CWI Tracts

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

The CWI is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

CWI Tract 25

Tree network and planar rectilinear location theory

A.J.W. Kolen

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

1980 Mathematics Subject Classification:
90C10, 05C05, 05C70.
ISBN 90 6196 300 1

PREFACE

When I was studying mathematics at the University of Technology in Eindhoven I became interested in combinatorics and operations research. By the time I finished my study I looked for an opportunity to combine both my interests. This became possible when prof. dr. J.H. van Lint (my supervisor at the university) told me of a position which was available at the Department of Operations Research of the Mathematisch Centrum (now the Centre for Mathematics and Computer Science (CWI)) in Amsterdam. On July 1, 1978 I started to work on combinatorial optimization at the Mathematisch Centrum. Jan Karel Lenstra and Alexander Rinnooy Kan suggested location theory as a possible area for research. I am very grateful to them for this suggestion because location theory turned out to be a nice and fruitful area to work in.

Some results which were obtained during the first three years at the Mathematisch Centrum were written down in my doctoral thesis in January 1982 under the supervision of prof. dr. G. de Leve and Jan Karel Lenstra.

These results would not have been possible without the opportunity given to me by the Mathematisch Centrum to study combinatorial optimization. Special thanks also go to Andries Brouwer who helped me prove one of the main results and to Lex Schrijver who increased my interest in combinatorial optimization and who is largely responsible for my current interest in polyhedral combinatorics. I am looking forward to many stimulating discussions in the future.

This monograph is a revision of my doctoral thesis as a CWI Tract. I have benefited from discussions with Richard Francis, Tim Lowe, Pitu Mirchandani, Pierre Hansen, Martin Farber, Louis Hakimi and especially Arie Tamir.

I thank the CWI for the opportunity to publish this monograph in the series CWI Tracts and all those at the CWI who have contributed to its technical realization. I also thank the Erasmus University in Rotterdam (my current employer) for enabling me to write this revision.

CONTENTS

CHAPTER 0

INTRODUCTION

Location theory dates back to the beginning of the seventeenth century, when Fermat posed his famous problem: given three points in a plane, find a fourth point such that the sum of its distances to the three given points is minimum. This problem was generalized by Simpson in his "Doctrine and Application of Fluxions" (London 1750), who asked for a point with minimum weighted sum of distances to three given points. Almost all the work on location theory, however, has taken place between 1957 and the present. Papers on the subject appear in a remarkably diverse collection of periodicals in the field of operations research, management science, industrial and civil engineering, transportation and regional science, geography, economics and planning. As a consequence, many results have been rediscovered over the years and results obtained by researchers in one field were not known to those in other fields. The latter is especially true for results in graph theory and combinatorial optimization. As we will show in this monograph, results on perfect graphs and totally-balanced matrices have only very recently found their way into location theory and provide a uniform framework in which old as well as new results can be derived. Communication in the field of location theory has increased since ISOLDE I, the First International Symposium On Locational Decisions, held in Banff, Canada in April 1976, ISOLDE II, held in Skodborg, Denmark in June 1981, and ISOLDE III held in Boston, USA in June 1984.

In the location models described in this monograph there exists a set of clients who have a demand for services or commodities. Facilities will be established to serve these clients. Facilities are assumed to have sufficient capacity to serve all clients.

We treat two fields of location theory. The first one is *discrete location theory* in which we assume that the underlying structure is a graph we will concentrate on tree graphs. The *distance* between two points on the

graph is defined to be the length of the shortest path connecting them. The second field is *planar location theory* in which the underlying structure is the plane. As the distance between two points we take the rectilinear distance.

We consider two types of costs. First we have the *setup cost*, i.e., the cost of establishing a facility. The second cost is the *transportation cost*, i.e., the cost of transporting the services between a facility and a client. The transportation cost is assumed to be a nondecreasing function of the distance between a client and the facility serving it.

There are two types of communications between clients and facilities. Normally there is only communication between a client and the closest facility. Whenever in a location problem there is communication between clients and more than one facility as well as between facilities themselves we call this a location problem with *mutual communication*.

Three different objective functions are considered: minimize the maximum transportation cost (*center problems*) minimize the sum of the transportation cost (*median problems* and minimize the sum of the setup cost and the transportation cost (*plant location problems*).

Only one constraint will be considered, namely the *budget constraint*, i.e., an upperbound on the total setup cost. In case of equal setup cost this constraint translates into an upperbound on the number of facilities to be established. This will usually be reflected in the name of the location problem (like in the "p-center problem" where p is the number of facilities to be established).

The theory of computational complexity provides the tools to make a formal distinction between *well-solved* problems, which are solvable by an algorithm whose running time is bounded by a polynomial function of problem size, and NP-*hard* problems, for which the existence of such an algorithm is highly unlikely. The exposition below is adapted from LAWLER and LENSTRA (1982). Computational complexity theory deals primarily with *decision* problems, which require a yes/no answer, rather than with optimization problems. By way of example consider the following decision problem that will occur in Chapters 3 and 4:

CLIQUE

Instance: A graph $G = (V,E)$ and an integer c.

Question: Does there exist a subset $C \subset V$ of cardinality c such that $[j,k] \in E$ if $\{j,k\} \subset C$?

An instance of a decision problem is said to be *feasible* if the question can be answered affirmatively. Feasibility is usually characterized by the existence of an associated *structure* which satisfies a certain property. E.g., in the case of CLIQUE the structure is a set of c pairwise adjacent vertices.

Two important problem classes are defined as follows. A decision problem is in the class P if, for any instance, one can determine its feasibility or infeasibility in polynomial time. It is in the class NP if, for any instance, one can determine in polynomial time whether a given structure affirms its feasibility. E.g., CLIQUE is a member of NP, since for any clique C one can verify that all its vertices are pairwise adjacent in $O(c^2)$ time.

A problem R is *reducible* to a problem Q if for any instance of problem R a corresponding instance of problem Q can be constructed in polynomial time such that solving the latter instance will solve the instance of R as well. In the class NP, a class of NP-*complete* problems is identified which have the property that: (i) they belong to NP; and (ii) every other problem in NP is reducible to them. A polynomial-time algorithm for an NP-complete problem Q could be used to solve any problem in NP in polynomial time by reducing it to Q. Since the class NP contains many notorious combinatorial problems for which, in spite of considerable research efforts, no polynomial-time algorithms have been found so far, it is very unlikely that $Q \in P$ for any NP-complete Q.

In dealing with the computational complexity of *optimization* problems, one usually reformulates the problem of finding a feasible solution of, say, minimum value as the problem of deciding whether there exists a feasible solution with value at most equal to a given threshold. If this decision problem is NP-complete, then the optimization problem is said to be NP-*hard* in the sense that the existence of a polynomial-time algorithm for its solution would imply that $P = NP$.

Most location problems on graphs, where the distance between two points is defined as the length of a shortest path between these two points, have been proved to be NP-hard. This justifies the investigation of restricted versions of these problems. We will make the assumption that the graph does not contain cycles, i.e., that it is a tree. It turns out that most location problems on trees can be solved in polynomial time. It is not well understood what it is about tree location problems that makes them so tractable. We will partially answer this question by relating tree location problems

to convexity, perfect graphs and totally-balanced matrices.

The research on location problems has frequently been justified as having direct applicability to a wide variety of real world location and logistical problems. Yet very few applications of location models are discussed or even mentioned in the literature. Although there are indications that the techniques are incidentally being applied both in the private and public sector, it seems clear that if models and solution algorithms were commonly used there would be more evidence of this in the literature. There are, of course, built-in biases against the publication of applications. Most researchers are theoreticians and not practitioners, and that practitioners have a lesser propensity to publish. There is also a lag effect between invention and adoption of algorithms. Although tree location problems have a nice mathematical structure, it remains to be tested how useful the developed algorithms are in more complex real world problems. They could be used for example in branch-and-bound algorithms as well as in heuristics for more general problems. Not much work in this direction has been done.

At this point we give a preview of what follows. In Chapter 1 we treat center problems with equal setup costs. In this case the number of facilities to be located is prespecified. We introduce the concept of a chordal graph and indicate how it can be used to derive duality results for center problems including the p-center problem and the round-trip p-center problem.

In Chapter 2 we treat center problems with unequal setup costs. It is shown that these problems can be solved as a sequence of set covering problems defined on special structured matrices (matrices in standard greedy form). A polynomial time algorithm to solve this type of set covering problem is presented. The simple plant location problem on a tree is shown to be solvable using this algorithm.

In Chapters 3 and 4 we mention problems with mutual communication. We treat the p-median problem in Chapter 3 and the p-center problem in Chapter 4. Convexity of the distance function on trees will be used to develop polynomial time algorithms to solve the problems with mutual communication on trees.

In Chapter 5 we show how Farkas' Lemma and generalizations of it can be used to solve center problems in the plane using rectilinear distances. An efficient algorithm to find all efficient points in the plane is presented.

In Chapter 6 we treat totally-balanced matrices and chordal graphs. We discuss the equivalence between chordal graphs and intersection graphs of subtrees of a tree. The class of totally-balanced matrices is defined and shown to be identical to the class of matrices which can be transformed into standard greedy form. Some extensions of the set covering problem on matrices in standard greedy form (see Chapter 2) are discussed.

We assume the reader of this monograph to be familiar with the basic principles of mathematical programming. Allthough concepts of graph theory are defined when needed, some familiarity with graph theory will be useful. Each chapter of this monograph can be read independently of the others; cross references are restricted as much as possible.

CHAPTER 2

1. CENTER PROBLEMS WITH EQUAL SETUP COST

In the first four chapters of this monograph we discuss discrete location problems. In contrast to *planar* location problems in which we assume that clients and facilities are located in the plane, we assume in discrete location problems that the underlying structure is a graph. One can think of a graph as a road network.

A graph is defined to be a pair (V,E) where V is a set of elements called *vertices*, and E is a set of unordered pairs of vertices, called *edges*. Two vertices v_i and v_j are *adjacent* if $[v_i,v_j] \in E$; v_i and v_j are the *endpoints* of the edge $[v_i,v_j]$. A q-*chain* is a sequence $[e_1,e_2,\ldots,e_q]$ of edges such that each edge e_i $(i = 2,\ldots,q-1)$ in the sequence has one endpoint in common with its predecessor in the sequence and its other endpoint with its successor in the sequence. The endpoint of $e_1(e_q)$ which is not an endpoint of $e_2(e_{q-1})$ is an *endpoint of the chain*. A *cycle* is a chain such that: (1) no edge appears twice in the sequence: and (2) the two endpoints of the chain are the same vertex. An *induced subgraph* of the graph G = (V,E) is defined by a subset $V_1 \subseteq V$ and all edges of E with both endpoints in V_1.

The graph $G = (\{v_1,v_2,v_3,v_4\},\{[v_1,v_2],[v_2,v_3],[v_3,v_4],[v_4,v_1],[v_1,v_3]\})$ is given by Figure 1.1. The chain $[[v_1,v_2],[v_2,v_3],[v_3,v_4],[v_4,v_1]]$ is a cycle.

Given a graph G = (V,E) each edge $e \in E$ will be given a positive length. If we consider an edge $[v_i,v_j]$ as a line segment, then we can identify a *point* on the edge by defining its distance to v_i or v_j. For the graph of Figure 1.1 the edge lengths are as indicated. The point x on the graph on $[v_1,v_2]$ is at distance 1 from v_1, y is the point on $[v_1,v_3]$ at distance 6 from v_1. The *distance* d(x,y) between two points x and y on the graph is defined to be the length of a shortest path P(x,y) between x and y. The *distance* d(y,X) between a point y on G and a set of points X on G (denoted by $X \subseteq G$)

is defined to be the distance between y and the closest point in the set X. In Figure 1.1 we have $d(x,y) = 4$ and $d(v_2,\{x,y\}) = 2$.

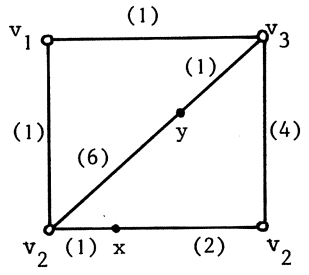

Figure 1.1. Example of a graph.

In this chapter we will assume that the graph is a *tree*, i.e., it is connected and has no cycles. In the two location problems we discuss in this chapter facilities may be established anywhere on the tree. Each facility has a sufficient capacity to serve all clients. We have an upperbound, say p, on the number of facilities. This upperbound comes from the fact that in case of equal setup costs (i.e., the cost of establishing a facility) a budget constraint translates into an upperbound on the number of facilities.

In both problems considered we locate p facilities so as to minimize the maximum transportation cost. The first problem is the p-*center problem* in which clients are located at the vertices of the tree and the transportation cost is a monotone increasing function of the distance between the client and the closest established facility. The second problem is the *round-trip p-center problem* in which a client corresponds to a pair of vertices and the transportation cost is a monotone increasing function of the round-trip distance. The *round-trip distance* is the sum of the distances between the facility serving the client and the vertices corresponding to it, and the distance between the vertices. This model is appropriate when a vehicle located at the facility has to travel to one vertex to pick up goods and deliver them to the other vertex before returning to the facility. The round-trip distance is the total distance traveled by the vehicle. The round-trip 1-center problem with linear transportation cost was solved by Chan and Francis (1976). The same problem in the rectilinear plane was solved by Chan and Hearn (1977).

Let $T = (V,E)$ be a tree with n vertices v_i, $i = 1,\ldots,n$. In the p-*center problem* we are locating p facilities anywhere on the tree so as

to minimize the maximum transportation cost. We can formulate this problem as

$$(1.2) \qquad \min_{X \subseteq T:\ |X| = p} \{ \max_{i=1,\ldots,n} \{ f_i(d(v_i,X)) \} \},$$

where X corresponds to the set of p facilities, and $f_i(d(v_i,X))$ is the transportation cost associated with client i located at vertex v_i. Note that the transportation cost is a monotone increasing function f_i of the distance between client i and the closest facility. Another way of formulating this problem is the following:

$$(1.3) \qquad \begin{aligned} &\min r \\ &\text{s.t. } f_i(d(v_i,X)) \le r, \ i = 1,2,\ldots,n, \\ &\qquad\qquad X \subseteq T, \\ &\qquad\qquad |X| = p. \end{aligned}$$

Since we are minimizing the value of r it will be equal to the maximum transportation cost.

We say that client i is *served within distance* r by a set of facilities X if $f_i(d(v_i,X)) \le r$. A problem related to the p-center problem is the *covering problem* in which for a given value of r we want to determine the minimum number of facilities such that each client is covered within distance r. A formulation of the covering problem is given by

$$(1.4) \qquad \begin{aligned} &\min |X| \\ &\text{s.t. } f_i(d(v_i,X)) \le r, \quad i = 1,2,\ldots,n, \\ &\qquad\qquad X \subseteq T. \end{aligned}$$

It is easy to see that the optimum value r_p of the p-center problem is the smallest valve r for which (1.4) has an optimum value less than or equal to p.

Let $J_i = \{a_i,b_i\}$ be a pair of vertices of the tree corresponding to client i, i = 1,2,...,m. The *round-trip distance* between client i and a facility at a point $x \in T$ is given by $d(a_i,x) + d(x,b_i) + d(a_i,b_i)$. Let u(x) be the unique point on the shortest path from a_i to b_i which is closest to x. (u(x) is unique since we have a tree). The round-trip distance is equal to $2d(a_i,b_i) + 2d(u(x),x)$. Hence d(u(x),x) the distance from x to the shortest path from a_i to b_i is the part in the round-trip distance

depending on the choice of x. Define

$$(1.5) \qquad D(J_i,x) := \tfrac{1}{2}[d(a_i,x) + d(b_i,x) - d(a_i,b_i)],$$

Note that $D(J_i,x) = d(u(x),x)$. Also define

$$(1.6) \qquad D(J_i,X) := \min_{x\in X} \{D(J_i,x)\}.$$

In the *round-trip p-center problem* we are locating p facilities on the tree so as to minimize the maximum transportation cost. We can formulate this problem as

$$(1.7) \qquad \min_{X\subseteq T:|X|=p} \{ \max_{i=1,\ldots,m} \{f_i(D(J_i,X))\}\},$$

where X denotes the set of p facilities, and $f_i(D(J_i,X))$ is the transportation cost associated with client i corresponding to J_i. Note that the transportation cost is a monotone increasing function f_i of the distance from the path corresponding to client i and the closest facility. Another way of formulating this problem is the following:

$$(1.8) \qquad \begin{aligned} &\min r \\ &\text{s.t. } f_i(D(J_i,X)) \le r, \quad i = 1,2,\ldots,m \\ &\qquad\qquad X \subseteq T, \\ &\qquad\qquad |X| = p. \end{aligned}$$

We say that client i is *served within distance* r by a set of facilities X if $f_i(D(J_i,X)) \le r$. A problem related to the round-trip p-center problem is the round-trip covering problem in which for a given value of r we want to determine the minimum number of facilities needed to serve each client within distance r. A formulation of the round-trip covering problem is given by

$$(1.9) \qquad \begin{aligned} &\min |X| \\ &\text{s.t } f_i(D(J_i,X) \le r, \quad i = 1,\ldots,m, \\ &\qquad\qquad X \subseteq T. \end{aligned}$$

It is easy to see that the optimum value r_p of the round-trip p-center problem is the smallest value r for which (1.9) has an optimum value less than

or equal to p.

We will now study some properties of trees which will enable us to derive a duality result for the covering problems (1.4) and (1.9). These results will then be used to derive duality results for the p-center problems.

We start by fixing one vertex of the tree, called the *root* r. For each vertex v there is a unique shortest path to the root, denoted by p(v,r). If v_j is adjacent to v_i and v_j lies on $p(v_i,r)$, then v_j is called the *parent* of v_i, and v_i is called a *child* of v_j. We index the vertices of the tree in such a way that a child has a smaller index than its parent. In the tree rooted at v_7 of Figure 1.10, v_1,v_2,v_3 are children of v_5.

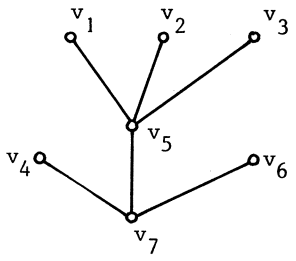

Figure 1.10. A rooted tree.

A *subtree* of a tree is any connected subset of points on the tree, not necessarily an induced subgraph. To prove the properties of subtrees we need in order to prove the duality results, we may assume without loss of generality that the subtrees are induced subtrees. If this is not the case, then we can add the endpoint of a subtree on an edge to the vertex set. With each subtree we associate a *root* defined to be the vertex with the largest index belonging to the subtree. Assume that T_i and T_j are subtrees of a tree T with roots $v_{r(i)}$ and $v_{r(j)}$ respectively, $r(i) < r(j)$. Then $v_{r(i)}$ will be on $p(v_{r(j)},x)$ for any point $x \in T_i$. Hence we have the following observation.

OBSERVATION 1.11. Let T_i and T_j be subtrees of a tree T with roots $v_{r(i)}$ and $v_{r(j)}$ respectively, $r(i) \leq r(j)$. Then $T_i \cap T_j \neq \emptyset$ iff $v_{r(i)} \in T_j$.

This observation enables us to prove our first result.

LEMMA 1.12 (Helly Property). *Let* $T_1,\ldots,T_m$ *be subtrees of a tree* T *such that* $T_i \cap T_j \neq \emptyset$ *for all* i,j. *Then* $\cap_{i=1}^{m} T_i \neq \emptyset$.

