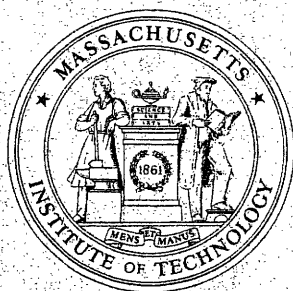


# OPERATIONS RESEARCH CENTER

working paper



**MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY**



A SURVEY OF NETWORK DESIGN PROBLEMS

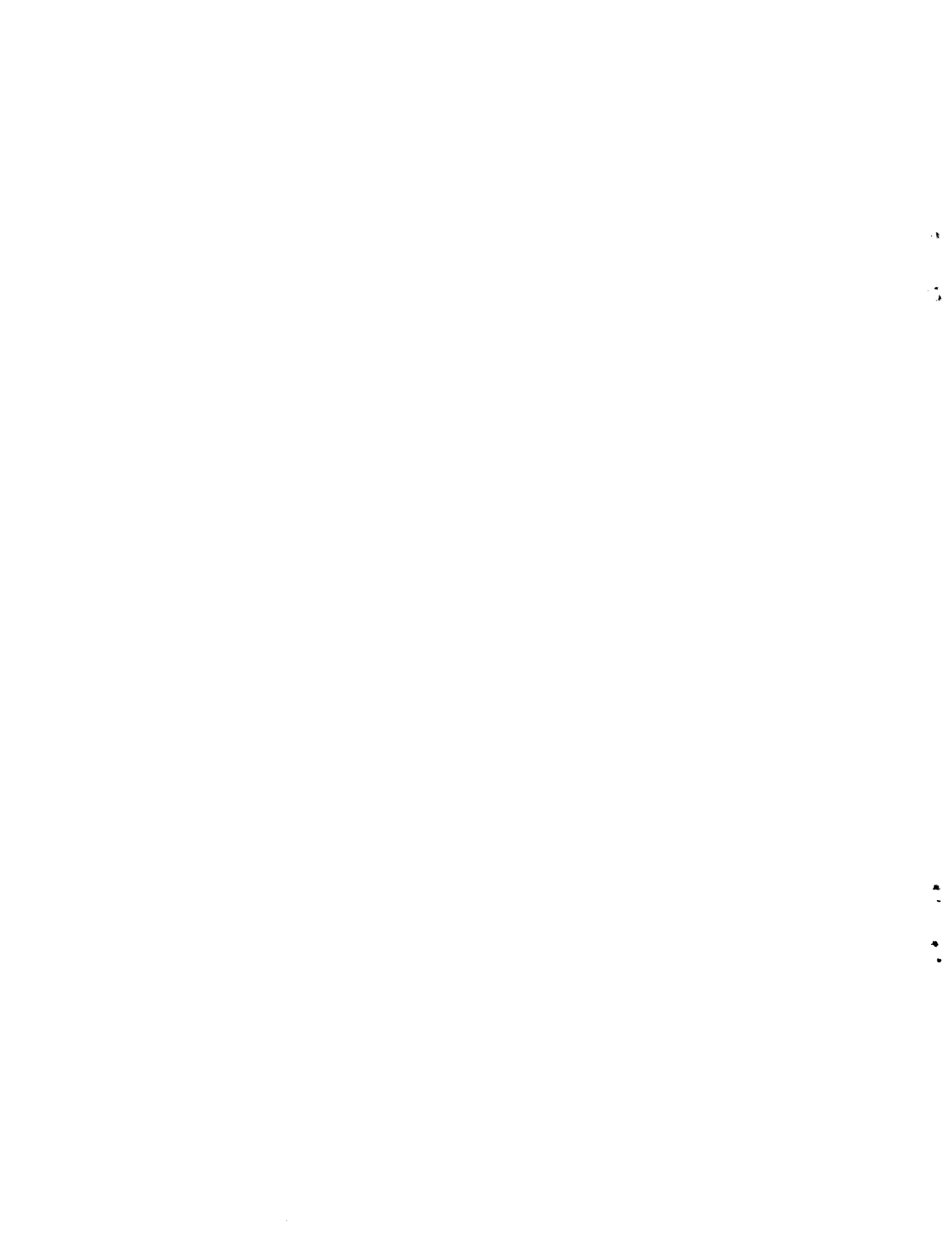
by

Richard T. Wong

OR 053-76

May 1976

Supported in part by the U.S. Department of Transportation under contract DOT-TSC-1058, Transportation Advanced Research Program (TARP).



ABSTRACT

This report is a survey of the design of various types of networks that frequently occur in the study of transportation and communication problems. The report contains a general framework which facilitates comparisons between problems. We discuss a large number of different network design problems and give computational experience for the various solution techniques.

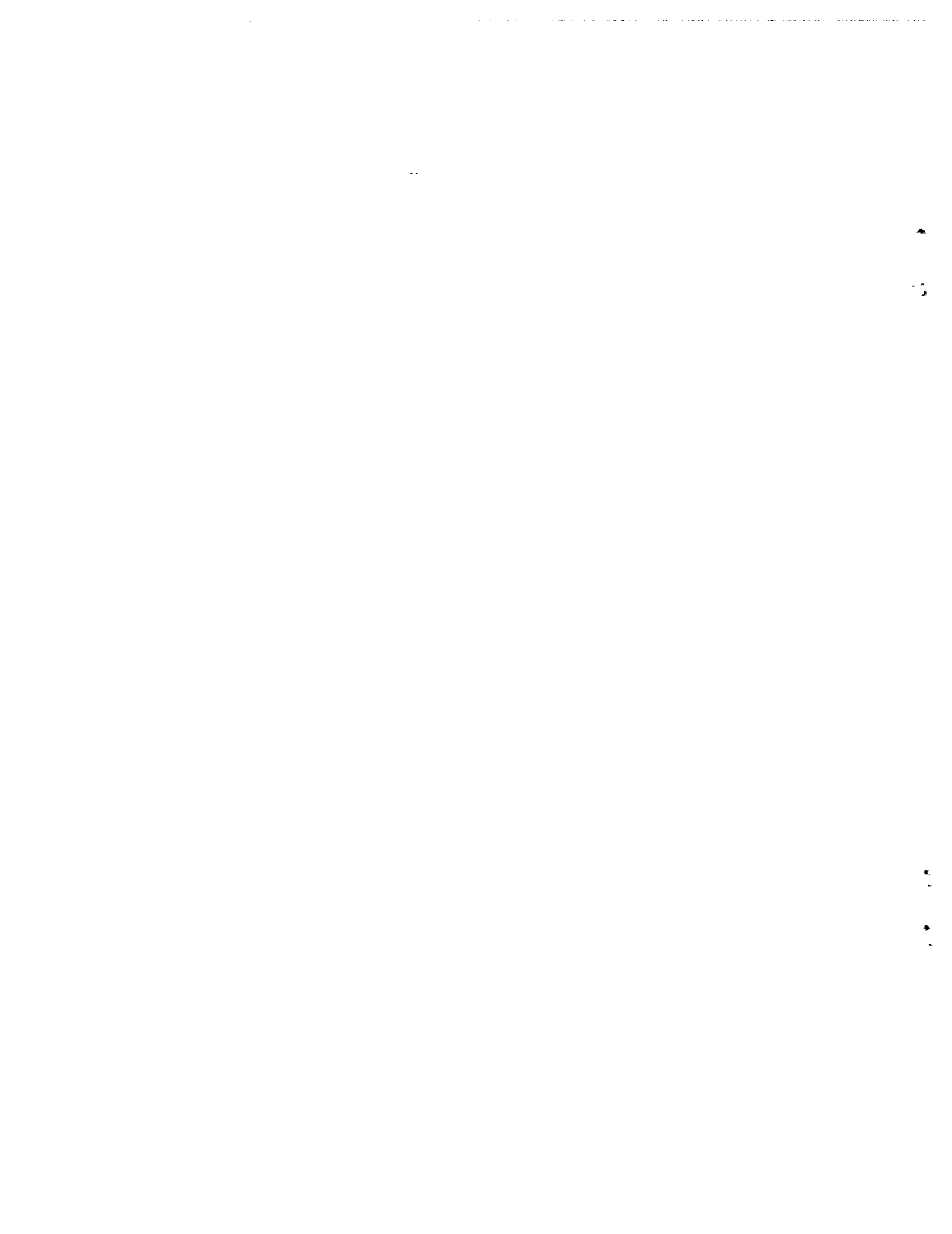


Table of Contents

	<u>Page</u>	
1.1	Introduction	4
1.2	Infinite Capacity Network Design Problems	9
1.2.1	Infinite Capacity Network Synthesis Problems	9
1.2.2	Fixed Charge Transportation Problem	18
1.2.3	Infinite Capacity Network Improvement Problems	20
1.3	Capacitated Network Improvement Problems	28
1.4	Network Design Problems with User Equilibrium Routing	38
1.5	Communication Network Synthesis Problems	48
1.6	Maximum Flow Improvement Problems	66
1.7	Final Remarks	71
1.8	Acknowledgements	79
	References	80
	Appendix 1	88

## 1.1 Introduction

In this survey we will discuss the design of various types of networks that frequently occur in the study of transportation and communications problems.

In particular, we will analyze networks that satisfy the following general description. First, we must specify a set of nodes and a set of arcs (directed or undirected) in the network. Each node will have a specified capacity which limits the total amount of flow that can pass through the node. Associated with each arc is a set of possible arc capacities. For instance, an arc capacity could be a binary variable taking on the values zero or some capacity  $C$ . Alternatively, the arc capacity could be a continuous variable assuming any value from zero to some upper bound  $C'$ . Note that setting an arc capacity to zero is equivalent to eliminating the arc from the network. In this survey we will often refer to increasing an arc capacity from zero as "adding" an arc or "constructing" an arc in the network.

In these network design problems we will also have a set of required flows that must be routed through the network. For example, there could be required flows between pairs of nodes. In most problems, as is the case for the above example, the required flows will be multi-commodity flows in the sense that there will be several types of flow to route through the network.



For each arc given in the design problem there is a cost function associated with setting the arc capacity variable at a particular level. We will refer to these costs as construction costs. For every arc in the network there is also a routing cost function that depends on the amount of flow routed through the arc. A very common routing cost function is one where the cost is proportional to the amount of flow traveling through the arc.

At this point we should note that there are two different flow routing policies that will be used in this survey. One possible routing policy is Wardrop's "Principle of Overall Minimization" [WAR1]. Utilizing this policy means that the flow routing will be done so that the sum of the routing costs for all of the required flows is minimized. Such a routing process is optimal from the viewpoint of the entire system.

In contrast, another possible routing policy uses Wardrop's "Principle of Equal Travel Times" [WAR1]. Under this routing principle, each unit of flow will seek to minimize its own origin to destination routing cost. An optimal traffic flow assignment according to this strategy has the property that no unit of flow can improve its routing cost by taking an alternative route between its origin and destination. We will refer to this type of routing as user equilibrium routing (in the literature this type of routing is sometimes referred to as descriptive flow assignment). Unless otherwise noted, all of the network problems discussed in this survey will use the system optimal routing strategy.

For network design problems that fit the general description given above, we shall discuss two types of design problems, network synthesis and network improvement problems. For a network synthesis problem, all the arc capacities are initially zero. That is, we start the design process without any part of the network constructed. For network improvement problems, we begin with a network configuration through which the required flows can already be routed. The problem is to add additional capacities to the network arcs in order to improve the performance of the network.

For either type of design problem, a solution consists of decisions on how to set the arc capacity levels so that all required flows can be routed through the network and that a specified objective function is minimized. There are several kinds of objective functions that are used to evaluate proposed network configurations. One consists of the sum of the routing and construction costs for all the arcs in the network. Another type of objective function consists of only the sum of the routing costs for the arcs but with the constraint that the sum of the construction costs for all the arcs does not exceed a given budget. We will describe other objective functions in the course of this survey as they are required.

We can now summarize the above description of network design problems in the following general framework:

(1.1.1) OBJECTIVE: minimize a given objective function

CONSTRAINTS:

- 1) ARC TYPE - here we specify whether the arcs in the network are directed and/or undirected
- 2) ARC CAPACITIES - here we describe the set of possible arc capacity levels
- 3) CONSTRUCTION COSTS - here we specify the set of arc construction cost functions
- 4) ROUTING COSTS - here we specify the set of arc routing cost functions
- 5) REQUIRED FLOWS - here we describe the set of required flows in the problem
- 6) SPECIAL CONSTRAINTS - here we discuss any other constraints of the problem; e.g., a budget constraint on the total construction costs.

Note that in our general framework we assume that a set of nodes has already been specified. Unless otherwise stated, we will also assume that all nodes have infinite flow capacity.

It should be easy to see that the general framework in (1.1.1) encompasses a large number of different design problems. In the following sections we will discuss various special cases of this general network synthesis and improvement problem.

Previous survey work in the area of network design problems includes reports by Schwartz [SCH1], Stairs [STA1], and Steenbrink [ST2, Chapter 4].

## 1.2 Infinite Capacity Network Design Problems

In this section we will discuss problems that have two common traits. First, the network arc capacities can either be zero or infinite. So, once an arc is added to the network, any amount of flow can be routed through it. Second, all arc routing costs are linear functions of the flow routed through the arc.

Note that with these two properties, the arc routing cost per unit of flow is constant and independent of the level of flow through the arc. Under this condition, both the system optimal and user equilibrium routing policies will produce flow assignments for the problems in this section that incur the same amount of routing cost. So, although we are assuming that the routing is done according to a system optimal policy, the network design results of this section are valid for either routing policy.

Now we will examine some of the various infinite capacity network design problems.

### 1.2.1 Infinite Capacity Network Synthesis Problems

Billheimer and Gray [BIL1] formulated the first type of infinite capacity network synthesis problem which we will present. We discuss this problem first because it contains as a special case a variety of combinatorial problems including the Steiner

tree problem on a graph [HAK3, DRE2], the simple plant location problem [MAN1, EFR1], the optimum communication spanning tree problem [HUT3], and the minimum spanning tree problem [KRU1]. Later in this survey, we will consider these special cases.

In terms of our general framework, we can describe Billheimer and Gray's problem in the following way:

(1.2.1.1) OBJECTIVE: minimize total routing and construction costs

CONSTRAINTS:

- 1) ARC TYPE - directed and/or undirected
- 2) ARC CAPACITIES - zero or infinite
- 3) CONSTRUCTION COSTS - a fixed cost for constructing an arc with infinite capacity
- 4) ROUTING COSTS - linear functions of the arc flows
- 5) REQUIRED FLOWS - there are required flows between all pairs of nodes in the network

Note that an infinite capacity arc represents an arc which is capable of carrying every possible flow in the network. So it is possible to replace an infinite capacity arc with an arc whose finite capacity is sufficiently large.

