OPERATIONS RESEARCH CENTER

Working Paper

Polyhedral Properties of the Network Restoration Problem-with the Convex Hull of a Special Case

by
T. L. Magnanti and
Y. Wang

OR 323-97 November 1997

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

Polyhedral Properties of the Network Restoration Problem-with the Convex Hull of a Special Case

by

T. L. Magnanti and
Y. Wang

OR 323-97 November 1997

Polyhedral Properties of the Network Restoration Problem – with the Convex Hull of a Special Case

Thomas L. Magnanti
Sloan School of Management and
Department of Electrical Engineering and Computer Science,
MIT, Cambridge, MA 02139

Yi Wang
Operations Research Center, MIT, Cambridge, MA 02139

November, 1997

Abstract

The network restoration problem is a specialized capacitated network design problem in which spare capacity must be installed in a network to fully restore disrupted demands in the event of any link failure. We consider the installation of spare capacity using a single type of capacitated facility. The problem is to determine the number of facilities to be installed on the edges of the network so that it is capable of routing point-to-point traffic when any single edge fails. This paper develops a new family of facets for an integer programming formulation for the problem and shows that these facets completely characterize the convex hull of feasible integer solutions for a special case, the *parallel path network restoration problem*, which arises in larger networks if we aggregate nodes.

1. Introduction

Our motivation for studying the network restoration problem arose in the context of the telecommunication industry. Because of their high bandwidth, new fiber transmission systems can carry large amounts of data on a few strands of fiber. As a result, the failure of a single transmission facility such as a link in the network can create a severe service loss to the customers.

Many businesses with high at-risk revenues or critical communication needs, such as financial institutions and airlines, seek to protect themselves from a major network disaster by requesting increasingly reliable service or by acquiring and managing their own networks. A few examples illustrate the practical importance of providing uninterrupted service. AT&T's Intelligent Network, the world's largest, handles more than 150 million calls daily. In the past several years, AT&T's network has failed several times, affecting millions of customers in the United States and abroad. The power outage on September 17, 1991 disabled AT&T's New York City switching station. The nine hour breakdown created havoc for long-distance customers in many parts of the US. To prevent this kind of catastrophe, AT&T has invested billions of dollars to develop the Fast Automated Restoration (FASTAR) system. This system uses the Digital Access and Cross-Connect System (DACS) and intelligent real time routing to restore 67,200 voice circuits in minutes[1]. Dynamic restoration schemes use pre-assigned spare capacity in the network to accommodate the traffic when any equipment or link fails. The United Services Automobile Association (USAA) built its private network based on AT&T Garrison System architecture, which automatically protects both link and node failures[2]. A disaster recovery system is essential to banks which rely heavily on networks and communications. The Office of the Comptroller of the Currency mandates that in case of a disaster, banks must be able to reinitiate their business within a prescribed period of time. To survive another disaster of the magnitude of the Mississippi Valley flood in summer of 1993's, many banks adopted AT&T's dynamic reconfiguration solutions. Recently, AT&T has developed a service that lets corporate networks build in as much backup capacity as they want but pay only when disaster recovery is needed[3].

To address this situation, both telecommunication carriers and companies that maintain private networks would like to design cost effective networks that can readily survive failures by rerouting traffic using pre-installed spare capacity. The objective of the *network restoration problem* (also known as the spare capacity assignment problem) is to determine where and how much spare capacity to install in a network at minimum cost. In many cases, planners have a choice of different facilities offering different levels of capacity at costs that exhibit economies of scale. The planners then need to determine the optimal combination of the facilities to install on the edges of the network.

Researchers have studied various versions of the network restoration problem. Balakrishnan, Magnanti, Sokol and Wang [1] describe much of the relevant literature. In particular, Sakauchi, Nishimura and Hasegawa [6], Grover, Billodeau and Venables [3], and Balakrishnan et al. [1] study the network restoration prob-

[1] "Network restoration," Telephone Engineer & Management, August, 1993
[2] "Single-source Survivability," Telephone Engineer & Management, February, 1993
[3] "Bargain Backup: Captures and Decodes," Data Communications, March, 1996

lem using a line restoration scheme and a single facility type. We will describe the line restoration (also known as the local rerouting scheme) briefly in the next section. Sakauchi et al. [6] and Grover et al. [3] focus on various solution approaches to the problem. Balakrishnan et al. [1] present the first polyhedral analysis and provide a better understanding of the problem by generating valid inequalities and using them computationally. Veerasamy, Venkatesan and Shah [8] describe various restoration schemes and present an approximation scheme for solving the path restoration problem (as known as the global rerouting scheme). Lisser, Sarkissian and Vial [4] study the dimensioning of the reserve network to restore flows in the event of single node or edge failures using a global rerouting scheme. Stoer and Dahl [7] examine an integrated problem of allocating both the working and spare capacity so that the network can route all the point-to-point traffic under single node or edge failures.

In this paper, assuming a line restoration scheme and a single facility type, we develop a new family of facets for the network restoration problem. We also completely characterize the convex hull of feasible solutions for a special case, the *parallel path network restoration problem.* This special case arises when we aggregate any larger network into a two-node network and consider the polyhedral structure across a single cut.

Recently, in a parallel and independent investigation, Bienstock and Muratore [2] have examined the polyhedral structure of network restoration problem subject to node failures. They have discovered some results that are similar to those presented in Balakrishnan et al. [1] and in this paper.

This paper is organized as follows. In Section 2, we describe the underlying assumptions of the network restoration problem and present an integer programming formulation of the problem. In Section 3, we derive a new family of valid inequalities and show that they are facet defining under certain conditions. In Section 4, we introduce the parallel path problem and examine properties of the linear relaxation of its integer programming formulation. We also present polynomial time algorithms for solving both the linear relaxation and the integer programming formulation of the parallel path problem. In Section 5, we completely characterize the convex hull of feasible solutions for the parallel path problem. Section 6 summarizes our results.

2. Problem Description

2.1. Problem assumptions

The network restoration problem we examine in this paper is based upon several assumptions:

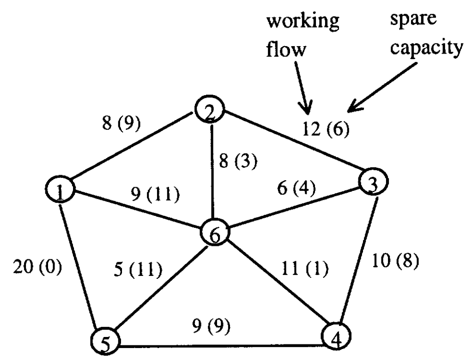

Figure 2.1: A network with working flow and spare capacity

(1) On each edge e, the network has a current flow of d_e which we refer to as the working flow or demand on edge e. (In this paper, we use the working flow and demand interchangeably.)

(2) Only one edge fails at a time;

(3) The network restores traffic using a line restoration scheme, that is, the interrupted traffic creates a local demand between the end points of the failed edge, and the system routes this demand from one end point to the other using spare capacity in the network (not on the failed edge itself);

(4) Network contains no existing spare capacity on any edges;

(5) We use only one type of facility with fixed capacity $C = 1$;

(6) The restoration traffic can travel on unlimited number of alternate paths;

(7) Once the facilities are installed, the system incurs no additional routing cost.

Example 2.1. Figure 2.1 illustrates a small network with working flow and spare capacity. For instance, edge $\{2, 6\}$ has 8 units of working flow and 3 units of spare capacity. When the edge fails, we must reroute the interrupted working flow from node 2 to node 6, not on edge $\{2, 6\}$ itself. The network contains a path from node 2 to node 6 via node 1 with enough spare capacity to restore edge $\{2, 6\}$. When edge $\{1, 2\}$ fails, the network can reroute the 8 units of working flow on the following two paths: 3 units from node 1 to node 2 via node 6, and 5 units via nodes 6, 5, 4, and 3. Note that the alternate paths are not necessarily disjoint.

2.2. Integer programming model

If we let y_e denote the amount of spare capacity loaded on edge $e \in E$ in our given network $G = (N, E)$, and let d_e denote the working flow (demand) on edge e, then the following model is one natural formulation of the network restoration problem.