PROOF. Assume w.l.o.g. $r(1) \leq r(2) \leq .. \leq r(m)$. Then $T_1 \cap T_i \neq \emptyset$ implies $v_{r(1)} \in T_i$ for all i. (Observation 1.11). Hence $v_{r(1)} \in \cap_{i=1}^{m} T_i$. □

We define the *intersection graph* $G = (\{1,...,m\},E)$ of subtrees $T_1,...,T_m$ by $[i,j] \in E$ iff $T_i \cap T_j \neq \emptyset$. So we have an edge in the intersection graph if the corresponding two subtrees intersect.

A *chordal graph* is a graph with the property that every cycle of length at least four has a *chord*, i.e., an edge connecting two vertices which are not adjacent in the cycle. The graph in Figure 1.14 is an example of a chordal graph.

An *independent set* of a graph is a set of pairwise nonadjacent vertices. For example {1,4} is an independent set of the graph in Figure 1.14. The maximum cardinality of an independent set of a graph G is denoted by $\alpha(G)$.

A *clique* of a graph is a set of pairwise adjacent vertices. For example {3,4,5} is a clique of the graph in Figure 1.14.

A *clique cover* is a set of cliques such that each vertex is in at least one clique. For example {1,2,3},{3,4,5},{4,5,6} form a clique cover in Figure 1.14. The minimum cardinality of a clique cover is denoted by $\theta(G)$.

Let us consider an arbitrary independent set I and an arbitrary clique cover C of a graph G. Since any two vertices in the independent set are not adjacent they must be in different cliques of the clique cover. Hence $|I| \leq |C|$. We have just proved the following result

(1.13) Weak duality: $\alpha(G) \leq \theta(G)$ for every graph G.

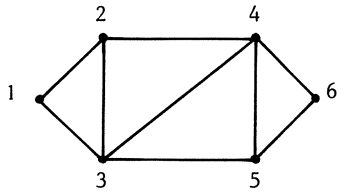

Figure 1.14. A chordal graph.

The next two Lemmas are needed to establish that for the intersection graph G of subtrees of a tree we have strong duality, i.e., $\alpha(G) = \theta(G)$.

LEMMA 1.15. *The intersection graph* $G = (\{1,...,m\},E)$ *of subtrees* $T_1,...,T_m$ *of a tree* T *is a chordal graph.*

PROOF. Without loss of generality assume $r(1) \leq r(2) \leq \ldots \leq r(m)$. Let C be a cycle of length at least four, let i be the smallest vertex in the cycle, and let j,k be the two vertices in the cycle adjacent to i. Note that since the length of the cycle is at least four, [j,k] is not an edge of the cycle. Since $i < j$, $i < k$ we have $v_{r(i)} \in T_j \cap T_k$ (Observation 1.11). Hence $[j,k] \in E$ is a chord of the cycle. □

If we take a closer look at the proof of Lemma 1.15 we see that for the intersection graph of subtrees of a tree there exists a numbering of the vertices (corresponding to nondecreasing roots) with the property that for all i the following holds:

(1.16) All vertices j(j>i) adjacent to i form a clique in the intersection graph.

A numbering of the vertices of the graph such that (1.16) holds for all i is called a *perfect scheme*. We shall prove in Chapter 6 that a graph has a perfect scheme iff it is a chordal graph. We shall also prove in Chapter 6 that the converse of Lemma 1.15 is also true, i.e., each chordal graph can be obtained as the intersection graph of subtrees of a tree. In the next Lemma we shall explore the fact that a perfect scheme exists to prove that for a chordal graph G $\alpha(G) = \theta(G)$. We restrict ourselves to intersection graphs.

LEMMA 1.17. *For the intersection graph* $G = (\{1,\ldots,m\},E)$ *of subtrees* $T_1,\ldots,T_m$ *of a tree* T *we have* $\alpha(G) = \theta(G)$.

PROOF. Assuming that $r(1) \leq r(2) \leq \ldots \leq r(m)$ we give a constructive proof of strong duality. We start with an empty independent set I and an empty set of cliques C. Then we repeat the following procedure until the graph has no more vertices: Add the smallest vertex, say i, to the independent set I; add the set consisting of i and all adjacent vertices to C (By property 1.16 these vertices form a clique); delete all vertices in the clique and their adjacent edges from G (this ensures that the next vertex we add to I is not adjacent to a vertex in I). After completion of the procedure we have an independent set I and a clique cover C. □

Note that all we used in Lemma 1.17 was the existence of a perfect scheme. So $\alpha(G) = \theta(G)$ for any chordal graph G.

Consider the following six induced subtrees of the tree given by Figure 1.10. The subtrees are given by their vertex set $T_1 = \{1\}$, $T_2 = \{1,2,5\}$, $T_3 = \{1,3,5,7\}$ $T_4 = \{5,6,7\}$, $T_5 = \{4,6,7\}$, $T_6 = \{6\}$. The intersection graph is given by Figure 1.14. A maximum independent set is $\{1,4\}$, and a minimum clique cover is given by $\{1,2,3\}$, $\{4,5,6\}$.

Let us now go back to the covering problems (1.4) and (1.9). Instead of (1.4) we consider the covering problem

$$(1.18) \qquad \begin{aligned} &\min |X| \\ &\text{s.t. } d(v_i,X) \le r_i, \quad i = 1,\ldots,n, \\ &\qquad X \subseteq T \end{aligned}$$

where $r_i \ge 0$, $i = 1,\ldots,n$, and instead of (1.9) we consider the covering problem

$$(1.9) \qquad \begin{aligned} &\min |X| \\ &\text{s.t. } D(J_i,X) \le r_i, \ i = 1,\ldots,m \\ &\qquad X \subseteq T, \end{aligned}$$

where $r_i \ge 0$, $i = 1,\ldots,m$. At the end of this chapter we shall show that the original covering problems (1.4) and (1.9) are equivalent to covering problems of type (1.18) and (1.19) respectively.

With respect to (1.18) let us define T_i to be the set of all points of T with the property that a facility at such a point can serve client i within distance r_i, i.e., $T_i = \{y \in T \mid d(v_i,y) \le r_i\}$, $i = 1,2,\ldots,n$. Note that $r_i \ge 0$ implies $T_i \neq \emptyset$.

LEMMA 1.20. $T_i \cap T_j \neq \emptyset$ *iff* $d(v_i,v_j) \le r_i + r_j$.

PROOF. If $T_i \cap T_j \neq \emptyset$, then there is an $x \in T_i \cap T_j$. Hence $d(v_i,x) + d(x,v_j) \le r_i + r_j$.

Conversely, let $d(v_i,v_j) \le r_i + r_j$. If $r_i \ge d(v_i,v_j)$, then $v_j \in T_i \cap T_j$. If $r_i < d(v_i,v_j)$, then define x to be the point on $P(v_i,v_j)$ at distance r_i from v_i. We have $x \in T_i$ and $d(v_j,x) = d(v_i,v_j) - d(v_i,x) = d(v_i,v_j) - r_i \le r_j$ so that $x \in T_j$. We conclude that $T_i \cap T_j \neq \emptyset$. □

If we construct the intersection graph $G = (\{1,\ldots,n\},E)$ corresponding to $T_1,\ldots,T_n$, then $[i,j] \in E$ iff $d(v_i,v_j) \le r_i + r_j$.

If two clients i and j can be served by the same facility at a point $x \in T$, then $x \in T_i \cap T_j$. It follows that all clients which can be served by the same facility form a clique in the intersection graph.

Conversely for any clique in the intersection graph G we can find a

point $x \in T$ which lies in the intersection of all subtrees corresponding to the clique (Helly property). It follows that all clients corresponding to that clique can be served by one facility, namely a facility located at the point x.

We reached the following conclusion.

CONCLUSION 1.21. There is a 1-1 correspondence between cliques in the intersection graph $G = (\{1,\ldots,n\},E)$ of $T_1,\ldots,T_n$, where $T_i = \{y \in T \mid d(v_i,y) \leq r_i\}$, $i = 1,2,\ldots,n$, and clients which can be served within distance r_i by the same facility.

Using the strong duality result of Lemma 1.17 we see that the minimum number of facilities needed to serve client i within distance r_i, $i = 1,\ldots n$, is equal to the maximum cardinality of an independent set. Since two vertices i and j in the intersection graph are nonadjacent if $d(v_i,v_j) > r_i + r_j$ the problem of finding a maximum independent set can be formulated as

$$\begin{aligned} (1.22) \qquad & \max |I| \\ & \text{s.t } d(v_i,v_j) > r_i + r_j,\ i,j \in I,\ i \neq j, \\ & \quad I \subseteq \{1,2,\ldots,n\}. \end{aligned}$$

We have obtained the following strong duality result:

$$\begin{aligned} (1.23) \qquad & \max \{|I| \mid d(v_i,v_j) > r_i + r_j,\ i,j \in I,\ i \neq j,\ I \subseteq \{1,2,\ldots,n\}\} = \\ & \min \{|X| \mid d(v_i,X) \leq r_i, \quad i = 1,2,\ldots,n,\ X \subseteq T\}. \end{aligned}$$

To demonstrate how the strong duality result (1.23) can be used to obtain a strong duality result for the p-center problem we restrict ourselves to linear transportation cost. We come back to the nonlinear case at the end of this chapter.

THEOREM 1.24. $\max_{I\subseteq\{1,\ldots,n\}:|I|=p+1}\{\min_{i,j\in I:i\neq j}\{w_iw_jd(v_i,v_j)/(w_i+w_j)\}\} =$

$$\min_{X\subseteq T:|X|=p}\{\max_{i=1,\ldots,n}\{w_id(v_i,X)\}\}.$$

PROOF. We shall prove that the righthand side is less than or equal to r iff the lefthand side is less than or equal to r, for all $r \geq 0$. This shows that the two expressions are in fact equal.

The first equivalence is obtained by stating in two different ways that there exists p facilities such that we can cover each client within

distance r. We have

$$(1.25) \qquad \min_{X\subseteq T:|X|=p}\{\max_{i=1,\ldots,n}\{w_i d(v_i,X)\}\} \le r$$

iff

$$(1.26) \qquad \min\{|X| \mid \max_{i=1,\ldots,n}\{w_i d(v_i,X)\} \le r\} \le p.$$

Using (1.23) with $r_i = r/w_i$ we find that (1.26) holds iff (1.27) holds, where (1.27) is given by

$$(1.27) \qquad \max\{|I| \mid w_i w_j d(v_i,v_j)/(w_i+w_j) > r,\ i,j \in I,\ i \neq j\} \le p.$$

Expression (1.27) states that there are at most p vertices such that for any two of those vertices v_i, v_j $w_i w_j d(v_i,v_j)/(w_i+w_j) > r$. Another way of saying this is that for every set of p+1 vertices there are at least two of those vertices, say v_i, v_j, such that $w_i w_j d(v_i,v_j)/(w_i+w_j) \le r$. Hence (1.27) holds iff (1.28) holds, were (1.28) is given by

$$(1.28) \qquad \max_{I\subseteq\{1,\ldots,n\}:|I|=p+1}\{\min_{i,j\in I:i\neq j}\{w_i w_j d(v_i,v_j)/(w_i+w_j)\}\} \le r. \ \square$$

Let us derive similar duality results for the round-trip covering problem (1.19) and the linear round-trip p-center problem. Consider the shortest path $P(a_i,b_i)$ and $P(a_j,b_j)$. If these paths intersect, then we define the distance $D(P_i,P_j)$ between these paths to be zero. If the paths do not intersect, then define $c_i(c_j)$ to be the unique point on $P(a_i,b_i)$ $(P(a_j,b_j))$ closest to $P(a_j,b_j)(P(a_i,b_i))$. (See Figure 1.29). The points c_i, c_j are unique since we have a tree.

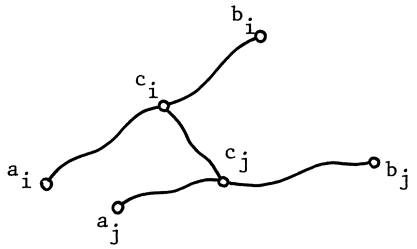

Figure 1.29. $D(P_i,P_j) = d(c_i,c_j)$.

If $P(a_i,b_i)$ and $P(a_j,b_j)$ do not intersect, then we define $D(P_i,P_j) = d(c_i,c_j)$. We conclude that in both cases

$$(1.30) \qquad D(P_i,P_j) = \tfrac{1}{2}[d(a_i,a_j)+d(b_i,b_j)-d(a_i,b_i)-d(a_j,b_j)].$$

With respect to (1.19) let us define T_i to be the set of all points of T with the property that a facility at such a point can serve client i within distance r_i, i.e., $T_i = \{y \in T \mid \frac{1}{2}[d(a_i,y)+d(b_i,y)-d(a_i,b_i)] \leq r_i\}$, $i = 1,2,\ldots,m$. Note that $r_i \geq 0$ implies $T_i \neq \emptyset$.

LEMMA 1.3.1. $T_i \cap T_j \neq \emptyset$ *iff* $D(P_i,P_j) \leq r_i + r_j$.

PROOF. If $x \in T_i \cap T_j$, then $D(P_i,P_j) = \frac{1}{2}[d(a_i,a_j)+d(b_i,b_j)-d(a_i,b_i)-d(a_j,b_j)]$ $\leq \frac{1}{2}[d(a_i,x)+d(x,a_j)+d(b_i,x)+d(x,b_j)-d(a_i,b_i)-d(a_j,b_j] \leq r_i + r_j$.

Conversely if $D(P_i,P_j) \leq r_i + r_j$, then if $P(a_i,b_i)$ and $P(a_j,b_j)$ intersect any point x in the intersection satisfies $x \in T_i \cap T_j$. If $P(a_i,b_i)$ and $P(a_j,b_j)$ do not intersect, then $d(c_i,c_j) \leq r_i + r_j$, where c_i,c_j are defined in Figure 1.29. According to Lemma 1.20 there exist a point x such that $d(c_i,x) \leq r_i$ and $d(c_j,x) \leq r_j$. Then

$$\begin{aligned} d(a_i,x)+d(x,b_i)-d(a_i,b_i) &\leq d(a_i,c_i)+d(c_i,b_i)+2d(c_i,x)-d(a_i,b_i) \\ &= 2d(c_i,x) \leq 2r_i. \end{aligned}$$

Hence $x \in T_i$. Equivalently $x \in T_j$. Hence $x \in T_i \cap T_j$. □

If we construct the intersection graph $G = (\{1,\ldots,m\},E)$ corresponding to $T_1,\ldots,T_m$, then $[i,j] \in E$ iff $D(P_i,P_j) \leq r_i + r_j$. Equivalent to (2.23) we obtain the strong duality

$$\begin{aligned} (1.32)\qquad &\max\{|I| \mid D(P_i,P_j) > r_i + r_j,\ i,j \in I,\ i \neq j,\ I \subseteq \{1,\ldots,m\}\} = \\ &\min\{|X| \mid D(J_i,X) \leq r_i,\ i = 1,\ldots,m,\ X \subseteq T\}. \end{aligned}$$

If we define the transportation cost of the linear round-trip p-center problem by

$$(1.33)\qquad w_i D(J_i,X), \quad i = 1,\ldots,m,$$

then we obtain the following strong duality result for the linear round-trip p-center problem:

$$\begin{aligned} (1.34)\qquad &\max_{I\subseteq\{1,\ldots,m\}:|I|=p+1}\{\min_{i,j\in I:i\neq j}\{w_i w_j D(P_i,P_j)/(w_i+w_j)\}\} = \\ &\min_{X\subseteq T:|X|=p}\{\max_{i=1,\ldots,m}\{w_i D(J_i,X)\}\}. \end{aligned}$$

The duality results of Theorem 1.24 and of (1.34) are important since it provides us with a set (of polynomial size) of values to which the optimum value r_p of the p-center problems belongs. For the p-center problem this is the set R given by

$$(1.35) \qquad R = \{w_i w_j d(v_i,v_j)/(w_i+w_j) \mid i,j = 1,2,\ldots,n\}.$$

Since the optimum value r_p of the p-center problem is the smallest value r in the set R for which the covering problem (1.18) with $r_i = r/w_i$ has an optimal solution less than or equal to p, and since there exists an O(n) algorithm to solve the covering problem (1.18) (KARIV and HAKIMI 1979). We can solve the p-center problem efficiently as a sequence of covering problems.

In case $w_i = 1$, $i = 1,2,\ldots,n$, an O(nlogn) implementation of this approach is given by FREDERICKSON and JOHNSON [1983]. It is based on an efficient search of the set R (see (1.35)) without explicitly generating it. Note that generating the set R takes $O(n^2)$ time.

In case of arbitrary weights w_i, $i = 1,\ldots,n$ an $O(n\log^2 n \log\log n)$ implementation is given by MEGIDDO and TAMIR [1983].

For the round-trip p-center problem the optimum value r_p belongs to the set S given by

$$(1.36) \qquad S = \{w_i w_j D(P_i,P_j)/(w_i+w_j) \mid i,j = 1,\ldots,m\}.$$

The optimum value r_p of the round-trip p-center is the smallest value r in the set S for which the covering problem (1.19) with $r_i = r/w_i$ has an optimal solution less than or equal to p. We will describe next an O(nm) algorithm to solve the covering problem (1.19). Using a *median finding procedure* on the set S we can solve the round-trip p-center problem as a sequence of O(logm) covering problems. The median finding procedure works as follows: First we find the median of the set S (the *median* of a set of k elements is the $\lceil \frac{k}{2} \rceil$ largest element. It can be found in O(k) time (see AHO, HOPCROFT and ULLMAN [1974])) and solve the covering problem for this value. If the solution of the covering problem is less than or equal to p, then all values greater than the median can be deleted from the set S, else all values smaller than the median can be deleted from S. Since in each step we delete half the total number of elements from the set S after O(logm) iteration we have found the optimum value. The complexity of

finding all the medians is $O(m^2)+O(\frac{1}{2}m^2)+O(\frac{1}{4}m^2)+\ldots+O((\frac{1}{2})^k m^2) = O(m^2)$, where k is such that $(\frac{1}{2})^k m^2 \leq 1$. Note that $k = O(\log m)$.

Let us now describe the round-trip covering algorithm.

During each iteration of the algorithm the original tree is partitioned into two subgraphs: one green, the other brown. The green subgraph is always a tree, denoted by GT, while the brown subgraph consists of one or more subtrees of the original tree T, each of which is "rooted" at a vertex of the green tree. By convention a root t will be in both GT and the associated brown subtree, denoted by BT(t). Given a brown subtree BT(t) we have the following sets and values assigned to the root t:

- I_t^1 denotes the set of all indices i such that either a_i or b_i belongs to BT(t), and such that client i is not served within distance r_i by any facility x located so far at BT(t), i.e. $d(a_i,x)+d(b_i,x)-d(a_i,b_i) > 2r_i$.
- I_t^2 denotes the set of all indices i such that both a_i and b_i belong to BT(t), and such that client i is not served within distance r_i by any facility x located so far at BT(t).
- $c_i(t) = r_i - \frac{1}{2}[d(a_i,t)+d(b_i,t)-d(a_i,b_i)]$ for every $i \in I_t^2$.