This particular problem is quite complex. The formulation of (1.2.1.1) as a mixed integer program results in a very large problem. For a network design with 50 nodes and 200 possible directed arcs, the corresponding mixed integer program will have 2,700 rows, 10,000 continuous variables and 200 binary variables. Because of the complexity of the problem, Billheimer and Gray propose a heuristic procedure for obtaining a solution.

The procedure begins with all possible arcs in the network constructed. Then the procedure applies two iterative algorithms. The first algorithm reduces the total cost (routing and construction) at each iteration by eliminating from the network the arc which will produce the largest cost reduction. The other algorithm reduces the total cost at each iteration by adding to the network the arc which will produce the largest cost reduction. Each algorithm continues to remove or add an arc until no further cost reduction can be obtained. Then the other algorithm is applied. These two algorithms are used repeatedly until a local optimum is reached. At this local optimum we cannot reduce the cost of the network configuration by the addition or deletion of a single arc.

The heuristic procedure has been tested on a problem with 68 nodes and 476 arcs. The method reached a local optimum after about 3 minutes of computation time on an IBM 360/67 computer. It is difficult to judge the quality of heuristic's solutions

since no satisfactory method is known for solving problems of that size.

It is interesting to see that special cases of (1.2.1.1) encompass a wide range of combinatorial problems. If all arc construction costs are set to zero, then the (1.2.1.1) becomes a series of shortest path problems [DRE1]. If all arc routing costs are set to zero, then the problem becomes either the minimum spanning tree problem [KRU1] or Steiner's tree problem on a graph (STPG) [HAK3, DRE2]. The problem will be a STPG when the required flows necessitate that there be a path between every pair of nodes in some subset of the nodes in the network. When this subset is the entire set of network nodes, (1.2.1.1) becomes the minimum spanning tree problem.

Since (1.2.1.1) contains the STPG as a special use, we can be sure that it is very difficult to solve. Karp [KAR1] has shown that the STPG belongs to the class of NP-complete problems (this class of problems is also referred to as P-complete problems and polynomially complete problems). This implies that the STPG is as difficult to solve as such combinatorial problems as the traveling salesman problem [BEL1], the maximum clique problem [HAR2] and the 0-1 integer programming problem (see [KAR1, KAR2] for a full discussion of the various NP-complete problems). In view of the lack of success in solving any of the above problems on a large scale, it appears unlikely that there is an efficient algorithm for the STPG or



or for (1.2.1.1). In fact, it can be shown that (1.2.1.1) belongs to the class of NP-complete problems. (This follows from the fact that the STPG is a special case of (1.2.1.1)).

If the arc construction costs are all equal and totally dominate the routing costs (i.e., the optimal network design must be a tree), then (1.2.1.1) becomes the optimum communication spanning tree problem defined by Hu [HUT3].

Another special case of Billheimer and Gray's problem is the fixed charge plant location problem [MAN1, EFR1]. The plant location problem is normally associated with the placement of facilities on the nodes of a graph. Efraymson and Ray [EFR1] describe it in the following way: "In its simplest form, plant location can be posed as a transportation problem with no constraint on the amount shipped from any source. However, there is a cost associated with each source (plant). This cost (called a fixed cost or fixed charge) is zero if nothing is shipped from the plant, i.e., the plant is 'closed'. It is positive and independent of the amount shipped if any shipment from the plant takes place, i.e., the plant is 'open'." However, it is possible to convert the plant location problem to a network synthesis problem. This can be done in the following way: add a special node to the plant location network. This node will be the source of all the flow required by the customer nodes. Also, add a set of special arcs leading from the special node to each potential plant site (see figure 1.2.1.1).

"SPECIAL"  
NODE

POTENTIAL  
PLANT LOCATIONS

CUSTOMERS

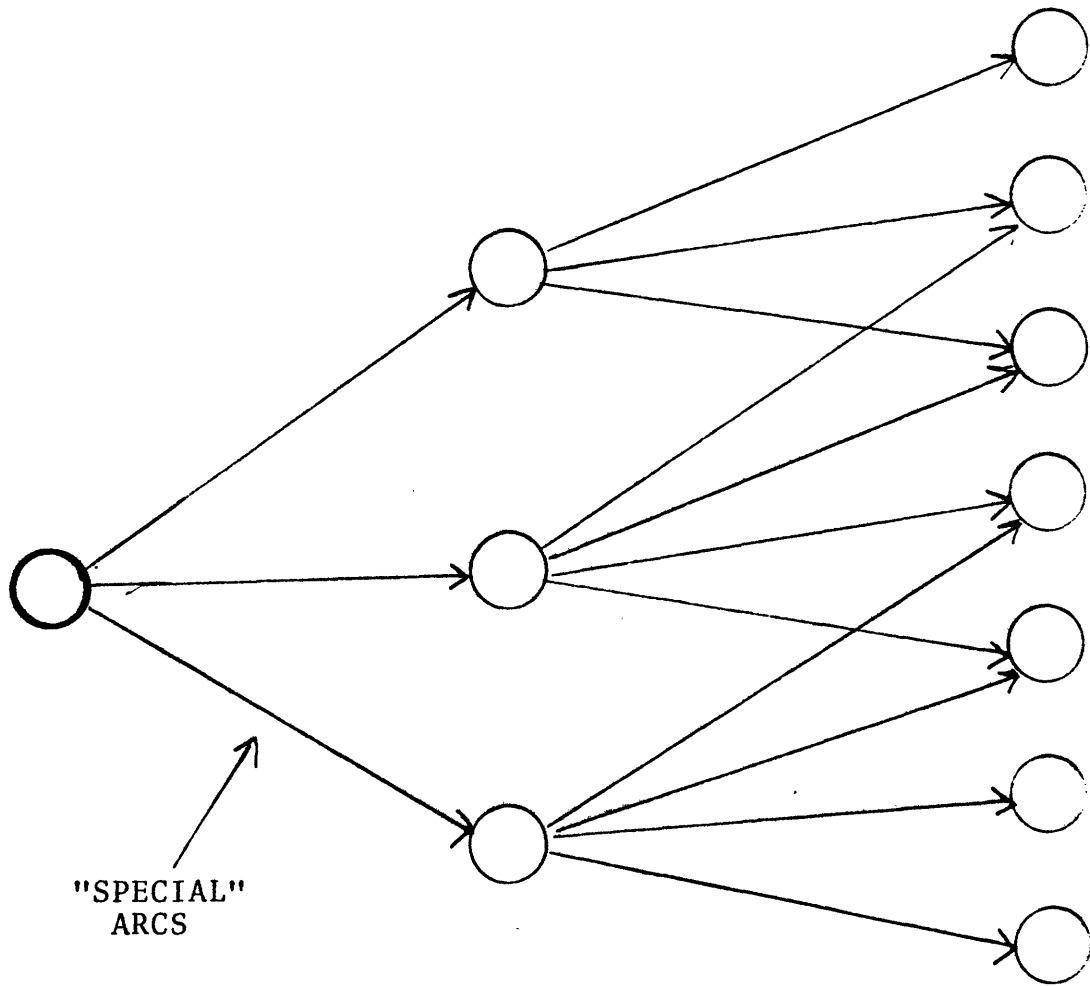


FIGURE 1.2.1.1

PLANT LOCATION AS AN ARC SYNTHESIS PROBLEM

A special arc connecting the factory to a plant site has a construction cost equal to the fixed charge associated with opening the site. These special arcs will have no routing costs. Arcs connecting plant locations with customers have no construction costs. However, they will have a routing cost equal to the transportation cost from the plant location to the customer. So now the corresponding synthesis problem is to design the minimum total cost (construction plus routing cost) network so that all the flow requirements between the special node and the customers are satisfied. Thus, the fixed charge plant location problem is a special case of (1.2.1.1) where arcs either have non-zero construction costs or routing costs but not both. Also, in this special case, the required flows are a single commodity, whereas in the general case of (1.2.1.1), the required flows are multi-commodity.

Viewing the fixed charge plant location problem as a special case (1.2.1.1) gives us additional insight into the network synthesis problem. For instance, Billheimer and Gray give some methods for partially characterizing the optimal network configuration. In particular, they give a procedure for identifying arcs which definitely must or must not be constructed in the optimal solution. Efraymson and Ray give a procedure for determining if a plant must or must not be opened in the optimal solution. By comparing the two procedures, we can see that

Billheimer and Gray's techniques are a generalization of Efroymsen and Ray's techniques.

Note that by using a similar construction as illustrated in figure 1.2.1, we can show that many other different facility location problems are special cases of the network design problems specified by our general framework. For example, if we have a capacitated plant location problem, the node capacity constraint can be represented by a capacity constraint on one of the "special" arcs added to the network. Since there has been so much work done in the area of facility location problems (see [REV1, FR1]), it may be possible to generalize some of the techniques developed in order to apply them to network design problems. The rules given by Billheimer and Gray and Efroymsen and Ray are one example of such a generalization.

Scott [SC01, SC02, SC03] discusses another network synthesis problem that is closely related to (1.2.1.1). The problem, called the optimal network problem by Scott, has the following description:

(1.2.1.2) OBJECTIVE: minimize total routing costs

CONSTRAINTS: same as (1.2.1.1) with an additional constraint:

6) SPECIAL CONSTRAINT

a) Construction Budget - total construction costs cannot exceed a given budget.

Scott proposes several methods for solving the optimal network problem. The first approach is to formulate the problem as a mixed integer programming problem [SC01]. Since the formulations for this problem are generally quite large, Scott does not develop this approach any further. The second method proposed by Scott [SC02] is a branch and backtrack procedure [GOU1]. The method was tested on a series of 26 problems each containing from 7 to 10 nodes with undirected arcs connecting all possible pairs of nodes. The total solution time ranged from under one minute to over one hour of IBM 360/65 computer time. Because of the excessive solution time required for this method, Scott [SC02] introduces a heuristic procedure which involves a series of arc exchanges, additions and deletions. The procedure is similar to Billheimer and Gray's heuristic method for (1.2.1.1). Scott tested the procedure on the group of problems described above. In all cases the solution time was less than one minute and all solutions obtained were within 3% of the global optimum.

Boyce et al. [BOY1] utilize a branch and bound algorithm to solve (1.2.1.2). They are able to solve problems with 10 nodes and 45 arcs in 3 to 400 seconds of IBM 360/75 computer time depending on the value of the given construction budget.

As was the case for Billheimer and Gray's problem, Scott's problem contains several interesting combinatorial problems as special cases. If all construction costs are set to zero, then Scott's problem reduces to a set of shortest path problems. If all construction costs are set to one, and if the construction budget is taken as  $(N-1)$ , where  $N$  is the number of nodes in the network, so that the optimal network must be a tree, then the problem reduces to Hu's optimum communication spanning tree problem. Using a construction similar to the one depicted in figure 1.2.1.1, it can be shown that the P-median problem [HAK1, HAK2] is also a special case of Scott's problem.

### 1.2.2 Fixed Charge Transportation Problem

The next network synthesis problem that we will discuss is the well known fixed charge transportation problem [BAL1, KUH1, SPI1, BAI1]. This problem arises when we consider a Hitchcock transportation problem [DAN1] with fixed charges added to the flow variables. Since the Hitchcock transportation problem can be formulated as a linear programming problem, the fixed charge transportation problem is a special case of the fixed charge problem that has been studied by Dantzig and Hirsch [DAN1].

In terms of our general framework, the problem has the following description:

(1.2.2.1) OBJECTIVE: minimize total construction and routing costs

CONSTRAINTS:

- 1) ARC TYPE - directed
- 2) ARC CAPACITIES - zero or infinite
- 3) CONSTRUCTION COSTS - a fixed cost for constructing an arc with infinite capacity
- 4) ROUTING COSTS - linear functions of the arc flows
- 5) REQUIRED FLOW - a single commodity that must be routed between the set of source nodes and the set of destination nodes (see below)
- 6) SPECIAL CONSTRAINT
  - a) Arc Restrictions - the set of nodes is divided into a set of source nodes and a set of destination nodes. Only arcs between source and destination nodes are allowed in the network.

So (1.2.2.1) is similar to (1.2.1.1) except that instead of having required flows between pairs of nodes, the fixed charge transportation problem has a required flow between two sets of nodes.

Many solution methods have been proposed for (1.2.2.1). For instance, Balinski [BAL1] and Kuhn and Baumol [KUH1] proposed heuristic solution methods. Spielberg [SPI1] suggested an exact solution method that solves a mixed integer programming formulation of (1.2.2.1) by using Benders' decomposition procedure [BEN1]. We will not discuss these methods here. The interested reader may consult Bair and Hefley's [BAI1] survey of the fixed charge transportation problem.

### 1.2.3 Infinite Capacity Network Improvement Problems

In this section we will discuss some network improvement problems that are closely related to the infinite capacity network synthesis problems described in the previous section.

Ridley [RID1] gives the following version of the infinite capacity network improvement problem:

(1.2.3.1) OBJECTIVE: minimize total routing cost (same as  
in (1.2.1.2))

CONSTRAINTS: same as (1.2.1.2) with the addition  
of the following constraints



6) SPECIAL CONSTRAINTS

- b) Initial Arc Capacities - same arc capacities are initially set at infinity. They constitute the initial unimproved network.
- c) Integer Values - all fixed charge construction costs and the given budget must be integer valued.

The main feature of this version is that there will generally be more than one candidate arc between a pair of nodes. Each arc will have a different routing cost function. So the optimal decision involves deciding not only whether to connect a pair of nodes with an arc but also deciding how "good" an arc to construct. Notice that by constructing another arc between a pair of nodes which initially has an arc connecting them, we can "upgrade" service between the two nodes.

It is possible to view Ridley's problem as a special case of Scott's optimal network problem where all the arcs in the initial network have zero construction cost. Thus, any method that solves Scott's problem theoretically will solve Ridley's problem as well.

