nodes from only one node set T_i, for $i = 1, 2, 3, 4, 5$, and we direct each of them away from the root node r_i of node set T_i. Figure 6(b) shows the forest after we have applied the directing procedure.

For notation, if $j \in T_i$, we let $\rho(j) = r_i$ denote the root node of the node set T_i that contains node j. We refer to r_i as node j's root node. We also define $T = T_1 \cup T_2 \cup \ldots T_P$, and let R be the set of all root nodes, that is, $R = \{r_1, r_2, \ldots, r_P\}$.



(a) Undirected forest.



(b) Direct each forest away from the lowest indexed root node that it contains.

Figure 6: Example of the directing procedure.

8.2 Improved Flow Formulation for the Steiner Forest Problem

We model the Steiner forest problem using multicommodity flows. Since the network we obtain after directing the Steiner forest contains a directed path from the lowest indexed root node in a component to all other nodes in that component, we can send a unit of flow from the root node of each directed component to every node in that component.

For each node $j \in T_i$, with $j \neq r_i$, and for each $p \leq i$, we define a commodity with origin node r_p and destination node j, and for each root node r_i and for each $p < i$, we define a commodity with origin node r_p and destination node r_i. In the optimal solution, it is possible to send a unit of flow from the lowest indexed root node of a component to each required node in that component. Let $CO(q)$ denote the set of all commodities that have node q as their origin, and $CD(q)$ denote the set of all nodes that have node q as their destination and let K denote the set of all commodities. We also define $\mathcal{H} = \{S : S \subset K, \text{ and} \, |S \cap CO(r_j)| = 1 \, \text{for all } j = 1, \ldots, P\}$. That is, each member of $\mathcal{H}$ is a set of commodities with exactly one commodity having each root node as its origin.

Let x_{ij} be 1 if the network design contains edge $\{i, j\}$ and be 0 otherwise. The improved undirected flow formulation for the Steiner forest problem has the following form.

Improved undirected flow formulation for Steiner forest problem:

$$\text{Minimize} \quad \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{16a}$$

$$\text{subject to} \quad \sum_{j\in N} f_{ji}^k - \sum_{l\in N} f_{il}^k \begin{cases} \geq \; -1 & \text{if } i = O(k); \\ \leq \; 1 & \text{if } i = D(k); \\ = \; 0 & \text{otherwise}; \end{cases} \left.\begin{array}{c} \forall i \in N; \\ \text{and} \\ k \in K; \end{array}\right\} \tag{16b}$$

$$\sum_{k\in CD(i)} \sum_{j\in N} f_{jD(k)}^k = 1 \qquad \text{for all } i \in T \backslash R \tag{16c}$$

$$\sum_{j\in N} f_{jD(k)}^k \leq \sum_{j\in N} f_{jD(k^*)}^{k^*} \qquad \begin{array}{l} \forall \; i \in T\backslash R, \forall k \in CD(i), \\ \text{s.t. } O(k) = O(k^*), \text{ and} \\ D(k^*) = \rho(i); \end{array} \tag{16d}$$

$$\sum_{k\in H} f_{ij}^k + \sum_{k\in \bar{H}} f_{ji}^k \leq x_{ij} \qquad \begin{array}{l} \text{for all } \{i,j\} \in E, \text{ and} \\ \text{all } H, \bar{H} \text{ pairs in } \mathcal{H} \end{array} \tag{16e}$$

$$\sum_{i\in N} \sum_{k\in H} f_{ij}^k \leq 1 \qquad \begin{array}{l} \text{for all } j \in N, \text{ and} \\ \text{all } H \text{ in } \mathcal{H} \end{array} \tag{16f}$$

$$f_{D(k)l}^k = 0 \qquad \begin{array}{l} \text{for all } l \in N, \text{ and} \\ k \in K \end{array} \tag{16g}$$

$$\left.\begin{array}{c} f_{ij}^k \\ f_{ji}^k \end{array}\right\} \geq 0 \qquad \begin{array}{l} \text{for all } \{i,j\} \in E, \text{ and} \\ k \in K \end{array} \tag{16h}$$