The following observation clarifies the meaning of the value $c_i(t)$, which will always be nonnegative. Since client i is not served by a facility located at BT(t), it must be served by a facility x not located at BT(t), i.e., $d(a_i,x)+d(x,b_i)-d(a_i,b_i) \leq 2r_i$. Since $a_i,b_i \in BT(t)$ we have $d(a_i,x)+d(b_i,x)-d(a_i,b_i) \leq 2r_i$. Since $a_i,b_i \in BT(t)$ we have $d(a_i,x)+d(b_i,x) = d(a_i,t)+d(b_i,t)+2d(t,x)$. It follows that x serves client i within distance r_i iff $d(t,x) \leq c_i(t)$.

- $c_0(t) = \min_{i \in I_t^2} \{c_i(t)\}$, $c_0(t) = \infty$ if $I_t^2 = \emptyset$.
- d(t) denotes the distance from t to the nearest facility located at BT(t), $d(t) = \infty$ if there is no facility located at BT(t).

Let k denote the number of facilities located so far and let $I \subseteq \{1,2,\ldots,m\}$ denote the set of indices i for which we have not yet determined whether client i is served within distance r_i by a facility located so far.

At each iteration we select a tip vertex t of GT if GT = {t}, then k := k+1, x_k = t and we stop. If GT $\neq$ {t}, then let v(t) be the vertex of GT adjacent to t. We delete [t,v(t)] from GT and add this together with BT(t) to BT(v(t)) to form the new BT(v(t)). If $c_0(t) < d(t,v(t))$, then k:= k+1 and x_k is defined to be the point on [t,v(t)] at distance $c_0(t)$ from t. We calculate the new sets and values $I_{v(t)}^1, I_{v(t)}^2, d(v(t))$, $c_i(v(t))$ for all $i \in I_{v(t)}^2$, and the set I. If $I = \emptyset$, then we stop. Otherwise we

continue with the next iteration.

Let us now consider the problem of finding the sets and values assigned to v(t) during an iteration.

Case A. $c_0(t) < d(t,v(t))$.

Case A.1. $\{a_i,b_i\}$ is included in BT(v(t)). There are three possibilities

Case A.1.1. Both vertices are in BT(t), i.e., $i \in I_t^2$.
Since $d(t,x_k) = c_0(t) \le c_i(t)$ for all $i \in I_t^2$ client i can be served by x_k and i can be deleted from I.

Case A.1.2. Both vertices are in the old BT(v(t)), i.e., $i \in I_{v(t)}^2$.
Client i is not served by a facility at the old BT(v(t)), but it is served by a facility at the new BT(v(t)) iff it is served by x_k, i.e., iff $d(v(t),x_k) \le c_i(v(t))$ (or equivalently $d(t,v(t)) - c_0(t) \le c_i(v(t))$). If this condition holds, then i can be deleted from I, else $i \in I_{v(t)}^2$.

Case A.1.3. One vertex is in BT(t) and the other vertex is in the old BT(v(t)), i.e., $i \in I_t^1 \cap I_{v(t)}^1$. Since x_k is on the shortest path $P(a_i,b_i)$ client i is served by x_k and i can be deleted from I.

The values of $c_i(v(t))$ for those i which are not deleted remain the same. Furthermore we have $i(v(t)) = \min\{d(v(t)), d(t,v(t)) - c_0(t)\}$.

Case A.2. Either a_i or b_i is contained in BT(v(t)). There are two possibilities.

Case A.2.1. The vertex is in BT(t), i.e., $i \in I_t^1 \setminus I_{v(t)}^1$.
Since x_k is on $P(a_i,b_i)$ client i is served by x_k and i can be deleted from I.

Case A.2.2. The vertex is in the old BT(v(t)).
Client i is not served by a facility in the old BT(v(t)) but it served by a facility in the new BT(v(t)) iff it is served by x_k. Since $d(a_i,x)+d(b_i,x)-d(a_i,b_i) =$
$= d(a_i,v(t))+d(b_i,v(t))+2d(v(t),x_k)-d(a_i,b_i) = 2d(v(t),x_k)$
client i can be served by x_k iff $d(v(t),x_k) \le r_i$ (or equivalently $d(t,v(t))-c_0(t) \le r_i$). If this condition holds, then i can be deleted from I, else $i \in I_{v(t)}^1$.

Case B $c_0(t) \ge d(t,v(t))$.

Case B.1. $\{a_i,b_i\}$ is included in BT(v(t)). There are three possibilities.

Case B.1.1. Both vertices are in BT(t), i.e., $i \in I_t^2$
Client i is not served by any facility located at BT(t), but is served by a facility in the new BT(v(t)) iff it is served by the facility x in the old BT(v(t)) closest to v(t), i.e., iff $d(t,v(t)) + d(v(t)) \leq c_i(t)$. If this condition holds, then i can be deleted from I, else $i \in I_{v(t)}^2$.

Case B.1.2. Both vertices are in the old BT(v(t)), i.e., $i \in I_{v(t)}^2$.
Equivalent to Case B.1.1 we find that i can be deleted from I iff $d(t,v(t))+d(t) \leq c_i(v(t))$, else $i \in I_{v(t)}^2$.

Case B.1.3. One vertex is in BT(t) and the other vertex in BT(v(t)), i.e., $i \in I_t^1 \cap I_{v(t)}^1$.
Client i is not served by a facility in BT(v(t)) and $c_i(v(t)) := r_i$.

We have $d(v(t)) = \min\{d(v(t)),d(t,v(t))+d(t)\}$.

Case B.2. Either a_i or b_i is contained in BT(v(t)). There are two possibilities.

Case B.2.1. The vertex is in BT(t), i.e. $i \in I_t^1 \setminus I_{v(t)}^1$.
Client i is not served by a facility in BT(t), but is served by a facility in the new BT(v(t)) iff it is served by the facility x in BT(v(t)) closest to v(t). Since

$$d(a_i,x)+d(b_i,x)-d(a_i,b_i) = d(a_i,v(t))+d(b_i,v(t))+2d(v(t),x)-$$
$$-d(a_i,b_i) = 2d(v(t),x) = 2d(v(t)).$$

client i is served by x iff $d(v(t)) \leq r_i$. If this condition holds, then i can be deleted from I, else $i \in I_{v(t)}^1$.

Case B.2.2. The vertex is in the old BT(v(t)). i.e., $i \in I_{v(t)}^1 \setminus I_t^1$.
Equivalent to Case B.2.1 i can be deleted from I iff $d(t)+d(t,v(t)) \leq r_i$, else $i \in I_{v(t)}^1$.

Each iteration requires O(m) calculations. Since there are at most n iterations, where n is the number of vertices in the tree, the total complexity is O(mn). We now proceed to prove the correctness of the algorithm.

LEMMA 1.3.7. *The algorithm constructs a feasible solution* X *to the round-trip covering problem with* $|X| \leq m$.

PROOF. An index i is deleted from $\{1,2,\ldots,m\}$ only when client i is served by a facility located so far. The algorithm stops when all indices have been deleted or when the green tree consists of a single point t. In the former

case all clients are served. In the latter case the set I of clients which are not yet served is equal to I_t^2. Since $c_i(t) \geq 0$ for all $i \in I_t^2$ these clients are served by the facility we locate at t. Hence X is a feasible solution. $|X| \leq m$ since each time we locate a facility at least one index is deleted from I. □

Let the algorithm construct a feasible solution $X = \{x_1, x_2, \ldots, x_p\}$. We define t_i to be the tip vertex of GT chosen by the algorithm in the iteration during which x_i was located. Note that after this iteration all pairs $\{a_k, b_k\}$ which have at least one vertex in $BT(t_i)$ are served. If x_i is located at distance $c_0(t_i)$ from t_i on $[t_i, v(t_i)]$, then we assume for notational convenience that the minimum in $c_0(t_i)$ is attained for $c_i(t_i)$. If x_p is located when the green tree consists of a single point t_p, then we assume for notational convenience that $p \in I_{t_p}^2$.

LEMMA 1.38. *Under the assumption made above we have* $D(P_i, P_j) > r_i + r_j$.

PROOF. First assume that x_p is located when the green tree GT consists of a single point t_p. Since $p \in I_{t_p}^2$ client p is not served by facility x_i, $i = 1, \ldots, p-1$. This implies

(1) Both a_p and b_p are not in $BT(t_i)$

(2) $d(a_p, x_i) - d(a_p, b_p) + d(b_p, x_i) > 2r_p$.

Since both a_i and b_i are in $BT(t_i)$ and $d(a_i, x_i) - d(a_i, b_i) + d(b_i, x_i) = 2r_i$.

We have
$$\begin{aligned} D(P_i, P_p) &= \tfrac{1}{2}[d(a_i, a_p) + d(b_i, b_p) - d(a_i, b_i) - d(a_p, b_p)] \\ &= \tfrac{1}{2}[d(a_i, x_i) + d(x_i, a_p) + d(b_i, x_i) + d(x_i, b_p) - d(a_i, b_i) - d(a_p, b_p)] \\ &= \tfrac{1}{2}[2r_i + d(x_i, a_p) + d(x_i, b_p) - d(a_p, b_p)] > r_i + r_p. \end{aligned}$$

If both facilities x_i, x_j $(i<j)$ are located in the case GT consists of at least one edge we consider two cases.

Case 1. If $BT(t_i) \subseteq BT(t_j)$, then $a_j, b_j \notin BT(t_i)$ and client j is not served by x_i, i.e., $d(a_j, x_i) - d(a_j, b_j) + d(b_j, x_i) > 2r_j$. We have

$$\begin{aligned} D(P_i, P_j) &= \tfrac{1}{2}[d(a_i, a_j) + d(b_i, b_j) - d(a_i, b_i) - d(a_j, b_j)] \\ &= \tfrac{1}{2}[d(a_i, x_i) + d(x_i, a_j) + d(b_i, x_i) + d(x_i, b_j) - d(a_i, b_i) - d(a_j, b_j)] \\ &> r_i + r_j. \end{aligned}$$

Case 2. If $BT(t_i) \not\subseteq BT(t_j)$, then we have

$$\begin{aligned} D(P_i,P_j) &= \tfrac{1}{2}[d(a_i,a_j)+d(b_i,b_j)-d(a_i,b_i)-d(a_j,b_j)] \\ &= \tfrac{1}{2}[d(a_i,x_i)+d(x_i,x_j)+d(x_j,a_j)+d(x_i,x_j)+d(x_j,b_j)+d(b_i,x_i) \\ &\quad - d(a_i,b_i)-d(a_j,b_j)] = \tfrac{1}{2}(2r_i+2r_j+2d(x_i,x_j)) > r_i+r_j. \quad \square \end{aligned}$$

According to Lemma 1.31 two clients i and j can only be served by the same facility iff $D(P_i,P_j) \le r_i+r_j$. We conclude from Lemma 1.38 that we need at least p facilities to serve all clients. Since we have a feasible solution with p facilities this solution must be optimal.

Let us return to the nonlinear round-trip p-center problem

$$\begin{aligned} (1.39) \qquad & \min r \\ & \text{s.t. } f_i(D(J_i,X) \le r, \ i = 1,\dots,m \\ & \qquad X \subseteq T, \\ & \qquad |X| = p. \end{aligned}$$

Define $\delta_i = \max\{D(J_i,x) \mid x \in T\}$, $\eta = \min_{i=1,\dots,m}\{f_i(\delta_i)\}$, $\alpha = \max_{i=1,\dots,m}\{f_i(0)\}$. We may assume without loss of generality that $\alpha < \eta$. If $\alpha = f_s(0) \ge f_t(\delta_t) = \eta$, then $f_s(D(J_s,X)) \ge f_t(D(J_t,X))$ for all X and the constraint corresponding to f_t is redundant in (1.39) and can therefore be deleted.

The function $f_i^{-1} + f_j^{-1}$ is increasing and has as domain $[\max[f_i(0),f_j(0)], \min[f_i(\delta_i),f_j(\delta_j)]]$. Define $L_{ij} = (f_i^{-1}+f_j^{-1})(\max[f_i(0), f_j(0)])$ and $U_{ij} = (f_i^{-1}+f_j^{-1})(\min[f_i(\delta_i),f_j(\delta_j)])$. Then $(f_i^{-1}+f_j^{-1})^{-1}$ has as domain $[L_{ij},U_{ij}]$.

Define $\varepsilon_{ij} = (f_i^{-1}+f_j^{-1})^{-1}(\max[L_{ij},D(P_i,P_j)])$. In order for ε_{ij} to be well defined $D(P_i,P_j) \le U_{ij}$ must hold. Without loss of generality assume $f_i(\delta_i) \le f_j(\delta_j)$. We shall prove $D(P_i,P_j) \le \delta_i = \delta_i+f_j^{-1}(f_j(0)) \le$ $\le \delta_i+f_j^{-1}(f_i(\delta_i)) = U_{ij}$ (Note that f_j^{-1} is increasing and $f_j(0) < f_i(\delta_i)$ since $\alpha < \eta$). Choose $x \in P(a_j,b_j)$. Then $D(P_i,P_j) = \tfrac{1}{2}[d(a_i,a_j)+d(b_i,b_j)-d(a_i,b_i)-d(a_j,b_j)] \le \tfrac{1}{2}[d(a_i,x)+d(x,b_i)+d(b_j,x)+$ $d(x,a_j)-d(a_i,b_i)-d(a_j,b_j)] = \tfrac{1}{2}[d(a_i,x)+d(b_i,x)-d(a_i,b_i)] \le \delta_i$. ε_{ij} is so defined that for a value r, $\alpha \le r \le \eta$ $D(P_i,P_j) \le f_i^{-1}(r)+f_j^{-1}(r)$ is equivalent to $\varepsilon_{ij} \le r$.

We obtain the following duality result for the nonlinear round-trip p-center problem.

THEOREM 1.40. $\max_{I\subseteq\{1,\ldots,m\} : I = p+1}\{\max\{\alpha,\min_{i,j\in I:i\neq j}\{\varepsilon_{ij}\}\}\} =$

$$\min_{X\subseteq T:|X|=p}\{\max_{i=1m}\{f_i(D(J_i,X))\}\}.$$

PROOF. Note that both sides are between α and η. We shall prove that the righthand side is less than or equal to r iff the lefthand side is less than or equal to r, for all r, $\alpha \le r \le \eta$.

$$\min_{X\subseteq T:|X|=p}\{\max_{i=1,\ldots,m}\{f_i(D(J_i,X))\}\} \le r$$

iff

$$\min\{|X| \mid X \subseteq T,\ f_i(D(J_i,X)) \le r,\ i = 1,\ldots,m\} \le p$$

iff

$$\min\{|X| \mid X \subseteq T,\ D(J_i,X) \le f_i^{-1}(r),\ i = 1,\ldots,m\} \le p$$

(since $\alpha \le r \le \eta$ we have

$$f_i(D(J_i,X)) \le r \quad \text{iff} \quad D(J_i,X) \le f_i^{-1}(r))$$

iff

$$\max\{|I| \mid I\subseteq\{1,\ldots,m\},\ D(P_i,P_j) > f_i^{-1}(r)+f_j^{-1}(r),\ i,j \in I,\ i \neq j\}\le p$$

(Lemma 1.31). Since

$$D(P_i,P_j) \le f_i^{-1}(r)+f_j^{-1}(r) \quad \text{iff} \quad \varepsilon_{ij} \le r$$

the last statement is equivalent to

$$\max_{I\subseteq\{1,\ldots,m\}:|I|=p+1}\{\min_{i,j\in I:i\neq j}\{\varepsilon_{ij}\}\} \le r. \quad \square$$

We can obtain similar results for the nonlinear p-center problem. These results can be found in Tansel, Francis, Lowe and Chen [1980].

CHAPTER 2

2. PROBLEMS WITH DIFFERENT SETUP COSTS

Let $T = (V,E)$ be a tree with n vertices v_i, $i = 1,2,\ldots,n$ and let c_j be the setup cost for locating a facility at vertex v_j, $j = 1,2,\ldots,n$. In the *center problem* with different setup costs we are locating facilities at the vertices so as to minimize the maximum transportation cost under the condition that the total setup costs may not exceed the upperbound B. The transportation cost is assumed to be an increasing function of the distance between a client and the closest facility. Each facility has a sufficient capacity to serve all clients. For ease of exposition we will assume the transportation cost to be a linear function of the distance between client and facility, but this is by no means a restriction. We can formulate this center problem as

$$(2.1) \qquad \min_{S\subseteq V, S\neq\emptyset:\sum_{j:v_j\in S} c_j \leq B} \{\max_{i=1,\ldots,n} \{w_i d(v_i,S)\}\},$$

where S denotes the subset of vertices at which facilities will be located, $\sum_{j:v_j\in S} c_j$ is the corresponding setup cost, and $w_i d(v_i,S)$ is the transportation cost associated with client i located at v_i. Another way of formulating this problem is the following.

$$(2.2) \qquad \begin{aligned} &\min r \\ &\text{s.t. } w_i d(v_i,S) \leq r, \quad i = 1,\ldots,n, \\ &\qquad \sum_{j:v_j\in S} c_j \leq B, \\ &\qquad S \subseteq V,\ S \neq \emptyset. \end{aligned}$$

Since we are minimizing the value of r it will be equal to the maximum transportation cost. We say that client i is served within *distance* r by a set of

facilities S iff $w_i d(v_i,S) \leq r$. A problem related to the center problem is the *covering problem* in which for a given value of r we want to minimize the setup costs needed to serve each client within distance r.
A formulation of the covering problem is given by

$$(2.3) \qquad \min \sum_{j:v_j \in S} c_j$$

$$\text{s.t. } w_i d(v_i,S) \leq r, \quad i = 1,\ldots,n,$$
$$S \subseteq V,\ S \neq \emptyset.$$

As in Chapter 1 it is easy to see that the optimum value of the center problem is the smallest value r in the set $\{w_i d(v_i,v_j) \mid i,j = 1,\ldots,n\}$ for which the covering problem (2.3) has an optimum value less than or equal to B. Therefore a polynomial algorithm to solve the covering problem would result in a polynomial time algorithm to solve the center problem.

We can formulate (2.3) as a set covering problem. Define the (0,1)-matrix $A = (a_{ij})$ by

$$(2.4) \qquad a_{ij} = 1 \text{ iff } w_i d(v_i,v_j) \leq r.$$

Then the following set covering problem is equivalent to (2.3).

$$(2.5) \qquad \min \sum_{j=1}^{n} c_j x_j$$
$$\text{S.t. } \sum_{j=1}^{n} a_{ij} x_j \geq 1,\ i = 1,2,\ldots,n,$$
$$x_j \in \{0,1\},\ j = 1,2,\ldots,n,$$

where $x_j = 1$ iff a facility is established at vertex j; $\sum_{j=1}^{n} a_{ij} x_j$ counts the number of established facilities which can serve client i within distance r.

Another problem with different setup costs is the simple plant location problem. In the *simple plant location problem* we locate facilities at the vertices so as to minimize the sum of the setup costs and the transportation costs. We can formulate this problem as

$$(2.6) \qquad \min_{S \subseteq V, S \neq \emptyset} \{\sum_{j:v_j \in S} c_j + \sum_{i=1}^{n} w_i d(v_i,S)\}.$$

We shall demonstrate next that the simple plant location problem can be

formulated as a set covering problem.

