

**The Convex Hull of Two Core Capacitated
Network Design Problems**

by

*Thomas L. Magnanti, Prakash Mirchandani
and Rita Vachani*

OR 217-90

June 1990

**THE CONVEX HULL OF TWO CORE
CAPACITATED NETWORK DESIGN PROBLEMS**

Thomas L. Magnanti
Sloan School of Management, MIT, Cambridge, MA 02139

Prakash Mirchandani
Katz Graduate School of Business, University of Pittsburgh, Pittsburgh, PA 15260

and

Rita Vachani
GTE Laboratories, Waltham, MA 02254

June 1990

Abstract

The network loading problem (NLP) is a specialized capacitated network design problem in which prescribed point-to-point demand between various pairs of nodes of a network must be met by installing (loading) a capacitated facility. We can load any number of units of the facility on each of the arcs at a specified arc dependent cost. The problem is to determine the number of facilities to be loaded on the arcs that will satisfy the given demand at minimum cost.

This paper studies two core subproblems of the NLP. The first problem, motivated by a Lagrangian relaxation approach for solving the problem, considers a multiple commodity, single arc capacitated network design problem. The second problem is a three node network; this specialized network arises in larger networks if we aggregate nodes. In both cases, we develop families of facets and completely characterize the convex hull of feasible solutions to the integer programming formulation of the problems. These results in turn strengthen the formulation of the NLP.

Keywords: Convex hull, Facets, Network design, Capacitated facilities.

Abbreviated title: CORE NETWORK DESIGN PROBLEMS

1. Introduction

In this paper, we study the polyhedral structure of two core subproblems of a capacitated network design problem that we call the *network loading problem* (NLP). The network loading problem models situations in which (i) the variable flow costs are zero, and (ii) facilities of fixed capacity -- in our case a single type of facility -- are available to carry flow. We can install (load) this facility on any of the arcs of a network. The problem is to determine the number of facilities to be loaded on each of the arcs of the network that will meet given point-to-point demand at minimum cost. In this introductory section, we provide a mixed integer programming formulation of the NLP, briefly note its applications, and provide motivation for our research.

Problem P(NLP):

$$\text{minimize } \sum_{\{i,j\} \in A} b_{ij} y_{ij}$$

subject to:

$$\sum_{j \in N} f_{ji}^k - \sum_{j \in N} f_{ij}^k = \begin{cases} -u_k & \text{if } i = O(k) \\ u_k & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } i \in N, \text{ for all } k \in Q \quad (1)$$

$$\sum_{k \in Q} (f_{ij}^k + f_{ji}^k) \leq C y_{ij} \quad \text{for all } \{i,j\} \in A \quad (2)$$

$$y_{ij} \geq 0 \text{ and integer for all } \{i,j\} \in A; f_{ij}^k, f_{ji}^k \geq 0 \text{ for all } \{i,j\} \in A, \text{ for all } k \in Q. \quad (3)$$

In this formulation, N denotes the set of nodes in the network, A the set of arcs, and Q the set of commodities; commodity k has origin $O(k)$, destination $D(k)$ and demand u_k . The variable f_{ij}^k models the flow of commodity k on arc $\{i,j\}$ in the direction i to j and the variable y_{ij} models the integer number of facilities that are loaded on arc $\{i,j\}$. The parameters C and b_{ij} represent the capacity of the facility and the cost of loading a single facility on arc $\{i,j\}$, and the objective function measures the total cost incurred in loading all the facilities. Constraints (1) correspond to the usual *flow conservation* constraints for each of the commodities at each node. The *capacity* constraints (2) model the requirement that the total flow (in both directions) on an arc cannot exceed the capacity loaded on that arc. We assume that the demands of all the commodities and the capacity C are integers.

For a more detailed discussion of a generalized version of this model, we refer the reader to Magnanti, Mirchandani and Vachani (1990).

This network loading problem is a special version of the general capacitated network design problem and arises in several application contexts. In the telecommunications industry, it represents the design of private networks that use digital transmission facilities (called T1 circuits) to carry voice and data traffic between locations. In the transportation industry, the facilities might represent trucks of fixed size and a slight variation of the model would prescribe a *load plan* (the assignment of trucks to routes and the loading of freight onto the trucks; see Powell and Sheffi, 1983). A generalization of the NLP, which has nonzero flow costs in the objective function, models many other network design problems, such as capacitated plant location (see Leung and Magnanti, 1989).

The NLP is strongly NP-hard (Magnanti, Mirchandani and Vachani, 1990) and, therefore, there is little hope of developing a theoretically efficient algorithm for solving it. Nonetheless, computational evidence in the literature suggests the possibility of developing pragmatically efficient algorithms for the problem if we can develop a better understanding of the problem's structure and the structure of its subproblems. For example, Crowder, Padberg and Johnson (1983) showed that minimal cover cuts from single constraints of a zero-one integer program are extremely effective in solving the (larger) original problem. This observation is the primary motivation for this research. Consequently, we study the polyhedral structure of two core subproblems of the NLP. The first subproblem arises when we adopt a Lagrangian relaxation approach to solve NLP; as we will see, the structural properties of the Lagrangian subproblem prove valuable in understanding the structure of NLP. The second subproblem arises when we restrict the number of nodes in the network; understanding the structure of this smaller problem, as well as the Lagrangian subproblem, permits us to strengthen the integer programming formulation of the general NLP. For each subproblem, we provide a complete characterization of the convex hull of its feasible solutions and comment briefly on how these results strengthen the formulation of NLP.

We might note that despite extensive study, researchers have been able to describe the convex hull of feasible solutions to core, polynomially solvable integer programming problems (other than problems like network flow that are inherently integer) for only a few problem classes: for example, 1-matching and matroids (Edmonds, 1965, 1970, 1971), perfectly matchable subgraphs of bipartite and general graphs (Balas and Pulleyblank, 1983, 1987), and uncapacitated production lot sizing (Barany, Van Roy and Wolsey,

1984). Although our problems are more specialized than those cited above, the results in this paper are applicable to a wide variety of fixed charge problems. Furthermore, for most of the polyhedra that researchers have been able to completely characterize in previous studies, the integer variables are restricted to be binary, whereas, for the problems that we study in this paper, the integer variables can assume any nonnegative value.

This paper is organized as follows. In Section 2, we describe the two subproblems. In Section 3, we discuss the polyhedral properties of the first subproblem. We show that the convex hull of feasible solutions to this problem has full dimension, introduce a family of facets for the problem, and show that with the addition of these facets, the linear programming relaxation of the problem describes the convex hull of its solutions. Our proof constructs a primal feasible solution and demonstrates the existence of a dual feasible solution that satisfies complementary slackness. We depart from the conventional “primal-dual” convex hull proofs that construct specific values of the primal and dual solutions. One reason for adopting this approach is that our primal problem is degenerate and so its linear programming optimal solution corresponds to several linear programming bases, each of which defines a potential dual optimal solution. Section 4 studies the second subproblem; in this case as well, we show how to strengthen the formulation and develop a complete and nonredundant characterization of the convex hull of integer solutions. Section 5 summarizes our results.

2. Description and Formulation of Subproblems

Single Arc Design Problem

When using a Lagrangian relaxation approach to solve the NLP, we can dualize either constraints (1) or (2). If we choose to relax constraints (2), the resulting Lagrangian problem is a network flow problem; thus, this problem satisfies the integrality property (i.e., it has integer extreme points) and, hence, the Lagrangian lower bound equals the bound provided by the linear programming relaxation of NLP (Geoffrion, 1974). It is easy to see that, in general, the linear programming relaxation provides a weak lower bound for the problem. (See Magnanti, Mirchandani and Vachani (1990) for confirming empirical evidence.) On the other hand, if we relax constraints (1) using multipliers v_1^k , then the resulting Lagrangian problem, with $v_{O(k)}^k = 0$, is

Problem P(LAG):

$$\text{minimize } \sum_{\{i,j\} \in A} \left\{ b_{ij} y_{ij} + \sum_{k \in Q} (f_{ji}^k - f_{ij}^k) (v_i^k - v_j^k) \right\} + \sum_{k \in Q} v_{D(k)}^k u_k$$

subject to:

(2) and (3).

To this problem, we add the following upper bound constraints which are redundant in the original formulation but improve the Lagrangian lower bound.

$$\left. \begin{array}{l} f_{ij}^k \leq u_k \\ f_{ji}^k \leq u_k \end{array} \right\} \text{ for all } \{i,j\} \in A, \text{ for all } k \in Q. \quad (4)$$

This problem does not satisfy the integrality property and therefore, we might expect the lower bound obtained from this Lagrangian dual to be stronger than that obtained from the linear programming relaxation. Vachani (1988) uses this Lagrangian relaxation strategy to solve a more general version of the NLP in which the network designer has a choice of two types of facilities. Her results show that the lower bounds from this relaxation improve upon the linear programming relaxation bounds considerably. This empirical evidence suggested that knowledge about the polyhedral structure of problem P(LAG) might permit us to strengthen the linear programming relaxation of the NLP.