Ridley gives a branch and bound procedure for solving (1.2.3.1). Suppose the construction budget is  $b$  units, where  $b$  is an integer. The procedure starts by increasing the budget

so that the best possible network (in terms of smallest routing costs) can be constructed. Then the procedure lowers the budget by one unit and finds the optimal set of arcs to construct subject to the new budget constraint. The above process is repeated until the budget is decreased to exactly  $b$  units. The optimal solution is the set of arcs chosen on the last iteration. Since a larger budget can never increase our routing costs, a lower bound on the routing costs of any iteration of the solution process is the routing cost of the previous iteration. Stairs [STAl] points out that this procedure is quite sensitive to the size of the budget used. A large budget would require a large number of iterations and a great amount of computation. So it does not appear that Ridley's method will be able to solve medium or large problems unless the construction budget is quite small. Stairs reports that Ridley's method has been used to solve problems that have up to a dozen nodes. She does not give any computation times for these problems.

Next we will consider a special case of (1.2.3.1) where there is only one required flow that must be routed. An equivalent statement of this problem is: suppose we have a network with  $V$  nodes. The "length" of every arc  $(i,j)$  can be decreased to any one of  $L_{ij}$  different values. Decreasing the length of an arc incurs some construction cost. Subject to a construction budget, find the optimal investment policy that achieves the

best improvement in the length of the shortest path between nodes 1 and V.

Goldman and Nemhauser [GOL1] give an interesting method for solving the above shortest path improvement problem. Given a network with the various possible levels of arc improvement, they show how to form a special expanded network in which a shortest path provides the optimal investment policy. The transformation they give can be described as follows:

- Let  $N$  = set of nodes in the original problem network
- $A$  = set of arcs in the original problem network
- $A(i,j)$  = set of non-negative integers which are the construction costs associated with decreasing the length of arc  $(i,j)$  to one of  $L_{ij}$  levels.
- $R$  = value of the construction budget
- $N^*$  = set of nodes in the enlarged network
- $A^*$  = set of arcs in the enlarged network
- $f_{ij}(u)$  = length of arc  $(i,j)$  after investing  $u$  units.

Note that  $f_{ij}(0)$  is the original length of arc  $(i,j)$ . Also,  $f_{ij}(u)$  is assumed to be a non-increasing function.

$$\begin{aligned} \text{Now } N^* &= \left\{ (i,u) : i \in N, 0 \leq u \leq R, u \text{ integer} \right\} \\ A^* &= \left\{ ((i,u),(j,v)) : (i,j) \in A, (v-u) \in A(i,j) \right\}. \end{aligned}$$

Finally, let  $d^*(x,y)$  be the length of arc  $(x,y)$  in the expanded network.

$$\text{Then } d^*((i,u),(j,v)) = f_{ij}(v-u).$$

Many of the nodes and arcs in the expanded network may be unnecessary. However, they are included in the above description to keep the notation uniform.

So, if we wish to improve the shortest path between nodes 1 and  $v$  in the original network, then the problem is to find the shortest path between nodes  $(1,0)$  and  $(v,R)$  in the expanded network. If arc  $((i,u),(j,v))$  is part of this shortest path in the expanded network, then the optimal improvement policy for the original network is to spend  $(v-u)$  units on arc  $(i,j)$ .

Figures 1.2.3.1 and 1.2.3.2 depict a small example of this network expansion procedure. Figure 1.2.3.1 shows the original network. Solid lines denote the original network arcs. Dotted lines denote the various levels at which an arc can be improved. The numbers beside the arcs are the arc lengths. The numbers placed within squares are the construction costs. Figure 1.2.3.2 shows the expanded network. For this example  $R$ , the construction budget equals 2. The problem is to improve the shortest path between nodes 1 and 3. So in the expanded network, the problem is to find the shortest path between nodes

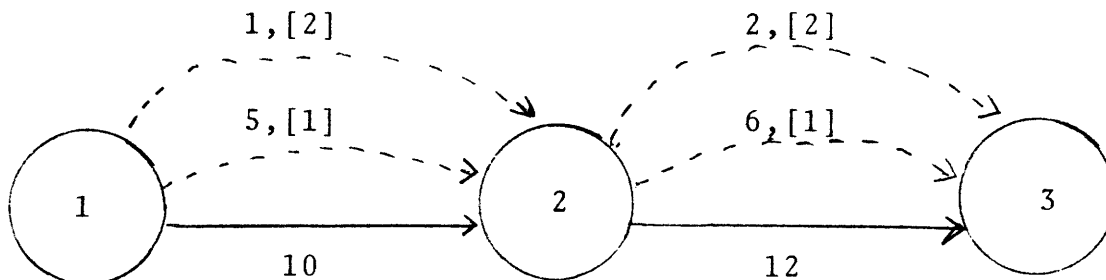


FIGURE 1.2.3.1

EXAMPLE OF NETWORK EXPANSION PROBLEM

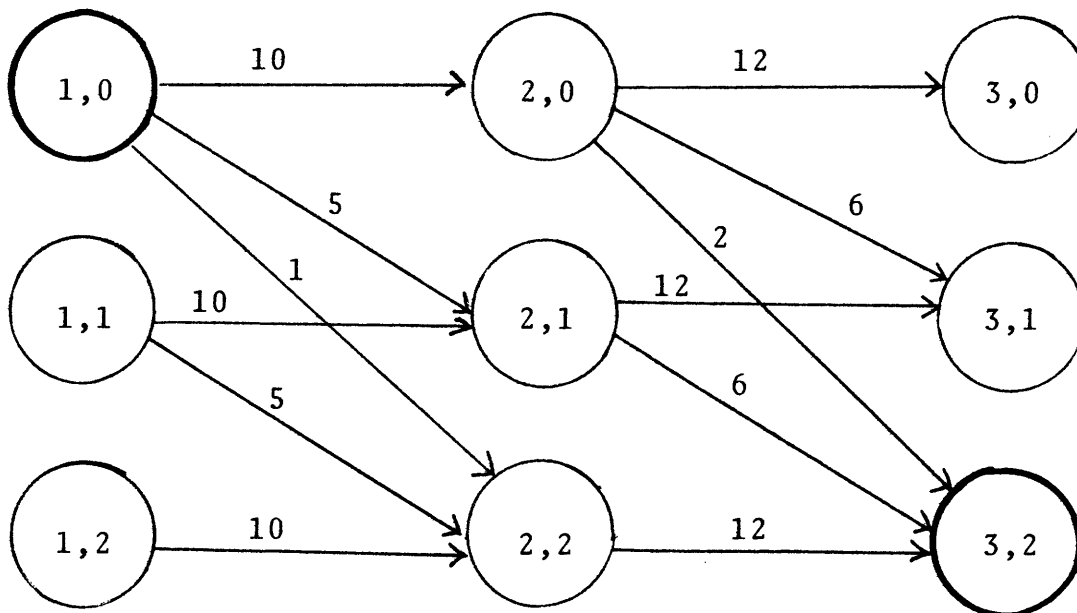


FIGURE 1.2.3.2

EXPANDED NETWORK REPRESENTATION OF FIGURE 1.2.3.1

(1,0) and (3,2). Note that nodes (1,1), (1,2), (3,0), and (3,1) are not necessary for the problem.

The expanded network can become quite large especially if  $R$  is large. So Goldman and Nemhauser show how to avoid the brute force application of a shortest path procedure to the expanded network. They adapt various shortest path algorithms to exploit the special structure of the expanded network.

Wollmer [WOL1] and Ridley [RID2] give efficient procedures for solving special cases of the shortest path improvement problem. However, it can be shown that they are just special cases of Goldman and Nemhauser's procedure.

Stairs [STA1] formulates a network improvement problem that is identical to Billheimer and Gray's infinite capacity network synthesis problem except for two constraints: first, an initial network which can already handle the required flows is given. Second, it is possible to close down an arc (set its capacity to zero) as well as to construct one. For an arc that could be closed down, the construction cost would be negative. This would represent the savings in costs that would occur if the arc were closed. Aside from these two differences, the objective function and the constraints for the two problems are the same.

Stairs suggests the use of an interactive computer program to solve her problem. Under her approach, the user chooses which arcs to open or close. The computer then computes the cost of the proposed solution. Utilizing this information, the

user modifies his proposed solution. By repeating this interaction between user and computer it is hoped that a reasonable solution will be found. Stairs states that the interactive procedure could be a useful tool for evaluating medium-sized networks. A traffic network designer could use it to gain some insight into the operation of a given network. The procedure could be used for a sensitivity analysis of a design problem.

A test problem involving a network with 35 nodes and 10 projects (a project is an arc whose capacity may be increased or decreased) has been solved using Stairs' procedure.

It may be possible that the interactive program approach could also be applied to large network problems. With Stairs' approach the interaction between the user and computer is comparatively simple. Krolak et al. [KRO1, KRO2] have designed more sophisticated exchanges of information between user and computer. They stress structuring and displaying the problem data in a way which complements the human problem solving process. Using their approach they are able to find solutions to 200 city traveling salesman problems that are about 4% from the optimum. So, perhaps with a more sophisticated approach, large scale network design problems could be handled by an interactive program.

Funk and Tillman [FUN1, SC03] also consider a variant of the infinite capacity network improvement problem. However, we will not discuss their work in this survey.

### 1.3 Capacitated Network Improvement Problems

We will now discuss some network improvement problems that are natural extensions of the network improvement problems discussed in section 1.2.3. The networks discussed in this section will contain arcs that have finite flow capacities. This feature is a more realistic assumption about the structure of most networks than the infinite capacity arcs of the previous network design problems. For example, a road in a traffic network certainly has a finite flow capacity.

Several different formulations of this problem will be described.

Roberts and Funk [ROB1] discuss a version of the capacitated network improvement problem that has the following formal description:

(1.3.1) OBJECTIVE: minimize total construction and routing costs

CONSTRAINTS:

- 1) ARC TYPE - directed
- 2) ARC CAPACITIES - the capacity of arc (i,j) can be either zero or  $C_{ij}$



- 3) CONSTRUCTION COSTS - a fixed charge for constructing an arc with non-zero capacity
- 4) ROUTING COSTS - linear functions of the arc flows
- 5) REQUIRED FLOWS - there are required flows between all pairs of nodes in the networks
- 6) SPECIAL CONSTRAINTS
  - a) Initial Arc Capacities - some arcs are initially set to their non-zero capacity value. They constitute the initial unimproved network.
  - b) Construction Budget - the total construction costs cannot exceed a given budget.

Roberts and Funk formulate (1.3.1) as a mixed integer programming problem. However, they give no specific method for solving the formulation.

Ochoa [OCH1] treats another version of the network improvement problem that can be considered as a generalization of the Roberts-Funk model. Ochoa analyzes the improvement of a network over  $K$  time periods. An arc may be added to the network before the start of any of  $K$  time periods. Now there is a construction

budget constraint for each of the K time periods. The objective is to minimize the sum of the routing costs for the K time periods. Otherwise the problem is identical to (1.3.1).

Note that if the number of time periods is equal to one, then the model is the same as the Roberts-Funk model except that the objective function does not include the arc construction costs. Ochoa also formulates this problem as a mixed integer programming problem. Since the size of the formulation is quite large, even for small networks, Ochoa suggests the use of Benders decomposition procedure [BEN1] for solving the problem. However, he does not report any computational results.

Hershdorfer [HER1] considers a third variant of the network improvement problem that is much more detailed than (1.3.1). His model has the following features:

(1.3.2) OBJECTIVE: minimize total construction costs

CONSTRAINTS:

- 1) ARC TYPE - directed
- 2) ARC CAPACITIES - each arc can be chosen to be one of L possible values
- 3) CONSTRUCTION COSTS - arbitrary
- 4) ROUTING COSTS - the routing cost of an arc is a convex piecewise linear function

of the flow. All arcs have two linear segments in their cost functions

5) REQUIRED FLOWS - there are required flows between all pairs of nodes in the network

6) SPECIAL CONSTRAINTS

- a) Initial Arc Capacities - some arc capacities are initially set at non-zero values. They constitute the initial unimproved networks.
- b) Routing Cost Reduction - the routing cost must be decreased by a certain given amount from the routing cost for the original unimproved network.
- c) Constant Ratio - the ratio of the capacities for the two linear segments of the routing cost function must remain constant no matter what the total arc capacity is.

Note that the type of cost function that appears in (1.3.1) is used to model the non-linear relationship between travel time and

traffic volume in a transportation network. The convex routing cost function reflects the effects of congestion in a network.

Hershdorfer formulates (1.3.2) as a large mixed-integer programming problem. He uses a branch and bound procedure which involves solving a series of linear programming problems. The largest network improvement problem that he solved successfully contained 12 nodes.

Carter and Stowers [CAR1] consider a problem that is very similar to (1.3.2). We do not discuss their work here.

In a recent doctoral thesis, Agarwal [AGA1] describes another version of the network improvement problem. Agarwal's problem has the following description:

(1.3.3) OBJECTIVE: minimize total routing costs

CONSTRAINTS:

- 1) ARC TYPE - directed
- 2) ARC CAPACITIES - the capacity of an arc  $(i,j)$  can range between 0 and some upper bound  $C_{ij}$
- 3) CONSTRUCTION COSTS - linear functions of arc capacity increases
- 4) ROUTING COSTS - the routing cost of an arc is a convex piece-

- wise linear function  
of the flow
- 5) REQUIRED FLOWS - there are required  
flows between all  
pairs of nodes in the  
network
- 6) SPECIAL CONSTRAINTS
- a) Initial Arc Capacities - some arc  
capacities are initially set at non-zero  
values. They constitute the initial un-  
improved network.
- b) Construction Budget - total construction  
costs cannot exceed a given budget.

Agarwal uses a variety of solution techniques in gaining computational experience. However, the results are quite discouraging.

A test problem with 24 nodes and 38 arcs was formulated as a linear program with 667 rows and 1938 variables. The solution of the linear problem required over 14 minutes of CDC 6400 computer time. In an effort to reduce computation time, Agarwal applied Dantzig-Wolfe decomposition [DAN1] and the Boxstep method [MAR1]. Neither approach was able to solve the problem in a reasonable amount of time. Agarwal decided that neither method was effective because of the arc capacity constraints present in the problem. The difficulty caused by the capacity constraints

should not be surprising. Consider the problem of finding the routing cost for a particular proposed network solution. For all of the above versions of the network improvement problem, if the capacity constraints were removed, then computing the routing cost would only involve solving a series of shortest path problems (note that this is exactly the case for the infinite capacity network design problems which do not have any capacity constraints). If the capacity constraints are kept, then computing the routing cost for the Roberts-Funk version of the problem requires the solution of a difficult minimum cost multi-commodity flow problem [AS1, KEN1, TOM1]. For the last two versions (1.3.2 and 1.3.3) with their piecewise-linear routing cost functions, the computation of the routing cost is even more difficult. Since the problem of evaluating a proposed solution is so difficult, it should not be surprising that the problem of finding the optimal solution is also very difficult.