For each vertex v_i we denote by $0 = r_{i1} \le r_{i2} \le \ldots \le r_{in}$ the sorted sequence of distances from v_i to all vertices, including v_i itself. We define $r_{i,n+1} = \infty$. If there is no established facility within distance r_{ik} from v_i, then the closest established facility is at a distance of at least $r_{i,k+1}$. In this case the transportation cost for client i is increased from $w_i r_{ik}$ to at least $w_i r_{i,k+1}$. Therefore we can consider the difference $w_i(r_{i,k+1}-r_{ik})$ as a cost for not establishing a facility within distance r_{ik} from v_i. This is the motivation behind the set covering problem. Define the $n^2 \times n$ (0,1)-matrix $A = (a_{ik,j})$ by

$$a_{ik,j} = 1 \quad \text{iff} \quad d(v_i,v_j) \le r_{ik}, \quad i,j,k = 1,\ldots,n. \tag{2.7}$$

The set covering fromulation of the simple plant location problem is given by

$$\begin{aligned} \min \ & \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{n} \sum_{k=1}^{n} w_i(r_{i,k+1}-r_{ik}) z_{ik} \\ \text{s.t.} \ & \sum_{j=1}^{n} a_{ik,j} x_j + z_{ik} \ge 1, \qquad i,k = 1,\ldots,n, \\ & z_{ik} \in \{0,1\}, \quad i,k = 1,\ldots,n, \\ & x_j \in \{0,1\}, \quad j = 1,\ldots,n, \end{aligned} \tag{2.8}$$

where $x_j = 1$ iff a facility is established at vertex v_j, $z_{ik} = 1$ iff there is no facility established within distance r_{ik} from client i located at v_i.

To see that (2.8) is a correct formulation of the simple plant location problem first note that since $r_{i,n+1} = \infty$ we have $z_{in} = 0$, i.e. $\sum_{j=1}^{n} a_{in,j} x_j = \sum_{j=1}^{n} x_j \ge 1$. In order to prove the correctness of (2.8) we shall fix the x_j variables (Let $S = \{v_j | x_j=1\}$, $S \neq \emptyset$) and show that if we minimize over the z_{ik} variables the optimum value is given by $\sum_{j:v_j \in S} c_j + \sum_{i=1}^{n} w_i d(v_i,S)$. This shows that (2.8) is equivalent to (2.6).

Let t(i) be the smallest index such that $r_{it(i)} = \min_{j:x_j=1}\{d(v_i,v_j)\}$. Then the number of facilities established within distance r_{ik} from v_i is zero for $k < t(i)$ and at least one for $k \ge t(i)$. Since the coefficient of z_{ik} in the objective function is nonnegative the optimum value of z_{ik} is

given by $z_{ik} = 0$ for $k \geq t(i)$ and $z_{ik} = 1$ for $k < t(i)$. Substituting this in the objective function gives the value

$$\sum_{j:v_j \in S} c_j + \sum_{i=1}^{n} \sum_{k=1}^{t(i)-1} w_i(r_{i,k+1} - r_{ik}) =$$

$$\sum_{j:v_j \in S} c_j + \sum_{i=1}^{n} w_i r_{it(i)} = \sum_{j:v_j \in S} c_j + \sum_{i=1}^{n} w_i d(v_i,S).$$

(Note that $r_{i1} = 0$).

We claim that we can solve the set covering problems (2.5) and (2.8) in polynomial time. Since the set covering problem is known to be NP-hard the matrix A in (2.5) and (2.8) must have a special structure in order to enable us to solve the problem in polynomial time.

Define a *neighborhood subtree* N(x,r) of a tree to be all points of the tree which are within distance r (called the *radius*) of the point x (called the *center*). For the matrix A in (2.5) and (2.8) rows correspond to neighborhood subtrees and columns correspond to vertices and we have a one in the corresponding position iff the vertex is in the neighborhood subtree, i.e., (2.4) is equivalent to

$$(2.9) \qquad a_{ij} = 1 \quad \text{iff} \quad v_j \in N(v_i, r/w_i),$$

and (2.7) is equivalent to

$$(2.10) \qquad a_{ik,j} = 1 \quad \text{iff} \quad v_j \in N(v_i, r_{ik}).$$

We say that the matrix A in (2.5) and (2.8) is the *intersection matrix* of neighborhood subtrees versus vertices.

The polynomial time algorithm to solve set covering problems like (2.5) and (2.8) defined on the intersection matrix of neighborhood subtrees of a tree versus the vertices of a tree, consists of two phases. In the *first phase* we permute the rows and permute the columns of the matrix such that the transformed matrix is in standard greedy form. A (0,1)-matrix is in *standard greedy form* if it does not contain any two rows and any two columns such that the corresponding 2×2 submatrix has the form

(2.11) $\begin{bmatrix} 11 \\ 10 \end{bmatrix}$.

In the *second phase* of the algorithm we solve the set covering problem on a matrix in standard greedy form.

Let us first concentrate on the first phase. In order to find the desired permutations we have to study the properties of neighborhood subtrees.

A neighborhood subtree of a tree T can be viewed as a generalization of a subinterval of a fixed interval [a,b]. Neighborhood subtrees have some of the nice properties of subintervals. For example the intersection of two neighborhood subtrees is again a neighborhood subtree.

LEMMA 2.12. *Let* $N(x_1,r_1)$ *and* $N(x_2,r_2)$ *be neighborhood subtrees of a tree* T *with* $N(x_1,r_1) \cap N(x_2,r_2) \neq \emptyset$. *Then the intersection is a neighborhood subtree.*

PROOF. According to Lemma 1.20 $N(x_1,r_1) \cap N(x_2,r_2) \neq \emptyset$ iff $d(x_1,x_2) \leq r_1 + r_2$. without loss of generality assume $r_1 \leq r_2$.

If $r_1+d(x_1,x_2) \leq r_2$, then $N(x_1,r_1) \subseteq N(x_2,r_2)$. This follows from the triangular inequality since $d(y,x_1) \leq r_1$ implies $d(y,x_2) \leq d(y,x_1)+d(x_1,x_2) \leq r_1+d(x_1,x_2) \leq r_2$.

If $r_2 < r_1+d(x_1,x_2)$, then define z to be the point on $P(x_1,x_2)$ at distance $\frac{1}{2}(r_1-r_2+d(x_1,x_2))$ from x_1 and define $s = \frac{1}{2}(r_1+r_2-d(x_1,x_2))$. We claim that $N(x_1,r_1) \cap N(x_2,r_2) = N(z,s)$. This can be proved as follows.

Let $y \in N(x_1,r_1) \cap N(x_2,r_2)$. Since $z \in P(x_1,x_2)$ we have $z \in P(x_1,y)$ and/or $z \in P(x_2,y)$. (Here we use the tree property). Assume $z \in p(x_2,y)$ (The case $z \in P(x_1,y)$ can be treated similarly). We have

$$\begin{aligned} d(y,z) = d(x_2,y) - d(x_2,z) &\leq r_2 - d(x_2,z) \\ &= r_2 - \tfrac{1}{2}(r_2-r_1+d(x_1,x_2)) \\ &= \tfrac{1}{2}(r_1+r_2-d(x_1,x_2)) = s. \end{aligned}$$

Conversely, let $d(y,z) \leq s$. Then we have

$$d(y,x_1) \leq d(y,z)+d(z,x_1) \leq s + \tfrac{1}{2}(r_1-r_2+d(x_1,x_2)) = r_1.$$

Equivalent we prove that $d(y,x_2) \leq r_2$. Hence $y \in N(x_1,r_1) \cap N(x_2,r_2)$. □

The most important property for neighborhood subtrees is a generalization of a property which is trivial for subintervals and which states that for two subintervals I_1,I_2 of $[a,b]$ containing a, we have $I_1 \subseteq I_2$ or $I_2 \subseteq I_1$.

THEOREM 2.13 (KOLEN 1983). *Let* $N(x_1,r_1)$ *and* $N(x_2,r_2)$ *be two neighborhood subtrees containing an endpoint* t_1 *of a longest path* $P(t_1,t_2)$ *in the tree. Then* $N(x_1,r_1) \subseteq N(x_2,r_2)$ *or* $N(x_2,r_2) \subseteq N(x_1,r_1)$.

PROOF. Define s_i to be the point in $N(x_i,r_i)$ closest to t_2 on $P(t_1,t_2)$. Without loss of generality assume $d(t_2,s_2) \leq d(t_2,s_1)$. We claim that $N(x_1,r_1) \subseteq N(x_2,r_2)$. If $s_2 = t_2$, then $N(x_2,r_2) = T$ (see Lemma 2.17) and the result holds.

If $s_2 \neq t_2$, then

$$d(x_i,s_i) = r_i, \quad i = 1,2. \tag{2.14}$$

If $d(x_i,s_i)$ was strictly less than r_i, then there is a point in $N(x_i,r_i)$ closer to t_2 on $P(t_1,t_2)$ than $d(t_2,s_i)$; contradicting the definition of s_i.

Let z_i be the midpoint of $P(t_1,s_i)$, $i = 1,2$. Since $d(x_i,s_i) = r_i \geq d(x_i,t_1)$ we have $z_i \in P(x_i,s_i)$, i.e.,

$$d(z_i,x_i) + d(z_i,s_i) = d(x_i,s_i) = r_i, \quad i = 1,2. \tag{2.15}$$

We claim that $N(x_i,r_i) = N(z_i,d(z_i,s_i))$, $i = 1,2$.

Let us first prove that $N(z_i,d(z_i,s_i)) \subseteq N(x_i,r_i)$. Let $y \in N(z_i,d(z_i,s_i))$. Then $d(x_i,y) \leq d(y,z_i) + d(z_i,x_i) \leq d(z_i,s_i) + d(z_i,x_i) = r_i$ (see (2.15)).

To prove $N(x_i,r_i) \subseteq N(z_i,d(z_i,s_i))$ choose $y \in N(x_i,r_i)$. Since $z_i \in P(x_i,s_i)$ we have $z_i \in P(y,x_i)$ and/or $z_i \in P(y,s_i)$. If $z_i \in P(y,s_i)$, then $d(y,z_i) \leq d(z_i,t_1)$ since otherwise $P(y,t_2)$ would be a longer path than $P(t_1,t_2)$. Since $d(z_i,t_1) = d(z_i,s_i)$ we have $d(z_i,y) \leq d(z_i,s_i)$. If $z_i \in P(y,x_i)$, then $d(y,z_i) = d(y,x_i)-d(x_i,z_i) \leq r_i-d(x_i,z_i) = d(z_i,s_i)$ (see (2.15)).

It is now a simple matter to show that $N(x_1,r_1) \subseteq N(x_2,r_2)$ (or equivalently $N(z_1,d(z_1,s_1)) \subseteq N(z_2,d(z_2,s_2))$). We have the situation of Figure 2.16.

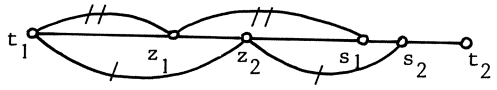

Figure 2.16. Situation on $P(t_1,t_2)$.

Let $y \in N(z_1,d(z_1,s_1))$. Then we have $d(z_2,y) \leq d(z_2,z_1) + d(z_1,y) \leq d(z_2,z_1)+d(z_1,s_1) = d(z_2,z_1)+d(z_1,t_1) = d(z_2,t_1) = d(z_2,s_2)$. Hence $y \in N(z_2,d(z_2,s_2))$. □

LEMMA 2.17. *If* $N(x,r)$ *is a neighborhood subtree containing a longest path* $P(t_1,t_2)$ *of the tree* T, *then* $N(x,r) = T$.

PROOF. Let z be the midpoint on $P(t_1,t_2)$. It is easy to see that $N(z,d(z,t_1)) = T$. If there exists an $y \in T$ such that $d(z,y) > d(z,t_1)$ then either $P(y,t_1)$ or $P(y,t_2)$ is a longer path than $P(t_1,t_2)$. Let $y \in T$. Without loss of generality assume $z \in P(x,t_1)$. Then

$$\begin{aligned} d(y,x) \leq d(y,z)+d(z,x) &= d(y,z)+d(x,t_1)-d(z,t_1) \\ &\leq d(x,t_1) \quad (\text{since } y \in N(z,d(z,t_1)) = T) \\ &\leq r \qquad (\text{since } t_1 \in N(x,r)). \quad \square \end{aligned}$$

The result of Theorem 2.13 is no longer true if we consider two neighborhood subtrees which have an arbitrary *tip vertex*, i.e. a vertex which is adjacent to only one other vertex, in common. This is shown by the following example on the tree given by Figure 2.18. $N(x_1,3)$ and $N(x_2,3)$ both contain vertex 4 but none is contained in the other.

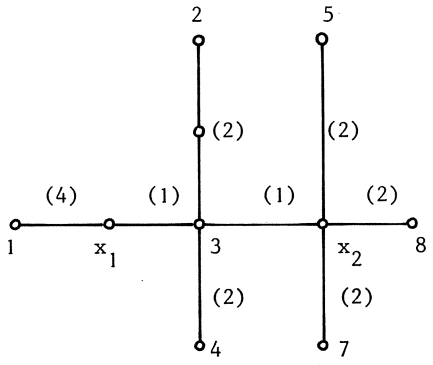

Figure 2.19. Tree of example.

We are now able to describe the permutation of the columns of the intersection matrix of neighborhood subtrees versus vertices, needed to transform this matrix into standard greedy form.

The first column corresponds to an endpoint of the longest path in the tree. Then we delete this vertex and the corresponding edge from the tree. As our second column we take and endpoint of a longest path in the resulting tree and repeat the procedure. Another way of obtaining the permutation is as follows. Fix one vertex v of the tree. Calculate the distances from v to all vertices and order them in nonincreasing order, say $d(t_1,v) \geq d(t_2,v) \geq \ldots \geq d(t_n,v)$, where $t_n = v$. Then t_k is an endpoint of a longest path in the subtree induced by $\{t_k,t_{k+1},\ldots,t_n\}$ for $k = 1,\ldots,n$. This is proved in Lemma 2.20. Hence if we let column k correspond to t_k we have obtained the desired permutation in O(nlogn) time.

LEMMA 2.20. *Let* v *be a vertex of a tree* T *and let* t_1 *be the vertex of* T *at the largest distance from* v. *Then* t_1 *is endpoint of a longest path in the tree* T.

PROOF. Suppose t_1 is not endpoint of a longest path. Let P(p,q) be a longest path and let w be the unique point on P(p,q) closest to v. We have $w \in P(t_1,p)$ and/or $w \in P(t_1,q)$. Assume without loss of generality that

$w \in P(t_1,p)$. Then $d(t_1,w) < d(q,w)$ since otherwise $P(t_1,p)$ would be at least as long as $P(p,q)$; contradicting the fact that t_1 is not endpoint of a longest path. We have $d(t,v) \leq d(t_1,w) + d(w,v) < d(q,w) + d(w,v) = d(q,v)$, contradicting the definition of t_1 as the point at largest distance from v. We conclude that t_1 is endpoint of a longest path in the tree. □

Let us go back to the obtained permutation of the vertices of the tree and study the property of this permutation. We know from Theorem 2.13 that all neighborhood subtrees containing t_1 (the endpoint of a longest path) can be totally ordered by inclusion, i.e., for any two neighborhood subtrees one is contained in the other. When we delete t_1 and the corresponding edge from the tree and from all neighborhood subtrees we are left with subtrees of the resulting tree. Fortunately these subtrees are again neighborhood subtrees of the resulting tree. (Lemma 2.22). Therefore all neighborhood subtrees containing t_2 (the endpoint of a longest path in the resulting tree) can be totally ordered by inclusion when restricted to the resulting tree. Repeating this argument we find that the intersection matrix of neighborhood subtrees versus vertices, where the vertices are permutated such that t_i is endpoint of a longest path in the subtree induced by $\{t_i,t_{i+1},\ldots,t_n\}$, $i = 1,\ldots,n$, has the nest ordering property for rows.

(2.21) A $m \times n$ (0,1)-matrix has the *nest ordering property* for rows if for all $j(1 \leq j \leq n)$ the following holds: all rows having a one in column j can be totally ordered by inclusion when restricted to columns $\{j,j+1,\ldots,n\}$. An equivalent definition is to say that the matrix does not contain any of the following two submatrices:

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

LEMMA 2.22. *Let* N(x,r) *be a neighborhood subtree of a tree* T *containing a tip vertex* t. *Let* T_1 *be the tree resulting when we delete* t *and the adjacent edge from* T. *If* $N(x,r) \cap T_1 \neq \emptyset$, *then it is a neighborhood subtree of* T_1.

PROOF. Assume $N(x,r) \cap T_1 \neq \emptyset$. Let v(t) be the vertex of T adjacent to t. If $x \in [t,v(t)]$, then define $y = v(t)$ and $s = r - d(x,v(t))$. If $x \notin [t,v(t)]$,

then define y = x and s = r. It is easy to see that $N(y,s) = N(x,r) \cap T_1$. □

It is now only one more step to transform the intersection matrix of neighborhood subtrees versus vertices into standard greedy form.

Let x and y be two (0,1)-vectors in $\mathbb{R}^n$. We say that y is *lexical larger* than x if in the last coordinate in which they differ x has a zero and y has a one. Note that this concept is different from lexicographically larger in which the first position in which they differ is important. We say that $x_1,x_2,\ldots,x_m$ are ordered in *lexical nondecreasing order* if $x_i = x_{i+1}$ or x_{i+1} is lexical larger than x_i, i = 1,...,m-1. A lexical ordering can be obtained in O(mn) time by a radix sort procedure (AHO, HOPCROFT and ULLMAN 1974).

LEMMA 2.23. *If a (0,1)-matrix has the nest ordering property for rows and the rows are in lexical nondecreasing order, then the matrix is in standard greedy form.*

PROOF. Suppose that a forbidden 2×2 submatrix exists with rows i_1,i_2 $(i_1<i_2)$ and columns $j_1,j_2(j_1<j_2)$. (See Figure 2.24). Since row i_2 is lexical larger than row i_1 the last column j_3 in which they differ has a one in row i_2 and a zero in row i_1. Since $j_3 > j_2$ the matrix of Figure 2.24 contradicts the nest ordering property for rows (see (2.21)). We conclude that A must be in standard greedy form. □

$$\begin{array}{c} \\ i_1 \\ i_2 \end{array} \begin{array}{c} \begin{array}{ccc} j_1 & j_2 & j_3 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \end{array}$$

Figure 2.24. Forbidden submatrix.

This completes the first phase of the algorithm in which we transformed the intersection matrix of neighborhood subtrees versus vertices into standard greedy form.