The Lagrangian problem, P(LAG), separates by arc for a given set of multipliers. Furthermore, in an optimal solution to the Lagrangian problem either $f_{ij}^k > 0$ or $f_{ji}^k > 0$. Defining $g_{ij}^k = \max(f_{ij}^k, f_{ji}^k)$ and $a_{ij}^k = \min(v_i^k - v_j^k, v_j^k - v_i^k)$ and, for convenience, dropping the arc subscripts and denoting commodity k by a subscript of the appropriate variable, we obtain the *single arc design problem* (SADP).

Problem P(SADP):

$$\text{minimize } by + \sum_{k \in Q} a_k g_k$$

subject to:

$$g_k \leq u_k \quad \text{for all } k \in Q \quad (5)$$

$$\sum_{k \in Q} g_k \leq Cy \quad (6)$$

$$g_k \geq 0 \text{ for all } k \in Q, y \geq 0 \text{ and integer.}$$

Although our study of this single arc design problem has been motivated by the fact that it arises as a Lagrangian subproblem of the NLP, the problem is of greater interest because it is a Lagrangian subproblem for the general capacitated network design problem with nonzero flow costs as well (see, for example, Leung, Magnanti, and Singhal, 1990).

Vachani (1988) shows how to solve the single arc design problem efficiently using a greedy procedure. In this paper, we show how to enhance the formulation to describe the convex hull of feasible solutions to the problem.

To conclude this subsection, we might comment on a related problem: a single node flow problem with an exogenous supply of u units and several outflow arcs incident to the node. Each arc has capacity C_j , carries flow f_j , and has a variable y_j associated with it which equals 1 if the arc is available to carry flow and is 0 otherwise. Padberg, Van Roy and Wolsey (1983) have studied the polyhedral structure of this single node flow problem and identified facets for this model and its generalizations. However, unlike the single arc design problem, the single node flow problem is NP-hard; a transformation from the knapsack problem establishes this result. Thus, these two core problems, which both arise in the context of capacitated network design, differ significantly in their computational complexity. This result has implications concerning the underlying polyhedral structure of the problem. In particular, unless $P = NP$, it is unlikely that the research community would be able to completely characterize the convex hull of the single node flow problem.

Three Node Network Problem

The second problem, $P(3N)$, we study in this paper is a three node version of the NLP. The NLP is defined over an undirected network, i.e., the total *bidirectional* flow on any arc determines its installed capacity. Suppose we obtain instance P2 of the NLP from

instance P1 by interchanging the origin and destination of some commodity, say q^* . Further, suppose we are given a feasible solution to instance P1. We can obtain a feasible solution to instance P2 by reversing the flow of commodity q^* on all the arcs and by keeping the remaining variable at the same values as in the feasible solution to instance P1. In particular, the optimal objective function values of both instances P1 and P2 are equal. Thus, we can view the three node network loading problem, with demand between every pair of nodes, as a problem in which all commodities originate at either of two nodes. Papernov (1976) has shown that a capacitated multicommodity flow problem, whose commodities all have either of two nodes as their origin node, has a feasible solution if and only if the capacity of every cutset is at least as large as the demand across the cutset. We can use this Papernov's result to eliminate the flow variables and obtain a formulation of the three node problem in the subspace of y variables only.

Let us denote the nodes of the network as nodes 1, 2 and 3, and let u_{12} , u_{13} and u_{23} denote the demands between nodes 1 and 2, 1 and 3, and 2 and 3, respectively. The formulation of the three node version of the NLP, projected into the subspace of y variables, is

Problem P(3N):

$$\text{minimize } b_{12}y_{12} + b_{13}y_{13} + b_{23}y_{23}$$

subject to:

$$\begin{aligned} y_{12} + y_{13} &\geq \frac{u_{12} + u_{13}}{C} \\ y_{12} + y_{23} &\geq \frac{u_{12} + u_{23}}{C} \\ y_{13} + y_{23} &\geq \frac{u_{13} + u_{23}}{C} \\ y_{ij} &\geq 0 \text{ and integer for all } \{i,j\}. \end{aligned}$$

In general, the linear programming relaxation of this projected formulation does not have integer extreme points. In fact, since the lefthand side of each inequality is an integer for any feasible solution, we can strengthen these inequalities to obtain the *cutset* inequalities:

$$\begin{aligned} y_{12} + y_{13} &\geq \left\lceil \frac{u_{12} + u_{13}}{C} \right\rceil \\ y_{12} + y_{23} &\geq \left\lceil \frac{u_{12} + u_{23}}{C} \right\rceil \\ y_{13} + y_{23} &\geq \left\lceil \frac{u_{13} + u_{23}}{C} \right\rceil. \end{aligned}$$

Although the cutset inequalities define facets of the NLP (see Magnanti, Mirchandani and Vachani, 1990), they are not sufficient, in general, for generating integer solutions of the y variables. For example, if $C = 10$, $u_{12} = u_{13} = u_{23} = 5$, $b_{12} = 2$, $b_{13} = 3$ and $b_{23} = 4$, then $y_{12} = y_{13} = y_{23} = 1/2$ is an optimal solution to the linear programming relaxation of the problem with the cutset inequalities included. This solution costs 4.5; the optimal solution to the network loading problem is a minimum spanning tree that costs 5 units. Notice that the arcs corresponding to the half-integral y 's in the linear programming solution form a cycle. This phenomenon occurs in larger networks for the same reason: a “half-cycle” satisfies the cutset constraints but might be cheaper than any integral solution. The facet inequalities presented in Section 4 are useful for cutting off these half-integral solutions. In particular, for the three node network, they, along with the cutset inequalities, are sufficient for describing the convex hull of feasible solutions.

3. The Single Arc Design Problem.

In this section, we study the single arc design problem and show how results about its structure are useful in strengthening the formulation of the NLP. The linear programming relaxation of P(SADP) does not satisfy the integrality restriction on y . Our objective is to identify valid inequalities for P(SADP) that strengthen its linear programming relaxation. In fact, we will give a complete description of the convex hull of the feasible integer solutions to the problem, that is, if $\text{Conv}(P)$ denotes the convex hull of the feasible solutions to Problem (P), we identify linear inequalities that completely describe $\text{Conv}(\text{SADP})$. For Problem (P), let $\dim(P)$ denote the dimension of $\text{Conv}(P)$.

We first state some results about $\text{Conv}(\text{SADP})$ and about the constraints of formulation P(SADP). Let $q = |Q|$, the number of commodities.

Proposition 3.1. *Conv(SADP) is a full dimensional polyhedron. That is, $\dim(\text{Conv}(\text{SADP}))$ equals $q + 1$.*

Proof. See Appendix A.

Proposition 3.2.

1. For every $k \in Q$, $g_k \geq 0$ defines a facet of $\text{Conv}(\text{SADP})$.
2. For every $k \in Q$, $g_k \leq u_k$ defines a facet of $\text{Conv}(\text{SADP})$.
3. If $\sum_{k \in Q} u_k > C$, then $\sum_{k \in Q} g_k \leq Cy$ defines a facet of $\text{Conv}(\text{SADP})$.

Proof. See Appendix A.

New Inequalities

We next identify a set of additional inequalities (facets) of $\text{Conv}(\text{SADP})$ that are necessary for completely describing it. We first note that if $u_k \leq C$ for any k , then we can tighten the upper bound constraint $g_k \leq u_k$ for small values of y (i.e., when $0 \leq y \leq 1$) by replacing it with the stronger *forcing* inequality $g_k \leq u_k y$ which, in fact, defines a facet of $\text{Conv}(\text{SADP})$. This idea of replacing a variable upper bound constraint by a forcing constraint is an old idea in the mathematical programming literature – for example, this replacement arises in the context of the uncapacitated facility location problem (see Cornuejols, Sridharan and Thizy, 1990). This observation implies that if $\sum_{k \in Q} u_k \leq C$, then $\sum_{k \in Q} g_k \leq Cy$ is no longer a facet of $\text{Conv}(\text{SADP})$ since the constraint $\sum_{k \in Q} g_k \leq (\sum_{k \in Q} u_k)y$, which is an aggregation of $g_k \leq u_k y$ for all $k \in Q$, dominates it. On the other hand, if $u_k > C$ for any k , then $g_k \leq u_k y$ is not a facet of $\text{Conv}(\text{SADP})$. In the following discussion, we develop a class of inequalities that generalize the forcing constraint $g_k \leq u_k y$ in two ways: (i) by permitting an arbitrary value of u_k , and (ii) by considering aggregations defined over an arbitrary subset $P \subseteq Q$ of variables rather than just a single variable g_k . To avoid some technicalities in the following discussion, we assume that $\sum_{k \in Q} u_k > C$.