Steenbrink [STE1, STE2] discusses another variant of the capacitated network improvement problem which has the following description:

(1.3.4) OBJECTIVE: minimize total construction and routing costs

CONSTRAINTS:

- 1) ARC TYPE - directed
- 2) ARC CAPACITIES - the capacity of an arc (i,j) can range between 0 and some upper bound  $C_{ij}$
- 3) CONSTRUCTION COSTS - arbitrary
- 4) ROUTING COSTS - arbitrary
- 5) REQUIRED FLOWS - there are required flows between all pairs of nodes in the network
- 6) SPECIAL CONSTRAINT
  - a) Initial Arc Capacities - some arc capacities are initially set at non-zero values. They constitute the initial unimproved network.

Steenbrink formulates this problem as a mathematical programming problem with linear constraints and a non-linear objective function. He does not propose an exact solution for the problem. Instead, he suggests a heuristic procedure that will hopefully produce a reasonable solution to the problem.

Steenbrink's method involves decomposing the original problem into a master problem and a series of subproblems. Each

subproblem concerns finding the optimal capacity for an arc given the total flow through it. The master problem is to route the required flows through the network with all capacity constraints removed. (This master problem is actually a multi-commodity flow problem [AS1]). Steenbrink solves the master problem by using a stepwise assignment procedure to route the flows. Let  $\alpha_1, \dots, \alpha_J$  be fractions such that

$$\sum_{i=1}^J \alpha_i = 1 \quad \text{and} \quad \alpha_i \geq 0 \quad i = 1, \dots, J.$$

On the  $i^{\text{th}}$  iteration of the stepwise assignment procedure, the fraction  $\alpha_i$  of each of the required flows is routed through the network. For example, let  $R_{12}$  be the required flow between nodes 1 and 2. At iteration  $i$  we route  $(\alpha_i \cdot R_{12})$  units of flow from node 1 and 2 via its shortest route in a specially defined network. In this specially defined network, the "length" of arc  $(k,1) =$

$$\left. \frac{dF_{k1}}{dx} \right|_{x = x_{i-1}^{k1}}$$

where

$F_{k1}$  = routing cost function for arc  $(k,1)$

$x_{i-1}^{k1}$  = the total amount of flow routed through arc  $(k,1)$  after the  $(i-1)^{\text{st}}$  iteration.



So, in effect, the routing cost of an arc in this special network is the marginal routing cost for the arc after the previous iteration. After J iterations all the required flows will be routed. Then the arc capacities are adjusted so that the construction costs are minimized (this is the solution of the subproblems).

Steenbrink shows that this process does not always terminate with an optimal solution. Hopefully, the solution generated will be a reasonable one.

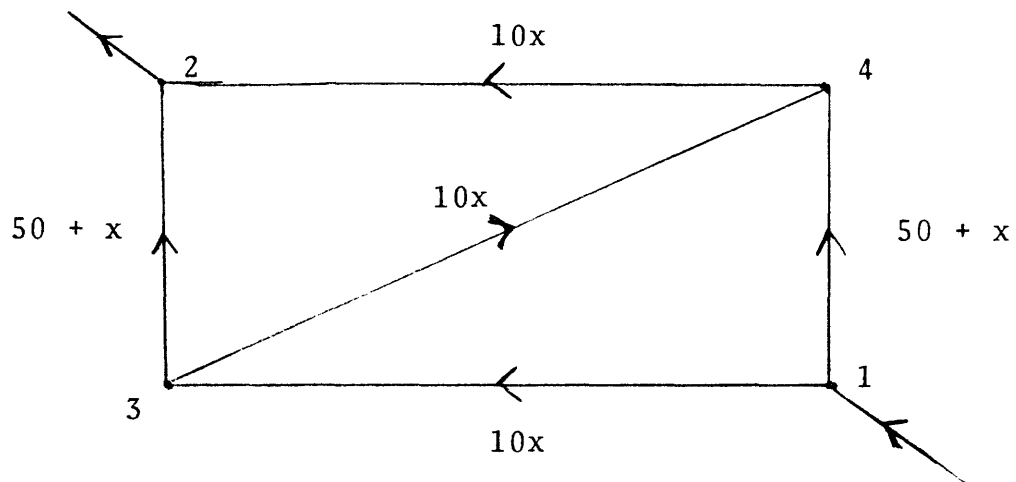
Steenbrink applied this method to a Dutch roadway design problem which was modelled as a network with 2000 nodes and 6000 arcs. The stepwise assignment procedure used 4 iterations to assign the flows. Each iteration required about 12 minutes of IBM 360/65 computer time. Of course, there is no way to evaluate how close Steenbrink's solution is to the optimal solution. Finding the optimal solution would require the solution of a very large non-linear program. In fact, it should be noted that this problem seems to be by far the largest network design problem attempted in the literature.

Steenbrink [STE2] discusses in great detail the stepwise assignment procedure and its application to the design of a Dutch roadway network. He also describes many practical details of the procedure's implementation.

#### 1.4 Network Design Problems with User Equilibrium Routing

In this section we discuss network design problems that have a user equilibrium routing (UER) policy instead of a system optimal routing policy. So, instead of routing flows in order to minimize the total routing cost for all flows, we seek a traffic flow assignment which has the property that no unit of flow can improve its routing cost by taking an alternative route between its origin and destination. First, we describe a major difference between network design problems with UER and those with system optimal routing. Braess [MUR1] (for the original German article by Braess see [BRA1]) was the first one to document this difference. For a network with system optimal routing, the addition of an arc to the network can never increase the total flow routing costs. Since we can always choose to use the flow routing pattern that was used before the new arc was added, the total routing cost can never increase and will usually decrease. Somewhat surprisingly, for a network with UER, the addition of an arc can lead to an increase in the total flow routing costs. This phenomenon is known as Braess' paradox.

We will now describe an example of Braess' paradox taken from [MUR1]. Figure 1.4.1 gives a sketch of the directed network that will be discussed. Six units of flow must be routed



path 1 = arcs (1, 3) and (3, 2)  
path 2 = arcs (1, 4) and (4, 2)  
path 3 = arcs (1, 3), (3, 4) and (4, 2)

FIGURE 1.4.1

Network Example of Braess' Paradox

from nodes 1 to 2. We also have:

$$\text{routing cost for arc } (i,j) = x_{ij} \cdot f_{ij}(x_{ij})$$

where  $x_{ij}$  = flow on arc  $(i,j)$

$$f_{13}(x) = 10x$$

$$f_{32}(x) = 50 + x$$

$$f_{34}(x) = 10 + x$$

$$f_{42}(x) = 10x$$

$$f_{14}(x) = 50 + x$$

The first situation that we will analyze is when arc  $(3,4)$  does not exist. By symmetry, the UER policy is to send 3 units of flow via paths 1 and 2. The total routing cost is 498. If we consider the network with arc  $(3,4)$  present, the UER policy is to send 2 units of flow via paths 1, 2, and 3. The total routing cost is 552. With the addition of arc  $(3,4)$  to the network, the routing cost increases by about 11%. It is not known how prevalent this counter-intuitive behavior is in networks that have a UER policy. However, Murchland [MUR1] reports on a recent experience by Knödel, "Knödel remarks that the example (of Braess) may seem contrived, but a recent experience in Stuttgart shows that it can occur in reality. Major road investments in the city centre, in the vicinity of the Schlossplatz, failed to yield the benefits expected. They were only obtained when a cross street, the lower part of Königstrase, was sub-

sequently withdrawn from traffic use." (see [KN01] for Knödel's original German article.)

Braess' paradox indicates that great care should be used in evaluating proposed improvements to a network with UER.

Now we discuss some work that has dealt with the design of networks with UER. All of this work concerns the area of network improvement problems.

The first version of the network improvement problem with UER has the following description:

(1.4.1) OBJECTIVE: minimize total routing costs

CONSTRAINTS:

- 1) ARC TYPE - directed
- 2) ARC CAPACITIES - zero or infinite
- 3) CONSTRUCTION COSTS - a fixed charge for constructing an arc with infinite capacity
- 4) ROUTING COSTS - arbitrary
- 5) REQUIRED FLOWS - there are required flows between all pairs of nodes in the network
- 6) SPECIAL CONSTRAINTS
  - a) Initial Arc Capacities - some arc capacities are initially set to infinity.

They constitute the initial unimproved network.

- b) Construction Budget - total construction costs cannot exceed a given budget.
- c) Network Restriction - the routing cost cannot increase when additional arcs are added to the network. This constraint forbids networks such as the one used to demonstrate Braess' paradox.

(1.4.1) is similar to Ridley's network improvement problem (1.2.3.1) except that (1.4.1) uses UER and in general, the routing cost functions used will be more complicated than Ridley's linear functions. Constraint 6c does not appear to be very practical. For complicated networks it will be very difficult to verify that the constraint is satisfied. However, constraint 6c is a crucial assumption for the solution technique about to be given.

Ochoa and Silva [OCH2] suggest using a branch and bound procedure to solve (1.4.1). At each vertex in the search tree, a decision is made whether or not to construct a particular candidate arc. Also, at each vertex the procedure computes lower bounds on the routing cost. This is done by adding to the network all arcs for which the construction decision has not yet been made and then computing the minimum UER cost for

this network. Of course, this lower bound is only valid if constraint 6c is satisfied (if constraint 6c is not satisfied, it might be possible to achieve a lower UER cost by adding only a subset of the remaining arcs). At each vertex a calculation is made to see if constraint 6b (the construction budget constraint) is satisfied. This is accomplished by computing a lower bound on the arc construction costs. The lower bound is merely the sum of the construction costs for all arcs which have already been added to the network. Ochoa and Silva do not give any computation experience for their procedure.

Chan [CHA1] also analyzes a network improvement problem that is very similar to (1.4.1).

Recently, Leblanc [LEB1] considered the following problem:

(1.4.2) Same as (1.4.1) except that constraint 6c is eliminated.

So this version of the network improvement problem does not require a monotonicity assumption for the routing cost function. Leblanc formulates (1.4.2) as a large nonlinear programming problem. He suggests a branch and bound procedure to solve it. His branch and bound procedure is identical to the one used by Ochoa and Silva except for a new way to obtain lower bounds on the routing costs. At a vertex in the search tree, a lower bound on the total routing costs is obtained without any assumptions about the behavior of the routing costs. This is

accomplished by adding all arcs for which the construction decision has not yet been made and then computing the minimum total routing cost for the network with a system optimal routing policy. Leblanc proves that this will always be a valid lower bound. His proof utilizes an important relationship between system optimal routing and UER. For any network, a system optimal routing policy will never have a higher total routing cost than a UER policy. This is because a UER flow assignment is always a feasible solution to the system optimal routing problem.

Leblanc uses his procedure to solve a sample problem with a network that has 24 nodes, 71 arcs and 5 arcs that could be added to the network. Finding an optimal solution to the problem required about 2 1/4 minutes of CDC 6400 computer time.

Although branch and bound has emerged as a very good technique for determining the exact solution of small and medium sized optimization problems, it is much less successful with large sized problems. So the branch and bound techniques given here for solving (1.4.1-2) will probably be unable to handle large network improvement problems that have dozens of candidate arcs. The next article that we discuss gives a problem formulation and solution technique that can be used with very large networks. The solution technique is not an exact one, but it is hoped that reasonable solutions will be generated.



Barbier [STA1, STE2, also see BAR1 for the original French article by Barbier] defines the following network improvement problem:

(1.4.3) OBJECTIVE: minimize total construction and routing costs

CONSTRAINTS: same as (1.4.1) except 6b and 6c are eliminated and constraint 1 is replaced by:

1<sup>1</sup>) ARC TYPE - undirected.

(1.4.3) is similar to Stairs' network improvement problem given in section 1.2.3. The only differences are that (1.4.3) uses UER and in general, the routing cost functions used will be more complicated.

Barbier gives an iterative algorithm that will hopefully generate reasonable, although not necessarily optimal solutions to (1.4.3). Barbier's algorithm has the following steps:

Step 1 Add all possible candidate arcs to the original unimproved network. Then assign the required flows according to the UER policy.

Step 2 Change the cost of routing flow through a candidate arc from just the routing cost to the routing cost plus a fixed charge equal to the construction cost of the arc.

Step 3 All arc flow levels are at their previously determined level (either determined in step 1 or 4). Choose a required flow  $R_{ij}$  that goes from node  $i$  to  $j$ . Find a path from  $i$  to  $j$  for  $R_{ij}$  so that the cost of routing is minimized. (Remember that some routing costs have an added fixed charge). Reassign the flow  $R_{ij}$  along this path.

Step 4 Take any candidate arc which appears on the path and eliminate the fixed charge from its cost of routing. (Since we have routed some flow through the candidate arc, we essentially have "paid" for its construction cost. So now we can eliminate the construction cost). Repeat steps 3 and 4 for every required flow.

Step 5 Eliminate from the network all candidate arcs which have zero flow through them. Now take the resultant network and reassign all the flows according to the UER policy.

Repeat steps 2-5 until a stable network is found.

This final network is the proposed solution.

Barbier uses this method to study additions to the Paris rail network. The method is applied to a network with 36 nodes, over 30 arcs and over 50 candidate arcs. Steenbrink [STE2] reports that Haubrich used a revised version of Barbier's method

to study the Dutch rail network. Haubrich's procedure obtained a solution to a network with about 1250 nodes and about 8000 arcs. The method required less than 40 minutes of IBM 360/65 computer time.

### 1.5 Communication Network Synthesis Problems

In this section we consider a different type of network synthesis problem. For these problems, no flow routing costs are involved. Instead, we must design a minimum construction cost network so that all the required flows can be routed through the network. Also, the required flows for this kind of problem are more complicated than the ones described in previous problems. Usually there are several different time periods. During each time period a subset of the required flows must be routed through the network. This type of problem is called a communication network synthesis problem since this kind of network usually occurs in the context of communication network design.

Several versions of the communication synthesis problem have been formulated. Gomory and Hu [GOM1] and Chien [CHI1] deal with the simplest version. In terms of our general framework, this problem has the following description:

(1.5.1) OBJECTIVE: minimize total arc construction costs

CONSTRAINTS:

- 1) ARC TYPE - undirected
- 2) ARC CAPACITIES - any value from zero to infinity

- 3) CONSTRUCTION COSTS - equal to the arc capacity
- 4) ROUTING COSTS - none
- 5) REQUIRED FLOWS - there is a set  $\{R_{ij}\}$  of required flows between nodes. The network must be designed so that any one particular  $R_{ij}$  can be routed through it.