In the second phase we assume that the matrix $A = (a_{ij})$ is in standard greedy form and we solve the set covering problems related to (2.5) and (2.8). Let us first solve the set covering problem to (2.5), i.e.

$$(2.25)\qquad \min \sum_{j=1}^{m} c_j x_j$$
$$\text{s.t. } \sum_{j=1}^{m} a_{ij} x_j \geq 1, \quad i = 1,\ldots,n,$$
$$x_j \in \{0,1\}, \quad j = 1,\ldots,m.$$

The algorithm works on the Lp-relaxation of the set covering problem (2.25) given by

$$(2.26)\qquad \min \sum_{j=1}^{m} c_j x_j$$
$$\text{s.t. } \sum_{j=1}^{m} a_{ij} x_j \geq 1, \quad i = 1,\ldots,n,$$
$$x_j \geq 0, \quad j = 1,\ldots,m,$$

and its dual problem given by

$$(2.27)\qquad \max \sum_{i=1}^{n} y_i$$
$$\text{s.t. } \sum_{i=1}^{n} y_i a_{ij} \leq c_j, \quad j = 1,\ldots,m$$
$$y_i \geq 0, \quad i = 1,\ldots,n.$$

We will see that (2.26) has an optimal (0,1)-solution. The complementary slackness relations for these LP-problems are

$$(2.28)\qquad x_j(\sum_{i=1}^{n} y_i a_{ij} - c_j) = 0, \quad j = 1,2,\ldots,m,$$

and

$$(2.29)\qquad y_i(\sum_{j=1}^{m} a_{ij} x_j - 1) = 0, \quad i = 1,2,\ldots,n.$$

In the *first step* of the algorithm we calculate a feasible solution to problem (2.27). This is done by the *greedy algorithm*, i.e., we calcualte y_i by increasing index of i and take it to be as large as possible with respect to the constraints. Hence

$$(2.30)\qquad y_i = \min_{j:a_{ij}=1} \{c_j - \sum_{k=1}^{i-1} y_k a_{kj}\}, \quad i = 1,\ldots,n.$$

As an example consider the matrix in standard greedy form given by (2.31) and let $c_1 = 2$, $c_2 = 3$, $c_3 = 3$, $c_4 = 4$, $c_5 = 3$. Following the procedure we find $y_1 = 2$, $y_2 = 0$, $y_3 = 1$, $y_4 = 3$, $y_5 = 0$.

$$(2.31) \qquad \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

In order to describe the *second step* of the algorithm in which we construct a (0,1)-solution x_j, $j = 1,2,\ldots,m$, we define the set $I = \{i \mid y_i > 0\}$ and the set $J = \{j \mid \sum_{i=1}^{n} y_i a_{ij} = c_j\}$. Relation (2.28) tells us that we can only take $x_j = 1$ for $j \in J$. Relation (2.29) tells us that if for some $i \in I$ we have defined $x_j = 1$ for which $a_{ij} = 1$, then for all other indices k with $a_{ik} = 1$ we must have $x_k = 0$.

We construct the solution x_j, $j = 1,2,\ldots,m$ as follows. Define $x_j = 1$ for the largest index j in the set J. Delete from J all indices k for which $a_{ij} = a_{ik} = 1$ for some $i \in I$ (Note that we are forced to do this in order to satisfy (2.29)) and repeat the procedure until $J = \emptyset$. All values x_j which have not been defined to be equal to one are set to zero.

Let us go back to our example (2.31). We have $I = \{1,3,4\}$ and $J = \{1,2,5\}$. We set $x_5 = 1$ and delete 5 from J. Then we set $x_2 = 1$ and delete $\{1,2\}$ from J and stop. An optimal solution with value six is obtained by taking $x_2 = x_5 = 1$.

In order to prove that x_j, $j = 1,\ldots,m$ is a feasible solution which together with y_i, $i = 1,\ldots,n$ satisfies the complementary slackness relations (and hence is optimal) we have to show that (1). Each row $i \in I$ is covered exactly once by a column j for which $x_j = 1$, and (2) Each row $i \notin I$ is covered at least once by a column j for which $x_j = 1$. A row i is *covered* by column j if $a_{ij} = 1$.

Since y_i, $i = 1,\ldots,n$, was determined by the greedy algorithm (see (2.30)) $y_i > 0$ implies that there exists an index j such that y_i *saturates* constraint j, i.e., $\sum_{k=1}^{i-1} y_k a_{kj} < c_j$ and $\sum_{k=1}^{i} y_k a_{kj} = c_j$. If y_i saturates constraint j, then j is a *blocking constraint* for y_i. Note that blocking constraints belong to J and have the following property.

(2.32). If constraint j is a blocking constraint for y_i, $i \in I$, then for all $k > i$ with $a_{kj} = 1$ we have $y_k = 0$.

LEMMA 2.33. *Row* i, $i \in I$ *is covered exactly once by a column* j *for which* $x_j = 1$. *Furthermore* $j \geq k$ *for all blocking constraint* k *for* y_i.

PROOF. When $x_j = 1$ then all columns $k \leq j$ which cover a row $i \in I$ also covered by column j are deleted from the set J. Therefore each row $i \in I$ is covered at most once. Let $i \in I$. Consider the largest blocking constraint k for y_i. We set $x_k = 1$ unless column k is deleted from J because there is a $j > k$ with $x_j = 1$ and a $p \in I$ such that $a_{pk} = a_{pj} = 1$. It follows from (2.32) that $p < i$. Since the matrix is in standard greedy form $a_{ik} = a_{pk} = a_{pj} = 1$ imply that $a_{ij} = 1$. We conclude that row $i \in I$ is covered exactly once by a column j (with the property that $j \geq k$ for all blocking constraint k for y_i). □

LEMMA 2.3.4. *Each row* $i \notin I$ *is covered at least once by a column* j *for which* $x_j = 1$.

PROOF. Since $y_i = 0$ it follows from the greedy algorithm that there exists a $j \in J$ such that $\sum_{k=1}^{i-1} y_k a_{kj} = c_j$. Since $c_j > 0$ there exists a $p < i$ such that $y_p > 0$ and y_p saturates constraint j. Let k be the column covering row p. It follows from Lemma 2.33 that $k \geq j$. If $k = j$, then the Lemma holds. If $k > j$, then $a_{pj} = a_{pk} = a_{ij} = 1$ imply that $a_{ik} = 1$. Hence column k covers row i. □

Let us next look at the more complicated set covering problem corresponding to (2.8), i.e.,

$$(2.35) \qquad \min \sum_{j=1}^{m} c_j x_j + \sum_{i=1}^{n} d_i z_i$$

$$\text{s.t.} \sum_{j=1}^{m} a_{ij} x_j + z_i \geq 1, \qquad i = 1,\ldots,n.$$
$$z_i \in \{0,1\}, \quad i = 1,\ldots,n,$$
$$x_j \in \{0,1\}, \quad j = 1,\ldots,m.$$

The algorithm to solve (2.35) is very similar to the algorithm to solve (2.25). It works on the LP-relaxation of the set covering problem given by

$$(2.36)\qquad \min \sum_{j=1}^{m} c_j x_j + \sum_{i=1}^{n} d_i z_i$$

$$\text{s.t.} \sum_{j=1}^{m} c_j x_j + z_i \geq 1, \quad i = 1,\ldots,n,$$
$$z_i \geq 0, \quad i = 1,\ldots,n,$$
$$x_j \geq 0, \quad j = 1,\ldots,m,$$

and its dual problem given by

$$(2.37)\qquad \max \sum_{i=1}^{n} y_i$$

$$\text{s.t.} \sum_{i=1}^{n} y_i a_{ij} \leq c_j, \quad j = 1,\ldots,m,$$
$$0 \leq y_i \leq d_i, \quad i = 1,\ldots,n.$$

The complementary slackness relations are given by

$$(2.38)\qquad y_i\left(\sum_{j=1}^{m} a_{ij}x_j + z_i - 1\right) = 0, \quad i = 1,\ldots,n,$$

$$(2.39)\qquad x_j\left(\sum_{i=1}^{n} y_i a_{ij} - c_j\right) = 0, \quad j = 1,\ldots,m,$$

$$(2.40)\qquad z_i(y_i - d_i) = 0, \quad i = 1,\ldots,n.$$

The algorithm starts by applying the greedy approach to the dual problem (2.37), i.e., we calculate y_i by increasing index of i and take it to be as large as possible with respect to the constraints. Hence

$$(2.41)\qquad y_i = \min\{d_i, \min_{j:a_{ij}=1}\}\{c_j - \sum_{k=1}^{i-1} y_k a_{kj}\}, \quad i = 1,\ldots,n.$$

A value y_i *saturates* constraint j if $\sum_{k=1}^{i-1} y_k a_{kj} < c_j$ and $\sum_{k=1}^{i} y_k a_{kj} = c_j$, constraint j is called a *blocking constraint* for y_i. Let I be the set of indices i such that y_i saturates a constraint. Define
$J = \{j \mid \sum_{i=1}^{n} y_i a_{ij} = c_j\}$.

We construct the solution x_j, $j = 1,\ldots,m$, z_i, $i = 1,\ldots,n$ as follows. Define $x_j = 1$ for the largest index j in the set J. Delete from J all indices k for which $a_{ij} = a_{ik} = 1$ for some $i \in I$, and repeat the procedure until $J = \emptyset$. All values x_j which have not been defined to be equal to one are set to zero. If row i is not covered by a column j for which $x_j = 1$, then $z_i = 1$, else $z_i = 0$.

Clearly x,z and y are feasible solutions. In order to prove optimality (by showing that the complementary slackness relations hold) it is sufficient

to show that (1) Each row $i \in I$ is covered exactly once by a column j with $x_j = 1$ (hence $z_i = 0$), (2) Each row $i \notin I$, $y_i = 0$ is covered at least once by a column j with $x_j = 1$ (hence $z_i = 0$) and (3) Each row $i \notin I$, $y_i = d_i$ is covered at most once by a column j with $x_j = 1$.

To prove (1) and (2) we can copy Lemma 2.33 and Lemma 2.34.

LEMMA 2.42. *Row* i, $y_i = d_i$, $i \notin I$ *is covered at most once by a column* j *with* $x_j = 1$.

PROOF. Suppose there is a column k with $x_k = 1$ covering row i and let k be the smallest column with this property. Since $i \notin I$, there is an index $p > i$ which saturates constraint k (Hence $p \in I$). If row i was also covered by a column j; $j > k$ with $x_j = 1$, then since the matrix is in standard greedy form $a_{ij} = a_{ik} = a_{pk} = 1$ imply that $a_{pj} = 1$. This contradicts the fact that row $p \in I$ is covered exactly once. We conclude that row i is covered at most once. □

This completes the description of the algorithms to solve set covering problems on matrices in standard greedy form. In HOFFMAN, KOLEN and SAKAROVITCH [1984] a more general algorithm was given to solve problems like (2.25) and (2.35) where the right hand side is replaced by b_i, $i = 1,\ldots,n$ satisfying $b_1 \geq b_2 \geq \ldots \geq b_n$. In Chapter 6 we study matrices which can be transformed into standard greedy form.

CHAPTER 3

3. THE p-MEDIAN PROBLEM WITH MUTUAL COMMUNICATION

Let $T = (V,E)$ be a tree with vertices v_i, $i = 1,\ldots,n$. In the p-median problem with mutual communication we want to locate p new facilities on the tree such that the sum of the transportation costs is minimal. There is communication between a client and perhaps more than one new facilities. Furthermore there is communication between new and existing facilities as well as between new facilities themselves. Assuming that the transportation cost are linear with respect to the distance travelled we can formulate the p-median problem with mutual communication as

$$(3.1) \qquad \min_{X=\{x_1,x_2,\ldots,x_p\}}\{F(x)\},$$

$$\text{with } F(x) = \sum_{i=1}^{n}\sum_{j=1}^{p} \alpha_{ij}d(v_i,x_j) + \tfrac{1}{2}\sum_{j=1}^{p}\sum_{k=1}^{p} \beta_{jk}d(x_j,x_k),$$

where α_{ij}, $i = 1,\ldots,n$, $j = 1,\ldots,p$ and $\beta_{jk} = \beta_{kj}$ $j,k = 1,\ldots,p$, are non-negative integers, called *weights*. We will also use the terminology existing facility for a client. The weights are zero when there is no communication between the corresponding facilities.

The p-median problem with mutual communication has been studied extensively in the plane using rectilinear distances. The *rectilinear distance* $d(A,C)$ between points $A = (a,b)$ and $C = (c,d)$ in the plane is defined by $d(A,C) = |a-c| + |b-d|$. This problem can be separated into two independent problems on a line, one for each coordinate. Let $a_1,a_2,\ldots,a_n$, $a_i < a_{i+1}$, $i = 1,2,\ldots,n-1$, be existing facility locations on a line. The p-median problem with mutual communication on a line can be formulated as

(3.2) $\min_{X=\{x_1,x_2,\ldots,x_p\}}\{G(x)\}$,

with $G(x) = \sum_{i=1}^{n}\sum_{j=1}^{p}\alpha_{ij}|a_i-x_j| + \frac{1}{2}\sum_{j=1}^{p}\sum_{k=1}^{p}\beta_{jk}|x_j-x_k|$.

There are essentially two approaches to solve the problem on a line.

The first one is the direct search approach by PRITSKER & CHARE (1970), RAO (1973), JUEL & LOVE (1976), SHERALI & SHETTY (1978), and KOLEN (1981). The direct search approach is as follows. Start with some solution X^1 with new facilities located at existing facility locations. Consider the set N_q of new facilities located at some existing facility location a_q $(1\leq q\leq n)$. If there is a subset $S \subseteq N_q$ that can be moved to an adjacent existing facility location (a_{q-1} or a_{q+1}) such that the new solution X^2 satisfies $G(X^2) < G(X^1)$, move that subset to the corresponding adjacent facility location. Then repeat this procedure with X^2. If no such subset exists at any location we have found an optimal solution. A justification of the algorithm can be found in the linear programming formulation of the problem. RAO [1973] has proved through the negation of various alternatives that a single non-degenerate simplex pivot can only result in the movement of a subset of new facilities at a given existing facility location to an adjacent facility location. We will show that the same result holds for the problem on a tree. In all algorithms proposed, the crucial point is to find a subset of new facilities located at an existing facility which, when moved to an adjacent facility location, gives the largest reduction in the value of G. None of these algorithms, with a single exception (KOLEN 1981), has a polynomial running time. It was shown by KOLEN (1981) how the direct search algorithms could be adjusted to run in polynomial time. We will generalize this algorithm for the problem on a tree. In contrast, the problem on an arbitrary graph will be shown to be NP-hard.

The second approach is the cut approach due to PICARD and RATLIFF (1978). The complexity of this algorithm is $O(np^3)$, which is the same as that of the direct search algorithm given by KOLEN (1981). This not surprising since it was shown in the latter reference that the two approaches are equivalent.

The direct search algorithm for the p-median problem with mutual communication on a tree is based on the following necessary and sufficient conditions for optimality:

(3.3). A solution $X, X \subseteq V, |X| = p$ is an optimal solution for the p-median problem with mutual communication iff there is no subset of facilities

which can be moved to an adjacent vertex such that the objective value decreases.

The condition formulated in (3.3) guarantees local optimality and therefore is necessary for optimality. In order to prove that it is also sufficient for optimality we shall first prove that the function F(x) defined by (3.1) is a convex function.

Let x and y be two different points on the tree. The point $\lambda x+(1-\lambda)y$ $(0\leq\lambda\leq 1)$ is defined to be the point on the shortest path P(x,y) between x and y at distance $(1-\lambda)d(x,y)$ from x. Let v, x_1,x_2,y_1 and y_2 be points on the tree. DEARING, FRANCIS and LOWE 1976 proved that $d(v,\lambda x_1+(1-\lambda)y_1) \leq \lambda d(v,x_1)+(1-\lambda)d(v,y_1)$ and $d(\lambda x_1+(1-\lambda)y_1,\lambda x_2+(1-\lambda)y_2) \leq$ $\leq \lambda d(x_1,x_2)+(1-\lambda)d(y_1,y_2)$.

LEMMA 3.4. *Let* x,y *and* v *be points on the tree* T. *Then* $d(v,\lambda x+(1-\lambda)y) \leq$ $\leq \lambda d(v,x)+(1-\lambda)d(v,y)$ *for all* $0 \leq \lambda \leq 1$.

PROOF. Let $z = \lambda x+(1-\lambda)y$. Then $d(x,z) = (1-\lambda)d(x,y), d(y,z) = \lambda d(x,y)$. Since $z \in P(x,y)$ we have $z \in P(y,v)$ and/or $z \in P(x,v)$. Without loss of generality assume $z \in P(x,v)$. Then

$$d(v,z) = d(v,x)-d(x,z) = d(v,x)-(1-\lambda)d(x,y) = \lambda d(v,x)+(1-\lambda)[d(v,x)-$$
$$d(x,y)] \leq \lambda d(v,x)+(1-\lambda)d(v,y). \quad \square$$

LEMMA 3.5. *Let* x_1,x_2,y_1,y_2 *be points on a tree* T. *Let* $z_i = \lambda x_i+(1-\lambda)y_i$, $i = 1,2$, $(0\leq\lambda\leq 1)$. *Then* $d(z_1,z_2) \leq \lambda d(x_1,x_2)+(1-\lambda)d(y_1,y_2)$.

PROOF. Since $z_1 \in P(x_1,y_1)$ we have $z_1 \in P(y_1,z_2)$ and/or $z_1 \in P(x_1,z_2)$. Without loss of generality assume $z_1 \in P(x_1\ z_2)$. Then

$$\begin{aligned} d(z_1,z_2) &= d(x_1,z_2)-d(x_1,z_1) \\ &\leq \lambda d(x_1,x_2)+(1-\lambda)d(x_1,y_2)-d(x_1,z_1) \text{ (Lemma 3.4)} \\ &= \lambda d(x,x_2)+(1-\lambda)[d(x_1,y_2)-d(x_1,y_1)] \\ &\leq \lambda d(x_1,x_2)+(1-\lambda)d(y_1,y_2). \quad \square \end{aligned}$$

If $X = \{x_1,\ldots,x_p\}$ and $Y = \{y_1,\ldots,y_p\}$, then $Z = \lambda X+(1-\lambda)Y$ $(0\leq\lambda\leq 1)$ is defined as $Z = \{z_1,\ldots,z_p\}$ with $z_i = \lambda x_i+(1-\lambda)y_i$, $i = 1,\ldots,p$. Using Lemma 3.4 and Lemma 3.5 we have proved convexity of the function F, i.e.,

(3.6) $F(\lambda X+(1-\lambda)Y) \leq \lambda F(X)+(1-\lambda)F(Y), \quad 0 \leq \lambda \leq 1,$

Let X^1 be a given solution and let a,b be two points on the same edge such that there are no facilities lying on the edge between a and b. Deleting the part of the edge between a and b from the tree results in two subtrees. Let $T_a(T_b)$ denote the subtree containing a(b). Let X^2 be the solution we get from X^1 be moving a subset S of new facilities located at a to b. Then the distance between new facilities in S and facilities, both existing and new, which in both solutions are lying on $T_a(T_b)$ is increased (decreased) with d(a,b). So the difference between $F(X^2)$ and $F(X^1)$ is a constant times d(a,b), where the constant does not depend on the exact location of a facility, but only on whether the facility is on T_a or T_b. If before moving the subset of new facilities S from a to b we move a subset of new facilities disjoint from S in T_a or T_b, then this does not affect the change in the objective value when we move S. We call two movements *independent* if the change in the objective value when one movement is performed is the same whether the other movement has been performed first or not.