Model (NRP)

Minimize

$$\sum_{e \in E} c_e y_e$$

subject to

$$\sum_{j \in K \setminus \{e\}} y_j \geq d_e \text{ for all edges } e \in E \text{ and all cutsets } K \text{ containing edge } e. \quad (2.1)$$

$$y_e \geq 0 \text{ and integer for all edges } e \in E.$$

We refer to the inequalities (2.1) in this model as *cutset capacity inequalities*. They state that if any edge fails, the total spare capacity on the remaining edges in any cutset separating the end points of that edge must be at least as large as the working flow on that edge.

An easy application of the max-flow min-cut theorem shows that if y is a feasible solution to this model (whether integer or not), then a network with capacity y_e installed on each edge $e \in E$ has sufficient capacity to restore each edge in E (that is, route d_e units of flow between the end points of edge e without flowing anything on this edge). Balakrishnan et al. [1] have examined a version of this model with explicit flow variables.

3. A New Family of Valid Inequalities

In this section we describe a new set of valid inequalities for the convex hull of the network restoration problem. By adding these inequalities, we can strengthen the linear program formulation of Model (*NRP*).

Let $K = \{1, 2, ..., k\}$ be the edge set of any cutset in the network. For any subset Q of K, let $D_Q = \sum_{j \in Q} d_j$ and $y_Q = \sum_{j \in Q} y_j$. Let $\bar{Q} = K \setminus Q$ and $y_{\bar{Q}} = \sum_{j \in \bar{Q}} y_j$. We also let $q = |Q|$, and $r_Q = D_Q \bmod (q-1)$.

Key fact. By summing the cutset capacity inequalities corresponding to the indices $j \in Q$, we obtain the following valid aggregate inequality:

$$(q-1)y_Q + q y_{\bar{Q}} \geq D_Q. \quad (3.1)$$

We can strengthen this inequality by adding the following Q–subset inequalities:

$$r_Q y_Q + (r_Q + 1) y_{\bar{Q}} \geq r_Q \lceil D_Q/(q-1) \rceil . \tag{3.2}$$

In this expression, $r_Q = D_Q \bmod (q-1)$ and $r_Q = (q-1)$ if D_Q is a multiple of $(q-1)$.

Example 2.1 continued. Consider the network shown in Figure 2.1. Let K be the cutset defined by the edges incident to node 6 and let Q be defined by edges $\{1,6\}$, $\{2,6\}$, $\{3,6\}$ and $\{4,6\}$. Then $D_Q = 34$, $q = 4$, $r_Q = 34 \bmod 3 = 1$, and $\lceil D_Q/(q-1) \rceil = 12$. The corresponding Q-subset inequality is given by

$$y_{16} + y_{26} + y_{36} + y_{46} + 2y_{56} \geq 12.$$

For this example, the cutset capacity inequality for edge $\{5,6\}$ and cutset K and the aggregate inequality (3.1) are given by

$$y_{16} + y_{26} + y_{36} + y_{46} \geq 5$$

and

$$4(y_{16} + y_{26} + y_{36} + y_{46}) + 5y_{56} \geq 34.$$

Note that when $y_{56} = 0$, the Q-subset inequality is stronger than either of these inequalities.

The following theorem establishes the validity of the Q-subset inequality and provides a geometric interpretation of it.

Theorem 3.1. *The Q-subset inequality (3.2) is valid for the network restoration problem.*

Proof. If $r_Q = (q-1)$, (3.2) becomes (3.1) and is obviously true.

Assume $r_Q < (q-1)$.

We plot (3.1) and (3.2) in the space of y_Q and $y_{\bar{Q}}$ for some subset Q of K. In Figure (3.1), line 1 corresponds to (3.1) and line 2 corresponds to (3.2). When $y_{\bar{Q}} \geq r_Q$, line 2 lies to the left of line 1. That is, (3.1) dominates (3.2). When $y_{\bar{Q}} = r_Q$, line 1 and 2 intersect at an integral solution corresponding to Point A with

$$\left\{ y_Q = \lfloor D_Q/(q-1) \rfloor - r_Q, y_{\bar{Q}} = r_Q \right\}.$$

Assume $0 < y_{\bar{Q}} < r_Q$. We want to show that the shaded area when $0 < y_{\bar{Q}} < r_Q$ in Figure (3.1) between line 1 and line 2 contains no integral solutions so no feasible solution to Model (*NRP*). Since the distance between Point A and B on

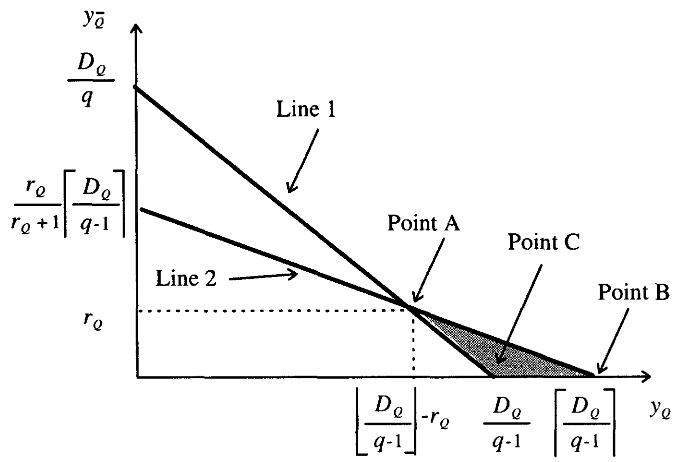

Figure 3.1: Plot in the $y_Q - y_{\bar{Q}}$ space

the y_Q-axis is $(r_Q + 1)$ and the slope of line 2 is $(-\frac{r_Q}{r_Q+1})$, and since point B is an integral point, we know that there are no integral points between A and B on line 2. Consider the integral coordinates of y_Q between point A and C. They are of the form

$$y_Q = \lfloor D_Q/(q-1) \rfloor - r_Q + h,$$

for some integer h satisfying the condition $1 \leq h \leq r_Q$. The corresponding value of $y_{\bar{Q}}$ on line 2 is of the form

$$\begin{aligned} y_{\bar{Q}}^2 &= r_Q - \frac{r_Q}{r_Q+1}h \\ &= r_Q - h + \frac{h}{r_Q+1}. \end{aligned}$$

Since $\frac{h}{r_Q+1} < 1$, we know that

$$\left\lfloor y_{\bar{Q}}^2 \right\rfloor = r_Q - h.$$

The corresponding value of $y_{\bar{Q}}$ on line 1 is of the form

$$\begin{aligned} y_{\bar{Q}}^1 &= r_Q - \frac{q-1}{q}h \\ &= r_Q - h + \frac{h}{q}. \end{aligned}$$

Since $h < q$, we have

$$y_{\bar{Q}}^1 > r_Q - h = \left\lfloor y_{\bar{Q}}^2 \right\rfloor .$$

Therefore, there are no integral points between Point A and C in the shaded area. It is easy to see that no points between Point C and B (excluding B) have an integral y_Q coordinate. Therefore, we have shown that shaded area contains no integral solutions. ■

Remark 1. *Balakrishnan et al. [1] define a cardinality-k cutset to be any cutset in the network consisting of k edges and they study the cardinality-k cutset inequality*

$$y_K \geq \left\lceil \frac{D_K}{k-1} \right\rceil .$$

In this expression, y_K and D_K are total capacity and total demand on cutset K. Note that when $q = k$ and $\bar{Q} = \emptyset$, (3.2) becomes

$$r_K y_K \geq r_K \left\lceil \frac{D_K}{k-1} \right\rceil .$$

Since $r_K \neq 0$, we obtain the cardinality-k cutset inequality by dividing both sides of this expression by r_K. Therefore, the inequality (3.2) is a generalization of the cardinality-k cutset inequality.

Since Balakrishnan et al. [1] have proved that the cardinality-k cutset inequality is facet defining in the space of y variables under certain conditions, in the following theorem, we show that (3.2) is facet defining under certain conditions when Q is the proper subset of K, that is, $\bar{Q} \neq \emptyset$.

We say a network is *restorable* if for any failed edge, the network contains a path from one end of the edge to the other. If we wished, we could install sufficient spare capacity on this path to restore flow on the failed edge.