Since constraint 3 implies that all arc construction cost functions are identical, an equivalent objective is to minimize the total arc capacity of the network.

Both Gomory and Hu and Chien give simple efficient solutions to (1.5.1). We will describe the method given by Gomory and Hu.

First, Gomory and Hu show that, if the flow routing constraint for the network is satisfied for a subset of the  $\{R_{ij}\}$  (known as the dominant requirement tree), then it is satisfied for the entire set of  $R_{ij}$ . The  $R_{ij}$  in the dominant requirement tree can be found in the following way: consider the network of our synthesis problem with all arcs  $(i,j)$  weighted by the flow requirement  $R_{ij}$ , then the dominant requirement tree is the maximal spanning tree of this network. (Note that the maximal spanning tree problem can be solved by a procedure that is

completely analagous to the minimal spanning tree solution procedure [KRU1]).

Using this fact about the synthesis problem, Gomory and Hu give the following synthesis procedure:

- 1) Find the dominant requirement tree of the network.
- 2) Decompose this tree into a "sum" of a "uniform" requirement tree (where all arcs in the tree have equal arc weight) and a remainder (which is a forest of two or more trees). This decomposition is done by subtracting the smallest arc weight from every arc weight in the tree.
- 3) Take each tree in the remainder and go back to step 2.
- 4) For each uniform requirement tree formed, synthesize it by forming a cycle through its nodes. Each arc in the cycle will have a capacity equal to one-half the requirement of the uniform tree.

An example of this procedure, taken from [FOR1], is sketched in figures 1.5.1-1.5.4. Figure 1.5.1 contains the given network with arc  $(i,j)$  labeled with flow requirement  $R_{ij}$ . The members of the dominant requirement tree are denoted by heavy lines. Figure 1.5.2 shows the result of the first execution of step 2. Figure 1.5.3 shows the result after all the iterations of step 2 have been completed. Figure 1.5.4 shows the final result of step 4. Notice that the network in figure 1.5.4 is the sum of 4 cycles, since there are 4 uniform requirement trees.

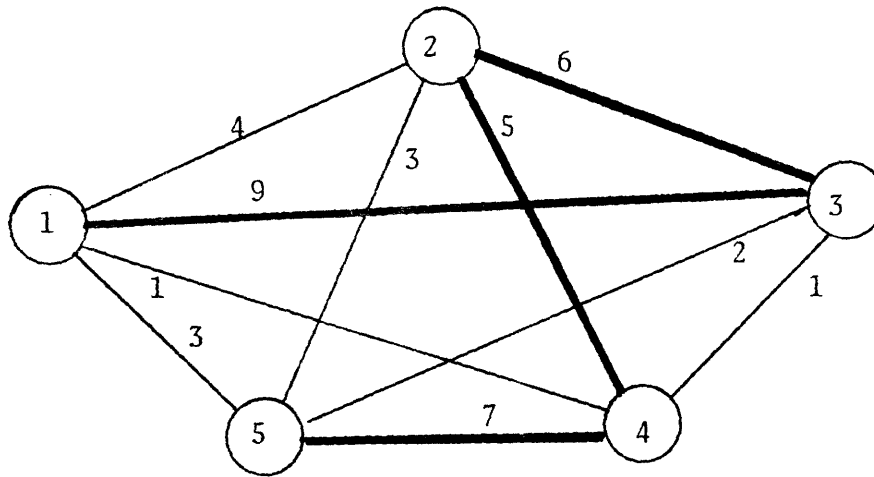


FIGURE 1.5.1

COMMUNICATION NETWORK SYNTHESIS PROBLEM

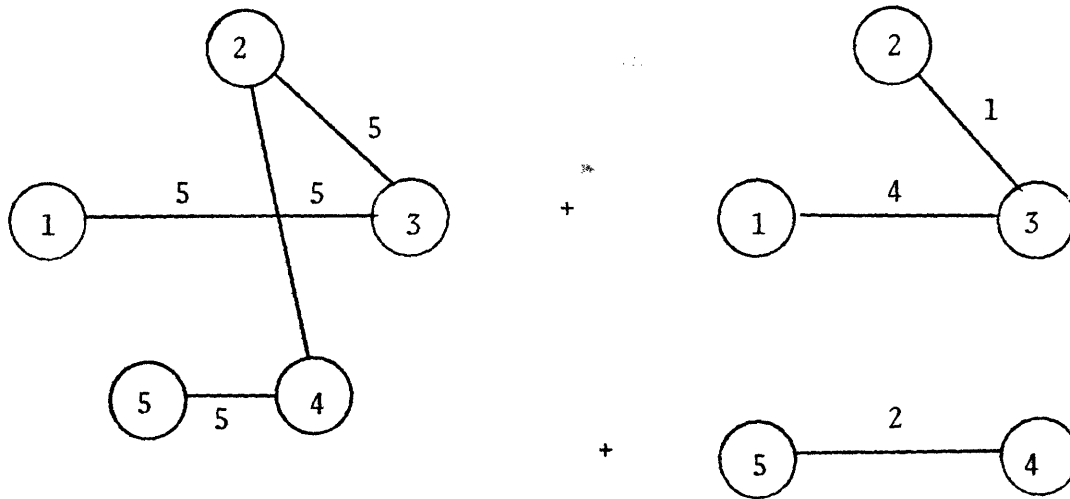


FIGURE 1.5.2

PARTIAL DECOMPOSITION OF COMMUNICATION PROBLEM

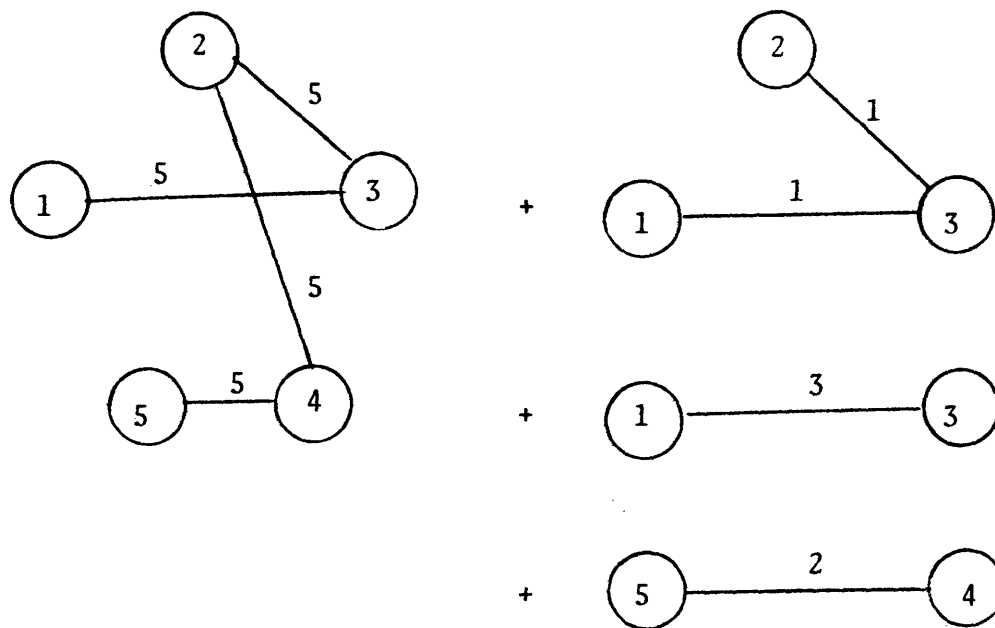


FIGURE 1.5.3

DECOMPOSITION OF COMMUNICATION NETWORK PROBLEM  
INTO UNIFORM REQUIREMENT TREES

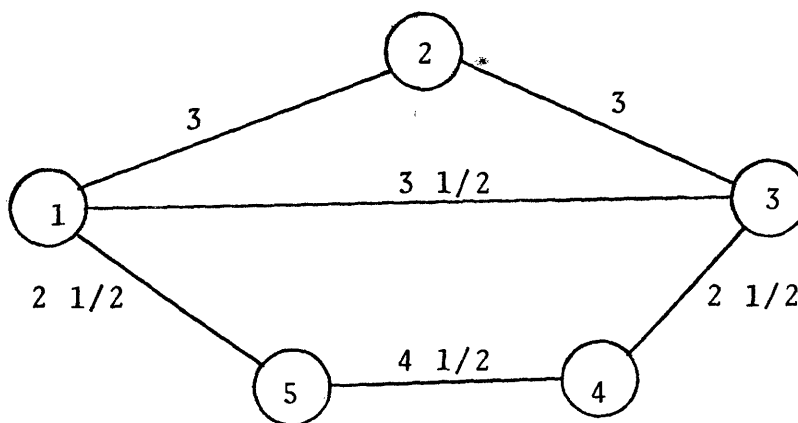


FIGURE 1.5.4

FINAL NETWORK DESIGN



Although Gomory and Hu do not give any computational results for their procedure, it is easy to see that the procedure is quite efficient. The worst case computation time of the procedure is proportional to  $N^2$ .

Gomory and Hu [GOM2] treat the following generalization of (1.5.1):

(1.5.2) The same objective and constraints as (1.5.1) except that constraint 3 is replaced by:

3<sup>1</sup>) CONSTRUCTION COSTS - linear functions of the arc capacities.

So, in this more difficult version of the problem, the arc construction cost functions can all be different.

The simple algorithm used to solve (1.5.1) cannot be applied to (1.5.2). Instead, Gomory and Hu formulate the synthesis problem as the following linear program:

$$(1.5.3) \text{ minimize: } \sum_{i=1}^N \sum_{j=1}^N c_{ij} b_{ij}$$
$$\text{s.t. } K_{ij}^{\alpha} \geq R_{ij} \quad \text{for all } \alpha \text{ with arc } (i,j) \in T.$$
$$c_{ij} \geq 0 \quad \text{for all } i,j.$$

where  $c_{ij}$  = capacity of arc (i,j)  
 $b_{ij}$  = cost of constructing a unit of capacity for arc (i,j)  
 $K_{ij}^\alpha$  = capacity of the  $\alpha^{\text{th}}$  cut separating nodes i and j  

$$= \sum_{(r,\Delta) \in a} c_{rs}$$
  
 $T$  = the dominant requirement tree of the network.

Note that the result concerning the dominant requirement tree of the network is also applicable to (1.5.2).

Since there are only (N-1) arcs in T (where N is the total number of nodes in the problem), only (N-1) flow requirements must be satisfied. However, since the number of cuts separating two nodes is very large (even for small networks), the number of rows in P1 is enormous.

Now, if we take the dual of (1.5.3), we obtain the following linear program:

$$(1.5.4) \quad \text{maximize:} \quad \sum_{\alpha} \sum_{A_{ij} \in T} R_{ij} \Pi_{ij\alpha}$$

$$\text{s.t.} \quad \sum_{\alpha} \sum_{A_{ij} \in T} P_{ij\alpha} \Pi_{ij\alpha} \leq B$$

$$\Pi_{ij\alpha} \geq 0$$

where  $P$  = incidence matrix of cuts  $\alpha_{ij}$  separating nodes  $i$  and  $j$  vs. arcs  $(k,1)$  in the network.

$B$  = vector with components  $b_{ij}$ .

(1.5.4) has many columns and a number of rows equal to the number of arcs in the network. Gomory and Hu use Dantzig - Wolfe decomposition [DAN1] to solve the problem. At each iteration after solving the restricted version of (1.5.4), they perform  $(N-1)$  maximal flow calculations in order to generate additional columns. The  $(N-1)$  maximal flow calculations check to see if the  $(N-1)$  flow requirements in the dominant requirement tree are satisfied.

The above procedure is a dual method so a feasible solution is not generated until the optimal network is found. Gomory and Hu also describe a primal solution method which has the advantage of producing a feasible solution method even if terminated before an optimum is reached. However, the primal procedure appears to be less efficient than the dual procedure.

Gomory and Hu do not give any computational results for either method.

Gomory and Hu [GOM3] describe another version of the communication network synthesis problem.

(1.5.5) The same objective and constraints as (1.5.2) except that constraint 5 is replaced by:

5<sup>1</sup>) REQUIRED FLOWS - there is a set  $\{R_{ij}\}$  of required flows between nodes. The network must be designed so that all  $R_{ij}$  can be simultaneously routed through the network.

Gomory and Hu suggest a very simple algorithm to solve (1.5.5). Start with all arc capacities set equal to zero. Assign a "distance" to each arc which equals the cost of increasing the arc capacity by one unit. Then for each pair of nodes  $i$  and  $j$ , find the shortest path between  $i$  and  $j$ , and increase the capacity of each arc along this shortest path by  $R_{ij}$  units. So this algorithm only requires the calculation of all the shortest path pairs in the network.

Finally, Gomory and Hu [GOM3] give a fourth version of the problem which is a generalization of the previous three problems.

(1.5.6) The same objective and constraints as (1.5.2) except that constraint 5 is replaced by:

5<sup>1</sup>) REQUIRED FLOWS - there are  $T$  sets  $R_{ij}^t$ , ( $t = 1, \dots, T$ ), of required flows between nodes. There are  $T$

different time periods.  
During a particular time  
period  $t_1$ , all required  
flows in the set

$\left\{ \begin{array}{l} t_1 \\ R_{ij} \end{array} \right\}$  must be routed  
through the network.

In (1.5.1) and (1.5.2) the optimal network only had to handle one  $R_{ij}$  during any particular time period. In (1.5.5) the optimal network had more than one  $R_{ij}$  to route during a time period, but there was only one time period to consider. So (1.5.6) can be considered a combination of the above problems.

Gomory and Hu formulate (1.5.6) as a linear program with a large number of rows. The columns represent the values of the network arc capacities. The rows represent the various flow requirements that the network must satisfy. The linear program is solved using the dual simplex method. A series of  $T$  subproblems is solved to find constraints that are violated. Each subproblem is a check to see if the candidate network (represented by the present value of the arc capacities) can feasibly route the flows for a particular time period. These subproblems, which contain a large number of columns, are solved using a column generation procedure.

Computational experience for a ten node and twenty arc network with flow requirements for two time periods is given. Finding an optimal solution required ten minutes of IBM 7094 computer time.