Let X ba a nonoptimal solution. Then there is a solution Y such that F(y) < F(x). Define q to be the length of the shortest edge. Then there is a $\lambda_1, 0 \leq \lambda_1 < 1$ such that for all $\lambda, \lambda_1 \leq \lambda < 1$ we have
$d(x_i, \lambda x_i+(1-\lambda)y_i) = (1-\lambda)d(x_i,y_i) < \frac{1}{2}q, \quad i = 1,\ldots,p.$
Let $Z = \lambda X+(1-\lambda)Y$ for some $\lambda, \lambda_1 \leq \lambda < 1$. Note that $F(Z) \leq \lambda F(X)+(1-\lambda)F(Y)$ $< \lambda F(X)+(1-X)F(X) = F(X)$. In order to obtain Z from X we only have to move subsets of new facilities along parts of the edges. All movements starting from different vertices as well as movements starting from the same vertex along different edges are independent. Since F(Z) < F(X) we know that there must be a vertex v_s and an edge $[v_s,v_t]$ such that performing all movements from v_s on this edge reduces the objective value. Let in Z subsets of new facilities S_i be located at the point u_i on the edge $[v_s,v_t]$, $i = 1,2,\ldots,k$, where $0 < d(v_s,u_i) < d(v_s,u_{i+1}) < \frac{1}{2}q$ $i = 1,2,\ldots,k-1$. We can perform the movements in k steps. In step i we move the subset of new facilities $\cup_{j=i}^{k} S_j$ from u_{i-1} to u_i, $i = 1,2,\ldots,k$, where $u_0 = v_s$. Since the objective value is decreased after these movements at least one of the movements gives a decrease in the objective value. Without loss of generality let this be the movement performed at step i ($1 \leq i \leq k$). The objective

value increases with $d(u_{i-1},u_i)$ times a constant, where the constant only depends on which facilities not belonging to $U_{j=i}^{k} S_j$ are located in $T_{u_{i-1}}$ or T_{u_i}. We know by assumption that this constant is negative. We claim that when we move the subset $U_{j=i}^{k} S_j$ of new facilities located at v_s in X to v_t, the objective value decreases. This proves that the necessary conditions are also sufficient for optimality. When we move the subset $U_{j=i}^{k} S_j$ from v_s to v_t the value of objective value increases with $d(v_s,v_t)$ times the same constant we had before when we moved the same subset in the solution Z from u_{i-1} to u_i. The two constants are the same since the facilities not belonging to $U_{j=i}^{k} S_j$ located at $T_{u_{i-1}}(T_{u_i})$ in Z are the same as the facilities located at $T_s(T_t)$ in X.

The medain problem with mutual communication on a tree is defined by the parameters T,P,α, and β, where T is the tree, P is the index set of new facilities, α is the vector of weights between existing and new facilities, and β is the vector of weights between new facilities themselves. The algorithm to solve the median problem with mutual communication generalizes the 1-median algorithm of GOLDMAN (1971) and can be described recursively as follows:

Algorithm (T,P,α,β)
if T consist of a single vertex
then locate all facilities in P at that vertex
else select a tip vertex v_s of T. Locate all new facilities in P at v_s and determine the subset S of P which when moved to the unique adjacent vertex v_t gives the largest positive decrease in the objective value. If no such subset exists, then the current solution is optimal and stop, else define $x_j = v_s$ for all $j \in P\backslash S$, $\alpha_{tk} := \alpha_{tk}+\alpha_{sk}+\sum_{j\in P\backslash S} \beta_{jk}$ for all $k \in S$, P := S, T := T with v_s and the edge $[v_s,v_t]$ deleted from it. Call algorithm (T,P,α,β).

We use induction on the number of vertices of the tree to prove that the algorithm constructs an optimal solution. In case of one vertex this is trivial.

Assume the induction hypothesis is true for all trees with less than n vertices ($n\geq 2$). We choose a tip vertex v_s and determine the subset S of P which when moved to the adjacent vertex v_t gives a largest positive reduction in the objective value. Let $S_1(S_2)$ be a subset of new facilities located at $v_s(v_t)$ after we moved S. If in the final solution constructed by

the algorithm the movement of $S_1(S_2)$ to $v_t(v_s)$ would give a reduction in the objective value, then the movement of $S \cup S_1(S \backslash S_2)$ from v_s to v_t in the solution with all facilities located at v_s would give a larger reduction than the movement of S; contradicting the definition of S. We conclude that in the solution constructed by the algorithm no subset of facilities located at $v_s(v_t)$ can be moved to $v_t(v_s)$ such that the objective value decreases. It remains to be shown that no subset of facilities located in the tree T_t we obtain from T by deleting the edge $[v_s,v_t]$ can be moved from a vertex in T_t to an adjacent vertex in T_t such that the objective value decreases. If we move a subset of S in T_t from v_k to an adjacent vertex v_ℓ, then the change in the objective value depends on which facilities are located in T_k and which are located in T_ℓ and does not depend on their exact location. Since v_s and v_t always occur in the same subtree we might as well add the weights of the facilities located at v_s to the weight of v_t, i.e., $\alpha_{tk} := \alpha_{tk} + \alpha_{sk} + \sum_{j\in P\backslash S} \beta_{jk}$ for all $k \in S$. Using the induction hypothesis on T_t we know that the solution constructed by the algorithm on T_t for the problem with new adjusted weights α_{tk} $(k\in S)$ satisfies the property that no subset of S can be moved from a vertex in T_t to an adjacent vertex in T_t such that the objective value decreases. The previous observation shows that this is equivalent to the property that no subset of S can be moved in the original problem such that the objective value decreases. This proves that the solution constructed by the algorithm satisfies the sufficient condition for optimality and hence is optimal.

Let us now return to the problem of finding a subset $S \subseteq P$ which when moved from v_s to v_t gives the largest reduction in the objective value. When we move the subset S from v_s to v_t the objective value is increased by

$$d(v_s,v_t)\Big[\sum_{j\in S} \alpha_{sj} + \sum_{j\in S} \sum_{k\in P\backslash S} \beta_{jk} - \sum_{j\in S} \sum_{i:v_i\in T_t} \alpha_{ij}\Big]. \tag{3.7}$$

The problem we have to solve is given by

$$\min_{S\subseteq P} \Big\{\sum_{j\in S}\alpha_{sj} + \sum_{j\in S}\sum_{k\in P\backslash S} \beta_{jk} - \sum_{j\in S}\sum_{i:v_i\in T_t} \alpha_{ij}\Big]. \tag{3.8}$$

If this minimum is nonnegative, then the current solution is optimal, else we move the subset for which the minimum is attained. We transform problem (3.8) into an equivalent problem by adding the constant $\sum_{j\in P}\sum_{i:v_i\in T_t} \alpha_{ij}$, which does not depend on S. This gives the problem

(3.9) $\min_{S\subseteq P}\{\sum_{j\in S}\alpha_{sj}+\sum_{j\in S}\sum_{k\in P\setminus S}\beta_{jk}+\sum_{j\in P\setminus S}\sum_{i:v_i\in T_t}\alpha_{ij}\}$.

In order to solve (3.9) we need some results from network flow theory.

A *directed graph* is defined to be a pair (V,A) where V is a set of element called *vertices*, and A is a set of ordered pairs of vertices, called *arcs*. Consider the directed graph G of Figure 3.10 with vertex set $\{s,t,1,2,\ldots,p\}$ and are set $\{(s,j)\,|\,j = 1,\ldots,p\}\cup\{(j,t)\,|\,j = 1,\ldots,p\}\cup \{(i,j)\,|\,i,j = 1,\ldots,p,i \neq j\}$. Associated with each arc is a capacity. The capacity of the arc (s,j) is given by a_j, the capacity of (j,t) is b_j, $j = 1,\ldots,p$, the capacity of (i,j) is given by c_{ij}, $i,j = 1,\ldots,p$, $i \neq j$.

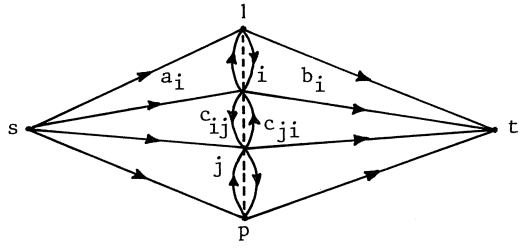

Figure 3.10. The directed graph G

In the *maximum flow* problem on the graph G we want to transport as much as possible from s (called the *source*) to t (called the *sink*) under the conditions that nothing is left behind in intermediate nodes and that the capacity restriction are respected. This can formulated as

(3.11) $\max v$

$$\text{s.t. } \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = \begin{cases} v & i = s \\ 0 & i \neq s,t \\ -v & i = t \end{cases}$$

$$0 \le x_{ij} \le c_{ij}, \qquad i,j = 1,\ldots,p,\ i \neq j,$$
$$0 \le x_{sj} \le a_j, \qquad j = 1,\ldots,p,$$
$$0 \le x_{jt} \le b_j, \qquad j = 1,\ldots,p.$$

An (s,t)-*cutset* is identified by a pair (S,T) of subsets of vertices with $s \in S$, $t \in T$ which partition the vertex set. The *capacity* of a cutset (S,T) is the sum of the capacities of all arcs with the predecessor in S and the successor in T.

A well-known result in the theory of maximum flows states that the

maximum flow from s to t is equal to the minimum capacity of an (s,t) cutset. The algorithm of KARZANOV [1974] finds a maximum flow and a minimum cutset in $O(p^3)$ time.

The (s,t) cutsets of the digraph G are defined by a subset $S \subseteq P$. The cutset defined by $S \subseteq P$ is given by $(\{s\} \cup (P \setminus S), \{t\} \cup S)$ and its capacity is equal to

$$(3.12) \qquad \sum_{j \in S} a_j + \sum_{j \in P \setminus S} b_j + \sum_{j \in P \setminus S} \sum_{k \in S} c_{jk}.$$

The problem of finding a minimum capacity cutset can be formulated as

$$(3.13) \qquad \min_{S \subseteq P} \{\sum_{j \in S} a_j + \sum_{j \in P \setminus S} b_j + \sum_{j \in P \setminus S} \sum_{k \in S} c_{jk}\}.$$

Note that problem (3.13) is equivalent to problem (3.9) if we define $a_j = \alpha_{sj}$, $b_j = \sum_{i:v_i \in T_t} \alpha_{ij}$ and $c_{jk} = \beta_{jk}$. Therefore problem (3.9) can be solved in $O(p^3)$ time.

The total complexity of the algorithm is $O(np^3)$ since we have to solve a minimum cut problem for each of the n-1 edges of the tree with n vertices.

In contrast with the problem on the tree the p-median problem with mutual communication on an arbitrary graph is NP-hard. It is sufficient to establish NP-completeness for the following decision problem.

MMC

Instance: A graph $G' = (V',E')$ with $V' = \{v'_1, v'_2, \ldots, v'_n\}$, $p \in \mathbb{N}$ weights $\alpha(v'_i, j) \in \mathbb{Z}^+$, $i = 1,2,\ldots,n$, $j = 1,\ldots,p$, $\beta(j,k) \in \mathbb{Z}^+$, $j,k = 1,\ldots,p$. lengths $\ell(e) \in \mathbb{Z}^+$ for all $e \in E'$, $B \in \mathbb{Q}^+$.

Question: Is there a set $X = \{x_1, x_2, \ldots, x_p\}$ on G' such that $\sum_{i=1}^{n} \sum_{j=1}^{p} \alpha(v'_i,j) d(v'_i,x_j) + \frac{1}{2} \sum_{j=1}^{p} \sum_{k=1}^{p} \beta(j,k) d(x_j,x_k) \leq B$?

We shall reduce the clique problem to MMC. The clique problem is known to be NP-complete (see GAREY and JOHNSON 1979). The problem is given by

CLIQUE

Instance: A graph $G = (V,E)$ and an integer c.

Question: Does there exist a subset $C \subseteq V$ of cardinality c such that $[j,k] \in E$ if $\{j,k\} \subseteq C$?

THEOREM 3.13. MMC *is* NP-*complete*.

PROOF. MMC belongs to the class *NP* since by using standard shortest path techniques one can test whether a given solution satisfies the bound B in polynomial time.

We shall reduce the problem CLIQUE to MMC. Let an instance of CLIQUE be given by G = (V,E) with V = $\{v_1,\ldots,v_n\}$ and integer c. The corresponding instance of MMC is defined by

$V' = V \cup W \cup U \cup \{p_1,p_2\}\cup\{q_1,q_2\}\cup\{r_1,r_2\}$, where $W = \{w_1,\ldots,w_n\}$ and $U = \{u_1,\ldots,u_n\}$, $E' = \{[p_1,v_i],[p_2,v_i],[q_1,w_i],[q_2,w_i],[r_1,u_i],[r_2,u_i],[v_i,w_i],[w_i,u_i] | i = 1,2,\ldots,n\}\cup\{[v_i,u_j] | [v_i,v_j] \in E,\ i < j\}$ (see Example 3.18), p = 3c,

$\alpha(p_1,j) = \alpha(p_2,j) = c-j+1, \alpha(q_1,j+c) = \alpha(q_2,j+c) = 2,\ \alpha(r_1,j+2c) = \alpha(r_2,j+2c) = j,\ j = 1,2,\ldots,c,\ \beta(j,j+c) = \beta(j+c,j+2c) = 2,\ j = 1,2,\ldots,c,\ \beta(i,j+2c) = 2,$ $i < j,\ i,j = 1,\ldots,c$, all other weights are zero,

$\ell(e) = 1$ for all $e \in E'$,

$B = 3(c^2+3c)$.

We claim that G = (V,E) contains a clique of size c iff there exist points $x_1,x_2,\ldots,x_{3c}$ on G' = (V',E') such that the weighted sum of distances is less than or equal to B, i.e.,

$$\begin{aligned}(3.14)\qquad &\sum_{j=1}^{c} [(c-j+1)(d(p_1,x_j)+d(p_2,x_j))+2(d(q_1,x_{j+c})+d(q_2,x_{j+c})) + \\ &\quad j(d(r_1,x_{j+2c})+d(r_2,x_{j+2c}))+2(d(x_j,x_{j+c})+d(x_{j+c},x_{j+2c})) + \\ &\quad \sum_{i=1}^{c}\sum_{j=i+1}^{c} 2d(x_i,x_{j+2c}) \le 3(c^2+3c).\end{aligned}$$

Let $x_1,x_2,\ldots,x_{3c}$ be points on G' = (V',E') such that (3.14) holds. We have the following inequalities

$$(3.15)\qquad d(p_k,q_k) \le d(p_k,x_j)+d(x_j,x_{j+c})+d(x_{j+c},q_k),\quad j = 1,\ldots,c,\ k = 1,2,$$

$$(3.16)\qquad d(q_k,r_k) \le d(q_k,x_{j+c})+d(x_{j+c},x_{j+2c})+d(x_{j+2c},r_k),\quad j = 1,\ldots,c,\ k = 1,2,$$

$$(3.17)\qquad d(p_k,r_k) \le d(p_k,x_i)+d(x_i,c_{j+2c})+d(x_{j+2c},r_k),\quad i < j,\ i,j = 1,\ldots,c,\ k = 1,2.$$

By adding all these inequalities together we find that the left hand side of (3.14) is greater than or equal to $c[d(p_1,q_1)+d(q_1,r_1)+d(p_2,q_2)+d(q_2,r_2)] + c(c-1)/2[d(p_1,r_1)+d(p_2,r_2)] = 3(c^2+3c)$. It follows that equality holds

in (3.14). Therefore equality holds for all inequalities (3.15), (3.16) and (3.17). It follows that $x_j \in V$, say $x_j = v_j$, then $x_{j+c} = w_j$ and $x_{j+2c} = u_j$, $j = 1,\ldots,c$. It follows from (3.17) that $[v_i,v_j] \in E$. Hence $\{v_j \mid j = 1,\ldots,c\}$ is a clique in $G = (V,E)$.

Let $\{v_j \mid j = 1,\ldots,c\}$ be a clique in $G = (V,E)$. Define $x_j = v_j$, $x_{j+c} = w_j$ and $x_{j+2c} = u_j$, $j = 1,..,c$. Then $x_1,\ldots,x_{3c}$ satisfies (3.14) with equality. □

EXAMPLE 3.18. Let $G = (V,E)$ be given by $V = \{v_1,v_2,v_3\}$ and $E = \{[v_1,v_2],[v_2,v_3]\}$. Then the graph $G' = (V'.E')$ of MMC is given by Figure 3.19

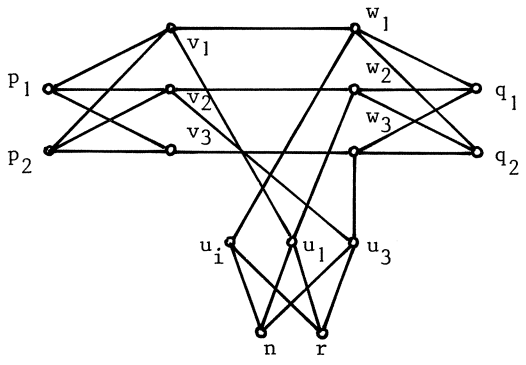

Figure 3.19. The graph $G' = (V',E')$.

CHAPTER 4

4. THE p-CENTER PROBLEM WITH MUTUAL COMMUNICATION

Let $T = (V,E)$ be a tree with $V = \{v_1,v_2,\ldots,v_n\}$ and let existing facilities be located at the vertices. We want to locate p new facilities on the tree such that the maximum over the weighted distances between existing and new facilities as well as between new facilities is minimized. This problem is known as the p-*center problem with mutual communication* and can be formulated as

$$(4.1)\qquad \min_{x_1,x_2,\ldots,x_p}\{\max\{\max_{i=1,\ldots,n,j=1,\ldots,p}\{\alpha_{ij}d(v_i,x_j)\},\ \max_{j,k=1,\ldots,p}\{\beta_{jk}d(x_j,x_k)\}\}\},$$

where α_{ij}, $i = 1,2,\ldots,n$, $j = 1,2,\ldots,p$ and $\beta_{jk}, j,k = 1,2,\ldots,p$ are positive integers.

We can reformulate the problem as

$$(4.2)\qquad \begin{aligned} &\min\ z\\ &\text{s.t. } d(v_i,x_j) \le z/\alpha_{ij}, \quad i = 1,\ldots,n,\ j = 1,\ldots,p,\\ &\qquad d(x_j,x_k) \le z/\beta_{jk}, \quad j,k = 1,\ldots,p. \end{aligned}$$

Problem (4.2) was solved by FRANCIS, LOWE and RATLIFF (1978). They first give necessary and sufficient conditions for the existence of points $x_1,x_2,\ldots,x_p$ on the tree such that the following distance constraints are satisfied:

$$(4.3)\qquad d(v_i,x_j) \le c_{ij} \quad = 1,\ldots,n,\ j = 1,\ldots,p,$$

$$(4.4)\qquad d(x_j,x_k) \le b_{jk}, \quad j,k = 1,\ldots,p,$$

where c_{ij}, $i = 1,\ldots,n$, $j = 1,\ldots,p$ and $b_{jk} = b_{kj}$ $j,k = 1,\ldots,p$ are non-negative numbers. These necessary and sufficient conditions for the existence of points $x_1,\ldots,x_p$ on the tree to satisfy (4.3) and (4.4), called separation conditions, lead to a closed form expression for the optimum value z^* of problem (4.2). They also give an algorithm to construct a feasible solution to (4.3), (4.4) if one exists. Substituting $z = z^*$ in (4.2) and using this algorithm gives an optimal solution to the p-center problem with mutual communication.

We will present a simple proof of the separation conditions. This proof differs from the one given by FRANCIS, LOWE and RATLIFF (1978).

Construct the graph BC as follows. We have vertices E_i corresponding to v_i, $i = 1,2,\ldots,n$, and N_j corresponding to x_j, $j = 1,2,\ldots,p$ edges $[E_i,N_j]$ of length c_{ij}, $i = 1,2,\ldots,n$, $j = 1,2,\ldots,p$, and $[N_j,N_k]$ of length b_{jk}, $j,k = 1,\ldots,p$, $j < k$. The graph BC for the case we have 3 vertices and $p = 3$ is given in Figure 4.5a.

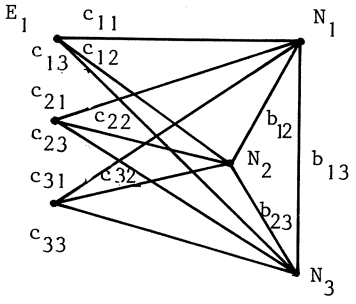

Figure 4.5. The graph BC.