Consider the inequality

$$g_P - r_P y \leq (\mu_P - 1)(C - r_P). \quad (7)$$

In this expression, for any $P \subseteq Q$, g_P denotes $\sum_{k \in P} g_k$ and u_P denotes $\sum_{k \in P} u_k$; $r_P = u_P \pmod{C}$ and $\mu_P = \lceil u_P / C \rceil$. By convention, we set r_P equal to C if u_P is a multiple of C . We refer to this inequality as the *arc residual capacity inequality* because $(C - r_P)$ is the

residual capacity left if we (i) set $y = \mu_P$, (ii) set the flow values for all the commodities in P at their upper bounds, and (iii) set the flow values for all commodities in $Q \setminus P$ at value zero.

We now show that the arc residual capacity inequalities are valid for $P(\text{SADP})$ and also show how they tighten the linear programming relaxation of the problem.

Validity

Suppose we write the arc residual capacity inequality as

$$g_P \leq r_P[y - (\mu_P - 1)] + (\mu_P - 1)C \equiv \text{RHS}(y).$$

As shown in Table I, the inequality is valid if $y = \mu_P - 1$ or $y = \mu_P$, it is dominated by the aggregate upper bound inequality $g_P \leq u_P$ if $y > \mu_P$, and it is dominated by the capacity inequality $g_Q \leq Cy$ if $y < \mu_P - 1$.

Table I. Bounds on the flow variable g_P from the arc residual capacity inequality

Value of y	RHS(y)
$< \mu_P - 1$	$> Cy$
$\mu_P - 1$	$(\mu_P - 1) C$
μ_P	$(\mu_P - 1) C + r_P = u_P$
$> \mu_P$	$> u_P$

Note that if $g_P = u_P$, then the arc residual capacity inequality forces y to be at least μ_P , and if $g_P = u_P - r_P$, then the inequality forces y to be at least $(\mu_P - 1)$. Thus, the inequality is tighter than the corresponding capacity constraint $g_Q \leq Cy$ whenever $g_{Q \setminus P} = 0$, $r_P \neq C$ and $u_P - r_P < g_P \leq u_P$. As the next result shows, this strengthening is sufficient for the inequality to be a facet $\text{Conv}(\text{SADP})$.

Proposition 3.3. *For a given nonempty set $P \subseteq Q$, the arc residual capacity inequality (7) defines a facet of $\text{Conv}(\text{SADP})$ if and only if the set P satisfies the following conditions.*

1. If $\mu_P = 1$, then $|P| = 1$.

2. If $r_P = C$, then $P = Q$.

Proof.

Necessity of the Conditions.

Suppose Condition 1 is not valid, i.e., $\mu_P = 1$ and $|P| \geq 2$. Let $P = \{1, 2, \dots, p\}$. Then the arc residual capacity inequality (7) is a linear combination of facet inequalities $g_k \leq u_k y$ for $1 \leq k \leq p$. Hence, (7) is not facet defining. On the other hand, if Condition 2 is not valid, then (7) is dominated by the capacity constraint $g_Q \leq C y$ and hence is not facet defining.

Sufficiency of the Conditions.

In light of Proposition 3.2 (part 3) and our remarks following it, we can assume that $\mu_P \geq 2$ and $r < C$. Now, consider the equation

$$g_P - r_P y = (\mu_P - 1)(C - r_P) \quad (7')$$

and let $L = \text{Conv}(\text{SADP}) \cap \{(y, g) : y, g \text{ satisfies } (7')\}$. We will show that inequality (7) is a facet of $\text{Conv}(\text{SADP})$ by showing that whenever any inequality

$$\lambda y + \beta g \leq \beta_0 \quad (8)$$

is valid for $\text{Conv}(\text{SADP})$ and satisfies the condition that $\lambda y + \beta g = \beta_0$ for all points in L , then the equation $\lambda y + \beta g = \beta_0$ is a scalar multiple of (7').

Without loss of generality, let $P = \{1, 2, \dots, p\}$ and let

$s = \text{argmax}_{k < p} \{ k : \sum_{j=1}^k u_j \leq C(\mu_P - 1) \}$ and $S = \{1, 2, \dots, s\}$. We adopt the convention that $s = 0$ and $S = \emptyset$ if $u_1 > C(\mu_P - 1)$.

Consider the following feasible point which belongs to L :

$$y = \mu_P; g_k = u_k \text{ for } k \leq p; g_k = 0 \text{ for all } k \geq p+1.$$

Since r_p is less than C , we can modify this solution to obtain another solution that belongs to L by increasing the flow g_k to 1 from 0 for some $k \geq p+1$. Substituting these two solutions in the equality $\lambda y + \beta g = \beta_0$ and subtracting, we obtain $\beta_k = 0$ for all $k \geq p+1$.

Now define the solution

$$y = \mu_P - 1; g_k = u_k \text{ for } k \leq s; g_{s+1} = C(\mu_P - 1) - \sum_{k=1}^s u_k < u_{s+1}; g_k = 0 \text{ for all } k > s+1.$$

Construct a new solution by decreasing the flow of some commodity k_1 belonging to S by 1 (or by decreasing the flow of commodity 1 if $S = \emptyset$), and increasing the flow of some commodity k_2 belonging to PS by the same quantity. Since this solution also belongs to L , by substituting both solutions in the equality $\lambda y + \beta g = \beta_0$ and subtracting, we obtain $\beta_{k_1} = \beta_{k_2}$. Using a similar argument on the other commodities belonging to P , we obtain $\beta_k = \alpha$ for all $k \in P$. It is now easy to see that $\lambda = -r_P \alpha$, and we are done. \otimes

Suppose we now add all of the arc residual capacity inequalities to the formulation $P(\text{SADP})$. The polyhedron defined by the linear programming relaxation of this problem is fairly complex: (i) as shown by Proposition 3.3, it has an exponential number of facets, and (ii) as shown by the following example, the polyhedron can be highly degenerate; that is, an excessive number of facets might intersect at a vertex of $\text{Conv}(\text{SADP})$ and, therefore, also at a vertex of the polyhedron defined by the linear programming relaxation.

Example 3.4. Let $q = 3$ with $u_1 = 8$, $u_2 = 7$ and $u_3 = 4$, and $C = 10$. Let $g_k = u_k$ for all k and $y = 2$. This solution satisfies the upper bound constraint $g_k \leq u_k$ as an equality for all k , as well as the arc residual capacity inequalities (7) as an equality for all two and three element subsets P of Q . Thus the primal solution to the linear programming relaxation can be highly degenerate. \otimes

We now establish the main result of this section.

Theorem 3.5. *The upper bound inequalities (5), the capacity inequality (6), the arc residual capacity inequalities (7), and the nonnegativity constraints describe the convex hull of the set of feasible solutions to $P(\text{SADP})$.*

Before proving this result, we provide a geometric interpretation.

Underlying Geometry

Figures 1 and 2 illustrate the geometry for a two-commodity version of the single arc design problem. Because the problem is a mixed integer program, the feasible region contains slices of continuous regions, one for each choice of the integer variable y . For any specific value of y , the feasible region $g_1 + g_2 \leq Cy$ over the nonnegative variables g_1 and g_2 is a triangle (it is a simplex in higher dimensions); as y becomes larger, so do the triangles. The upper bound constraints $g_1 \leq u_1$ and $g_2 \leq u_2$ will truncate some of the triangles, depending upon the value of y . As indicated in Figure 1, the overall feasible region is composed of these (truncated) triangles, one for each integral value of y . At $y = 0$, the triangle is a single point. For the example illustrated in the figure, $u_1 < C$, $C < u_2 < 2C$ and $2C < u_{\{1,2\}} < 3C$. Since $u_1 < C$, the upper bound $g_1 \leq u_1$ truncates the triangles in the g_1 direction for all $y \geq 1$. Since $C < u_2 < 2C$, the upper bound $g_2 \leq u_2$ truncates the triangles for all values of $y \geq 2$. Note that at $y = 3$, the box $0 \leq g_1 \leq u_1$ and $0 \leq g_2 \leq u_2$ lies strictly within the triangle $g_1 + g_2 \leq 3C$ so the feasible region is a box. (The feasible region is a box for larger values of y as well; the figures do not show this portion of the polyhedron.)

Figure 2 shows the polyhedron defined by the convex hull of the truncated triangles shown in Figure 1. This polyhedron has eight faces, four defined by the nonnegativity and upper bound constraints imposed upon the flow variables g , one defined by the capacity constraint $g_1 + g_2 \leq Cy$, and three defined by the arc residual capacity inequalities (one for each nonempty subset of $\{1, 2\}$). The three arc residual capacity inequalities have the following structure.

(i) Since $u_1 < C$, the arc residual capacity constraint for variable g_1 becomes $g_1 - r_1 y \leq 0$; the facet defined by this constraint is hidden in the figure (its extreme points are the origin and two points with $y = 1$ and $g_1 = r_1$; at one of these points $g_2 = 0$, and at the other $g_2 = C - r_1$).