For the rest of this section we consider a subclass of (1.5.1) where the arc capacities and required flows  $R_{ij}$  are restricted to integer values. Problems of this type are usually treated as combinatorial problems. So the solution techniques described are usually combinatorial algorithms.

Chou and Frank [CH01] consider the following discrete version of (1.5.1):

(1.5.7) The same objective and constraints as (1.5.1) except that constraints 2 and 5 are replaced by:

- 2<sup>1</sup>) ARC CAPACITIES - any integer value from zero to infinity
- 5<sup>1</sup>) REQUIRED FLOWS - there is a set of positive integers  $\{R_{ij}\}$  of required flows between nodes. The network must be designed so that any one particular  $R_{ij}$  can be routed through it.

The following problem is equivalent to (1.5.7): given a set of nodes and a set of integers  $\{R_{ij}\}$ , construct a network with a

minimum number of branches (we will sometimes refer to an arc with a capacity of one unit as a branch) so that there are at least  $R_{ij}$  branch disjoint paths between nodes  $i$  and  $j$ . Parallel branches between nodes are allowed but no new nodes are allowed. (By taking the optimal network for this problem and summing the number of parallel branches between nodes  $i$  and  $j$ , we get the capacity of arc  $(i,j)$  in the optimal network for (1.5.7)). This equivalent problem is sometimes known as the survivable communication network problem.

The algorithms of Gomory and Hu and of Chien for solving (1.5.1) cannot always be used to solve (1.5.7). Both methods will sometimes generate networks with non-integer capacities. For example, the Gomory and Hu algorithm will only generate a network with all integer capacities if every  $R_{ij}$  in the dominant requirement tree is an even number.

Chou and Frank give an algorithm to solve (1.5.7). The method is quite efficient and can probably solve very large problems (thousands of nodes) in a reasonable amount of time. They also formulate and give solutions to some related synthesis problems. One type of problem occurs when the network design is allowed to be a pseudosymmetric network instead of an undirected network (a pseudosymmetric network is a directed network in which the sum of the capacities of the incoming arcs is equal to the sum of the capacities of the outgoing arcs). Chou and Frank also consider the optimal realization of terminal capacity matrices for

symmetric and pseudosymmetric networks. This problem involves finding the minimum cost network so that the maximum flow capacity between nodes  $i$  and  $j$  is exactly  $R_{ij}$ .

Frank and Chou [FRA1, FRA2] consider a restricted version of (1.5.7).

(1.5.8) The same objective and constraints as (1.5.1) except that constraints 2 and 5 are replaced and a constraint 6 is added:

2<sup>1</sup>) ARC CAPACITIES - zero or one

5<sup>1</sup>) REQUIRED FLOWS - there is a set of positive integers  $\{R_{ij}\}$  of required flows between nodes. The network must be designed so that any one particular  $R_{ij}$  can be routed through it. (This is the same constraint as in (1.5.7)).

6) SPECIAL CONSTRAINT

a) Node Additions - if necessary, additional nodes may be added to the original set of nodes.

We can also describe a problem equivalent to (1.5.8). This problem is the same as the problem equivalent to (1.5.7) except that new nodes may be added, but no parallel branches between nodes are allowed.



Frank and Chou give a complicated but efficient algorithm to solve (1.5.8). Due to its excessive length, this algorithm will not be discussed here.

The logical generalizations of (1.5.7) and (1.5.8) are to allow the arc construction costs to be arbitrary linear functions of the capacity. This would create situations where some arcs are more "expensive" to build than others. However, it can be shown that these generalizations are very difficult problems. Consider the case where  $R_{ij} = 1$  for  $i$  and  $j$  belonging to  $S$  (where  $S$  is a subset of the nodes in the network) and  $R_{ij} = 0$  for all other  $i$  and  $j$ . With general linear construction costs this problem is exactly the Steiner tree problem on a graph [HAK3, DRE2]. So the generalizations of (1.5.7) and (1.5.8) contain the Steiner tree problem on a graph as a special case. Therefore, these generalizations are at least as hard as the class of NP - complete problems [KAR1, KAR2]. So it is not surprising that no work has been done on these generalizations.

Now we will discuss still another variation of (1.5.1). The problem can be described in the following way: given a set of nodes and a set of integers  $\{R_{ij}\}$ , construct a network with a minimum number of branches so that there are at least  $R_{ij}$  node disjoint paths between nodes  $i$  and  $j$ . In terms of our general framework, the problem can be stated as:

(1.5.9) The same objective and constraints as (1.5.1) except that constraints 2 and 5 are replaced and a constraint 6 is added.

2<sup>1</sup>) ARC CAPACITIES - any integer value from zero to infinity

5<sup>1</sup>) REQUIRED FLOWS - there is a set of positive integers  $\{R_{ij}\}$  of required flows between nodes. The network must be designed so that any one particular  $R_{ij}$  can be routed through it.

6) SPECIAL CONSTRAINT

a) Node Capacity - All nodes have a flow capacity of one unit.

Notice that (1.5.9) is similar to (1.5.7) except that (1.5.9) requires node disjoint paths instead of branch disjoint paths as does (1.5.7). Two paths are branch disjoint if they are node disjoint, but the opposite is not true. So (1.5.9) is a more restrictive problem than (1.5.7). Also, notice that any optimal solution to (1.5.9) will never have any arc capacities greater than one.

Harary [HAR1] and Boesch and Thomas [BOE1] give procedures for solving (1.5.9) when all the  $R_{ij}$  are equal. The complete version of (1.5.9) with the  $R_{ij}$  allowed to be arbitrary integers

has not been completely solved, although Frank has worked out many special cases [FRA4].

Steiglitz et al. [STE3] consider the extension of (1.5.9) where the arc construction costs are general functions of the capacity. This general problem is quite difficult. For the case when all  $R_{ij}$  equal two, this design problem is exactly the traveling salesman problem (since a hamiltonian tour creates two node disjoint paths between every pair of nodes). Since the traveling salesman problem is an NP - complete problem, this general design problem is probably intractable.

Instead of trying to find an exact solution, Steiglitz et al. give a heuristic approach to the problem. A procedure is given to generate a random feasible solution. Then a local transformation is applied to the feasible solution. If a feasible solution of lower cost is found, then the improved network is adopted and the local transformation is applied again. This continues until a feasible network is found which is locally optimal in the sense that no local transformations of the type considered result in a feasible network of lower cost.

The local transformation used in the procedure is a generalization of Lin's  $\lambda$ -change procedure [LIN1]. This local transformation called an X-change, takes a feasible network  $N_1$  that has branches  $(i,m)$  and  $(j,l)$ , but does not have the two branches  $(i,l)$  and  $(j,m)$ . Let  $d_{ij}$  be the cost of constructing branch  $(i,j)$ . If  $(d_{il} + d_{jm}) < (d_{im} + d_{jl})$ , then we transform  $N_1$  by

adding branches  $(i,1)$  and  $(j,m)$  while removing branches  $(i,m)$  and  $(j,1)$  (see figure 1.5.5). This paper also presents an efficient method to check if the transformed network remains a feasible solution.

Steiglitz et al. apply this series of transformations to a number of random initial networks. The best local optimum obtained from all these iterations is selected as the most appropriate network design.

The authors provide some computational results for their procedure. For a ten node problem, generating a single local optimum required about 3.4 seconds on the Univac 1108. For a 58 node problem, the computation time for a local optimum increased to 12 minutes.

In this section we have only considered a small number of the problems that could be discussed. There is a large body of literature that deals with network design problems that are related to the problems discussed in this section. Most of these other problems differ with respect to the type of objective functions and flow constraints used. The interested reader can find some of these problems discussed in FRA3 (chapters 5, 6 and 7). Also see the set of references given in Appendix 1.

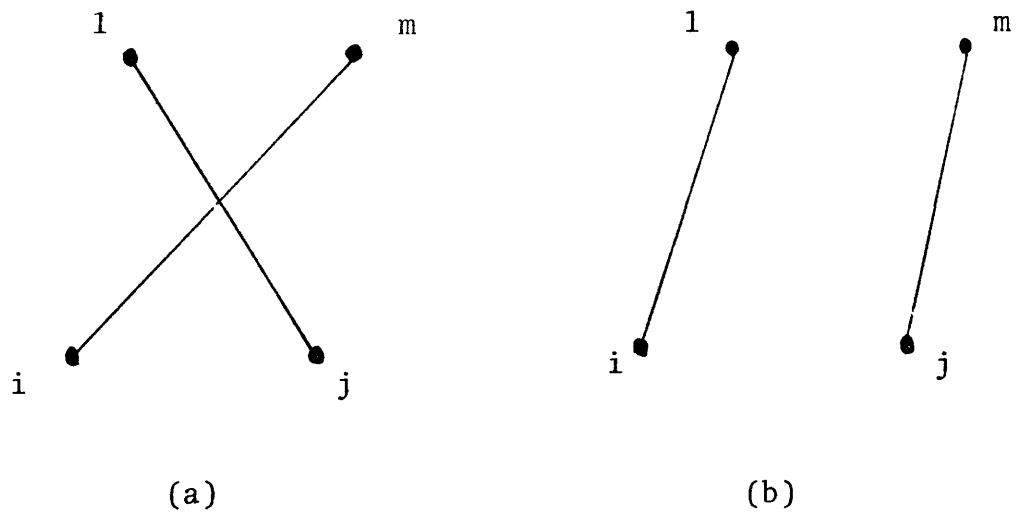


FIGURE 1.5.5

EXAMPLE OF X-CHANGE TRANSFORMATION

## 1.6 Maximum Flow Capacity Improvement Problems

The next type of problem we will discuss concerns the improvement of the maximum flow capacity of a network. The problem usually has the following description. The network consists of a set of nodes and a set of undirected and capacitated arcs. There is a source node  $S$  and terminal node  $T$  designated in the network. The goal of the problem involves improving the maximum flow capacity from  $S$  to  $T$  by improving the arc capacities in the network.

Since we are using our general framework (which involves the definition of required flows and routing costs) to classify problems, we will now restate the problem in terms of the general framework.

Our alternative way of stating the problem has the following description: the network consists of a set of nodes and undirected and capacitated arcs. There is also a special directed arc connecting nodes  $T$  and  $S$ . Figure 1.6.1 provides a sketch of a typical network. Each arc capacity is originally set at some level and may be increased. Associated with each arc is a construction cost function for increasing the arc capacity. There are no flow requirements in the network. There are no routing costs except on the special directed arc  $(T,S)$  which has a routing cost equal to the negative of the flow through the arc. It is easy to see that minimizing the routing cost in our modified net-

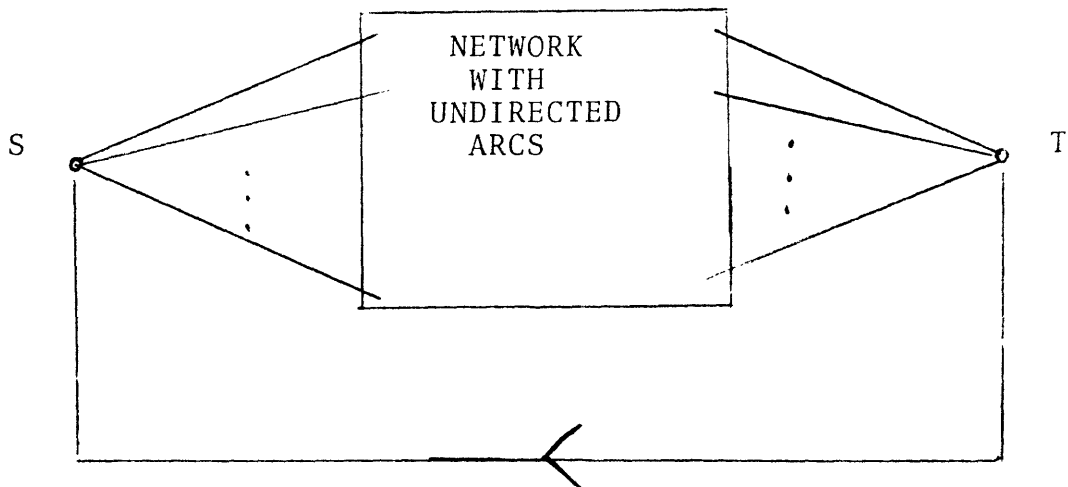


FIGURE 1.6.1

TYPICAL NETWORK FOR MAXIMUM FLOW IMPROVEMENT PROBLEMS

work is equivalent to maximizing the flow between S and T. By altering the type of construction cost functions and objective functions used, several different versions of the maximum flow capacity improvement problem are possible. We will now discuss several of them.

Fulkerson [FUL1] and Hu [HUT1] consider the simplest case of the flow improvement problem. It has the following description:

(1.6.1) OBJECTIVE: minimize total routing costs

CONSTRAINTS:

- 1) ARC TYPE - all arcs are undirected except for arc (T,S) which connects the source and sink nodes
- 2) ARC CAPACITIES - any value from zero to infinity
- 3) CONSTRUCTION COSTS - linear functions of the arc capacity increases
- 4) ROUTING COSTS - the only routing cost in the network is for arc (T,S). On this arc the routing cost is equal to the negative of the flow through the arc.



- 5) REQUIRED FLOWS - none
- 6) SPECIAL CONSTRAINTS
  - a) Initial Arc Capacities - some arcs are initially set to non-zero capacity values. They constitute the initial unimproved network.
  - b) Construction Budget - total construction costs cannot exceed a given budget.

Hu approaches (1.6.1) by solving a series of minimum cost flow problems. His algorithm starts by finding the maximum possible flow between nodes S and T without any increase in the arc capacities. Then the flow is augmented by solving a series of minimum cost flow problems until the entire budget is spent.

Christofides and Brooker [CHR1] consider a discrete version of (1.6.1) where the arc capacities can only take on discrete values. The problem has the following formal description:

(1.6.2) OBJECTIVE: same as (1.6.1)

CONSTRAINTS: same as (1.6.1) except constraints

2 and 3 are replaced by

- 2<sup>1</sup>) ARC CAPACITIES - limited to a discrete set of values for each arc
- 3<sup>1</sup>) CONSTRUCTION COSTS - arbitrary

Christofides and Brooker use a branch and bound procedure to solve (1.6.2). At each vertex in the search tree, they generate upper bounds on the total routing cost (i.e., maximum flow) by using the values of cuts that separate nodes S and T.