Let $L(E_i,E_j)$ denote the length of a shortest path in BC from E_i to E_j. The *separation conditions* are given by

$$d(v_i,v_j) \leq L(E_i,E_j), \quad i,j = 1,2,\ldots,n, \quad i \neq j.$$

THEOREM 4.6. *There exist points* $x_1,\ldots,x_p$ *satisfying the distance constraints* (4.3) *and* (4.4) *iff the separation conditions hold.*

PROOF. We use induction on the number p. In the case that $p = 1$, the theorem is equivalent to Lemma 1.20. Suppose the theorem is true for distance

constraints with less than p points ($p \geq 2$). Consider the distance constraints (4.3),(4.4). Suppose we have found points $x_1,\ldots,x_{p-1}$ satisfying

$$d(v_i,x_j) \leq c_{ij}, \quad i = 1,2,\ldots,n,\ j = 1,2,\ldots,p-1,$$
$$d(x_j,x_k) \leq b_{jk}, \quad j,k = 1,\ldots,p-1.$$

According to the induction hypothesis there exists a point x_p such that

$$d(v_i,x_p) \leq c_{ip}, \quad i = 1,2,\ldots,n,$$
$$d(x_j,x_p) \leq b_{jp}, \quad j = 1,\ldots,p-1,$$

if and only if

$$d(v_i,v_j) \leq c_{ip} + c_{jp}, \quad i,j = 1,2,\ldots,n, i \neq j,$$
$$d(v_i,x_j) \leq c_{ip} + b_{jp}, \quad i = 1,2,\ldots,n,\ j = 1,\ldots,p-1,$$
$$d(x_j,x_k) \leq b_{jp} + b_{kp}, \quad j,k = 1,\ldots,p-1.$$

We conclude that there exist points $x_1,x_2,\ldots,x_p$ satisfying (DC) if and only if

$$d(v_i,v_j) \leq c_{ip} + c_{jp}, \quad i,j = 1,2,\ldots,n, \quad i \neq j,$$

and there exist points $x_1,x_2,\ldots,x_{p-1}$ satisfying

$$d(v_i,x_j) \leq \min\{c_{ij},c_{ip} + b_{jp}\}, \quad i = 1,2,\ldots,n,\ j = 1,\ldots,p-1, \tag{4.7}$$

$$d(x_j,x_k) \leq \min\{b_{jk},b_{jp} + b_{kp}\}, \quad j,k = 1,\ldots,p-1. \tag{4.8}$$

Let BC' be the graph with vertices E_i', $i = 1,2,\ldots,n$, N_j', $j = 1,\ldots,p$ and edges $[E_i',N_j']$ of length $\min\{c_{ij},b_{jp} + c_{ip}\}$, $i = 1,2,\ldots,n$, $j = 1,\ldots,p-1$, $[E_i',N_p']$ of length c_{ip}, $i = 1,2,\ldots,n$, and $[N_j',N_k']$ of length $\min\{b_{jk},b_{jp} + b_{kp}\}$, $j,k = 1,\ldots,p-1$, $j < k$. The graph BC' for the case $n = p = 3$ is given in Figure 4.9

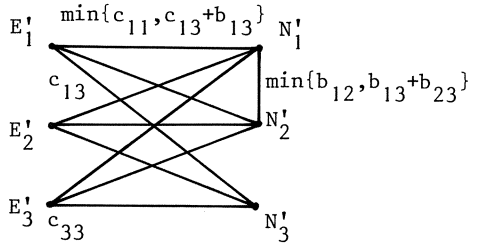

Figure 4.9. The graph BC'.

Let $L(E_i',E_j')$ denote the length of a shortest path from E_j' to E_j' in BC'. Then using the induction hypothesis on (4.7) and (4.8) and taking into account the constraints

$$d(v_i,v_j) \leq c_{ip} + c_{jp}, \quad i,j = 1,2,\ldots,n,\ i \neq j,$$

it follows that necessary and sufficient conditions for the existence of $x_1,\ldots,x_p$ satisfying (4.3) and (4.4) are

$$d(v_i,v_j) \leq L(E_i',E_j'), \quad i,j = 1,2,\ldots,n,\ i \neq j.$$

We shall prove in Lemma 4.10 that $L(E_i,E_j) = L(E_i',E_j')$, which completes the proof. □

LEMMA 4.10. *$L(E_i,E_j) = L(E_i',E_j')$ for all $i,j = 1,2,\ldots,n,\ i \neq j$.*

PROOF. Consider a path from E_i' to E_j' in BC'. We construct a path in BC of the same length. This proves $L(E_i,E_j) \leq L(E_i',E_j')$. An edge $[E_i',N_j']$, $j \neq p$ is replaced by the edges $[E_i,N_p][N_p,N_j]$ if $c_{ip} + b_{jp} < c_{ij}$. Similarly we replace $[E_j',N_k']$, $k \neq p$ by $[E_j,N_p][N_p,N_k]$ if $b_{jp} + b_{kp} < b_{jk}$. All other edges are replaced by the corresponding edges. In this way we have constructed a path in BC of the same length.

Consider the shortest path from E_i to E_j in BC. We construct a path in BC' of the same length. This proves that $L(E_i',E_j') \leq L(E_i,E_j)$. If the path contains an edge $[N_j,N_k]$ $(j,k \neq p)$, then we know that $b_{jk} \leq b_{jp} + b_{kp}$ and we take the corresponding edge in BC'. If the path contains an edge $[E_i,N_j]$ $(j \neq p)$, then $c_{ij} \leq c_{ip} + b_{jp}$ and we take the corresponding edge

in BC'. If $[E_i,N_p][N_p,N_j]$ occurs in the path, then $c_{ip} + b_{jp} \le c_{ij}$ and these two edges are replaced by $[E_i',N_j']$. If $[N_j,N_p][N_p,N_k]$ occurs in the path, then $b_{jp} + b_{kp} \le b_{jk}$ and these two edges are replaced by $[N_j',N_k']$. The constructed path in BC' has the same length as the shortest path in BC. □

Let us return to problem (4.2). We construct the graph BC with $c_{ij} = 1/\alpha_{ij}$, $i = 1,\ldots,n$, $j = 1,\ldots,p$ and $b_{jk} = 1/\beta_{jk}$, $j,k = 1,\ldots,p$, $j < k$. It follows from the separation conditions that (4.2) has a feasible solution iff

$$(4.11) \qquad d(v_i,v_j) \le zL(E_i,E_j), \quad i,j = 1,\ldots,n,\ i \ne j.$$

We conclude that

$$(4.12) \qquad z^* = \max_{i,j=1,\ldots,n,i\ne j}\{d(v_i,v_j)/L(E_i,E_j)\}.$$

In contrast with the problem on the tree the p-center problem with mutual communication on an arbitrary graph is NP-hard. It is sufficient to establish NP-completeness for the following decision problem

CMC

Instance: A graph $G' = (V',E')$ with $V' = \{v_1',v_2',\ldots,v_n'\}$, $p \in \mathbb{N}$, weights $\alpha(v_i',j) \in \mathbb{Z}^+$, $i = 1,2,\ldots,n$, $j = 1,\ldots,p$, $\beta(j,k) \in \mathbb{Z}^+$, $j,k = 1,\ldots,p$, lengths $\ell(e) \in \mathbb{Z}^+$, $e \in E'$, $B \in Q^+$.

Question: Is there a set $X = \{x_1,x_2,\ldots,x_p\}$ on G' such that

$$\alpha(v_j',j)d(v_i',x_j) \le B, \quad i = 1,\ldots,n,\ j = 1,\ldots,p,$$
$$\beta(j,k)d(x_j,x_k) \le B, \quad j,k = 1,\ldots,p?$$

We reduce the clique problem defined in Chapter 3 to CMC.

THEOREM 4.13. CMC *is* NP-*complete.*

PROOF. CMC belongs to the class *NP* since by using standard shortest path techniques one can test whether a given solution satisfies the bound B in polynomial time.

We shall reduce the problem CLIQUE to CMC. Let an instance of CLIQUE be given by $G = (V,E)$ with $V = \{v_1,\ldots,v_n\}$ and integer c. The corresponding

instance of CMC is defined by
$V' = V \cup W \cup U \cup \{p\} \cup \{q\} \cup \{r\}$, where $W = \{w_1,w_2,\ldots,w_n\}$ and $U = \{u_1,u_2,\ldots,u_n\}$,
$E' = \{[p,v_i],[q,w_i],[r,u_i],[v_i,w_i],[w_i,u_i] \mid i = 1,\ldots,n\} \cup$
$\cup \{[v_i,u_j] \mid [v_i,v_j] \in E, i < j\}$, $p = 3c$,
$\alpha(p,j) = \alpha(q,j+c) = \alpha(r,j+2c) = 1$, $j = 1,\ldots,c$, $\beta(j,j+c) = \beta(j+c,j+2c) = 1$,
$j = 1,\ldots,c$, $\beta(i,j+2c) = 1, i,j = 1,\ldots,c, i < j$, all other weights are zero,
$\ell(e) = 1$ for all $e \in E'$, $B = 1$.

We claim that $G = (V,E)$ contains a clique of size c iff there exist points $x_1,x_2,\ldots,x_{3c}$ on $G' = (V',E')$ such that the weighted distances are less than or equal to B, i.e.

$$(4.14) \qquad d(p,x_j) \le 1,\ d(q,x_{j+c}) \le 1,\ d(r,x_{j+2c}) \le 1,\quad j = 1,\ldots,c,$$

$$(4.15) \qquad d(x_j,x_{j+c}) \le 1,\ d(x_{j+c},x_{j+2c}) \le 1,\quad j = 1,\ldots,c,$$

$$(4.16) \qquad d(x_i,x_{j+2c}) \le 1,\quad i,j = 1,\ldots,c,\ i < j.$$

Let $x_1,x_2,\ldots,x_{3c}$ be points on $G' = (V',E')$ satisfying (4.14), (4.15) and (4.16). Since $d(p,q) = d(q,r) = 1$ we have from (4.14) and (4.15) that $x_j \in V$, say $x_j = v_j$, $x_{j+c} = w_j$ and $x_{j+2c} = u_j$, $j = 1,\ldots,c$. It follows from (4.16) that $[v_i,u_j] \in E'$, i.e., $[v_i,v_j] \in E$, $i < j$. Hence $\{v_j \mid j = 1,\ldots,c\}$ is a clique in the graph $G = (V,E)$.

Let $\{v_j \mid j = 1,\ldots,c\}$ be a clique of size c in G. Define $x_j = v_j$, $x_{j+c} = w_j$ ans $x_{j+2c} = u_j$, $j = 1,\ldots,c$. Then $x_1,x_2,..,x_{3c}$ satisfy (4.14), (4.15) and (4.16). □

EXAMPLE 4.17. Let $G = (V,E)$ be given by $V = \{v_1,v_2,v_3\}$ and $E = \{[v_1,v_2],[v_2,v_3]\}$. Then the graph $G' = (V' E')$ of CMC is given by Figure 4.18

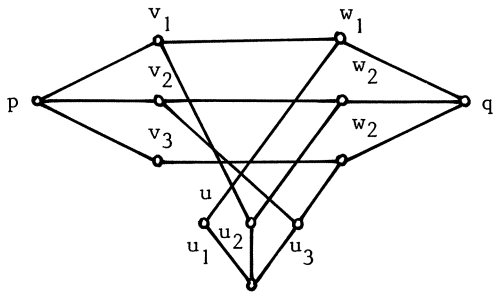

Figure 4.18. The graph $G' = (V',E')$.

CHAPTER 5

5. FARKAS' LEMMA IN RECTILINEAR LOCATION PROBLEMS

In 1902 Farkas proved what is known as a fundamental result in linear programming.

LEMMA 5.1 (Farkas' lemma). *The system of linear inequalities* $Ax \leq b$ has *a solution* x *iff* $yb \geq 0$ *whenever* $yA = 0$ *for any nonnegative vector* y. □

Several extensions of Farkas' lemma are known. In this chapter we will use an extension proved by FOURIER (1826), KUHN (1956) and MOTZKIN (1936) (see also STOER and WITZGALL (1970)).

LEMMA 5.2. *Let* A_1, A_2 *be matrices and* b_1, b_2 *vectors. Then there exists a vector* x *such that* $A_1x < b_1$ *and* $A_2x \leq b_2$ *iff for all nonnegative vectors* y_1 *and* y_2 *one has: if* $y_1A_1 + y_2A_2 = 0$, *then* $y_1b_1 + y_2b_2 \geq 0$ *and furthermore if* $y_1 \neq 0$, *then* $y_1b_1 + y_2b_2 > 0$. □

The best way to show how one can use Farkas' lemma in solving rectilinear center problems is by considering the following simple example, which is known as the 1-*center problem*.

$$(5.3) \qquad \min_{x,y}\{\max_{i=1,\ldots,n}\{w_i(|a_i-x|+|b_i-y|)+h_i\}\},$$

where (a_i,b_i) is a point in the plane, w_i, h_i are positive integers, $i = 1,\ldots,n$. We can reformulate this problem as

$$(5.4) \qquad \min z$$
$$\text{s.t. } w_i(|a_i-x|+|b_i-y|)+h_i \leq z, \quad i = 1,\ldots,n.$$

For the rectilinear distance we have

$$
\begin{aligned}
|a_i-x|+|b_i-y| &= \max\{x-a_i+y-b_i,\ a_i-x+b_i-y,\ x-a_i+b_i-y, a_i-x+y-b_i\} \\
&= \max\{\max\{x-a_i+y-b_i, a_i-x+b_i-y\}, \max\{x-a_i+b_i-y, \\
&\quad a_i-x+y-b_i\}\} \\
&= \max\{|(x+y)-(a_i+b_i)|, |(x-y)-(a_i-b_i)|\}
\end{aligned}
$$

if we define $(\hat{p},\hat{q}) = (p+q,p-q)$, then (5.4) is equivalent to

$$
\begin{aligned}
(5.5)\qquad & \min z \\
& \text{s.t. } w_i \max\{|\hat{x}-\hat{a}_i|, |\hat{y}-\hat{b}_i|\} + h_i \le z.
\end{aligned}
$$

This transformation of the coordinate plane, which corresponds with a rotation over 45^o, enables us to solve the 1-center problem in the plane as two independent problems on the line of the following type.

$$
\begin{aligned}
(5.6)\qquad & \min z \\
& \text{s.t. } |x-a_i| \le (z-h_i)/w_i, \quad i = 1,\ldots,n,
\end{aligned}
$$

or equivalently

$$
\begin{aligned}
(5.7)\qquad & \min z \\
& \text{s.t. } x \le \min_{i=1,\ldots,n}\{a_i+(z-h_i)/w_i\}, \\
& \quad -x \le \min_{j=1,\ldots,n}\{-a_j+(z-h_j)/w_j\}.
\end{aligned}
$$

Using Farkas' Lemma we have

$$
\exists x \begin{bmatrix} 1 \\ -1 \end{bmatrix} x \le \begin{bmatrix} \min_i\{a_i+(z-h_i)/w_i\} \\ \min_j\{-a_j+(z-h_j)/w_j\} \end{bmatrix} \iff
$$

$$
\forall y_1,y_2 \ge 0 [y_1-y_2 = 0 \Rightarrow y_1 \min_i\{a_i+(z-h_i)/w_i\} +
$$

$$
+ y_2 \min_j\{-a_j+(z-h_j)/w_j\} \ge 0] \iff
$$

$$
\min_i\{a_i+(z-h_i)/w_i\} + \min_j\{-a_j+(z-h_j)/w_j\} \ge 0 \iff
$$

$$
z \ge (|a_i-a_j| + h_i/w_i + h_j/w_j) w_i w_j/(w_i+w_j), \quad i,j = 1,2,\ldots,n,\ i < j.
$$

It follows that the optimum value z^* of the 1-center problem is given by

$$(5.8) \qquad z^* = \max_{i,j=1,2,\ldots,n,i<j}\{(|a_i-a_j| + h_i/w_i+h_j/w_j)w_iw_j/(w_i+w_j)\}.$$

This simple example has all the ingredients for a succesful use of Farkas' Lemma in solving rectilinear center problems. We formulate the problem as

$$(5.9) \qquad \text{determine } z^* = \min\{z \mid Ax \leq b(z)\},$$

where A is a matrix and b(z) is a vector each component of which is a function of the variable z. We then use Farkas' Lemma to derive a necessary and sufficient condition for the system $Ax \leq b(z)$ to have a solution x. This condition takes on the form $z \geq c$, where c is a constant depending only on the problem data. It is clear that $z^* = c$.

The second problem we consider is the *round-trip 1-center problem* in the rectilinear plane. The round-trip center problem is discussed in Chapter 1. The round-trip 1-center problem in the rectilinear plane was solved by CHAN and HEARN (1977). It can be formulated as

$$(5.10) \qquad \min_{x,y}\{\max_{1\leq i\leq n}\{w_i(|x-a_i| + |y-b_i| + |x-c_i| + |y-d_i|) + h_i\}\}$$

where (a_i,b_i), (c_i,d_i) are points in the plane and w_i,h_i are positive integers, $i = 1,2,\ldots,n$. An equivalent formulation of the problem is given by

$$(5.11) \qquad \begin{aligned} &\min z \\ &\text{s.t. } w_i(|x-a_i|+|y-b_i|+|x-c_i|+|y-d_i|) + h_i \leq z, \quad i = 1,\ldots,n. \end{aligned}$$

Each of the constraints in (5.11) can be written as a system of sixteen linear inequalities. This leads to the following formulation of the problem

$$(5.12) \qquad \begin{aligned} &\min z \\ &\text{s.t. } \tfrac{1}{2}\max_i\{a_i+b_i+c_i+d_i-(z-h_i)/w_i \leq x+y \leq \\ &\qquad \tfrac{1}{2}\min_j\{a_j+b_j+c_j+d_j+(z-h_j)/w_j\}, \\ &\qquad \tfrac{1}{2}\max_i\{a_i+c_i-b_i-d_i-(z-h_i)/w_i\} \leq x-y \leq \\ &\qquad \tfrac{1}{2}\min_j\{a_j+c_j-d_j-b_j+(z-h_j)/w_j\}, \end{aligned}$$

$$\tfrac{1}{2}\max_i\{a_i+c_i+|b_i-d_i|-(z-h_i)/w_i\} \le x \le$$
$$\tfrac{1}{2}\min_j\{a_j+c_j-|b_j-d_j|+(z-h_j)w,\},$$
$$\tfrac{1}{2}\max_i\{b_i+d_i+|a_i-c_i|-(z-h_i)/w_i\} \le y \le$$
$$\tfrac{1}{2}\min_j\{b_j+d_j-|a_j-c_j|+(z-h_j)/w_j\},$$
$$z \ge \max_i\{w_i(|a_i-c_i|+|b_i-d_i|)+h_i\}.$$

Using Farkas' Lemma we find that x,y satisfying the system of inequalities

$$a_1 \le x+y \le a_2,$$
$$b_1 \le x-y \le b_2,$$
$$c_1 \le x \le c_2,$$
$$d_1 \le y \le d_2,$$

exist if and only if the following twelve conditions are satisfied:

$$a_1 \le a_2, \quad c_1 \le \tfrac{1}{2}(a_2+b_2), \quad a_1 \le c_2+d_2,$$
$$b_1 \le b_2, \quad c_2 \ge \tfrac{1}{2}(a_1+b_1), \quad a_2 \ge c_1+d_1,$$
$$c_1 \le c_2, \quad d_1 \le \tfrac{1}{2}(a_2-b_1), \quad b_1 \le c_2-d_1,$$
$$d_1 \le d_2, \quad d_2 \ge \tfrac{1}{2}(a_1-b_2), \quad b_2 \ge c_1-d_2.$$

Applying this to the round-trip location problem leads to the following seven necessary and sufficient conditions:

$$z \ge \max_i\{w_i(|a_i-c_i|+|b_i-d_i|)+h_i\},$$
$$z \ge \max_{1\le i\le j\le n}\{(|a_i+c_i-a_j-c_j|+|b_i+d_i-b_j-d_j| +$$
$$+ h_i/w_i+h_j/w_j)w_iw_j/(w_i+w_j)\},$$
$$z \ge \max_{1\le i\le j\le n}\{|(a_i+c_i-a_j-c_j|+|b_i-d_i|+|b_j-d_j|+$$
$$+ h_i/w_i+h_j/w_j)w_iw_j/(w_i+w_j)\},$$
$$z \ge \max_{1\le i\le j\le n}\{|b_i+d_i-b_j-d_j|+|a_i-c_i|+|a_j-c_j|+$$
$$+ h_i/w_i+h_j/w_j)w_iw_j/(w_i+w_j)\}$$
$$z \ge \max_{i,j,k,i\le j}\{(|2(a_k+c_k)-a_i-c_i-a_j-c_j|+|b_i+d_i-b_j-d_j|+2|b_k-d_k|+$$
$$+ h_i/w_i+h_j/w_j+2h_k/w_k)w_iw_jw_k/(2w_iw_j+w_iw_k+w_jw_k)\},$$

$$z \geq \max_{i,j,k,i\leq j}\{(|2(b_k+d_k)-b_i-d_i-b_j-d_j|+|a_i+c_i-a_j-c_j|+2|a_k-c_k|+$$
$$+ h_i/w_i+h_j/w_j+2h_k/w_k)w_iw_jw_k/2w_iw_j+w_iw_k+w_jw_k)\},$$

$$z \geq \max_{i,j,k}\{(|a_i+c_i-a_k-c_k|+|b_j+d_j-b_k-d_k|+|a_j-c_j|+|b_i-d_i| +$$
$$+ h_i/w_i+h_j/w_j+h_k/w_k)w_iw_jw_k/(w_iw_j+w_iw_k+w_jw_k)\}.$$

The optimum value is equal to the maximum of the righthand sides of these seven inequalities.