$$x_{ij} \in \{0,1\} \qquad \text{for all } \{i,j\} \in E. \tag{16i}$$

Constraints (16b), (16c), and (16g) ensure that each node i in $T \backslash R$ obtains a unit of flow from either its root node, or the root node of a lower indexed node set. Constraints (16d) and (16g) ensure that if node $i \in T_j$, $i \neq r_j$, is supplied by a commodity k whose origin is not the root node of set T_j, then its root node also is supplied from the origin of commodity k (i.e., its root node belongs to the same component that it belongs to). Note that constraint (16g) simply states that flow of a commodity out of its destination node is zero, and so allows us to simplify notation in constraints (16c) and (16d) (they contain only terms for flow into the destination node). Constraint (16e) follows from the property that in an optimal solution flow travels in only one direction across an edge, and all the flow across an edge originates from the same source (the root node of the component the edge belongs to). Constraint (16f) follows from the fact that flow into any node in a component originates from a single node (the root node of that component).

Figure 7 shows that the improved undirected flow formulation is stronger than the cutset formulation (or undirected flow formulation, since they are equivalent) for the Steiner forest problem. In this example $T_1 = \{a, d\}$, $T_2 = \{b, c\}$, and each edge has a cost of 1 unit. The optimal solution to the cutset formulation sets $x_{ab} = x_{bd} = x_{dc} = x_{ca} = 0.5$ with a cost of 2 units. In the improved undirected flow formulation, we select node a as the root of node set T_1, and select node b as the root of node set T_2. This formulation contains four commodities. Commodities 1, 2, and 3 have origin node a and destinations nodes b, c and d. Commodity 4 has origin node b and destination node c. The optimal solution to the linear programming relaxation of the improved undirected flow formulation sets $x_{ab} = x_{bd} = x_{ad} = 1$, $f^2_{ac} = f^1_{ab} = f^3_{ab} = f^3_{bd} = 1$ (notice that all the commodities originate at the node a, the lowest indexed root node of the optimal solution), and has a cost of 3 units.

Researchers have previously shown that the linear programming relaxations of the improved undirected flow formulation provide integer solutions (i.e., the edge design variables are integer) for the Steiner tree problem when the underlying graph is series-parallel (see [PLG85]). The example in Figure 7 suggests that a similar result might be true for the Steiner forest problem (since the underlying graph in that example is series-parallel). However, adding a single edge $\{b, c\}$ with unit cost to the example in Figure 7 shows that this property is not true. The optimal solution to the linear programming relaxation of the improved undirected flow formulation is $x_{ab} = x_{bd} = x_{dc} = x_{ca} = x_{bc} = 0.5$, $f^3_{ab} = f^3_{bd} = f^3_{ac} = f^3_{cd} = f^1_{ab} = f^2_{ac} = f^4_{bc} = 0.5$, with a cost of 2.5. The optimal integer solution and the optimal solution to the linear programming relaxation of the cutset formulation (or undirected flow formulation) are the same as in the previous example with a cost of 3 units and 2 units respectively. Thus improved flow-based models for Steiner forest problems defined on series-parallel graphs need not have integer solutions.

In this section we modeled the directing procedure on an undirected graph by using directed commodity flows. It is also possible to implement this directing procedure by