Christofides and Brooker provide computational experience for their procedure. The results are quite satisfactory for their sample of medium-sized networks. A problem that had 50 nodes, 55 arcs, and 20 other arcs that could be improved, each to one of 3 possible levels, required about 35 seconds of CDC 6400 computer time. Christofides and Brooker also note the results "indicate a comparatively slow increase of computation time with problem size."

Bansal and Jacobsen [BAN1] consider a generalization of (1.6.1) where the arc construction costs are concave functions of the capacity increase. The set of feasible solutions for this problem forms a nonconvex set. They propose a solution procedure that uses a generalization of Benders decomposition procedure [GE01]. The relaxed master problems generated are also nonconvex. Bansal and Jacobsen give a finite algorithm for solving the relaxed master problems which involves solving a series of linear programs. No computational experience is given.

Price [PRI1], Hess [HES1], and Hammer [HAM1] also consider maximum flow capacity improvement problems that are similar to (1.6.1) and (1.6.2). However, we will not discuss their work in this survey.

## 1.7 Final Remarks

We have seen that a great many different network design problems can be accommodated by our general framework. Table 1.7.1 contains a brief summary of the problems discussed in this report.

For our final remarks, we will discuss the various network design techniques that have been used in the work surveyed by this paper. The principle design methods that have been used are mathematical programming techniques (especially decomposition methods), branch and bound procedures, efficient special purpose algorithms, and heuristic procedures.

Efficient special purpose algorithms have been used to solve only the most basic variants of the network design problems discussed (see Goldman and Nemhauser's version of (1.2.3.1), also see (1.5.1), (1.5.7), (1.5.8) and (1.6.1)). It does not seem likely that the more advanced versions of the problems discussed will be solvable via this method. Indeed, many of the more advanced versions contain as special cases such computationally intractable problems as the traveling salesman problem or the Steiner tree problem on a graph.

Various authors have used branch and bound procedures to solve several network improvement problems (see (1.2.1.2), (1.2.3.1), (1.3.2), (1.4.1), (1.4.2) and (1.6.2)). However,

TABLE 1.7.1  
SUMMARY OF NETWORK DESIGN PROBLEMS

AUTHORS	EQUATION NUMBER	ARC CAPACITY VARIABLES	OBJECTIVE	SOLUTION ALGORITHM	(# OF NODES IN TEST NETWORK, # OF ARCS, COMP. TIME) COMPUTATIONAL EXPERIENCE
1. Billheimer and Gray [BILL]	1.2.1.1	Discrete	Minimize total construction and routing costs	Heuristic	(68 nodes, 476 arcs, 180 seconds)
2. Scott [SC01, SC02, SC03]	1.2.1.2	Discrete	Minimize routing costs subject to a construction budget constraint	Heuristic and Branch and Backtrack	(10 nodes, 45 arcs, 60 seconds) for heuristic procedure
Boyce et al. [BOYL]				Branch and Bound	(10 nodes, 45 arcs, 200 seconds)
3. Ridley [RID1]	1.2.3.1	Discrete	Minimize routing costs subject to a construction budget constraint	Branch and Bound	(12 nodes, ?, ?)
4. Stairs [STAI]	see section 1.2.3	Discrete	Minimize total construction and routing costs	Interactive computer system	(35 nodes, ?, ?)

TABLE 1.7.1  
SUMMARY OF NETWORK DESIGN PROBLEMS (Cont.)

AUTHORS	EQUATION NUMBER	ARC CAPACITY VARIABLES	OBJECTIVE	SOLUTION ALGORITHM	(# OF NODES IN TEST NETWORK, # OF ARCS, COMP. TIME) COMPUTATIONAL EXPERIENCE
5. Roberts and Funk [ROB1]	1.3.1	Discrete	Minimize total construction and routing costs subject to a construction budget constraint	Mixed Integer Linear Programming	NONE
6. Hershendorfer [HER1]	1.3.2	Discrete	Minimize construction costs subject to a routing cost constraint	Branch and Bound	(12 nodes, ?, ?)
7. Agarwal [AGAL]	1.3.3	Continuous	Minimize routing costs subject to a construction budget constraint	3 Methods: Simplex Method, Dantzig-Wolfe Decomposition and Boxstep	(24 nodes, 38 arcs, 840 seconds) for simplex method
8. Steenbrink [STE1, STE2]	1.3.4	Continuous	Minimize total routing and construction costs	Heuristic	(2,000 nodes, 6,000 arcs, 2,880 seconds)

TABLE 1.7.1  
SUMMARY OF NETWORK DESIGN PROBLEMS (Cont.)

AUTHORS	EQUATION NUMBER	ARC CAPACITY VARIABLES	OBJECTIVE	SOLUTION ALGORITHM	(# OF NODES IN TEST NETWORK, # OF ARCS, COMP. TIME) COMPUTATIONAL EXPERIENCE
9. Leblanc [LEB1]	1.4.2	Discrete	Minimize routing costs subject to a construction budget constraint	Branch and Bound	(24 nodes, 71 arcs, 135 seconds)
10. Barbier [BAR1] Haubrich [STE2]	1.4.3	Discrete	Minimize total routing and construction costs	Heuristic	(36 nodes, 80 arcs, ?) for Barbier, (1,250 nodes, 8,000 arcs, 2,400 seconds) for Haubrich
11. Gomory and Hu [GOM1], Chien [CHI1]	1.5.1	Continuous	Minimize construction costs	Combinatorial Algorithm	NONE
12. Gomory and Hu [GOM2]	1.5.2	Continuous	Minimize construction costs	Dantzig-Wolfe Decomposition	NONE
13. Gomory and Hu [GOM3]	1.5.5	Continuous	Minimize construction costs	Combinatorial Algorithm-Shortest Path Procedure	NONE
14. Gomory and Hu [GOM3]	1.5.6	Continuous	Minimize construction costs	Dantzig-Wolfe Decomposition	(10 nodes, ?, 600 seconds)

TABLE 1.7.1

## SUMMARY OF NETWORK DESIGN PROBLEMS (Cont.)

AUTHORS	EQUATION NUMBER	ARC CAPACITY VARIABLES	OBJECTIVE	SOLUTION ALGORITHM	(# OF NODES IN TEST NETWORK, # OF ARCS, COMP. TIME) COMPUTATIONAL EXPERIENCE
15. Chou and Frank [CHO1]	1.5.7	Discrete	Minimize construction costs	Combinatorial Algorithm	NONE
16. Frank and Chou [FRA1, FRA2]	1.5.8	Discrete	Minimize construction costs	Combinatorial Algorithm	NONE
17. Steiglitz et al. [STE3]	see section 1.5	Discrete	Minimize construction costs	Heuristic	(58 nodes, ?, 720 seconds)
18. Hu [HU1] Fulkerson [FUL1]	1.6.1	Continuous	Minimize routing costs subject to a construction budget constraint	Combinatorial Algorithm	NONE
19. Christofides and Brooker [CHR1]	1.6.2	Discrete	Minimize routing costs subject to a construction budget constraint	Branch and Bound	(50 nodes, 55 arcs, 35 seconds)
20. Bansal and Jacobsen [BANI]	see section 1.6	Continuous	Minimize routing costs subject to a construction budget constraint	Generalized Benders Decomposition	NONE

most computational experience has been limited to small test networks. The only exception to this is the work of Christofides and Brooker [CHR1] with (1.6.2). They were able to solve medium sized networks (containing up to 60 nodes) in a reasonable amount of time. In view of this experience, and also the tendency of branch and bound computation times to increase greatly with problem size, it appears unlikely that branch and bound will be useful in solving large network improvement problems. Also, it does not seem that branch and bound will be any more successful if applied to network synthesis problems. Network synthesis problems generally have many more capacity variables to set than corresponding network improvement problems of the same size.

Mathematical programming techniques have been used to solve a variety of network design problems (see (1.2.2.1), (1.3.3), (1.5.2) and (1.5.6), also see Bansal and Jacobsen's version of (1.6.1)). Most network design problems require formulations that have large numbers of constraints and variables. Various types of decomposition procedures (Benders decomposition, generalized Benders decomposition, Dantzig-Wolfe decomposition, and Boxstep have been applied to these large problems. Most of the computational results for these techniques have not been encouraging. However, in view of the computational success achieved by Geoffrion and Graves [GE02] using Benders decomposition with strengthened cuts, the application of this technique



bears reconsideration. The author is currently studying the application of Benders decomposition to the infinite capacity network synthesis problems described in section 1.2.1. The recent advances in solving user equilibrium flow routing problems [GOL3, LEB2, NGU1] may make feasible the application of generalized Benders decomposition to the network improvement problems with user equilibrium routing described in section 1.4. In the application of the technique to these problems, the subproblems generated are user equilibrium flow routing problems which can now be solved efficiently. Also, recent advances in large scale system methodology, such as list processing techniques and network flow algorithms, may have some impact on the size of problems that can be solved practically. The reader may consult a recent report by Magnanti [MAG1] for a survey of these new advances.

Heuristic procedures have been the most frequently applied solution technique (see (1.2.1.1), (1.2.1.2), (1.2.2.1), (1.3.4), (1.4.3) and (1.5.9)). Most heuristic procedures proceed in the following way: starting from some initial feasible solution a local transformation is used to obtain another feasible solution which, hopefully, has lower cost. This continues until a local optimum is reached where the transformation cannot produce another feasible solution which might have lower cost. The only heuristic procedure which uses a different approach is the procedure for

(1.3.4) where the evaluation of a feasible solution's cost requires the solution of a difficult multi-commodity flow problem. Heuristic procedures have obtained non-optimal solutions to network design problems that would be impractical to solve with any other technique. However, it is not known how close these generated solutions are to the optimal solutions. Recently, Cornuejols, Fisher, and Nemhauser [COR1] have analyzed the worst case behavior of heuristics that solve uncapacitated facility location problems. It may be possible to perform a similar analysis for some network design heuristics.

1.8 Acknowledgements

The author wishes to thank Professor Thomas L. Magnanti for his encouragement and assistance in the preparation of this report. The author also wishes to thank Eileen McEnaney for her help in typing and editing this report. Preparation of this paper was supported by the Department of Transportation (Contract DOT-TSC-1058).

REFERENCES

- [AGA1] Agarwal, S.K., "Optimizing Techniques for the Interactive Design of Transportation Networks Under Multiple Objectives," Ph.D. Thesis, Department of Civil Engineering, Northwestern University, 1973.
- [AS1] Assad, A., "Multicommodity Network Flows - A Survey," Working Paper OR-045-75, Operations Research Center, M.I.T., December, 1975.
- [BAI1] Bair, B.O., and G.L. Hefley, "Fixed Charge Transportation Problem: A Survey," Technical Report No. 19, The Institute of Urban and Regional Research, University of Iowa, May, 1973.
- [BAL1] Balinski, M.L., "Fixed Cost Transportation Problem," Naval Research Logistics Quarterly, Vol. 8, January, 1961, pp. 41-54.
- [BAN1] Bansal, P., and S. Jacobsen, "An Algorithm for Optimizing Network Flow Capacity Under Economics of Scale," JOTA, Vol. 15, 1975, pp. 565-586.
- [BAR1] Barbier, M., "Le future réseau de transports en Région de Paris," Cahiers de l'Institut d'Aménagement et d'Urbanisme de la Région Parisienne, Vol. 4-5, No. 4, 1966.
- [BEL1] Bellmore, M., and G.L. Nemhauser, "The Travelling Salesman Problem - A Survey," Operations Research, Vol. 16, 1968, pp. 538-558.
- [BEN1] Benders, J.F., "Partitioning Procedures for Solving Mixed-Variables Programming Problems," Numerische Mathematik, Vol. 4, 1962, pp. 238-252.
- [BIL1] Billheimer, J., and P. Gray, "Network Design with Variable Cost Elements," Transportation Science, Vol. 7, 1973, pp. 49-74.

- [BOE1] Boesch, F.T., and R.E. Thomas, "Optimal Damage Resistant Nets," Proceedings of the IEEE International Conference on Communications, Vol. 4, 1968, pp. 688-693.
- [BOY1] Boyce, D.E., A. Farhi, and R. Weischedel, "Optimal Network Problem: A Branch-and-Bound Algorithm," Environment and Planning, Vol. 5, 1973, pp. 519-533.
- [BRA1] Braess, D., "Uber ein Paradoxen der Verkehrsplanung," Unternehmensforschung, Vol. 12, 1968, pp. 258-268.
- [CAR1] Carter, E.C., and J.R. Stowers, "Model for Funds Allocation for Urban Highway Systems Capacity Improvements," Highway Research Record, Vol. 20, 1963, pp. 84-102.
- [CHA1] Chan, Y.P., "Optimal Travel Time Reduction in a Transport Network: An Application of Network Aggregation and Branch-and-Bound Techniques," Technical Report No. R69-39, Department of Civil Engineering, M.I.T., July, 1969.
- [CHI1] Chien, R.T., "Synthesis of a Communication Net," IBM Journal of Research and Development, Vol. 4, 1960, pp. 311-320.
- [CHO1] Chou, W., and H. Frank, "Survivable Communication Networks and The Terminal Capacity Matrix," IEEE Transportation Circuit Theory, Vol. CT-17, 1970, pp. 192-197.
- [CHR1] Christofides, N., and P. Brooker, "Optimal Expansion of an Existing Network," Mathematical Programming, Vol. 6, 1974, pp. 197-211.
- [COR1] Cornuejols, G., M.L. Fisher, and G.L. Nemhauser, "An Analysis of Heuristics and Relaxations for the Uncapacitated Location Problem," Technical Report No. 271, Department of Operations Research, Cornell University, Ithaca, New York, 1975.
- [DAN1] Dantzig, G.B., Linear Programming and Extensions, Princeton University Press, Princeton, N.J., 1963.