(ii) Since u_2 lies between C and $2C$, the arc residual capacity constraint for the flow variable g_2 is $g_2 - r_2 y \leq C - r_2$. The facet defined by this inequality contains one extreme point with $y = 1$ ($g_1 = 0$ and $g_2 = C$) and two extreme points with $y = 2$ ($g_2 = C + r_2$, and $g_1 = 0$ or $g_1 = 2C - (C + r_2) = C - r_2$).

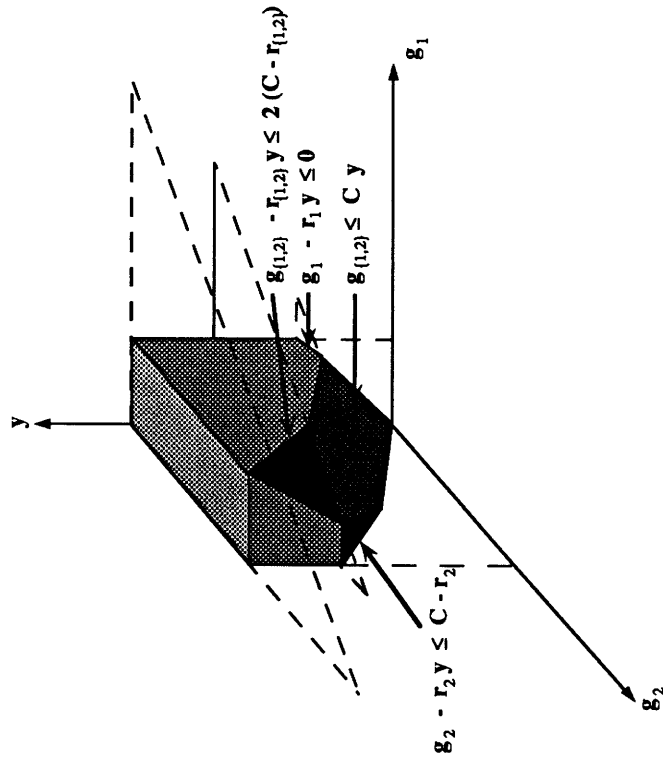


Figure 2. Single Arc Design Polyhedron

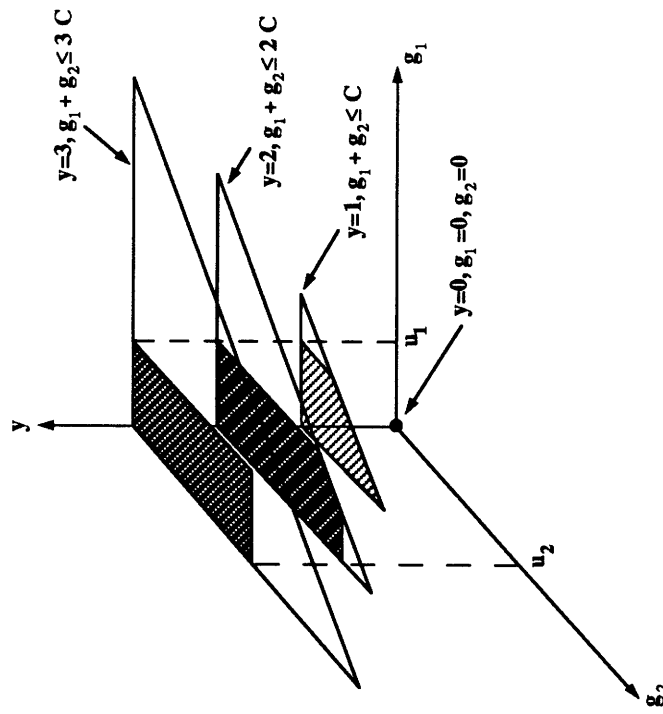


Figure 1. Single Arc Design Problem

(iii) Since $u_{\{1,2\}}$ lies between $2C$ and $3C$, $\mu_{\{1,2\}} = 3$ and the arc residual capacity constraint corresponding to the variable set $\{1,2\}$ is $g_{\{1,2\}} - r_{\{1,2\}}y \leq 2(C - r_{\{1,2\}})$. The facet defined by this inequality has one extreme point with $y = 3$ ($g_1 = u_1$ and $g_2 = u_2$) and two extreme points with $y = 2$ ($g_1 = u_1$, $g_2 = 2C - u_1$ and $g_2 = u_2$ and $g_1 = 2C - u_2$). These extreme points correspond to the "outermost" corner of the box shown at level $y = 3$ in Figure 1 and the two corner points where the constraint $g_1 + g_2 \leq 2C$ truncates the box at level $y = 2$.

Proof of Theorem 3.5. Let $P(\text{FACET})$ denote the problem defined by appending the arc residual capacity inequalities constraints (7) to the linear programming relaxation of $P(\text{SADP})$. Consider the dual $P(D)$ to this problem.

Problem $P(D)$:

$$\text{maximize } \sum_{k \in Q} u_k \lambda_k + \sum_{P: P \subseteq Q} \delta_P (\mu_P - 1)(C - r_P)$$

subject to:

$$\lambda_k + \alpha + \sum_{P: k \in P} \delta_P \leq a_k \text{ for all } k \in Q$$

$$-C\alpha - \sum_{P: P \subseteq Q} r_P \delta_P \leq b$$

$$\alpha \leq 0, \lambda_k \leq 0 \text{ for all } k \in Q, \delta_P \leq 0 \text{ for all } P \subseteq Q.$$

For any given choice of cost data, we will define an integer feasible solution to $P(\text{FACET})$. We will then show that some corresponding (but unspecified) dual solution satisfies complementary slackness. This result will prove the theorem. The proof is complicated by the possibility of degeneracy as illustrated by Example 3.4, which implies that the dual might have many potentially optimal solutions since many linear programming bases might correspond to the same degenerate primal optimal solution.

Before continuing with the proof, we make two observations.

Note 1. We can assume without loss of generality that $a_k < 0$ for all $k \in Q$. Suppose not. Let $M = \{k : a_k \geq 0, k \in Q\}$, $M \neq \emptyset$. Suppose we solve $P(\text{FACET})$ and $P(D)$ with Q'

= QM and let (y^*, g^*) and $(\lambda^*, \alpha^*, \delta^*)$ be the optimal solutions to these two problems (with Q' as the commodity set). It is easy to see that $(y^*; g_k = g_k^*$ for $k \in Q'$; $g_k = 0$ for $k \in M)$ and $(\lambda_k = \lambda_k^*$ for $k \in Q'$; $\lambda_k = 0$ for $k \in M$; $\alpha^*, \delta_S = \delta_S^*$ for $S \subseteq Q'$; $\delta_S = 0$ otherwise) are optimal for the original problems. Thus if (y^*, g^*) is integer for all values of b and all negative values of the coefficients a_k , then so is the solution to the problem when some of the values of a_k are nonnegative. ♦

Note 2. We will restrict ourselves to situations with $b > 0$. If $b < 0$, then the primal problem is unbounded and if $b = 0$, then the proof is trivial. ♦

Index the commodities so that $a_1 \leq a_2 \leq \dots \leq a_q < 0$. Let

$$M = \{k \in Q : a_k < -b/C\} = \{1, 2, \dots, m\};$$

$$QM = \{k \in Q : -b/C \leq a_k < 0\} = \{m+1, m+2, \dots, q\};$$

$$s = \operatorname{argmax}_{k < m} \left\{ k : \sum_{j=1}^k u_j \leq C (\mu_M - 1) \right\} \text{ and } S = \{1, 2, \dots, s\}. \text{ Note that if } y = \mu_M - 1, \text{ then it is possible to set all the variables } g_k \text{ for } k \in S \text{ to their upper bounds } u_k, \text{ but not all the variables } g_1, g_2, \dots, g_{s+1};$$

$$t = \operatorname{argmax}_{m \leq k \leq q} \left\{ k : \sum_{j=1}^k u_j \leq C \mu_M \right\}, T = \{1, 2, \dots, t\}.$$

Note that by definition of s and t , $s+1 \leq m \leq t$ and $\mu_{S'} = \mu_M$ for all $S' = \{1, 2, \dots, s'\}$ with $s+1 \leq s' \leq t$.

We will consider the following two mutually exclusive and exhaustive cases.

Case 1. $b > 0$, $a_k \geq -b/C$ for all k , i.e., $M = \emptyset$.

Notice that the condition on the flow costs implies that all the flows in the optimal solution to the linear programming relaxation of P(SADP) will be zero. Thus, the optimal flow values will be zero for P(SADP) as well. Define the following primal and dual solutions.

Primal solution. $y = 0$; $g_k = 0$ for all $k \in Q$.