In the next result and throughout rest of this paper, for a given cutset K of cardinality k, we let $A = D_K/(k-1)$ denote a modified average of the total demand on the edges of K.

Theorem 3.2. *When Q is a proper subset of the cutset K, the inequality (3.2) defines a facet for the network restoration problem in the space of the capacity variables if*

(i) The two subnetworks separated by the cutset K are restorable;

(ii) $r_Q < (q-1)$;

(iii) $d_e \leq \lfloor D_Q/(q-1) \rfloor$ *for all edges* $e \in K$; *and*

(iv) $A < \lfloor D_Q/(q-1) \rfloor$.

Proof. Assume $Q = \{1, 2, ..., q\}$ and $\bar{Q} = \{q+1, ..., k\}$.

We use the interchange argument in this proof. That is, we let $\sum_{e \in E} \alpha_e y_e = \gamma$ $(*)$ represent an arbitrary equation that is satisfied by every feasible solution y that satisfies (3.2) as an equality. We want to show that the coefficients α in $(*)$ are a multiple of those in (3.2). We first construct feasible solutions satisfying (3.2) as an equality. By substituting the solutions in $(*)$, and comparing the resulting expressions, we then derive a relationship between the coefficients appearing in $(*)$.

In this discussion, we let $Y = \sum_{e \in K} y_e$ denote the total spare capacity installed in the cutset K. Note that if we subtract this defining equation from the cutset capacity inequalities (2.1), they become

$$y_e \leq Y - d_e \text{ for all } e \in K. \tag{3.3}$$

These inequalities and the defining condition $Y = \sum_{e \in K} y_e$ are equivalent to the cutset capacity inequalities. The inequalities (3.3) state the obvious condition that in order to restore edge e, we must install at least d_e units of the total installed spare capacity on other edges in the cutset K. We say that the edge e is *saturated* if $y_e = Y - d_e$, or equivalently, the cutset capacity constraint defined by $K \backslash \{e\}$ is binding. Since $y_e \geq 0$, feasibility requires that $Y - d_e \geq 0$, or equivalently, $Y \geq d_e$ for all $e \in K$.

Since the two subnetworks separated by the cutset K are restorable, we can install a sufficiently large amount of spare capacity on all the edges in the two subnetworks to restore the edges in the subnetworks. To restore the edges in the cutset K, we construct two feasible integral solutions satisfying (3.2) as an equality as follows.

Solution (1) Let $Y^1 = \lfloor D_Q/(q-1) \rfloor$. We saturate all edges in the set Q by setting $y_j = Y^1 - d_j$ for $j = 1, ..., q$. By summing these expressions, we obtain

$$y_Q = q \lfloor D_Q/(q-1) \rfloor - D_Q = \lfloor D_Q/(q-1) \rfloor - r_Q.$$

Thus we must install the remaining r_Q units of capacity on the set $\bar{Q}$, that is set $y_{\bar{Q}} = r_Q$. We saturate edges in $\bar{Q}$ one at a time in any order until we have allocated all of r_Q. It is easy to show at least one of the edges in $\bar{Q}$ is not saturated. For suppose that the $(k-q)$ edges in $\bar{Q}$ are all saturated. Since the edges in Q are all saturated, we have $y_j = Y^1 - d_j$ for $j = 1, ..., k$.

Adding the equalities shows that $Y^1 = A$, and we reach the contradiction $Y^1 = A < \lfloor D_Q/(q-1) \rfloor = Y^1$. Note that this solution satisfies (3.2) as an equality.

Solution (2) Let $Y^2 = \lceil D_Q/(q-1) \rceil$. We saturate the edges in Q one at a time starting from edge 1 until we have allocated all of Y^2. It is easy to show that the q edges in Q are not all saturated. For suppose they are. Then $y_j = Y^1 - d_j$ for $j = 1, ..., q$ and $y_{\bar{Q}} = 0$. Adding the equalities shows that $Y^2 = D_Q/(q-1)$, and we reach the contradiction $Y^2 = D_Q/(q-1) < \lceil D_Q/(q-1) \rceil = Y^2$ since $r_Q < (q-1)$. Therefore, $y_Q = \lceil D_Q/(q-1) \rceil$ and $y_{\bar{Q}} = 0$. Note that this solution satisfies (3.2) as an equality.

By construction, the solution (1) is feasible since by assumption (iii), $Y^1 \geq d_e$ for all $e \in K$. Similarly, the solution (2) is feasible since $Y^2 \geq d_e$ for all $e \in K$.

1. Claim: $\alpha_j = 0$ for all $j \notin K$. Given any feasible solution y satisfying (3.2) as an equality, we can always increase y_j by 1 while keeping all other variables unchanged. The new solution y' is feasible and satisfies (3.2) as an equality. Substituting the y and y' values into (*) and subtracting shows that $\alpha_j = 0$. Thus, the coefficients of every edge not in K are zero in equation (*).

2. Claim: $\alpha_i = \alpha_j$ if $i, j \in \bar{Q}$. Consider solution (1) and assume that edge k is not saturated. We can increase y_k by 1 and decrease y_j by 1 for any $q+1 \leq j < k$. Thus $\alpha_i = \alpha_j$ if $i, j \in \bar{Q}$.

3. Claim: $\alpha_i = \alpha_j$ for all $i, j \in Q$. Consider solution (2) and assume that edge q is not saturated. We can increase y_q by 1 and decrease y_j by 1 for any $j < q$. Thus $\alpha_i = \alpha_j$ for all $i, j \in Q$.

4. Claim $(r_Q + 1)\alpha_i = r_Q \alpha_j$, if $i \in Q$, $j \in \bar{Q}$.

   If we decrease $y_{\bar{Q}}$ by r_Q to 0 and increase y_Q by $r_Q + 1$ in the solution (1), we obtain the solution (2). Since all the coefficients in Q are the same, and all the coefficients in $\bar{Q}$ are the same, we then have $(r_Q + 1)\alpha_i = r_Q \alpha_j$, whenever $i \in Q$, $j \in \bar{Q}$. ■

Example 2.1 continued. Consider again the example shown in Figure 2.1. Let K be the cutset defined by the edges incident to node 6 and let Q be defined by edges $\{1,6\}$, $\{2,6\}$, $\{3,6\}$ and $\{4,6\}$. Since $r_Q = 1$, $\lfloor D_Q/(q-1) \rfloor = 11$ and $A = D_K/(k-1) = 39/4$, the data satisfies conditions (ii), (iii) and (iv) in Theorem 3.2. After we remove the cutset K, one subnetwork contains a single node 6 and no edges, and thus is restorable. The other subnetwork contains nodes 1, 2, 3, 4 and 5. It is also restorable because the edges form a cycle. Therefore, the corresponding Q-subset inequality $y_{16} + y_{26} + y_{36} + y_{46} + 2y_{56} \geq 12$ is a facet.

We might note that condition (i) in Theorem 3.2 is sufficient, but not necessary. In Section 5, we show that inequalities (3.2) are facet defining for parallel path problems that satisfy conditions (ii), (iii) and (iv), even when the two subnetworks after the removal of any cutset need not be restorable.

As a special case when $r_Q = 1$, inequality (3.2) becomes

$$y_Q + 2y_Q \geq \lceil D_Q/(q-1) \rceil . \tag{3.4}$$

It is easy to provide an alternative derivation of this inequality as follows. We know that

$$(q-1)y_Q + qy_{\bar{Q}} \geq D_Q,$$

that is,

$$y_Q + \frac{q}{q-1} y_{\bar{Q}} \geq \frac{D_Q}{q-1}.$$

Since $\frac{q}{q-1} \leq 2$, we have

$$y_Q + 2y_Q \geq D_Q/(q-1).$$

Since the lefthand side of this expression is integer, we can round up the righthand side and obtain (3.4). We then conclude that (3.4) is always valid for any subset Q of the cutset K in the network restoration problem.

The next theorem proves that the cutset capacity inequality is facet defining under certain conditions.

Theorem 3.3. *Given a cutset K containing edge e, the cutset capacity inequality*

$$\sum_{j \in K \setminus \{e\}} y_j \geq d_e \tag{3.5}$$

is facet defining in the space of the y variables if the two subnetworks separated by the cutset K are restorable.