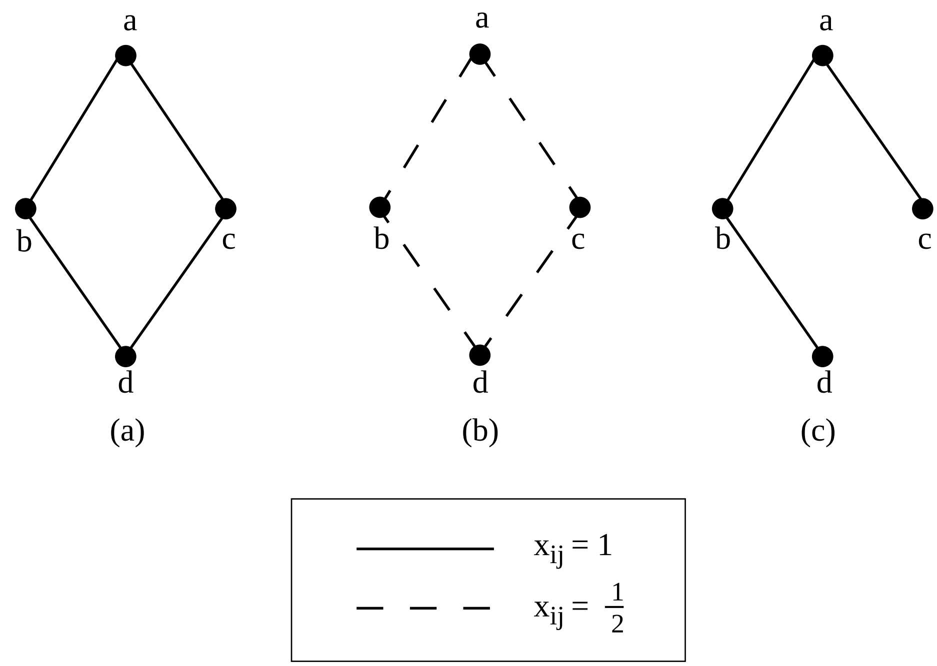


Figure 7: (a) Graph with unit edge costs and $T_1 = \{a, d\}$, and $T_2 = \{b, c\}$. (b) Optimal solution to LP relaxation of cutset formulation. (c) Optimal solution to LP relaxation of improved undirected flow formulation.

transforming the problem onto a directed graph. Raghavan [Rag95] describes equivalent directed flow formulations for the Steiner forest problem.

To conclude this section, we note that we modeled the directing procedure using commodity flows. Unlike the unitary NDC problem, there does not seem to be any obvious way to formulate a directed cut model. This shows the flexibility and power of flow models for modeling network design problems with connectivity constraints.

9 Directing the NDC Problem

In this section we show how to generalize the directing procedure we have just presented for the Steiner forest problem to obtain a directed model for all NDC problems. As a result, we obtain a stronger formulation for the NDC model with edge-connectivity requirements.

To sketch the basic idea underlying the directing procedure, consider any solution to the NDC problem. It consists of one or more connected components. By following the procedure described in Section 3 for each connected component of the integer solution to the NDC problem, we can direct each component. However, like the Steiner forest problem, because the problem is nonunitary, we do not know a priori the number of connected components in the optimal solution and the required nodes they contain. (We do know that if we delete the edges $\{s, t\}$ with $r_{st} = 0$ from the requirement spanning tree, then the nodes that belong to the same tree (component) of the requirement forest will be in the same component of the solution to the NDC problem.) By combining the directing procedure for the unitary NDC problem and the directing procedure for the Steiner forest problem, we obtain a directed model for the NDC problem.

The following commodity selection procedure outlines the essential idea of the directing procedure. We first use the directing procedure described in Section 3.2 to direct the problem for commodities with $r_{st} \geq 2$. We then apply the Steiner forest problem's directing procedure to direct the bridge edges in each component of the optimal integer solution to the NDC problem.

Commodity selection procedure for Formulation (17)

1. Find the requirement spanning tree.

2. Delete all edges with $r_{st} = 0$ from the requirement spanning tree.

3. For each edge $\{s, t\}$ of the requirement spanning tree with $r_{st} \geq 2$, create two commodities: one with origin node s and destination node t, and the other with origin node t and destination node s; each of these commodities has a flow requirement of $r_{st}/2$. Let L denote this set of commodities.