Dual solution. $\lambda_k = 0$ for all $k \in Q$; $\alpha = -b/C$; $\delta_P = 0$ for all $P \subseteq Q$.

This dual solution is feasible since, by assumption, $-b/C \leq a_k$ for all $k \in Q$. Furthermore, it is easy to see that these primal and dual solutions satisfy the complementary slackness conditions.

Case 2. $b > 0$, and $M \neq \emptyset$, i.e., $a_1 < -b/C$.

For this case, for notational convenience we define the following terms:

- (i) $a_{q+1} = 0$;
- (ii) $R_k = r_{\{1, 2, \dots, k\}}$.

If $M \neq \emptyset$, then by definition of M , the optimal solution to the linear programming relaxation of P(SADP) is given by $g_k = u_k$ for all commodities $k \in M$, $g_k = 0$ for all commodities $k \in Q \setminus M$, and $y = \sum_{k \in M} u_k / C$. Based on this observation, it is easy to check

that if $M \neq \emptyset$, then the optimal primal (and so integer) solution to P(SADP) will load either μ_M (which equals $\lceil \sum_{k \in M} u_k / C \rceil$) or $(\mu_M - 1)$ facilities on the arc, i.e., $y = \mu_M$ or $y = (\mu_M - 1)$ in an optimal solution. The variable y will equal $(\mu_M - 1)$ if b , the cost of the facility, exceeds the “maximum contribution” from flowing the commodities in QS on this facility. That is, in the integer programming solution, we can always set $g_k = u_k$ for all $k \in S$. If $y = (\mu_M - 1)$ we send a flow of value $(C - R_s) < u_{s+1}$ of commodity $s+1$, since $\sum_{k=1}^{s+1} u_k >$

$(\mu_M - 1)C$. If we set $y = \mu_M$, however, then we send a flow of the remaining R_{s+1} units of commodity $s+1$, send a flow of value equal to the respective demands of commodities $s+2, s+3, \dots, t$, and a flow of value $(C - R_t)$ of commodity $t+1$. Consequently, if

$$b > -R_{s+1}a_{s+1} - \sum_{k=s+2}^t u_k a_k - (C - R_t)a_{t+1} \text{ for } t < q,$$

or

$$b > -R_{s+1}a_{s+1} - \sum_{k=s+2}^t u_k a_k \text{ for } t = q$$

then $y = (\mu_M - 1)$ and the optimal flows are $g_k = u_k$ for all commodities $k \in S$, $g_{s+1} = C(\mu_M - 1) - \sum_{k=1}^s u_k$ and $g_k = 0$ for all $k > s+1$.

If the cost b is less than the additional flow contribution, then y will equal μ_M in the optimal primal solution. We consider the two cases corresponding to these two primal solutions separately.

Subcase 2(a).

$$(i) \quad b + R_{s+1}a_{s+1} + \sum_{k=s+2}^t u_k a_k + (C - R_t)a_{t+1} > 0 \text{ if } t < q,$$

or

$$(ii) \quad b + R_{s+1}a_{s+1} + \sum_{k=s+2}^t u_k a_k > 0 \quad \text{if } t = q.$$

Subcase 2 (b).

$$(i) \quad b + R_{s+1}a_{s+1} + \sum_{k=s+2}^t u_k a_k + (C - R_t)a_{t+1} \leq 0 \text{ if } t < q,$$

or

$$(ii) \quad b + R_{s+1}a_{s+1} + \sum_{k=s+2}^t u_k a_k \leq 0 \quad \text{if } t = q.$$

As Example 3.4 shows, a given primal solution might be highly degenerate, satisfying the arc residual capacity inequality as an equality for many subsets P of Q . We will deal with this degeneracy by parameterizing the dual variables α or Δ and showing that *some* dual feasible solution satisfies the complementary slackness conditions. (The dual problem may have multiple dual optimal solutions and we will not specify any particular one for this part of the proof.)

Subcase 2(a).

Primal solution. $y = \mu_M - 1$; $g_k = u_k$ for $k \leq s$; $g_{s+1} = C(\mu_M - 1) - \sum_{k=1}^s u_k < u_{s+1}$; $g_k = 0$ for all $k > s+1$.

Dual solution. Recall that, by definition of S and T , $\mu_{S'} = \mu_M$ for all $S' = \{1, 2, \dots, s'\}$ with $s+1 \leq s' \leq t$. Therefore, with the above definition of the primal variables, the arc residual capacity constraints

$$g_P - r_P y \leq (\mu_P - 1)(C - r_P)$$

are tight for all $P = \{1, 2, \dots, p\}$ with $s+1 \leq p \leq t$. We set $\delta_W = 0$ for all sets $W \subseteq Q$ not of this form. We also define $\Delta_k = \delta_{\{1, 2, \dots, k\}}$.

We now construct an optimal dual solution by considering the dual constraints that must be satisfied as an equality to satisfy complementary slackness. Since $y > 0$, we require

$$-R_{s+1}\Delta_{s+1} - R_{s+2}\Delta_{s+2} - \dots - R_t\Delta_t - C\alpha = b.$$

Since $g_k < u_k$ for all $k \geq s+1$, we set $\lambda_k = 0$ for these commodities. Now, consider the dual feasibility constraints for the variables $g_{s+1}, \dots, g_t, g_{t+1}$:

$$\begin{aligned} \Delta_{s+1} + \Delta_{s+2} + \dots + \Delta_t + \alpha &= a_{s+1} \\ \Delta_{s+2} + \dots + \Delta_t + \alpha &\leq a_{s+2} \\ &\vdots \\ \Delta_t + \alpha &\leq a_t \\ \alpha &\leq a_{t+1}. \end{aligned}$$

We have written the first of these constraints as an equality so that it satisfies complementary slackness (since $g_{s+1} > 0$). Notice that these constraints and the nonpositivity of the variables Δ_k imply that $a_{s+1} \leq \alpha \leq a_{t+1}$. For α in this range define

$$\Delta_k(\alpha) = \min \{0, a_k - \alpha - \Delta_{k+1}(\alpha) - \dots - \Delta_t(\alpha)\}$$

$$\text{and } v(\alpha) = -R_{s+1}\Delta_{s+1}(\alpha) - R_{s+2}\Delta_{s+2}(\alpha) - \dots - R_t\Delta_t(\alpha) - C\alpha.$$

We note the following properties of the dual variables and $v(\alpha)$. The first property assigns specific values to the variables $\Delta_k(\alpha)$ given a particular value of α . The second and third properties place lower and upper bounds on the function $v(\alpha)$.

Property 1. *Sequential difference property.*

Let $k(\alpha) = \max \{k : \alpha \geq a_k\}$ Then by definition of the $\Delta_k(\alpha)$ variables,

$$\Delta_k(\alpha) = \begin{cases} 0 & \text{for } k \geq k(\alpha) + 1 \\ a_k - \alpha & \text{for } k = k(\alpha) \\ a_k - a_{k+1} & \text{for } s+1 \leq k < k(\alpha). \end{cases}$$

In particular, $\Delta_{s+1}(\alpha) + \Delta_{s+2}(\alpha) + \dots + \Delta_t(\alpha) + \alpha = a_{s+1}$. ♦

Property 2. *Lower bounding property.*

$\Delta_k(a_{s+1}) = 0$ for all $k = s+1, \dots, t$. Thus $v(a_{s+1}) = -Ca_{s+1} > b$. ♦

Property 3. *Upper bounding property.*

$\Delta_k(a_{t+1}) = a_k - a_{k+1}$ for all $k = s+1, \dots, t$. Thus

$$\begin{aligned} v(a_{t+1}) &= -R_{s+1}(a_{s+1} - a_{s+2}) - \dots - R_t(a_t - a_{t+1}) - Ca_{t+1} \\ &= -R_{s+1}a_{s+1} - (R_{s+2} - R_{s+1})a_{s+2} - \dots - (R_t - R_{t-1})a_t - (C - R_t)a_{t+1} \\ &= -R_{s+1}a_{s+1} - u_{s+2}a_{s+2} - \dots - u_t a_t - (C - R_t)a_{t+1} \\ &< b \end{aligned}$$

by hypothesis of subcase 2(a)(i). Note that this expression is also valid for subcase 2(a)(ii) because $a_{q+1} = 0$. ♦

Since $v(\alpha)$ is a continuous function of α , Properties 2 and 3 and the mean value theorem imply that $v(\bar{\alpha}) = b$ for some value $\bar{\alpha}$ of α in the interval (a_{s+1}, a_{t+1}) . For $k = 1, 2, \dots, s$, define $\lambda_k = a_k - a_{s+1}$. Then $\lambda_k + \Delta_{s+1}(\bar{\alpha}) + \dots + \Delta_t(\bar{\alpha}) + \bar{\alpha} = a_k$ for $1 \leq k \leq s+1$ by Property 1. Also, $\bar{\alpha} < a_{t+1}$ implies $\bar{\alpha} < a_k$ for all $k \geq t+1$. Therefore, the dual variables corresponding to $\alpha = \bar{\alpha}$ are dual feasible and they, together with the primal solution, satisfy complementary slackness. Therefore, for case 2, subcase (a), the linear programming formulation P(FACET) always has an integer optimal solution.