Proof. We first construct feasible solutions satisfying (3.5) as an equality and then use the interchange argument as in the proof of Theorem 3.2 to prove the result.

We assign a spare capacity equal the largest demand in the network to all edges in the two subnetworks as well as to edge e. We select an arbitrary edge i from set $K\setminus\{e\}$ and assign d_e units of spare capacity to it. The resulting network contains sufficient spare capacity to restore all edges. Moreover, the solution satisfies (3.5) as an equality.

Since we can always obtain a new solution by increasing y_j by 1 for $j \notin K\backslash\{e\}$ while keeping other variables unchanged, the interchange arguments show that the coefficients of every edge not in $K\backslash\{e\}$ are zero. Since we select edge $i \in K\backslash\{e\}$ arbitrarily, the interchange argument shows that the coefficients of edges in $K\backslash\{e\}$ are the same. ■

4. Polyhedral Properties of a Special Case

4.1. Parallel path network restoration problem

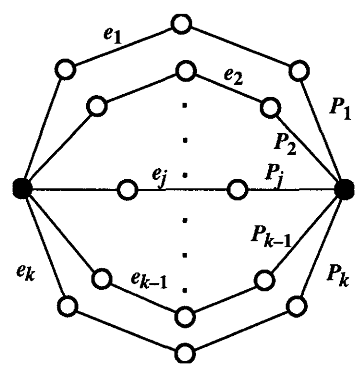

Figure 4.1: Parallel path network

In this section, we describe a special network, the *parallel path network*, as shown in Figure 4.1, with k parallel paths joining two distinguished nodes. This problem is of interest not only because this kind of network might occur in practice, but also because it provides insight into the polyhedral structure of a generic cut in the network. That is, suppose we separate the nodes into two groups S and T and consider the problem of determining how much restoration capacity we need on the cut they define (that is, the edges between these nodes sets) to restore the edge flows on the cut. This situation would, for example, arise if we simply were not concerned about the restoration capacity of edges within the node sets S and T. It also arises as a subset of the general restoration problem. If we aggregate the node sets S and T into two nodes, then the problem would be one with two nodes and parallel edges connecting those nodes. A number of studies of capacitated network design problems in other contexts have shown that valid inequalities based upon such cuts can be very effective in improving linear programming formulations.

Consider any path P_j for the situation of parallel paths shown in Figure 4.1, and suppose that the demand on edge e_j is as large as the demand of any edge on the path P_j. Note that

(i) the restoration for any edge e on the path P_j must flow though all the other edges on that path; and

(ii) if in restoring any edge $e \notin P_j$, we send flow through edge e_j, that flow must pass through every other edge of P_j.

These observations imply that for any edge $e \neq e_j$ of P_j, $y_e \geq \max\{y_{e_j}, d_{e_j}\}$. We say that an edge $e \neq e_j$ of P_j is *tight* if $y_e = \max\{y_{e_j}, d_{e_j}\}$. In any extreme point solution y to the problem, all edges must be tight. Since if some edge e is not tight, then we can either add or subtract an amount of capacity from y_e and the two resulting solutions y^1 and y^2 will be feasible; but since $y = (1/2)y^1 + (1/2)y^2$, y cannot be an extreme point. Therefore, $y_e = \max\{y_{e_j}, d_{e_j}\}$ for any edge $e \neq e_j$ of P_j.

Since we are assuming that d_{e_j} is an integer, this result shows that in any extreme point solution y_e is fractional if and only if y_{e_j} is fractional. This same conclusion applies to each path and shows that we can determine the restoration capacity for each edge by knowing the restoration capacity on the largest demand edge for each path.

This observation suggests that we contract all but one edge (an edge with the largest demand) on each parallel path and so consider a situation with two nodes and k parallel edges as shown in Figure 4.2. In the following sections, we investigate this two node parallel edge network in detail. We then indicate how to extend results for this special case to the parallel path problem.

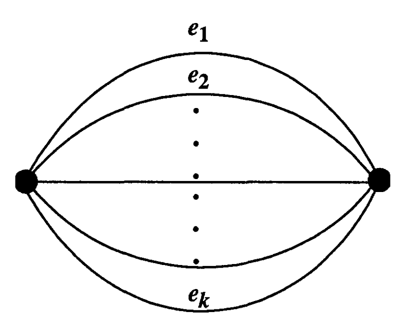

Figure 4.2: Two-node parallel edge network

4.2. Optimal solutions to the underlying linear and integer programs

For the two node parallel edge network in Figure 4.2, we enumerate the cutset capacity constraints for each edge across the single cut and obtain the following integer programming formulation in the space of y variables. For instance, the cutset capacity constraint for edge 1 states that the total spare capacity on edges 2 to k must be at least the demand d_1 on edge 1 so that the spare capacity network can reroute d_1 units of flow from one node to the other if edge 1 fails.

$$
\begin{array}{llll}
 & y_2 & +y_3 + \cdots + y_{k-1} + y_k & \geq d_1 \\
y_1 & & +y_3 + \cdots + y_{k-1} + y_k & \geq d_2 \\
\vdots & & & \vdots \\
y_1 & +y_2 & +y_3 + \cdots + y_{k-1} & \geq d_k \\
y_i & \geq 0 & \text{and } y_i \text{ integer for all } i & = 1, ..., k.
\end{array}
\tag{4.1}
$$

We start by studying the linear relaxation of system (4.1). Therefore, we relax the integrality constraints on the variable y_i. As before, we define $Y = y_1 + y_2 + ... + y_k$ as the total allocated spare capacity of any given solution y and subtract this defining equation from the cutset capacity inequalities in system (4.1), obtaining the following alternate linear relaxation formulation.

$$
\begin{array}{l}
y_1 + y_2 + ... + y_k = Y \\
0 \leq y_j \leq Y - d_j \text{ for all } 1 \leq j \leq k.
\end{array}
\tag{4.2}
$$

Since (4.2) contains the constraint $y_j \leq Y - d_j$ for all $1 \leq j \leq k$, feasibility also requires that $Y \geq d^* = \max_{1 \leq j \leq k} d_j$. Note that by adding the inequalities $y_j \leq Y - d_j$ we obtain the feasibility condition $Y \leq kY - D_k$ or $Y \geq A \equiv D_k/(k-1)$. Therefore, any feasible solution to the problem must satisfy the condition $Y \geq \max\{A, d^*\}$. In particular, if $Y = A$, then $y_j = A - d_j$ for all j and so each inequality is tight. Moreover, the solution allocates capacity on each edge if $A > d^*$.

We recall that the edge j is *saturated* if $y_j = Y - d_j$, or equivalently, the jth cutset capacity constraint R_j is binding. The following lemma indicates that we must saturate edges in increasing order of edge costs in the optimal linear programming solutions.

Lemma 4.1. *In the optimal linear programming solution y of system (4.1), suppose that $Q = \{1 \leq i \leq q : y_i > 0\}$ and that $c_i < c_j$ for some $i, j \in Q$. Then edge i is saturated, or equivalently, the cutset constraint R_i for the ith edge is binding.*

Proof. Suppose R_i is not binding. We can increase y_i by ε and decrease y_j by ε. The new solution is feasible and cheaper. ■

Properties of the value function $v(Y)$

As a function of the total allocated spare capacity Y, the linear relaxation of (4.2) is easy to analyze. Let $v(Y)$ equal the optimal value of the linear programming problem (4.2), or equivalently, (4.1), as a function of Y.

For any given value of Y and nonnegative cost vector c, to solve the linear relaxation of system (4.2) and determine the value of $v(Y)$, we order the edges so that $c_1 \leq c_2 \leq ... \leq c_k$. For $1 \leq j \leq k$, let $D_j = d_1 + d_2 + ... + d_j$. Let $D_0 = 0$. Then for some q, we allocate the installed capacity Y on the edges 1,2, ..., $q+1$, saturating the edges $j = 1, 2, ..., q$ (that is, set $y_j = Y - d_j$) and then allocate the remaining positive capacity (which is no more than $Y - d_{q+1}$) on edge $q+1$. Therefore,

$$y_{q+1} = Y - (y_1 + y_2 + ... + y_q) = Y - qY + D_q = D_q - (q-1)Y.$$

(Note that $y_{q+1} = Y - d_{q+1}$ is a possibility.)