4. Contract each edge $\{s, t\}$ with $r_{st} \geq 2$ in the requirement spanning tree, creating a forest F in which $r_{ij} = 1$ for all edges $\{i, j\}$. Identify the connected components $T_1, T_2, \ldots, T_P$ of this forest. Denote any node in F created by contraction by any of the nodes it contains in the original requirement spanning tree (e.g., if contracting nodes s and t creates a node in F, then we denote the contracted node by s). Select a contracted node in each set T_i as the root node r_i of the node set T_i. (If node set T_i does not contain a contracted node, then arbitrarily select any one of the nodes as the root node.) Now create commodities as described for the Steiner forest problem with node sets $T_1, T_2, \ldots, T_P$, and root nodes $r_1, r_2, \ldots r_P$. Let K denote this set of commodities.

Using this set of commodities we obtain the following improved undirected flow formulation.

Improved undirected flow formulation for NDC problem:

$$\text{Minimize} \quad \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{17a}$$

$$\text{subject to} \quad \sum_{j\in N} f_{ji}^k - \sum_{l\in N} f_{il}^k \begin{cases} \geq \ -1 & \text{if } i = O(k); \\ \leq \quad 1 & \text{if } i = D(k); \\ = \quad 0 & \text{otherwise}; \end{cases} \left.\begin{matrix} \forall i \in N; \\ \text{and} \\ k \in K; \end{matrix}\right. \tag{17b}$$

$$\sum_{j\in N} f^k_{ji} - \sum_{l\in N} f^k_{il} = \begin{cases} -q_k & \text{if } i = O(k); \\ q_k & \text{if } i = D(k); \\ 0 & \text{otherwise;} \end{cases} \quad \forall i \in N; \text{ and } k \in L; \tag{17c}$$

$$\sum_{k\in CD(i)} \sum_{j\in N} f^k_{jD(k)} = 1 \qquad \text{for all } i \in T\backslash R \tag{17d}$$

$$\sum_{j\in N} f^k_{jD(k)} \leq \sum_{j\in N} f^{k^*}_{jD(k^*)} \qquad \forall\ i \in T\backslash R,\ \forall k \in CD(i), \text{ s.t. } O(k) = O(k^*), \text{ and } D(k^*) = \rho(i); \tag{17e}$$

$$\sum_{k\in H} f^k_{ij} + \sum_{k\in \bar{H}} f^k_{ji} \leq x_{ij} \qquad \text{for all } \{i,j\} \in E, \text{ and all } H, \bar{H} \text{ pairs in } \mathcal{H} \tag{17f}$$

$$\sum_{i\in N} \sum_{k\in H} f^k_{ij} \leq 1 \qquad \text{for all } j \in N, \text{ and all } H \text{ in } \mathcal{H} \tag{17g}$$

$$f^k_{ij} + f^h_{ji} \leq x_{ij} \qquad \text{for all } k, h \in K \cup L \tag{17h}$$

$$f^k_{D(k)l} = 0 \qquad \text{for all } l \in N \text{ and } k \in K \tag{17i}$$

$$x_{ij} \leq 1 \qquad \text{for all } \{i,j\} \in E \tag{17j}$$

$$\left.\begin{matrix} f^k_{ij} \\ f^k_{ji} \end{matrix}\right\} \geq 0 \qquad \text{for all } \{i,j\} \in E, \text{ and } k \in K \cup L \tag{17k}$$

$$x_{ij} \geq 0 \qquad \text{and integer, for all } \{i,j\} \in E. \tag{17l}$$

In this formulation, $T = T_1 \cup \ldots \cup T_P$, $R = \{r_1, \ldots, r_P\}$, and $\rho(j)$ denotes node j's root node. $CO(k)$, $CD(k)$, and $\mathcal{H}$ are defined as for the Steiner forest problem for the commodities $k \in K$.

If the requirement spanning tree contains no edge $\{s,t\}$, with $r_{st} = 1$, Formulation (17) contains no commodities in K, and so the model contains only constraints (17c), (17h), (17j), (17k), and (17l). In this case, an argument similar to the one we used in Section 3.2 shows that this formulation is equivalent to the undirected flow formulation. Consequently, the improved formulation for the NDC problem is stronger than the undirected flow formulation only when the maximum spanning tree of the requirement graph contains some edge $\{s,t\}$ with $r_{st} = 1$.