Subcase 2(b).

Primal solution. $y = \mu_M$; $g_k = u_k$ for $k \leq t$; $g_{t+1} = C\mu_M - \sum_{k=1}^t u_k < u_{t+1}$; $g_k = 0$

for all $k > t+1$.

Dual solution. We again set $\delta_W = 0$ for all sets W that are not of the form $W = \{1, 2, \dots, p\}$ for some $s+1 \leq p \leq t$. Consider the complementary slackness conditions for the variables $g_{s+1}, g_{s+2}, \dots, g_t, g_{t+1}$, and y :

$$\begin{array}{rcccccccl} \lambda_{s+1} + & \Delta_{s+1} + & \Delta_{s+2} + \dots + & \Delta_t + & \alpha & = & a_{s+1} \\ \lambda_{s+2} + & & \Delta_{s+2} + \dots + & \Delta_t + & \alpha & = & a_{s+2} \\ \vdots & & \ddots & \vdots & \vdots & & \\ \lambda_t + & & & \Delta_t + & \alpha & = & a_t \\ & & & & \alpha & = & a_{t+1} \\ & & & & & & \\ & -R_{s+1}\Delta_{s+1} - R_{s+2}\Delta_{s+2} - \dots - R_t\Delta_t - C\alpha & = & b. \end{array}$$

Now, consider two solutions to the equalities corresponding to the g_k variables:

1. $\Delta_k^1 = 0$ and $\lambda_k^1 = a_k - a_{t+1}$ for all $s+1 \leq k \leq t$, $\alpha = a_{t+1}$.
2. $\Delta_k^2 = a_k - a_{k+1}$ and $\lambda_k^2 = 0$ for all $s+1 \leq k \leq t$, $\alpha = a_{t+1}$.

For $0 \leq \theta \leq 1$, let

$\Delta_k(\theta) = \theta\Delta_k^1 + (1 - \theta)\Delta_k^2$ and $\lambda_k(\theta) = \theta\lambda_k^1 + (1 - \theta)\lambda_k^2$ for all $s+1 \leq k \leq t$. Then

$\Delta_k(\theta), \lambda_k(\theta)$ and $\alpha = a_{t+1}$ satisfy the complementary slackness equalities corresponding to the variables $g_{s+1}, g_{s+2}, \dots, g_t$. Next note that

$-R_{s+1}\Delta_{s+1}(1) - R_{s+2}\Delta_{s+2}(1) - \dots - R_t\Delta_t(1) - C\alpha = -Ca_{t+1} \leq b$. Furthermore, as in subcase 2(a),

$$\begin{aligned} & -R_{s+1}\Delta_{s+1}(0) - R_{s+2}\Delta_{s+2}(0) - \dots - R_t\Delta_t(0) - C\alpha \\ & = -R_{s+1}a_{s+1} - R_{s+2}a_{s+2} - \dots - R_t a_t - (C - R_t)a_{t+1} \geq b \end{aligned}$$

by hypothesis. Therefore, for some value of $\bar{\theta}$ of θ , with $\bar{\alpha} = a_{t+1}$

$$-R_{s+1}\Delta_{s+1}(\bar{\theta}) - R_{s+2}\Delta_{s+2}(\bar{\theta}) - \dots - R_t\Delta_t(\bar{\theta}) - C\bar{\alpha} = b.$$

For $k \leq s$, set $\bar{\lambda}_k(\bar{\theta}) = a_k - \Delta_{s+1}(\bar{\theta}) - \Delta_{s+2}(\bar{\theta}) - \dots - \Delta_t(\bar{\theta}) - \bar{\alpha}$. Since $\Delta_{s+1}(\bar{\theta}) + \Delta_{s+2}(\bar{\theta}) + \dots + \Delta_t(\bar{\theta}) + \bar{\alpha} = a_{s+1} - \bar{\lambda}_{s+1}(\bar{\theta})$, $\bar{\lambda}_k(\bar{\theta}) = a_k - a_{s+1} + \bar{\lambda}_{s+1}(\bar{\theta}) \leq 0$. Also, $\bar{\alpha} = a_{t+1}$ implies $\bar{\alpha} \leq a_k$ for all $k \geq t+1$. Therefore, the dual variables corresponding to $\theta = \bar{\theta}$ are dual feasible and, they, together with the primal solution, satisfy complementary slackness. Therefore, for case 2, subcase (b), the linear programming formulation P(FACET) always has an integer optimal solution. \otimes

The arc residual inequalities for the single arc design problem are also valid for the NLP and hence could be used to strengthen its linear programming relaxation. Magnanti, Mirchandani and Vachani (1990) show that the linear programming relaxation of P(NLP), with inequalities (7) included, will provide a lower bound that is the same as that obtained from a Lagrangian approach that relaxes the flow conservation constraints (1). They identify additional facets for P(NLP) which further strengthen the formulation of NLP. Thus, the objective function value of the linear programming relaxation of their extended model is at least as strong as the bound obtained by the Lagrangian approach described in Section 2.

4. The Three-node Network Loading Problem

This section discusses the three-node network loading problem. As noted in Section 2, we will study P(3N), the projected formulation of the problem in the space of y variables. Recall that the nodes of the network are numbered 1, 2 and 3 and u_{12} , u_{13} , and u_{23} denote the demands between nodes 1 and 2, 1 and 3, and, 2 and 3 respectively. Let r_{ij} equal $u_{ij} \pmod C$. To consider interesting cases only, we assume that the demand values are nonzero and further, that they are not multiples of C , i.e., $0 < r_{ij} < C$. Our results for the three-node network apply to the general NLP if we partition the nodes for the larger problem into three sets. Thus, the facets developed for P(3N) are useful for tightening the linear programming relaxation of the general problem as well.

As mentioned in Section 2, the following set of *cutset* inequalities strengthen formulation P(3N):

$$\begin{aligned} y_{12} + y_{13} &\geq \left\lceil \frac{u_{12} + u_{13}}{C} \right\rceil \\ y_{12} + y_{23} &\geq \left\lceil \frac{u_{12} + u_{23}}{C} \right\rceil \\ y_{13} + y_{23} &\geq \left\lceil \frac{u_{13} + u_{23}}{C} \right\rceil. \end{aligned} \tag{9}$$

Although these cutset inequalities are facets of P(3N) (see Magnanti, Mirchandani and Vachani, 1990), in general, they are not sufficient to describe the convex hull of feasible solutions to the problem. We can view the cutset inequalities as (rank 1) Chvátal-Gomory inequalities derived by taking linear combinations of known inequalities and then tightening the aggregate inequality using integrality arguments. By applying the Chvátal-Gomory procedure to the cutset inequalities, we obtain the following additional (three-partition) valid inequality for P(3N):

$$y_{12} + y_{13} + y_{23} \geq \left\lceil \frac{1}{2} \left(\left\lceil \frac{u_{12} + u_{13}}{C} \right\rceil + \left\lceil \frac{u_{12} + u_{23}}{C} \right\rceil + \left\lceil \frac{u_{13} + u_{23}}{C} \right\rceil \right) \right\rceil. \tag{10}$$

We refer to this inequality as the three-partition inequality since it specifies a lower bound on the number of installed facilities on the arcs $\{1,2\}$, $\{1,3\}$ and $\{2,3\}$ that connect three nodes (a generalization of these three-partition inequalities with a set of nodes in place of each of the three nodes applies to the general network loading problem – see Magnanti, Mirchandani and Vachani, 1990).

This rank 2 Chvátal-Gomory inequality is a facet for P(3N) if it is nonredundant, i.e., if the righthand sides of the three inequalities in (9) sum to an odd integer. The next theorem shows that the cutset and three-partition inequalities completely characterize the convex hull of feasible solutions to P(3N).

Theorem 4.1. *The cutset inequalities (9), the three-partition inequality (10), and the nonnegativity constraints completely describe the convex hull of feasible solutions to P(3N). Furthermore, these inequalities provide a nonredundant description of the convex hull if the three-partition inequality (10) is nonredundant; otherwise, the cutset inequalities and the nonnegativity constraints provide a nonredundant description of the convex hull.*

Proof.