As we have just seen in the chosen linear programming solution corresponding to the total allocation Y,

$$\begin{aligned} &y_j = Y - d_j \text{ for } j = 1, 2, ..., q \\ &0 < y_{q+1} \leq Y - d_{q+1}. \end{aligned}$$

Suppose $0 < y_{q+1} < Y - d_{q+1}$. If we alter Y by an amount ε, then in the chosen optimal linear programming solution (there might be alternative solutions), we alter the allocated capacity on edges 1 to q by an amount ε and alter the allocated capacity on edge q by $-(q-1)\varepsilon$. Therefore, in the interval $0 < y_{q+1} < Y - d_{q+1}$, the value function v is differentiable with a derivative $v(Y)$ given by

$$v^{'}(Y) = c_1 + c_2 + ... + c_q - (q-1)c_{q+1}.$$

Observe that since $y_{q+1} = D_q - (q-1)Y$, we can restate the condition $0 < y_{q+1} \leq Y - d_{q+1}$ as

$$\frac{D_{q+1}}{q} \leq Y < \frac{D_q}{q-1}. \tag{4.3}$$

Moreover, note that since $y_{q+1} = D_q - (q-1)Y$, within the interval $[D_{q+1}/q, D_q/(q-1))$, as Y increases, y_{q+1} decreases and y_j increases for all $1 \leq j \leq q$. At $Y = D_{q+1}/q$, we have $y_{q+1} = Y - d_{q+1}$ and all the $(q+1)$ edges are saturated.

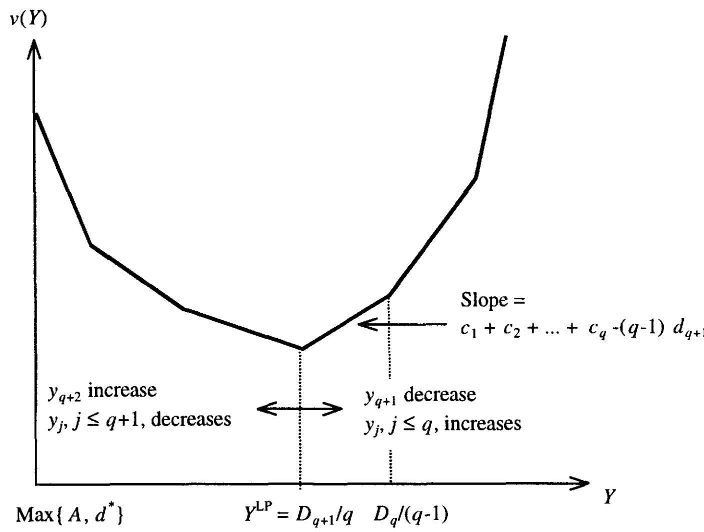

Figure 4.3: Solving the linear programming

Note further that since

$$\frac{D_{q+1}}{q} = \frac{D_q + d_{q+1}}{q} = \left(\frac{D_q}{q-1}\right)\left(\frac{q-1}{q}\right) + \left(\frac{d_{q+1}}{q}\right), \tag{4.4}$$

D_{q+1}/q is a convex combination, with weights $(q-1)/q$ and $1/q$, of $D_q/(q-1)$ and d_{q+1}. Therefore, if $d_{q+1} \le D_q/(q-1)$, then $D_{q+1}/q \le D_q/(q-1)$ and so the interval $[D_{q+1}/q, D_q/(q-1))$ is nonempty. Consequently, if for $j = 1, 2, ..., q$, $d^* \le D_j/(j-1)$, then $d_{j+1} \le D_j/(j-1)$ and each of the intervals $[D_{j+1}/j, D_j/(j-1))$ is nonempty.

Figure 4.3 illustrates the nature of the function $v(Y)$. The Y axis begins at $Y = \max\{A, D^*\}$ so that the values of Y to the right of the axis all satisfy the feasibility condition. If we consider the function $v(Y)$ from right to left, as we decrease the value of Y, we are increasing the number of positive variables $y_1, y_2, ..., y_k$ in turn. To solve the problem, we increase $q+1$ from value 2 as long as the derivative $v'(Y) = c_1 + c_2 + ... + c_q - (q-1)c_{q+1}$ is positive. When the derivative $v'(Y) = c_1 + c_2 + ... + c_{q+1} - qc_{q+2}$ becomes nonnegative, we have solved the problem. We obtain alternative optimal linear programming solutions for Y when the derivative $v'(Y) = c_1 + c_2 + ... + c_{q+1} - qc_{q+2} = 0$. At each break point, we include the next edge into the positive solution set and all the edges in the positive solution set are saturated. For example, at the lowest point in Figure 4.3, edges $1, 2, ..., q+1$ belong to the positive solution set and

$y_i = Y - d_i$ for all $i = 1, 2, ..., q+1$, or equivalently $Y = D_{q+1}/q$. The quantity $c_1 + c_2 + ... + c_q - (q-1)c_{q+1}$ is the right derivative $v^+(Y)$ of v at the lowest point $Y = D_{q+1}/q$, and $c_1 + c_2 + ... + c_{q+1} - qc_{q+2}$ is the left derivative $v^-(Y)$ of v at the lowest point.

As we have seen, in the open interval $(D_{q+1}/q, D_q/(q-1))$,

$$v'(Y) \geq 0 \text{ if and only if } c_{q+1} \leq (c_1 + c_2 + ... + c_q)/(q-1)$$

and

$$v'(Y) > 0 \text{ if and only if } c_{q+1} < (c_1 + c_2 + ... + c_q)/(q-1).$$

If $c_{q+1} \leq (c_1 + c_2 + ... + c_q)/(q-1)$ and $d^* < D_q/(q-1)$, then we will say that variable y_{q+1}, or index $q+1$, is *admissible.* Otherwise, we say that it is *inadmissible.* If $c_{q+1} < (c_1 + c_2 + ... + c_q)/(q-1)$ and $d^* < D_q/(q-1)$, then we will say that variable y_{q+1}, or index $q+1$, is *strictly admissible.*

Note that when $q = k-1$, by definition of A, the first inequality in the expression (4.3) becomes simply $A \leq Y$.

$v(Y)$ is a piecewise linear and convex function, since the slopes are constant in each interval, and the slopes are nondecreasing from left to right. To see that the slopes are nondecreasing, we compare the quantity $c_1 + c_2 + ... + c_{q+1} - qc_{q+2}$ with $c_1 + c_2 + ... + c_q - (q-1)c_{q+1}$. Since $c_{q+2} \geq c_{q+1}$, the first slope is less than or equal to the second. Therefore, to solve the linear programming problem min $\{v(Y){:}Y \geq d^*\}$, we determine an optimal value Y^{LP} of Y by finding the smallest feasible point Y for which the righthand derivative $v^+(Y)$ of $v(Y)$ is nonnegative.

Recall that feasibility requires that $Y \geq d^*$. Our observations to this point show that it is always optimal to set $Y = d^*$ or $Y = D_{q+1}/q$ for some value of $q \geq 1$.

Algorithm LP: for Solving the Linear Relaxation of System (4.1)

1. Sort the edges so that $c_1 \leq c_2 \leq ... \leq c_k$. If $c_1 < 0$, then terminate: the objective function is unbounded from below over the feasible region.

2. Find the largest value of the index q satisfying the property that all indices $j = 1, 2, ..., q+1$ are strictly admissible, that is, satisfy the condition $c_{j+1} < (c_1 + c_2 + ... + c_j)/(j-1)$. Set $Y^{\text{LP}} = max\{D_{q+1}/q, d^*\}$.

3. If $Y^{\text{LP}} = D_{q+1}/q$, saturate edge j for $j = 1, 2, ..., q, q+1$, that is, set $y_j = Y^{\text{LP}} - d_j$. Set $y_j = 0$ for $j \geq q+2$. If $Y^{\text{LP}} = d^*$, saturate edges one at a time, say up to p edges, and allocate the remaining capacity on edge $p+1$. All other variables remain zero. We can show that $p \leq q$ since

$d^* \geq D_{q+1}/q$, it lies to the right of D_{q+1}/q in Figure 4.3. Thus this solution needs less than $q+1$ positive variables.