10 Concluding Remarks

In this paper we showed how to improve formulations for NDC problems by generalizing directing procedures for NDC problems that have connectivity requirements r_{st} of 0 or 1. For unitary NDC problems, we generalized the directing procedure for the Steiner tree problem and for nonunitary NDC problems, we generalized a new directing procedure for the Steiner forest problem.

For unitary NDC problems we also showed that the projection of the new formulations

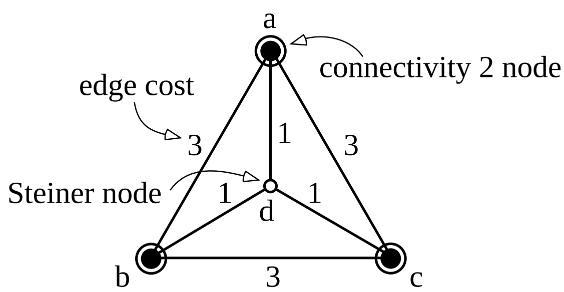


Figure 8: The flow formulation does not always give integer solutions. The LP solution is 7.5 and the optimal solution to the problem has cost 8.

onto the space of the edge design variables contains three classes of valid inequalities (partition, odd-hole, and combinatorial design) that are generalizations of valid inequalities for the Steiner tree problem. For nonunitary NDC problems, we have not fully investigated the projection of the improved models (the projection cones are quite complex!) to determine valid inequalities implied by them in the space of the original edge variables. This is one potential direction of future research.

Although we have shown that the improved flow formulation implies three classes of valid inequalities, some classes of facet defining inequalities cannot be obtained by projecting from the flow formulation. Consider the SND problem shown in Figure 8. In this SND problem, nodes a, b, and c have a connectivity requirement of 2. Figure 8 shows the underlying graph and the cost of the edges. One optimal integer solution to the problem has the edges $x_{ad} = x_{db} = x_{bc} = x_{ca} = 1$ and has cost 8. Assume that in the improved flow model, commodity 1 is from node a to b, commodity 2 is from node a to c, commodity 3 is from node b to a and commodity 4 is from node c to a. The optimal linear programming solution to the improved undirected flow model (5) is $f^{a1}_{ad} = f^{a1}_{db} = f^{c1}_{ad} = f^{c1}_{dc} = f^{a2}_{bd} = f^{a2}_{da} = f^{c2}_{cd} = f^{c2}_{da} = 0.5$, $f^{a1}_{ab} = f^{a1}_{ac} = f^{a1}_{cb} = f^{c1}_{ab} = f^{c1}_{ac} = f^{c1}_{bc} = f^{a2}_{ba} = f^{a2}_{ca} = f^{a2}_{bc} = f^{c2}_{ca} = f^{c2}_{cb} = f^{c2}_{ba} = 0.25$, $x_{ad} = x_{bd} = x_{cd} = 1$ and $x_{ab} = x_{bc} = x_{ca} = 0.5$ with cost 7.5. Grötschel, Monma and Stoer [GMS92b] introduced a class of inequalities for the SND problem, called r-cover inequalities, that eliminates such fractional solutions in the cutset model. It is unlikely that any polynomial size flow formulation implies the r-cover inequalities because, in a certain sense, the r-cover inequalities are complements of the blossom inequalities of the matching polytope. Therefore, if we can project out the r-cover inequalities from a polynomial size flow-based formulation, then we will have a polynomial size extended formulation for the matching polytope. However, Yanakakis [Yan88] showed that the matching polytope has no polynomial size extended formulation unless $\mathcal{P} = \mathcal{NP}$.

In this paper we did not consider node-connectivity requirements. Raghavan [Rag95] describes formulations and algorithms for NDC problems with node-connectivity requirements, as well as NDC problems with both edge- and node-connectivity requirements. We note however that because node connectivity implies edge connectivity, all the valid inequalities derived for the edge-connectivity version of the NDC problem are valid for the node-connectivity version. Consequently, the partition, odd-hole and combinatorial design inequalities are valid for the node-connectivity version of the unitary NDC problem, and in some instances they can be facet defining. For instance, Stoer [Sto92] shows that under certain conditions, the partition inequalities are facet defining for the node-connectivity version of the SND problem.