Let P(3LP) denote the following linear program

Problem P(3LP):

$$\text{minimize } b_{12}y_{12} + b_{13}y_{13} + b_{23}y_{23}$$

subject to:

$$\begin{aligned} y_{12} + y_{13} &\geq \left\lceil \frac{u_{12} + u_{13}}{C} \right\rceil \\ y_{12} + y_{23} &\geq \left\lceil \frac{u_{12} + u_{23}}{C} \right\rceil \\ y_{13} + y_{23} &\geq \left\lceil \frac{u_{13} + u_{23}}{C} \right\rceil \\ y_{12} + y_{13} + y_{23} &\geq \left\lceil \frac{1}{2} \left(\left\lceil \frac{u_{12} + u_{13}}{C} \right\rceil + \left\lceil \frac{u_{12} + u_{23}}{C} \right\rceil + \left\lceil \frac{u_{13} + u_{23}}{C} \right\rceil \right) \right\rceil \\ y_{12}, y_{13}, y_{23} &\geq 0 \end{aligned}$$

and let P(3D) denote its dual

Problem P(3D):

$$\begin{aligned} \text{maximize } &\lambda_1 \left\lceil \frac{u_{12} + u_{13}}{C} \right\rceil + \lambda_2 \left\lceil \frac{u_{12} + u_{23}}{C} \right\rceil + \lambda_3 \left\lceil \frac{u_{13} + u_{23}}{C} \right\rceil \\ &+ \pi \left\lceil \frac{1}{2} \left(\left\lceil \frac{u_{12} + u_{13}}{C} \right\rceil + \left\lceil \frac{u_{12} + u_{23}}{C} \right\rceil + \left\lceil \frac{u_{13} + u_{23}}{C} \right\rceil \right) \right\rceil \end{aligned}$$

subject to:

$$\lambda_1 + \lambda_2 + \pi \leq b_{12}$$

$$\lambda_1 + \lambda_3 + \pi \leq b_{13}$$

$$\lambda_2 + \lambda_3 + \pi \leq b_{23}$$

$$\lambda_1, \lambda_2, \lambda_3, \pi \geq 0.$$

To prove the theorem, we will construct an integer solution to P(3LP) and a corresponding dual solution to P(3D) that have the same objective function value (or, equivalently, that satisfy complementary slackness). We will assume that $b_{ij} \geq 0$ for all $\{i,j\}$, since otherwise the primal solution is unbounded. We also assume, without loss of generality, that $b_{23} \geq b_{12}$ and $b_{23} \geq b_{13}$. We consider the following two cases.

Case I. $b_{12} + b_{13} \leq b_{23}$.

This condition on the facility costs implies that an optimal solution to P(3N) will not load facilities on arc $\{2,3\}$; instead, the solution will route the demand between nodes 2

and 3 on arcs $\{1,2\}$ and $\{1,3\}$. Thus, we construct the following solutions to $P(3LP)$ and $P(3D)$, respectively.

$$\text{Primal solution. } y_{12} = \left\lceil \frac{u_{12} + u_{23}}{C} \right\rceil, y_{13} = \left\lceil \frac{u_{13} + u_{23}}{C} \right\rceil \text{ and } y_{23} = 0.$$

$$\text{Dual solution. } \lambda_1 = 0, \lambda_2 = b_{12}, \lambda_3 = b_{13}, \pi = 0.$$

Case 2. $b_{12} + b_{13} > b_{23}$.

The condition of this case implies that $y_{ij} \geq \lfloor u_{ij}/C \rfloor$ for all $\{i,j\}$ in an optimal solution to $P(3N)$; however, the exact number of facilities to be loaded on each of the arcs depends on the demand between the different pairs of nodes and we need to consider several subcases.

Subcase 2(a). $r_{12} + r_{13} \leq C, r_{12} + r_{23} \leq C$ and $r_{13} + r_{23} \leq C$.

Subcase 2(b). $r_{12} + r_{13} \leq C, r_{12} + r_{23} \leq C$ and $r_{13} + r_{23} > C$.

Subcase 2(c). $r_{12} + r_{13} \leq C, r_{12} + r_{23} > C$ and $r_{13} + r_{23} > C$.

Subcase 2(d). $r_{12} + r_{13} > C, r_{12} + r_{23} \leq C$ and $r_{13} + r_{23} \leq C$.

Subcase 2(e). $r_{12} + r_{13} > C, r_{12} + r_{23} \leq C$ and $r_{13} + r_{23} > C$.

Subcase 2(f). $r_{12} + r_{13} > C, r_{12} + r_{23} > C$ and $r_{13} + r_{23} > C$.

Note that because of symmetry between the variables y_{12} and y_{13} , we do not need to consider the remaining two subcases:

Subcase 2(g). $r_{12} + r_{13} \leq C, r_{12} + r_{23} > C$ and $r_{13} + r_{23} \leq C$.

Subcase 2(h). $r_{12} + r_{13} > C, r_{12} + r_{23} > C$ and $r_{13} + r_{23} \leq C$.

Consider subcase 2(a). The stated condition implies that

$$\begin{aligned} \left\lceil \frac{u_{12} + u_{13}}{C} \right\rceil &= \left\lceil \frac{u_{12}}{C} \right\rceil + \left\lceil \frac{u_{13}}{C} \right\rceil + 1 \\ \left\lceil \frac{u_{12} + u_{23}}{C} \right\rceil &= \left\lceil \frac{u_{12}}{C} \right\rceil + \left\lceil \frac{u_{23}}{C} \right\rceil + 1 \\ \left\lceil \frac{u_{13} + u_{23}}{C} \right\rceil &= \left\lceil \frac{u_{13}}{C} \right\rceil + \left\lceil \frac{u_{23}}{C} \right\rceil + 1 \end{aligned}$$

and, hence, the three-partition inequality (10) is nonredundant with righthand side equal to

$$\left\lfloor \frac{u_{12}}{C} \right\rfloor + \left\lfloor \frac{u_{13}}{C} \right\rfloor + \left\lfloor \frac{u_{23}}{C} \right\rfloor + 2.$$

Similarly, the three-partition inequality is nonredundant for subcases 2(c) and 2(e) and is redundant for the other three cases.

Subcases 2(a), 2(c) and 2(e). For these three subcases, we define the dual solution as follows:

$$\lambda_1 = 0, \lambda_2 = b_{23} - b_{13}, \lambda_3 = b_{23} - b_{12}, \pi = b_{12} + b_{13} - b_{23}.$$

This solution is feasible for P(3D) and satisfies each of the constraints as an equality. The optimal solutions to P(3LP) corresponding to this solution depend on the values of r_{ij} and are:

$$\begin{aligned} y_{12} &= \left\lfloor \frac{u_{12}}{C} \right\rfloor + 1 \text{ for all three subcases;} \\ y_{13} &= \begin{cases} \left\lfloor \frac{u_{13}}{C} \right\rfloor + 1 & \text{for subcases 2(a) and 2(c),} \\ \left\lfloor \frac{u_{13}}{C} \right\rfloor + 2 & \text{for subcase 2(e);} \end{cases} \\ y_{23} &= \begin{cases} \left\lfloor \frac{u_{23}}{C} \right\rfloor & \text{for subcases 2(a) and 2(e),} \\ \left\lfloor \frac{u_{23}}{C} \right\rfloor + 1 & \text{for subcase 2(c).} \end{cases} \end{aligned}$$

It is easy to verify that these solutions are feasible for P(3LP) for the respective cases and that these integer solutions and the dual solution satisfy complementary slackness.

Subcases 2(b), 2(d) and 2(f). For these subcases, the three-partition inequality is redundant; we set $\pi = 0$ and the remaining variables of P(3D) as follows for all three subcases:

$$\lambda_1 = \frac{b_{12} + b_{13} - b_{23}}{2}, \lambda_2 = \frac{b_{12} + b_{23} - b_{13}}{2} \text{ and } \lambda_3 = \frac{b_{13} + b_{23} - b_{12}}{2}.$$

This solution is feasible for P(3D) and satisfies each of the constraints as an equality. The optimal solutions to P(3LP) corresponding to this dual solution are:

$$\begin{aligned}
 y_{12} &= \begin{cases} \left\lfloor \frac{u_{12}}{C} \right\rfloor & \text{for subcase 2(b),} \\ \left\lfloor \frac{u_{12}}{C} \right\rfloor + 1 & \text{for subcase 2(d) and 2(f);} \end{cases} \\
 y_{13} &= \left\lfloor \frac{u_{13}}{C} \right\rfloor + 1 \text{ for all three subcases;} \\
 y_{23} &= \begin{cases} \left\lfloor \frac{u_{23}}{C} \right\rfloor + 1 & \text{for subcases 2(b) and 2(f),} \\ \left\lfloor \frac{u_{23}}{C} \right\rfloor & \text{for subcase 2(d).} \end{cases}
 \end{aligned}$$

Again, it is easy to verify that these solutions are feasible for P(3LP) for the respective cases and satisfy complementary slackness. \otimes

4. Conclusion

In this paper, we have studied two core subproblems of the Network Loading Problem. In both cases we completely characterized the convex hull of the set of feasible solutions by identifying additional classes of facets. Several generalizations of the results are possible. For example, a generalized version of the arc residual capacity inequality applies to practical situations in the telecommunications and trucking industries when two facilities of capacities 1 and C (representing two communication channels or two trucks) are available. More importantly, a generalization of Theorem 3.5 applies to these situations. This generalization has an algorithmic implication: for the two facility case, it permits us to develop a linear programming formulation of the NLP whose bound is at least as strong as the bound obtained by a Lagrangian relaxation approach that relaxes flow conservation constraints. Magnanti, Mirchandani and Vachani (1990) describe this result and its computational consequences in greater detail.