Examples

To illustrate the algorithm, we consider two examples. Example 4.1 in Table 4.1 has four strictly admissable edges $Q = \{1,2,3,4\}$ with $D_Q/3 = \frac{22}{3} > d^*$. Thus we set $Y^{\text{LP}} = \frac{22}{3}$. Example 4.2 in Table 4.2 has three strictly admissable edges $Q = \{1,2,3\}$ with $D_Q/3 = \frac{22}{2} < d^*$. Thus we set $Y^{\text{LP}} = d^* = 12$.

edge	1	2	3	4	5	6	Y^{LP}	total cost
cost	1	1	1	1	2	100	–	–
demand	5	5	6	6	6	7	–	–
admissable	Y	Y	Y	Y	N	N	–	–
solution	$\frac{7}{3}$	$\frac{7}{3}$	$\frac{4}{3}$	$\frac{4}{3}$	0	0	$\frac{22}{3}$	$\frac{22}{3}$

Table 4.1: Linear programming solution for Example 4.1

edge	1	2	3	4	5	6	Y^{LP}	total cost
cost	1	1	1	2	2	2	–	–
demand	8	12	2	10	10	3	–	–
admissable	Y	Y	Y	N	N	N	–	–
solution	4	0	8	0	0	0	12	12

Table 4.2: Linear programming solution for Example 4.2

Note that the fractional solution we obtained for the example in Table 4.1 violates the Q-subset inequalities we discussed in Section 3. If we let $Q = \{1,2,3,4\}$, $D_Q = 22$ and $r_Q = 1$, we can cut away this fractional solution by adding the following inequality

$$y_1 + y_2 + y_3 + y_4 + 2y_5 + 2y_6 \geq 8.$$

It is worthwhile to make several observations at this point based upon the solution to the linear program.

Observation 1.

Since, by assumption, each demand d_j is integer, whenever Y is integer, the procedure LP will also provide an integer solution to the problem.

This observation implies the following result.

Proposition 4.2. *The linear programming problem (4.2) has an integer optimal solution y if and only if in some optimal solution Y is integer.*

Proof. Clearly, if the problem has an integer solution y, then since each component of y is integer, Y is integer. If Y is integer in some optimal solution with an associated vector y, then with Y fixed at this value, the vector y^{LP} in the linear programming solution determined by the method we have just described is integer. Since $cy^{\mathrm{LP}} = cy$, the integer vector (y^{LP}, Y) solves the linear program (4.2). ■

Observation 2.

Given any $q+1$ variables for the problem, which by renumbering, we assume are variables $1, 2, ..., q+1$, if we set $c_j = 1$ for $j = 1, 2, ..., q+1$ and $c_j = 3$ for $j \geq q+2$. The variables 1, 2, ..., $q+1$ will be strictly admissible and the variable $q+2$ will be inadmissible. Therefore, if $D_{q+1}/q \geq d^*$, the solution to the linear program found by the algorithm LP with $Y^{\mathrm{LP}} = D_{q+1}/q$ will be the unique optimal solution to the problem. Therefore, this solution is an extreme point of the polyhedron defined by the constraints. Since a feasible point to any polyhedron is an extreme point if and only if it is the unique optimal solution for some choice of the objective function, we have established the following result.

Proposition 4.3. *Each break point in Figure 4.3 with $Y = D_{q+1}/q$ for some subset of $q+1$ variables corresponds to an extreme point of the linear program (4.2).*

Observation 3.

Expression (4.4) provides us with an easy $\mathrm{O}(k)$ procedure for computing the quantities D_{q+1}/q. Similarly, the expression

$$\frac{c_1 + c_2 + \cdots c_{q+1}}{q} = \left(\frac{c_1 + c_2 + \cdots c_q}{q-1}\right)\left(\frac{q-1}{q}\right) + \left(\frac{c_{q+1}}{q}\right)$$

provides us with an easy $\mathrm{O}(k)$ procedure for computing the quantities required to determine which variables are admissible. Therefore, the algorithm LP requires $\mathrm{O}(k)$ computations.

Observation 4.

> Since the value function $v(Y)$ is convex, whenever the solution Y^{LP} provided by the algorithm LP is fractional, to determine an optimal integer value of Y, we simply need to round up and down and choose the lower cost solution. Observations 2 and 3 show that we will have solved the integer programming problem in $\mathrm{O}(k)$ computations as well.

Algorithm IP: for Solving the Integer Programming Version of System (4.1)

1. Let Y^{LP} be the optimal solution to the linear programming version of the problem determined by the algorithm LP.
2. Choose Y^{IP} as the $\operatorname{argmin}\{v(Y) : Y = \lfloor Y^{\mathrm{LP}} \rfloor \text{ or } Y = \lceil Y^{\mathrm{LP}} \rceil\}$.
3. Saturate edges one at a time, set $y_j = Y^{\mathrm{IP}} - d_j$, say up to p edges, and allocate the remaining capacity on edge $p+1$. All other variables remain zero.

Example 4.1 continued. Consider the fractional linear programming solution we obtained for the example in Table 4.1. If we set $Y = \left\lfloor \frac{22}{3} \right\rfloor = 7$, the solution is $y = \{2, 2, 1, 1, 1, 0\}$ with a total cost of 8. If we set $Y = \left\lceil \frac{22}{3} \right\rceil = 8$, the solution is $y = \{3, 3, 2, 0, 0, 0\}$ with a total cost of 8. In this case, we have multiple optimal integer solutions.

5. The Integer Programming Convex Hull

We have shown that the parallel path network restoration problem is polynomially solvable. This result suggests that we might be able to completely characterize its convex hull by identifying all of its facets. In this section, we confirm this conjecture by completely characterizing the convex hull of the feasible solutions to the two-node parallel edge network problem assuming nonnegative integer demands. We then extend the convex hull results for the parallel path network. This result suggests that the cutset capacity inequalities and Q-subset inequalities might be valuable (are "strong" inequalities) for solving the general restoration problem.

We first state a version of Theorem 3.2 which shows that we can eliminate the condition (i) for the parallel path problem. Indeed, for the parallel path problem, the subnetworks are trees and so are not restorable.

Corollary 5.1. *Let K be the cutset consisting of the edges with the largest demand on each parallel path and suppose Q is a proper subset of the cutset K. The Q-subset inequality (3.2) defines a facet for the parallel path problem in the space of the capacity variables, if*

(i) $r_Q < (q-1)$;

(ii) $d_e \leq \lfloor D_Q/(q-1) \rfloor$ *for all edges* $e \in K$; *and*

(iii) $A < \lfloor D_Q/(q-1) \rfloor$.

Proof. Suppose we contract all of the edges except the edges with the largest demand on each parallel path and obtain a two-node parallel edge network. To construct feasible solutions satisfying (3.2) as an equality, we use the same method described in the proof of Theorem 3.2 to install capacity on the edges in K. Given any feasible solution to the two-node problem, it is easy to verify that the solution obtained by adding sufficiently large capacity (for example, the largest demand in the network) on the contracted edges is feasible in the original parallel path network. The interchange arguments are exactly the same as in the proof of Theorem 3.2. ■

Theorem 5.2. *In the two-node parallel edge network restoration problem with k parallel edges $K = \{1, 2, ..., k\}$, the following constraints completely describe the convex hull of feasible integer solutions.*

(i) The cutset capacity constraints $\sum_{j \in K \setminus \{e\}} y_j \geq d_e$ *for all edges* $e \in K$;

(ii) The Q-subset inequalities for all subsets Q of K (including K); and

(iii) The nonnegativity constraints.

Before we prove the theorem, we establish two lemmas.

Lemma 5.3. *Suppose y is the leftmost lowest point in Figure 4.3 (y could be the unique lowest point), $y_i > 0$ for all $1 \leq i \leq q$, $y_j = 0$ for all $j \geq q+1$, and $c_1 \leq c_2 \leq ... \leq c_k$ by reindexing if necessary. Then $c_{q+1} > c_q$.*

Proof. Since we saturate the edges in the order of increasing costs, it is obvious that $c_{q+1} \geq c_q$. Now suppose $c_{q+1} = c_q$.