For the Steiner tree problem, Goemans and Myung [GM93] show that the choice of the root node in the directing procedure does not affect the optimal objective value of linear programming relaxation of the improved undirected flow formulation. Using this result it is easy to show that the choice of the root node does not affect the value of the linear programming relaxation of the improved undirected flow formulation for the unitary NDC problem. A natural question to ask for the Steiner forest problem is whether (i) the choice of root node for each node set, and (ii) the order of node sets, affects the optimal objective value of the linear programming relaxation of the improved flow formulation for the Steiner forest problem. Although we have not proved this result in this paper, it is possible to establish this result. With some further argument, this result implies that for nonunitary NDC problems, the choice of root node for each set, and the order of node sets does not affect the optimal objective value of the linear programming relaxation of the improved flow formulation.

We note that the strong formulation for the Steiner forest problem also leads to a stronger formulation for a more general version of a problem called the Multi-Level Network Design problem that has been studied in the literature (see [BMM94]). In the two-level network design problem, we are given a network with two facility types available for each edge—primary and secondary—and a partition of the nodes into two types—primary nodes and secondary nodes. In this scenario, primary edges are more expensive than secondary edges. We wish to design a minimum cost connected network that connects the primary nodes to each other by primary edges. A more general version of this problem splits the primary nodes into primary sets $T_1, \ldots, T_P$ with $T_i \cap T_j = \phi$ for all primary node set pairs. We wish to design a minimum cost connected network that connects the nodes in each primary set by primary edges. We can interpret this problem as a Steiner forest problem overlayed on a tree (the primary edges in the optimal solution define a forest, and the edges in the optimal solution, both primary and secondary, define a tree). The stronger model for the Steiner forest problem yields a stronger model for this two-level network design problem (and for the multi-level network design problem).

Although we did not use Nash-Williams theorem to obtain stronger formulations for the unitary NDC problem, the Nash-Williams result proves useful when designing dual-ascent algorithms from the improved flow formulation. For example consider the NDLC problem.

In this case because of Nash-Williams result, we can transform and thus formulate the problem on a directed graph (recall this transformation is only valid when $r_{st} \in \{0, 1, \text{even}\}$)). This, in turn, significantly simplifies the dual-ascent algorithm (as compared to performing dual-ascent using the improved undirected flow formulation). In a companion paper [MR99] we consider the NDLC problem and derive a dual-ascent algorithm using a directed flow model on a directed graph. Computational experiments reported in that paper show that the dual ascent algorithm applied to the directed flow model is able to solve problems with up to 300 nodes and 3000 edges to within a few percent of optimality, indicating that the linear programming relaxation of the improved flow formulation provides a good approximation to this mixed integer program model.

References

[Ben62] J. F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[BH93] S. C. Boyd and T. Hao. An integer polytope related to the design of survivable communication networks. *SIAM Journal on Discrete Mathematics*, 6(4):612–630, 1993.

[BM93] M. Baïou and A. R. Mahjoub. The 2-edge connected Steiner subgraph polytope of a series-parallel graph. Technical report, Départment d'informatique, Université de Bretagne Occidentale, 6 Avenue Victor Le Gorgeu, B.P. 452, 29275 Brest Cedex, France, 1993.

[BMM94] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. A dual-based algorithm for multi-level network design. *Management Science*, 40:567–581, 1994.

[BP83] E. Balas and W. R. Pulleyblank. The perfectly matchable subgraph polytope of a bipartite graph. *Networks*, 13:495–516, 1983.

[Cho89] S. Chopra. On the spanning tree polyhedron. *Operations Research Letters*, 8(1):25–29, 1989.

[Cho92] S. Chopra. Polyhedra of the equivalent subgraph problem and some edge connectivity problems. *SIAM Journal on Discrete Mathematics*, 5:321–337, 1992.

[CMW89] R. H. Cardwell, C. L. Monma, and T. H. Wu. Computer-aided design procedures for survivable fiber optic networks. *IEEE Journal on Selected Areas in Communications*, 7:1188–1197, 1989.