The second subproblem that we have studied is a three-node network loading problem with a demand between every pair of nodes. Although the problem has little, if any, direct application, it is useful in the context of network loading problems defined on general networks if the nodes are partitioned into three subsets. We have noted that the cutset facet inequalities are not, in general, sufficient for describing the convex hull of feasible solutions to this problem. We identified an additional facet inequality, the three-partition inequality, and appended it to the formulation to describe the convex hull. This three-partition facet generalizes to facets for larger networks if we consider a three-partition of the node set; moreover, a generalization of this facet define valid inequalities for the two facility

case. These facets and valid inequalities can form the basis of a cutting plane procedure for solving the NLP.

Indeed, we are currently in the process of testing the computational efficacy of the arc residual capacity and the three-partition inequalities. Our preliminary results are quite encouraging: when applied to some practical multifacility network problems encountered in the telecommunications industry, the addition of these inequalities, together with some related cutset residual capacity inequalities, has reduced the gap between the objective values of the integer program and its linear programming relaxation by 75%. Magnanti, Mirchandani and Vachani (1990) give more details concerning these computational ramifications of the results presented in this paper.

Appendix A

In this appendix we provide proofs for two of the results stated in the body of this paper.

Proposition 3.1. *Conv(SADP) is a full dimensional polyhedron. That is, $\dim(\text{Conv}(\text{SADP}))$ equals $q + 1$.*

Proof. The constraint set of P(SADP) has $q + 1$ variables and no explicitly defined equations. Therefore, $\dim(\text{Conv}(\text{SADP}))$ is no more than $q + 1$. To show that this bound is tight, we list $q + 2$ affinely independent points belonging to $\text{Conv}(\text{SADP})$.

- (i) $y = 0; g_k = 0$ for all $k \in Q$.
- (ii) $y = 1; g_k = 0$ for all $k \in Q$.
- (iii) Let $j \in Q; y = \lceil u_j / C \rceil; g_k = u_k$ for $k = j$ and 0 otherwise. (There are q such solutions.) ⊗

Proposition 3.2.

1. For every $k \in Q, g_k \geq 0$ defines a facet of $\text{Conv}(\text{SADP})$.
2. For every $k \in Q, g_k \leq u_k$ defines a facet of $\text{Conv}(\text{SADP})$.
3. If $\sum_{k \in Q} u_k > C$, then $\sum_{k \in Q} g_k \leq Cy$ defines a facet of $\text{Conv}(\text{SADP})$.

Proof. In order to show that a valid inequality $\lambda y + \beta g \leq \beta_0$ for $\text{Conv}(\text{SADP})$ is also facet defining we must show that (a) $\{(y, g) \in \text{Conv}(\text{SADP}) : \lambda y + \beta g = \beta_0\} \neq \text{Conv}(\text{SADP})$, i.e., $\lambda y + \beta g \leq \beta_0$ defines a proper face of $\text{Conv}(\text{SADP})$ and (b) $(q+1)$ affinely independent points in $\text{Conv}(\text{SADP})$ satisfy $\lambda y + \beta g = \beta_0$.

It is easy to see that all three inequalities satisfy condition (a). Thus, we will establish only condition (b).

1. Without loss of generality, consider the inequality $g_1 \geq 0$. The $(q + 1)$ points listed in the proof of Proposition 3.1 with $j \in Q \setminus \{1\}$ satisfy $g_1 = 0$.

2. Without loss of generality, consider the inequality $g_1 \leq u_1$. The following solutions are affinely independent, belong to $\text{Conv}(\text{SADP})$, and satisfy $g_1 = u_1$.

- (i) $y = \lceil u_1/C \rceil$; $g_1 = u_1$; $g_k = 0$ for all $k \geq 2$.
- (ii) $y = \lceil u_1/C \rceil + 1$; $g_1 = u_1$; $g_k = 0$ for all $k \geq 2$.
- (iii) Let $j \in Q \setminus \{1\}$, $y = \lceil (u_1 + u_j)/C \rceil$; $g_1 = u_1$; $g_k = u_k$ for $k = j$ and 0 otherwise. (There are $q-1$ such solutions.)

3. Let P form a minimal cover of C , i.e., $\sum_{k \in P} u_k \geq C$ and $\sum_{k \in P \setminus \{j\}} u_k < C$ for any $j \in P$.

By assumption, the problem has a minimum cover. Without loss of generality, suppose that $P = \{1, 2, \dots, p\}$. Let $\delta = C - \sum_{k \in P \setminus \{p\}} u_k < u_p$. The following $q + 1$ solutions are affinely independent, belong to $\text{Conv}(\text{SADP})$, and satisfy $\sum_{k \in Q} g_k = Cy$.

- (i) $y = 0$; $g_k = 0$ for all $k \in Q$.
- (ii) $y = 1$; $g_k = u_k$ for all $1 \leq k \leq (p-1)$; $g_p = \delta$; and $g_k = 0$ for all $k \in Q \setminus P$.
- (iii) $y = 1$; $g_1 = u_1 - 1$; $g_k = u_k$ for all $2 \leq k \leq (p-1)$; $g_p = \delta + 1$; and $g_k = 0$, for all $k \in Q \setminus P$.
- (iv) $y = 1$; $g_1 = u_1 - 1$; $g_k = u_k$ for all $2 \leq k \leq (p-1)$; $g_p = \delta$; let $j \in Q \setminus P$, $g_k = 1$ for $k = j$ and 0 otherwise. (There are $(q - p)$ such solutions.)
- (v) $y = 1$; let $j \in P \setminus \{1\}$, $g_j = u_j - 1$; $g_k = u_k$ for $k \in P \setminus \{j\}$; $g_p = \delta + 1$; and $g_k = 0$ for $k \in Q \setminus P$. (There are $(p - 2)$ such solutions.) \otimes

References

- Balas, E., and W. R. Pulleyblank, "The perfectly matchable subgraph polytope of a bipartite graph," *Networks* 13 (1983) 486–516.
- Balas, E., and W. R. Pulleyblank, "The perfectly matchable subgraph polytope of an arbitrary graph," (1987). To appear in *Combinatorica*.
- Barany, I., T. J. Van Roy, and L. A. Wolsey, "Uncapacitated lot sizing: The convex hull of solutions," *Mathematical Programming Study* 22 (1984) 32–43.
- Crowder, H., E. L. Johnson, and P. W. Padberg, "Solving large-scale zero-one linear programming problems," *Operations Research* 31 (1983) 803–834.
- Cornuejols, G., R. Shridharan, and J.M. Thizy "A comparison of heuristics and relaxations for the capacitated plant location problem," *European Journal of Operations Research* (1990) to appear.
- Edmonds, J., "Maximum matching and a polyhedron with 0-1 vertices," *Journal of Research of the National Bureau of Standards* 69B (1965) 125–130.
- Edmonds, J., "Submodular functions, matroids and certain polyhedra." In *Combinatorial Structures and Their Applications*, R. Guy et al., eds. Gordon and Breach, (1970) 69–87.
- Edmonds, J., "Matroids and the greedy algorithm," *Mathematical Programming* 1 (1971) 127–136.
- Geoffrion, A. M., "Lagrangian relaxation for integer programming," *Mathematical Programming Study* 2 (1974) 82–114.
- Leung, J. M. Y, and T. L. Magnanti, "Valid inequalities and facets of the capacitated plant location problem," *Mathematical Programming* 44 (1989) 271–291.
- Leung, J. M. Y, T. L. Magnanti, and V. Singhal, "Routing in point to point delivery systems," (1990). To appear in *Transportation Science*.
- Magnanti, T. L., P. Mirchandani, and R. Vachani, "Modeling and solving the capacitated network loading problem," (1990). In preparation.
- Padberg, M. W., T. J. Van Roy, and L. A. Wolsey, "Valid inequalities for fixed charge problems," *Operations Research* 33 (1985) 842–861.
- Papernov, B. A., "Feasibility of multicommodity flows." In *Studies in Discrete Optimization*, A. Friedman, ed. Idzat. "Nauka", Moscow, (1976) 230–261 (in Russian).
- Powell, W. B., and Y. Sheffi, "The load planning problem of LTL motor carriers: Problem description and a proposed solution approach," *Transportation Research* 17A (1983) 471–480.
- Vachani, R., "Design of private communication networks," TM 0027-07-88-446 (1988) GTE Laboratories Incorporated, Waltham, MA 02254.