Since edge q is admissable, we know that

$$c_q \leq \frac{c_1 + ... + c_{q-1}}{q-2}.$$

Therefore,

$$\begin{aligned}
c_{q+1} = c_q &\leq \frac{c_1 + ... + c_{q-1}}{q-2} \\
(q-2)c_q &\leq c_1 + ... + c_{q-1} \\
(q-1)c_q &\leq c_1 + ... + c_{q-1} + c_q \\
c_q &\leq \frac{c_1 + ... + c_q}{q-1} \\
c_{q+1} &\leq \frac{c_1 + ... + c_q}{q-1}.
\end{aligned}$$

Therefore, edge $q+1$ is admissable. If it is strictly admissable, then the current point $Y = D_q/(q-1)$ is not the lowest point. If the data satisfies the last inequality as an equality, then $Y = D_{q+1}/q$ is an alternate solution lies to the left of the current point. This result contradicts the assumption that the current point is the leftmost lowest point. ■

Lemma 5.4. *In the optimal integer programming solution y of system (4.1), suppose that $Q = \{1 \leq i \leq q : y_i > 0\}$ and that $c_i < c_j$ for some $i, j \in Q$. Then edge i is saturated, or equivalently, the cutset constraint R_i for the ith edge is binding.*

Proof. Suppose R_i is not binding. We can increase y_i by 1 and decrease y_j by 1. The new solution is feasible and cheaper. ■

Proof of Theorem 5.2.

To prove the convex hull result, we use the optimal inequality argument from the field of polyhedral combinatorics (see, for example, Magnanti and Wolsey [5]). Let w be any n-vector of weight coefficients and consider the optimization problem $\max\{wx : x \in X\}$ defined over a finite set X. Let $Q = \{a_j x \leq b_j$ for $j = 1, 2, ..., m\}$ be any bounded polyhedron that contains X. An *optimal inequality* is an inequality $a_j x \leq b_j$ of the polyhedron Q satisfying the condition that all optimal solutions to the problem $\max\{wx : x \in X\}$ lie on it for a given choice of w. If the polyhedron Q contains an optimal inequality for every choice of $w \neq 0$, then the polyhedron is the convex hull of X.

Let y be the optimal solution to the linear programming. We first sort the edges in increasing order of edge costs. We discuss different types of cost vectors as follows.

Case 1. The linear programming has an integer optimal solution.

If the linear programming has a unique optimal, then this point also is the unique optimal solution to the integer programming. Since this solution is an

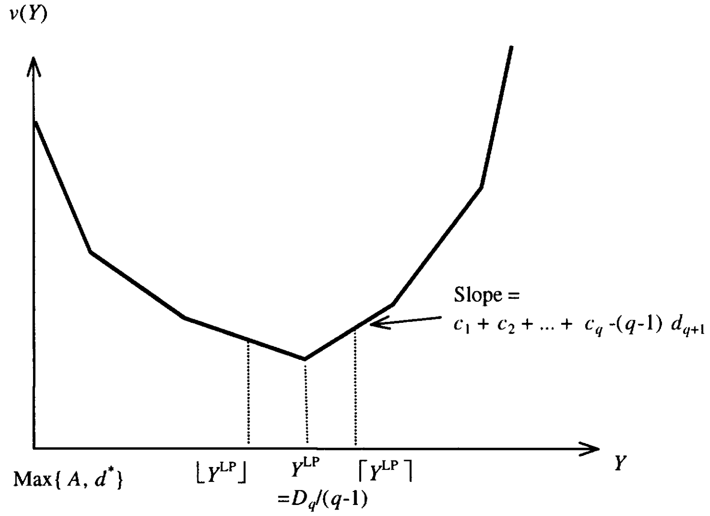

Figure 5.1: $v(Y)$ function with a unique lowest point

extreme point of the linear programming polyhedron, it lies on one of the inequalities in the linear programming system.

If the linear programming has multiple optimal solutions, then the cost vector c must be parallel to some face of the linear programming polyhedron. A face is a hyperplane defined by one or more constraints. Since some integer solution has the same cost, then all optimal integer points lie on the same face.

Case 2. The linear programming has no integer optimal solutions.
Subcase 2.1. For a given cost vector c, the function $v(Y)$ has a unique lowest point as in Figure 5.1.

Given a unique optimal linear programming solution y with $Y^{\mathrm{LP}} = D_q/(q-1)$. Let $Q = \{1 \leq j \leq q : y_j > 0\}$. To obtain an integer solution, we either set $Y^{\mathrm{IP}} = \lfloor Y^{\mathrm{LP}} \rfloor$ or $Y^{\mathrm{IP}} = \lceil Y^{\mathrm{LP}} \rceil$.

(1) Assume that $Y^{\mathrm{IP}} = \lfloor Y^{\mathrm{LP}} \rfloor$ and we saturate the first q edges, that is $y_j = \lfloor Y^{\mathrm{LP}} \rfloor - d_j$ for all $1 \leq j \leq q$. The total capacity installed on these edges is

$$y_Q = q \left\lfloor Y^{\mathrm{LP}} \right\rfloor - D_q = q \lfloor D_q(q-1) \rfloor - D_q = \lfloor D_q(q-1) \rfloor - r_Q.$$

Thus we must install the additional r_Q units of capacity on edges in the set $\bar{Q} = \{q+1, q+2, ..., k\}$. In Figure 5.1, we move from the lowest point to its left, thus increases the number of positive variables. There might be alternate ways to allocate the r_Q units of capacity in $\bar{Q}$ if we incur ties in the edge costs while sorting the edges, in which case we might have multiple solutions. We know from Lemma 5.3 that $c_{q+1} > c_i$ for all $i \in Q$. Thus by Lemma 5.4, R_i is binding for all $i \leq q$. Therefore, our assumption that the first q edges are saturated is valid. Furthermore, $y_Q = \lfloor D_q(q-1) \rfloor - r_Q$ and $y_{\bar{Q}} = r_Q$, so this type of solution satisfies the Q-subset inequality as an equality:

$$r_Q y_Q + (r_Q+1) y_{\bar{Q}} = r_Q(\left\lfloor \frac{D_Q}{q-1} \right\rfloor - r_Q) + (r_Q+1) r_Q = r_Q \left\lceil \frac{D_Q}{q-1} \right\rceil .$$

(2) Assume that $Y^{\mathrm{IP}} = \lceil Y^{\mathrm{LP}} \rceil$. In Figure 5.1, we move from the lowest point to its right, thus the number of positive variables does not increase. So we can allocate all the capacity on the edges in Q. There might be alternate ways to allocate the $Y^{\mathrm{IP}} = \lceil Y^{\mathrm{LP}} \rceil$ units of capacity in Q if we incur ties in the edge costs while sorting the edges, in which case we might have multiple solutions. Furthermore, $y_Q = \lceil D_q(q-1) \rceil$ and $y_{\bar{Q}} = 0$, so this type of solutions satisfy the Q-subset inequality as an equality:

$$r_Q y_Q + (r_Q+1) y_{\bar{Q}} = r_Q \left\lceil \frac{D_Q}{q-1} \right\rceil + (r_Q+1) 0 = r_Q \left\lceil \frac{D_Q}{q-1} \right\rceil .$$

It is possible that either $\lfloor Y^{\mathrm{LP}} \rfloor$ or $\lceil Y^{\mathrm{LP}} \rceil$ is the optimal integer programming solution or both are optimal. In both cases, the optimal solutions satisfy the inequality $r_Q y_Q + (r_Q+1) y_{\bar{Q}} \geq r_Q \lceil D_Q/(q-1) \rceil$ as an equality.

Subcase 2.2. For a given cost vector c, the function $v(Y)$ has multiple lowest points as in Figure 5.2.