[CR94a] S. Chopra and M. R. Rao. The Steiner tree problem I: Formulations, compositions and extensions of facets. *Mathematical Programming*, 64(2):209–229, 1994.

[CR94b] S. Chopra and M. R. Rao. The Steiner tree problem II: Properties and classes of facets. *Mathematical Programming*, 64(2):231–246, 1994.

[Fra94] A. Frank. Connectivity augmentation problems in network design. In John R. Birge and Katta G. Murty, editors, *Mathematical Programming: State of the Art 1994*. The University of Michigan, 1994.

[GH61] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *SIAM Journal of Applied Mathematics*, 9:551–570, 1961.

[GM90] M. Grötschel and C. L. Monma. Integer polyhedra arising from certain network design problems with connectivity constraints. *SIAM Journal on Discrete Mathematics*, 3:502–523, 1990.

[GM93] M. X. Goemans and Y.-S. Myung. A catalog of Steiner tree formulations. *Networks*, 23(1):19–28, January 1993.

[GMS92a] M. Grötschel, C. L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research*, 40:309–330, 1992.

[GMS92b] M. Grötschel, C. L. Monma, and M. Stoer. Facets for polyhedra arising in the design of communication networks with low-connectivity requirements. *SIAM Journal on Optimization*, 2, 1992.

[GMS95a] M. Grötschel, C. L. Monma, and M. Stoer. Design of survivable networks. In M. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 617–672. North-Holland, 1995.

[GMS95b] M. Grötschel, C. L. Monma, and M. Stoer. Polyhedral and computational investigations for designing communication with high survivability requirements. *Operations Research*, 43:1012–1024, 1995.

[Goe90] M. X. Goemans. *Analysis of linear programming relaxations for a class of connectivity problems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA 02139, 1990. Also Working paper OR 233-90, Operations Research Center, M.I.T.

[Goe94] M. X. Goemans. The Steiner tree polytope and related polyhedra. *Mathematical Programming*, 63(2):157–182, January 1994.

[Goe95] M. X. Goemans. Worst-case comparison of valid inequalities for the TSP. *Mathematical Programming*, 69(2):335–349, 1995.

[Jai98] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. Technical report, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, 1998.

[Mah94] A. R. Mahjoub. Two-edge connected spanning subgraphs and polyhedra. *Mathematical Programming*, 64:199–208, 1994.

[Men27] K. Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.

[MR99] T. L. Magnanti and S. Raghavan. A dual-ascent algorithm for low-connectivity network design. Technical report, OR-Center, Massachusetts Institute of Technology, Cambridge, MA 02139, 1999.

[MW95] T. L. Magnanti and L. A. Wolsey. Optimal trees. In M. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 503–615. North-Holland, 1995.

[NW60] C. St. J. A. Nash-Williams. On orientations, connectivity, and odd vertex pairings in finite graphs. *Canadian Journal of Mathematics*, 12:555–567, 1960.

[PLG85] A. Prodon, Th. M. Liebling, and H. Gröflin. Steiner's problem on two-trees. Technical Report RO850315, Départment de Mathématiques, EPF Lausanne, Lausanne, Switzerland, 1985.

[Rag95] S. Raghavan. *Formulations and Algorithms for Network Design Problems with Connectivity Requirements*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA 02139, 1995.

[RM97] S. Raghavan and T. L. Magnanti. Network connectivity. In M. Dell'Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 335–354. John Wiley & Sons, 1997.

[Sto92] M. Stoer. *Design of Survivable Networks*. Number 1531 in Lecture Notes in Mathematics. Springer-Verlag, 1992.

[Whi32] H. Whitney. Non-separable and planar graphs. *Transactions of the American Mathematical Society*, 34:339–362, 1932.

[Won84] R. T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Mathematical Programming*, 28:271–287, 1984.

[Yan88] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. In *Proceedings of the 20th Annual Symposium on the Theory of Computing*, pages 223–228, 1988.