- [DRE1] Dreyfus, S.E., "An Appraisal of Some Shortest-Path Algorithms," Operations Research, Vol. 17, May-June, 1969, pp. 395-412.
- [DRE2] Dreyfus, S.E., and R.A. Wagner, "The Steiner Problem in Graphs," Networks, Vol. 1, 1972, pp. 195-207.
- [EFR1] Efroymsen, M.A., and T.L. Ray, "A Branch-Bound Algorithm for Plant Location" Operations Research, Vol. 14, 1966, pp. 361-368.
- [FR1] Francis, R.L., and J.M. Goldstein, "Location Theory: A Selective Bibliography," Operations Research, Vol. 22, 1974, pp. 400-410.
- [FRA1] Frank, H., and W. Chou, "Connectivity Considerations in the Design of Survivable Networks," Technical Memo, Office of Emergency Preparedness, Washington, D.C., January, 1969.
- [FRA2] \_\_\_\_\_ and \_\_\_\_\_, "Connectivity Considerations in the Design of Survivable Networks," IEEE Transactions On Circuit Theory, Vol. CT-17, 1970,
- [FRA3] \_\_\_\_\_ and I. Frisch, Communication, Transmissions, and Transportation, Addison-Wesley, Reading, Ma., 1971.
- [FRA4] \_\_\_\_\_, private communication to K. Steiglitz, P. Weiner and D. Kleitman, cited in [STE3].
- [FUL1] Fulkerson, D.R., "Increasing the Capacity of a Network: The Parametric Budget Problem," Management Science, Vol. 5, 1959, pp. 472-483.
- [FUN1] Funk, M.L., and F.A. Tillman, "A Dynamic Programming Approach to Optimal Construction Staging," American Society of Civil Engineers, Journal of the Highway Division, Vol. 94, 1968, pp. 255-265.
- [GEO1] Geoffrion, A.M., "Generalized Benders Decomposition," Journal of Optimization Theory and Application, Vol. 10, 1972, pp. 237-260.
- [GEO2] \_\_\_\_\_ and G. Graves, "Multicommodity Distribution System Design by Benders Decomposition," Management Science, Vol. 20, 1974, pp. 822-844.

- [GOL1] Goldman, A.J., and G.L. Nemhauser, "A Transport Improvement Problem Transformable to a Best-Path Problem," Transportation Science, Vol. 1, 1967, pp. 295-307.
- [GOL2] Golomb, S.W., and L.D. Baumert, "Backtrack Programming," Journal of ACM, Vol. 12, 1965, pp. 516-524.
- [GOL3] Golden, B., "A Minimum-Cost Multi-Commodity Network Problem Concerning Imports and Exports," Networks, Vol. 5, 1975, pp. 331-356.
- [GOM1] Gomory, R., and T. Hu, "Multi-Terminal Network Flows," Journal of SIAM, Vol. 9, 1961, pp. 551-570.
- [GOM2] \_\_\_\_\_ and \_\_\_\_\_, "An Application of Generalized Linear Programming to Network Flows," Journal of SIAM, Vol. 10, 1962, pp. 260-283.
- [GOM3] \_\_\_\_\_ and \_\_\_\_\_, "Synthesis of a Communication Network," Journal of SIAM, Vol. 12, 1964, pp. 348-369.
- [HAK1] Hakimi, S.L., "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph," Operations Research, Vol. 12, 1964, pp. 450-459.
- [HAK2] \_\_\_\_\_, "Optimum Distribution of Switching Centers in a Communications Network and Some Related Graph Theoretic Problems," Operations Research, Vol. 13, 1965, pp. 462-475.
- [HAK3] \_\_\_\_\_, "Steiners' Problem in a Graph and Its Implications," Networks, Vol. 1, 1971, pp. 113-133.
- [HAM1] Hammer, P.L., "Increasing the Capacity of a Network," CORS Journal, Vol. 7, 1969, pp. 83-91.
- [HAR1] Harary, F., "The Maximum Connectivity of a Graph," Proceedings of the National Academy of Science U.S., Vol. 48, 1962, pp. 1142-1146.
- [HAR2] \_\_\_\_\_, Graph Theory, Addison-Wesley, Reading, Mass., 1969.

- [HER1] Hershendorfer, A.M., "Optimal Routing of Urban Traffic," Ph.D. Thesis, Department of City and Regional Planning, M.I.T., May, 1965.
- [HES1] Hess, M.L., "An Observation on the Branch and Bound Method of Price," CORS Journal, Vol. 6, No. 1, 1968, pp. 58-59.
- [HIR1] Hirsch, W.M., and G. Dantzig, "The Fixed Charge Problem," Naval Research Logistics Quarterly, Vol. 15, 1968, pp. 413-424.
- [HUT1] Hu, T.C., Integer Programming and Network Flows, Addison-Wesley, Reading, Mass., 1969.
- [HUT2] \_\_\_\_\_, "Optimum Communication Spanning Trees," SIAM Journal of Computing, Vol. 3, No. 3, September, 1974, pp. 188-195.
- [KAR1] Karp, R.M., "Reducibility Among Combinatorial Problems," in Complexity of Computer Computations, edited by R.E. Miller and J.W. Thatcher, Plenum Press, New York, 1972, pp. 85-104.
- [KAR2] \_\_\_\_\_, "On the Computational Complexity of Combinatorial Problems," Networks, Vol. 5, No. 1, 1975, pp. 45-68.
- [KEN1] Kennington, J., "Multi-Commodity Flows: A State-of-the-Art Survey of Linear Models and Solution Techniques," Technical Report IEOR 75-014, Southern Methodist University, 1975.
- [KN01] Knödel, W., "Graphentheoretische Methoden und ihre Anwendungen," Springer-Verlag, Berlin, Heidelberg and New York, 1969.
- [KRO1] Krolak, P.D., W.J. Felts, and G. Marble, "A Man-Machine Approach Toward the Travelling Salesman Problem," Commun. of the ACM, Vol. 14, 1971, pp. 327-334.
- [KRO2] \_\_\_\_\_, \_\_\_\_\_, and J. Nelson, "A Man-Machine Approach Toward Solving the Generalized Truck-Dispatching Problem," Transportation Science, Vol. 6, No. 2, 1972, pp. 149-169.



- [KRU1] Kruskal, J.B., "On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem," Proceedings of the American Mathematical Society, Vol. 7, 1956, pp. 48-50.
- [KUH1] Kuhn, H.W., and W.J. Baumol, "An Approximative Algorithm for the Fixed-Charge Transportation Problem," Naval Research Logistics Quarterly, Vol. 9, No. 1, 1962, pp. 1-16.
- [LEB1] Leblanc, L.J., "An Algorithm for the Discrete Network Design Problem," Transportation Science, Vol. 9, No. 3, 1975, pp. 183-287.
- [LEB2] \_\_\_\_\_, E.K. Morlok, and W.P. Pierskalla, "An Efficient Approach to Solving the Road Network Equilibrium Traffic Assignment Problem," Transportation Research, Vol. 9, 1975, pp. 309-318.
- [LIN1] Lin, S., "Computer Solutions of the Travelling Salesman Problem," Bell System Technical Journal, Vol. 44, 1965, pp. 2245-2269.
- [MAG1] Magnanti, T.L., "Optimization for Sparse Systems," Technical Report No. 119, Operations Research Center, M.I.T., 1975, to appear in Sparse Matrix Computations.
- [MAN1] Manne, A.S., "Plant Location Under Economies of Scale-Decentralization and Computation," Management Science, Vol. 11, 1964, pp. 213-235.
- [MAR1] Marsten, R.E., W.W. Hogan, and J.W. Blankenship, "The Boxstep Method for Large-Scale Optimization," Operations Research, Vol. 23, No. 3, 1975, pp. 389-405.
- [MUR1] Murchland, J.D., "Braess' Paradox of Traffic Flow," Transportation Research, Vol. 4, 1970, pp. 391-394.
- [NGU1] Nguyen, S., "An Algorithm for the Traffic Assignment Problem," Transportation Science, Vol. 8, 1974, pp. 203-216.
- [OCH1] Ochoa-Rosso, F., "Applications of Discrete Optimization Techniques to Capital Investment and Network Synthesis Problems," Research Report R68-42, Department of Civil Engineering, M.I.T., June, 1968.

- [OCH2] \_\_\_\_\_, and A. Silva, "Optimum Project Addition in Urban Transportation Networks Via Descriptive Traffic Assignment Models," Research Report R68-44, Department of Civil Engineering, M.I.T., June, 1968.
- [PRI1] Price, W.L., "Increasing the Capacity of a Network Where the Costs are Non-Linear: A Branch-and-Bound Algorithm," CORS Journal, Vol. 5, 1967, pp. 110-114.
- [REV1] Revelle, C., D. Marks, and J. Liebman, "An Analysis of Private and Public Sector Location Models," Management Science, Vol. 16, 1970, pp. 692-707.
- [RID1] Ridley, T.M., "Investment in a Network to Reduce the Length of the Shortest Route," Proceedings of the Third International Symposium on the Theory of Traffic Flow, Elsevier, New York, June, 1965, pp. 235-236.
- [RID2] \_\_\_\_\_, "An Investment Policy to Reduce the Travel Time in a Transportation Network," Transportation Research, Vol. 2, 1968, pp. 409-424.
- [ROB1] Roberts, P.O., and M.L. Funk, "Toward Optimum Method of Link Addition in Transportation Networks," Department of Civil Engineering, M.I.T., September, 1964.
- [SCH1] Schwartz, J.G., "Network Flow and Synthesis Models for Transportation Planning: A Survey," Research Report R68-49, Department of Civil Engineering, M.I.T., June, 1968.
- [SC01] Scott, A.J., "A Programming Model of an Integrated Transportation Network," Paper of the Regional Science Association, Vol. 19, 1967, pp. 215-222.
- [SC02] \_\_\_\_\_, "The Optimal Network Problem: Some Computational Procedures," Transportation Research, Vol. 3, 1969, pp. 201-210.
- [SC03] \_\_\_\_\_, Combinatorial Programming, Spatial Analysis and Planning, Methuen & Co., Ltd., London, England, 1971.

- [SPI1] Spielberg, K., "On the Fixed Charge Transportation Problem," Proceedings of the Nineteenth National Conference, ACM, 1964, pp. A1.1-1 to A1.1-13.
- [STA1] Stairs, S., "Selecting a Traffic Network," Journal of Transport Economics and Policy, Vol. 2, 1968, pp. 218-231.
- [STE1] Steenbrink, P.A., "Transport Network Optimization in the Dutch Integral Transportation Study," Transportation Research, 1974, Vol. 8, pp. 11-27.
- [STE2] \_\_\_\_\_, Optimization of Transport Networks, John Wiley & Sons, London, England, 1974.
- [STE3] Steiglitz, K., P. Weiner, and D.J. Kleitman, "Design of Minimum Cost Survivable Networks," IEEE Transactions On Circuit Theory, Vol. CT-16, 1969, pp. 455-460.
- [TOM1] Tomlin, J., "Minimum Cost Multi-Commodity Network Flow," Operations Research, Vol. 14, No. 1, 1966, pp. 45-51.
- [WAR1] Wardrop, J.G., "Some Theoretical Aspects of Road Traffic Research," Proceedings of the Institute of Civil Engineering, Vol. 1, Part II, 1952, pp. 325-362.
- [WOL1] Wollmer, R., "Removing Arcs from a Network," Operations Research, Vol. 12, 1964, pp. 934-940.

APPENDIX 1

ADDITIONAL REFERENCES FOR COMMUNICATION

NETWORK SYNTHESIS PROBLEMS

- [AY01] Ayoub, J.N., and I.T. Frisch, "Optimally Invulnerable Directed Communication Networks," IEEE Transactions On Com. Tech., Vol. Com. 18, 1970, pp. 484-489.
- [BOE1] Boesch, F.T., and R.E. Thomas, "On Graphs of Invulnerable Communication Nets," IEEE Transactions On Circuit Theory, Vol. CT-17, No. 2, May, 1970.
- [BOE2] \_\_\_\_\_, and A.P. Felzer, "On the Minimum m Degree Vulnerability Criterion," IEEE Transactions On Circuit Theory, Vol. CT-18, No. 2, 1971, pp. 224-228.
- [BOE3] \_\_\_\_\_, and \_\_\_\_\_, "On the Invulnerability of the Regular Complete K-Partite Graphs," SIAM Journal of Applied Mathematics, Vol. 20, 1971, pp. 76-82.
- [BOE4] \_\_\_\_\_, and \_\_\_\_\_, "A General Class of Invulnerable Graphs," Networks, Vol. 2, No. 3, 1972, pp. 261-283.
- [CHE1] Chen, C., "Realizability of Communication Nets: An Application of the Zadeh Criterion," IEEE Transactions On Circuits and Systems, Vol. CAS-21, 1974, pp. 150-151.
- [DEO1] Deo, N., and S.L. Hakimi, "Minimum Cost Increase of the Terminal Capacity of a Communication Network," IEEE Transactions On Com. Tech., Vol. COM-14, 1966, pp. 63-64.
- [EDM1] Edmonds, J., "Existence of K-Edge Connected Ordinary Graphs with Prescribed Degrees," Journal of Research National Bureau of Standards, Vol. B, No. 68B, 1964, pp. 73-74.
- [ESA1] Esau, L., and K. Williams, "A Method for Approximating the Optimal Network," IBM System Journal, Vol. 5, 1966, pp. 142-147.

- [FRI1] Frisch, I.T., and M. Malek-Zavarei, "Vulnerability of Directed Communication Nets," IEEE Transactions On Circuit Theory, Vol. CT-19, 1972, pp. 283-284.
- [FRI2] \_\_\_\_\_, and D.K. Sen, "Algorithm for Synthesis of Oriented Communication Nets," IEEE Transactions On Circuit Theory, Vol. CT-14, 1967, pp. 370-379.
- [FUL1] Fulkerson, D., and L.S. Shapley, "Minimal K-Arc Connected Graphs," Networks, Vol. 1, 1971, pp. 91-98.
- [HAK1] Hakimi, S.L., "An Algorithm for Construction of the Least Vulnerable Communication Network," IEEE Transactions On Circuit Theory, Vol. CT-16, 1969, pp. 229-230.
- [SOU1] Soukup, J., "On Minimum Cost Networks with Nonlinear Costs," SIAM Journal of Applied Mathematics, Vol. 29, 1975, pp. 571-581.
- [TAN1] Tang, D.T., "Communication Networks with Simultaneous Flow Requirements," IEEE Transactions on Circuit Theory, Vol. CT-9, 1962, pp. 176-182.
- [TAN2] Tang, D.T., and R.T. Chien, "Analysis and Synthesis Techniques of Oriented Communication Nets," IRE Transactions On Circuit Theory, Vol. CT-8, 1961, pp. 39-44.
- [WAN1] Wang, D.L., and Kleitman, D.J., "On the Existence of n-connected Graphs with Prescribed Degrees ( $n \geq 2$ )," Networks, Vol. 3, 1973, pp. 225-240.
- [WIN1] Wing, O., and R.T. Chien, "Optimal Synthesis of a Communication Net," IRE Transactions On Circuit Theory, Vol. CT-8, 1961, pp. 44-49.