By assumption, there are no integral Y values between the leftmost and rightmost lowest points of $v(Y)$. Suppose the leftmost point is $Y^{\mathrm{LP}} = D_q/(q-1)$. Let $Q = \{1 \leq j \leq q : y_j > 0\}$. To obtain an integer solution, we either set $Y^{\mathrm{IP}} = \lfloor Y^{\mathrm{LP}} \rfloor$ so that the integer solutions lie to the left of the leftmost point, or $Y^{\mathrm{IP}} = \lceil Y^{\mathrm{LP}} \rceil$ so that the integer solutions lie to the right of the rightmost point. If we round up or down any other lowest point on $v(Y)$, we will reach the same two points. The rest of the arguments are exactly the same as in Subcase 2.1. Therefore, all the optimal integer solutions satisfy the inequality $r_Q y_Q + (r_Q+1) y_{\bar{Q}} \geq r_Q \lceil D_Q/(q-1) \rceil$ as an equality, which concludes the proof of Theorem 5.2. ■

The next theorem presents a special result for the two-node parallel edge network problem with unit costs on each edge.

Theorem 5.5. *For a two-node parallel edge network with unit costs on each edge, the linear relaxation of the model (4.1) plus the cardinality-k cutset inequality has integer optimal solutions.*

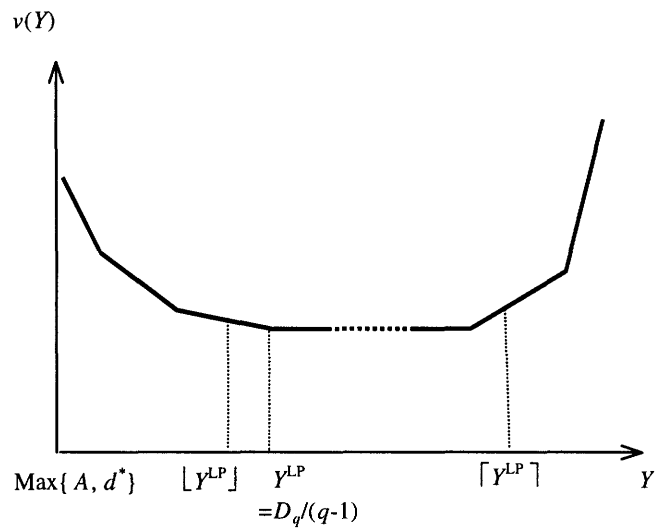

Figure 5.2: $v(Y)$ function with multiple lowest points

Proof. The cardinality-k cutset inequality states that $Y \geq \lceil A \rceil$. Define an augmented system composed of (4.1) plus the inequality $Y \geq \lceil A \rceil$. For situations with unit costs, we wish to minimize Y. Thus, the optimal solution to the linear program (4.1) always sets $Y^{\text{LP}} = \max\{A, d^*\}$. If Y^{LP} is integer, then $Y^{\text{LP}} \geq \lceil A \rceil$. By Proposition 4.2, the procedure LP provides an integer solution which is also optimal for the augmented system. Suppose Y^{LP} is fractional, then $Y^{\text{LP}} = A < \lceil A \rceil$ violates the inequality $Y \geq \lceil A \rceil$. Since we are minimizing Y, a solution to the augmented system is optimal when $Y = \lceil A \rceil$. We obtain a solution by saturating edges one at a time until we allocate all $\lceil A \rceil$ units of capacity. Since each demand d_j is integer, the optimal solution is integer. ■

We now extend the convex hull result in Theorem 5.2 to the parallel path network problem.

Theorem 5.6. *In the parallel path network restoration problem with k parallel paths $P_1, P_2, \ldots, P_k$, let e_j denote any edge on the jth path with the largest demand d_{e_j} and let $K = \{e_j,\ j = 1, \ldots, k\}$. For this problem, the following constraints completely describe the convex hull of feasible integer solutions.*

(i) The cutset capacity constraints $\sum_{u \in K \setminus \{e_j\}} y_u \geq d_{e_j}$ for all edges $e_j \in K$;

(ii) The Q-subset inequalities for all subsets Q of K (including K);

(iii) *The inequalities* $y_e \geq y_{e_j}$ *and* $y_e \geq d_{e_j}$ *for* $j = 1,2, \ldots, k$ *and every edge* $e \neq e_j \in P_j$; *and*

(iv) *The nonnegativity constraints.*

Proof. From the discussion in Section 4, we know that the system of constraints (i), (ii), (iii) and (iv) is a valid linear programming formulation for the parallel path network problem. From Theorem 5.2, we know that the constraints (i), (ii) and (iv) define a polyhedron with integral extreme points in the subspace of $y_{e_1}, \ldots, y_{e_k}$. We have shown that in any extreme point solution y for the parallel path problem, $y_e = \max\{y_{e_j}, d_{e_j}\}$ for any edge $e \neq e_j$ of P_j. Since we are assuming that d_{e_j} is an integer, y_e is integral if and only if y_{e_j} is integral. Therefore, the extreme points of the polyhedron defined by constraints (i), (ii), (iii) and (iv) are integral. ■

6. Conclusion

In this paper, we have developed a set of valid inequalities and facets for the general network restoration problem, and studied a special parallel path case of the problem. We completely characterized the convex hull of the set of feasible solutions of the parallel path problem. Several generalizations of the results are possible. For example, suppose we eliminate assumptions (4) and (5) in Section 2.1, that is, we permit existing spare capacity β_e on edge e and let $C > 1$. The cutset capacity constraints for edge $e \in K$ becomes

$$C \sum_{j \in K \setminus \{e\}} y_j + \sum_{j \in K \setminus \{e\}} \beta_j \geq d_e.$$

Inequality (3.1) becomes

$$C(q-1)y_Q + Cqy_{\bar{Q}} + (q-1)\beta_Q + q\beta_{\bar{Q}} \geq D_Q.$$

By rearranging the terms, we obtain the valid inequality

$$C(q-1)y_Q + Cqy_{\bar{Q}} \geq D_Q - (q-1)\beta_Q - q\beta_{\bar{Q}}.$$

By dividing both sides by C and rounding up the righthand side, we obtain

$$(q-1)y_Q + qy_{\bar{Q}} \geq \left\lceil (D_Q - (q-1)\beta_Q - q\beta_{\bar{Q}})/C \right\rceil.$$

Let $D'_Q = \left\lceil (D_Q - (q-1)\beta_Q - q\beta_{\bar{Q}})/C \right\rceil$ and $r'_Q = D'_Q \bmod (q-1)$. We then obtain the following modified Q-subset inequalities

$$r'_Q y_Q + (r'_Q + 1)y_{\bar{Q}} \geq r'_Q \left\lceil \frac{D'_Q}{q-1} \right\rceil.$$

Another generalized version of the Q-subset inequalities applies to situations when two facility types of capacities 1 and C are available. Wang [9] provides more details concerning this generalization. We have incorporated these facets in a cutting plane procedure for solving the network restoration problem. The computational results have been encouraging in the multiple facility case.

References

[1] A. Balakrishnan, T. Magnanti, J. Sokol, and Y. Wang. Modeling and solving the single facility line restoration problem. Working paper, Operations Research Center, MIT, Cambridge, MA, 1997.

[2] D. Bienstock and G. Muratore. Computational experiments with survivable network design problems. INFORMS Conference, Dallas, 1997.

[3] W. Grover, T. Billodeau, and B. Venables. Near optimal spare capacity planning in a mesh restorable network. *Proc. IEEE Globecom'91*, 2007–12, 1991.

[4] A. Lisser, R. Sarkissian, and J. Vial. Survivability in transmission telecommunication networks. Note Technique, CNET, Paris, France, 1995.

[5] T. Magnanti and L. Wolsey. Optimal trees. In M. Ball, T. Magnanti, C. Monma, and G. Nemhauser, eds., *Network Models. Handbooks in Operations Research and Management Science*, 503–615. Amsterdam, North-Holland, 1995.

[6] H. Sakauchi, Y. Nishimura, and S. Hasegawa. A self-healing network with an economical spare-channel assignment. *Globecom'90*, 438–43, 1990.

[7] M. Stoer and G. Dahl. A polyhedral approach to multicommodity survivable network design. *Numerische Mathematik*, 68(1):149–67, 1994.

[8] J. Veerasamy, S. Venkatesan, and J. Shah. Spare capacity assignment in telecom networks using path restoration. *Mascots'95*, 370–4, 1995.

[9] Y. Wang. *Modeling and Solving the Network Restoration Problem – Single and Multiple Facility Cases.* PhD thesis, Operations Research Center, MIT, Cambridge, MA, in progress.