Our third example is the p-*center problem with mutual communication* in the rectilinear plane. This problem can be formulated as

$$(5.13)\qquad \min_{x_1,\ldots,x_p,y_1,\ldots,y_p}\{\max\{\max_{1\leq i\leq n,\,1\leq j\leq p}\{\alpha_{ij}(|a_i-x_j|+|b_i-y_j|)\},$$
$$\max_{1\leq j\leq k\leq p}\{\beta_{jk}(|x_j-x_k|+|y_j-y_k|)\}\}\},$$

where (a_i,b_i) are points in the plane, $i = 1,2,\ldots,n$, α_{ij},β_{jk} are positive integers, $i = 1,\ldots,n, j, k = 1,\ldots,p$. Using the same transformation as in the case of the 1-center problem the problem can be separated into two independent problems on a line, of the type

$$(5.14)\qquad \min z$$
$$\text{s.t. } |a_i-x_j| \leq z/\alpha_{ij},\quad i = 1,\ldots,n,\ j = 1,\ldots,p,$$
$$|x_j-x_k| \leq z/\beta_{jk},\quad j,k = 1,\ldots,p.$$

In Chapter 4 we derived necessary and sufficient conditions for the more general case of a tree. Our objective is to show how these conditions can be derived from Farkas' Lemma in this special case of the problem on a line.

We consider the set of inequalities given by

$$(5.15)\qquad |a_i-x_j| \leq c_{ij},\quad i = 1,2,\ldots,n,\quad j = 1\ 2,\ldots,p,$$
$$|x_j-x_k| \leq b_{jk},\quad j,k = 1,\ldots,p,$$

where c_{ij} and b_{jk} are nonnegative numbers, $i = 1,2,\ldots,n$, $j,k = 1,\ldots,p$.

This set of inequalities is equivalent to

$$(5.16)\qquad \left.\begin{aligned} x_j &\le a_i + c_{ij} \\ -x_j &\le c_{ij} - a_i \end{aligned}\right\} \; i = 1,2,\ldots,n, \quad j = 1,\ldots,p,$$

$$\left.\begin{aligned} x_j - x_k &\le b_{jk} \\ -x_j + x_k &\le b_{jk} \end{aligned}\right\} \; j,k = 1,\ldots,p.$$

Let j_k denote a column vector of length k consisting of all ones, and let I_k denote the $k \times k$ unity matrix. Let A be the matrix corresponding to the set of inequalities and let b be the corresponding righthand side, i.e.,

$$A = \begin{bmatrix} j_n & & & & \\ & j_n & & & \\ & & \ddots & & \\ & & & j_n & \\ -j_n & & & & \\ & -j_n & & & \\ & & \ddots & & \\ & & & -j_n & \\ j_{p-1} & & -I_{p-1} & & \\ -j_{p-1} & & I_{p-1} & & \\ & j_{p-2} & & -I_{p-2} & \\ & -j_{p-2} & & I_{p-2} & \\ & & \ddots & & \\ & & & 1 & -1 \\ & & & -1 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_p \\ e_1 \\ e_2 \\ \vdots \\ e_p \\ f_1 \\ f_1 \\ f_2 \\ f_2 \\ \vdots \\ f_{p-1} \\ f_{p-1} \end{bmatrix},$$

where $d_i = (a_i + c_{1i}, \ldots, a_n + c_{ni})^T$, $e_i = (-a_1 + c_{1i}, \ldots, -a_n + c_{ni})^T$, $i = 1,2,\ldots,p$, and $f_i = (b_{i,i+1}, \ldots, b_{ip})^T$, $i = 1,2,\ldots,p-1$. Then the set of inequalities (5.15) is equivalent to $Ax \le b$.

According to Farkas' Lemma there exists a vector x such that $Ax \le b$ iff $yb \ge 0$ for all $y \ge 0$ such that $yA = 0$. By scaling we may assume that each component of y is less than or equal to 1. This leads to the following result

$$(5.19)\qquad \exists x[Ax \le b] \quad \text{iff} \quad \min\{yb \mid yA = 0, 0 \le y \le 1\} \ge 0.$$

A well-known result states that a (0,1-1)-matrix is totally unimodular if

(1) there are no more than two nonzero elements in each row and (2) the columns can be partitioned into two subsets Q_1 and Q_2 such that (a) if a row contains two nonzero elements with the same sign, one element is in each of the subsets, (b) if a row contains two nonzero elements of opposite sign, both elements are in the same subset (see GARFINKEL and NEMHAUSER (1972)).

Since the matrix A in (5.17) is totally unimodular there is an optimum solution y which is a (0,1)-solution.

Consider the directed graph G with vertices E_i, $i = 1,\ldots,n$, and N_j, $j = 1,\ldots,p$, and arcs (E_i,N_j) of length $a_i + c_{ij}$, (N_j,E_i) of length $-a_i+c_{ij}$, $i = 1 \ldots,n$, $j = 1,\ldots,p$, and arcs (N_j,N_k), (N_k,N_j) of length b_{jk}. Let us denote by $L(E_i,E_j)$ the length of a shortest path from E_i to E_j.

The necessary and sufficient condition for the existence of solution to (5.15) which we derived in Chapter 4 are equivalent to

$$L(E_i,E_j) \geq 0. \tag{5.18}$$

In order to prove that $\min\{yb \mid yA = 0,\ y \in \{0,1\}\} \geq 0$ is equivalent to (5.18) it is sufficient to show that each feasible solution y is the sum of a number of directed paths in G starting from and ending in $\{E_i \mid i = 1,\ldots,n\}$.

Let y be the vector given by $y = (t_1,\ldots,t_p,u_1,\ldots,u_p,v_1,w_1,\ldots,v_{p-1}, w_{p-1})$, where $t_i = (t_{1i},\ldots,t_{ni})$, $u_i = (u_{1i},\ldots,u_{ni})$, $i = 1,2,\ldots,p$, $v_i = (v_{i+1,i},\ldots,v_{pi})$, $w_i = (w_{i+1,i},\ldots,w_{pi})$, $i = 1,2,\ldots,p-1$. Let us associate the following interpretation to the parameters in the vector y:

$t_{ki} = 1$ if and only if vertex N_i is entered through arc (E_k,N_i),
$u_{ki} = 1$ if and only if vertex E_k is entered through arc (N_i,E_k),
$w_{ij} = 1$ if and only if vertex N_i is entered through arc (N_j,N_i) $(i>j)$,
$v_{ij} = 1$ if and only if vertex N_j is entered through arc (N_i,N_j) $(i>j)$.

The constraints yA = 0 imply that

$$\sum_{k=1}^{n} (t_{ki}-u_{ki}) - \sum_{j=1}^{i-1} (v_{ij}-w_{ij}) + \sum_{j=i+1}^{p} (v_{ji}-w_{ji}) = 0, \quad i = 1,2,\ldots,p. \tag{5.19}$$

Summing these equalities over all i and noting that

$$\sum_{i=1}^{p} \sum_{j=1}^{i-1} (v_{ij}-w_{ij}) = \sum_{i=1}^{p} \sum_{j=i+1}^{p} (v_{ji}-w_{ji})$$

we find that

$$(5.20) \qquad \sum_{i=1}^{p} \sum_{k=1}^{n} t_{ki} = \sum_{i=1}^{p} \sum_{k=1}^{n} u_{ki}.$$

Constraints (5.19) say that the number of times we enter a vertex N_i is equal to the number of times we leave the same vertex, i = 1,2,...,p. Constraint (5.20) says that the number of times we enter the set $\{E_i \; i = 1,2,\ldots,n\}$ is equal to the number of times we leave the same set. We conclude that we can interpret y as the sum of a number of directed paths starting from and ending in $\{E_i \; i = 1,2,\ldots,n\}$. The constraint $yb \geq 0$ says

$$(5.21) \qquad \sum_{i=1}^{p} \sum_{k=1}^{n} [t_{ki}(a_k+c_{ki})+u_{ki}(-a_k+c_{ki})]+\sum_{i=1}^{p} \sum_{j=i+1}^{p} (v_{ji}+w_{ji})b_{ij} \geq 0.$$

Constraint (5.21) says that the sum of the lengths of the paths corresponding to y is nonnegative.

This completes the proof of the separation conditions for the p-center problem with mutual communication in the plane.

In our last example of this chapter we shall use Lemma 5.2 to construct an O(nlogn) algorithm to find the set of all *efficient points* in the rectilinear plane (see also CHALMET, FRANCIS and KOLEN (1981)).

Let $P_i = (a_i,b_i)$, i = 1,2,...,n be given points in the plane. A point (x_1,y_1) is *dominated in position* k if and only if there is a point (x,y) such that

$$(5.22) \qquad \begin{aligned} &|a_i-x|+|b_i-y| \leq q_i, \quad i = 1,2,\ldots,n \; i \neq k, \\ &|a_i-x|+|b_k-y| < q_k, \end{aligned}$$

where $q_i = |a_i-x_1|+|b_i-y_1|$, i = 1,2,...,n.
A point which can not be dominated in any position is called an *efficient point*. We denote the set of all efficient points by S^*, and call S^* the *efficient set*.

WENDELL, HURTER and LOWE (1977) have introduced and studied the problem of finding S^* and discuss some application contexts, with emphasis on multiple objective problems. They develop two different algorithms for constructory S^*. These algorithms are of $O(n^2)$ and $O(n^3)$ time.

S^* may also be of value in carrying out sensitivity analysis for single objective location problems since such problems have the property

that their optimal solutions are efficient.

We will characterize efficient points using Lemma 5.2. An equivalent version of Lemma 5.2 is the following one.

(5.23) $\exists x[A_1x<b_1, A_2x\leq b_2]$ iff $[\min\{y_1b_1+y_2b_2 \mid y_1A_1+y_2A_2 = 0,\ 0\leq y_1,y_2\leq 1\} \geq 0$

and

$$\min\{y_1b_1+y_2b_2 \mid y_1A_1+y_2A_2 = 0,\ 0\leq y_1y_2\leq 1, y_1\neq 0\} > 0] .$$

An equivalent formulation of (5.22) is given by

(5.24)
$$\begin{aligned}
x+y &\leq \min_{i\neq k}\{a_i+b_i+q_i\},\\
x-y &\leq \min_{i\neq k}\{a_i-b_i+q_i\},\\
-x+y &\leq \min_{i\neq k}\{-a_i+b_i+q_i\},\\
-x-y &\leq \min_{i\neq k}\{-a_i-b_i+q_i\},\\
x+y &< a_k+b_k+q_k,\\
x-y &< a_k-b_k+q_k,\\
-x+y &< -a_k+b_k+q_k,\\
-x-y &< -a_k-b_k+q_k.
\end{aligned}$$

Define

$$A_1=A_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix}$$

Let $s = (s_1,s_2.s_3,s_4)$ and $t = (t_1,t_2,t_3,t_4)$ be nonnegative vectors. Then

(5.25) $sA_1+tA_2 = 0$ iff $(s,t)A = 0$,

where

$$A = \begin{bmatrix} 1 & 0 & 0 & -1 & 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 \end{bmatrix} .$$

Note that A is totally nimodular. Therefore in order to check the condition of (5.23) it suffices to restrict ourselves to (0,1)-solutions (s,t). There are eight vectors $(s,t) = (s_1,s_2,s_3,s_4,t_1,t_2,t_3,t_4)$ satisfying $(s,t)A = 0$. The eight vectors and the corresponding conditions are

(5.26)
$(1,0,0,1,0,0,0,0)$: $(a_k+b_k+q_k)+(-a_k-b_k+q_k) > 0$,
$(1,0,0,0,0,0,0,1)$: $(a_k+b_k+q_k)+\min_{i\neq k}\{-a_i-b_i+q_i\} > 0$,
$(0,1,1,0,0,0,0,0)$: $(a_k-b_k+q_k)+(-a_k+b_k+q_k) > 0$,
$(0,1,0,0,0,0,1,0)$: $(+a_k-b_k+q_k)+\min_{i\neq k}\{-a_i+b_i+q_i\} > 0$,
$(0,0,1,0,0,1,0,0)$: $(-a_k+b_k+q_k)+\min_{i\neq k}\{a_i-b_i+q_i\} > 0$,
$(0,0,0,1,1,0,0,0)$: $(-a_k-b_k+q_k)+\min_{i\neq k}\{a_i+b_i+q_i\} > 0$,
$(0,0,0,0,1,0,0,1)$: $\min_{i\neq k}\{a_i+b_i+q_i\}+\min_{j\neq k}\{-a_j+b_j+q_j\} \geq 0$,
$(0,0,0,0,0,1,1,0)$: $\min_{i\neq k}\{a_i-b_i+q_i\}+\min_{j\neq k}\{-a_j+b_j+q_j\} \geq 0$.

The first and third inequality imply $q_k > 0$, i.e. $(x_1,y_1) \neq (a_k,b_k)$. The second, fourth, fifth and sixth inequality imply $q_i+q_k > |a_i-a_k|+|b_i-b_k|$, $i \neq k$. The last two inequalities imply that $q_i+q_j \geq |a_i-a_j|+|b_i-b_j|$, $i,j \neq k$, which is always satisfied because of the triangular inequality. We have proved the following result.

LEMMA 5.27. *A point* X *is dominated in position* k *iff* $X \neq P_k$ and $d(P_i,P_k) < d(P_i,X)+d(X,P_k)$, *for all* $i \neq k$. □

Here $P_i = (a_i,b_i)$, $i = 1,..,n$ and the distance d is the rectilinear distance. Lemma 5.27 leads to the following characterization of an efficient point.

(5.28). A point X is efficient iff for every point P_i there exists a point P_j such that $d(P_i,X)+d(X,P_j) = d(P_i,P_j)$.

We shall use this characterization of an efficient point to develop an O(n log n) algorithm to find the set of all efficient points

Through each point P_i construct a horizontal line and a vertical line. The horizontal (vertical) line should extend at least as far right and as far left (as far up and as far down) as every P_i. Subsequently whenever we refer to a *line* we mean one of these constructed lines. Figure 5.29 illustrates a number of definitions to follow. For any vertical line we define the set of points *east* (*west*) of the line to be the union of the line with the set of points to the right (left) of the line. Similarly we define the

set of points *north*, and the set of points *south* of each horizontal line. (For convenience we use the abbreviations N,E,W and S for North, East, West, and South respectively). Given any two distinct adjacent horizontal lines H and H' with H N of H', and any two distinct adjacent vertical V and V' with V E of V', we call the set of points lying W of V, E of V', S of H and N of H', a *box*, and denote the box by B. We call the collection of all boxes between any two adjacent vertical (horizontal) lines a *column* (*row*). Each of the four intersections of the box with a line we call an *edge* of B. We say two boxes are *adjacent* if their intersection is an edge of each box. The collection of all points lying S of H' and E of V we call the SE *direction* of B (abbreviated SE(B)); similarly we define SW, NW and NE directions of B, and use the abbreviations SW(B), NW(B) and NE(B) respectively. We say that a direction of B is empty if there is no point P_i in the direction, i = 1,2,...,n.

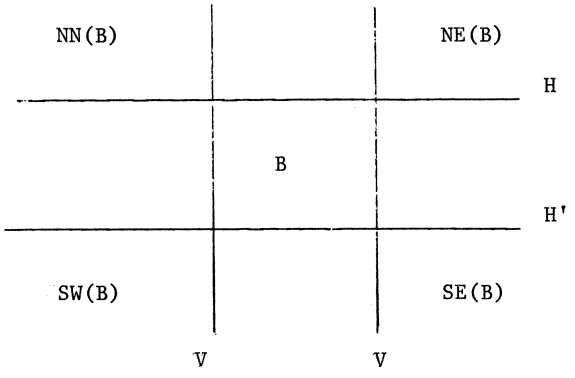

Figure 5.29. Directions of B.

We assume that not all the P_i lie on a single vertical, or on a single horizontal line, as in this case the set of efficient points is precisely the line segment joining the two P_i which are farthest apart.

We call a box a 0-*box*, 1-*box* or 2-*box* if 0,1, or 2 directions respectively are empty; it is possible to have only these three types of boxes. For any 1-box B we call the two edges opposite the empty direction the *leading edges* of B. It follows directly from the result (5.28) that each point in a 0-box or in a 2-box is an efficient point, and that only leading edges of a 1-box are possible candidates for containing efficient points.

Denote the horizontal lines by $H_1,H_2,\ldots,H_{p+1}$ from N to S and the rows by $R_1,\ldots,R_p$ from N to S. For R_i, $1 \le i \le p$, define $NW_i(SW_i)$ to be the

x-coordinate of the westmost point P_j which is N(S) of R_i. Define $NE_i(SE_i)$ to be the x-coordinate of the eastmost point P_j which is N(S) of R_i. For each R_i we define $W_i^* = \max\{NW_i, SW_i\}$ and $E_i^* = \min\{NE_i, SE_i\}$. For $1 \le i \le p+1$ denote by $W_i(E_i)$ the x-coordinate of the westmost (eastmost) point P_j on H_i. Let $V' < B < V$ mean that box B is E of V' and W of V. The following two lemmas are trivial.

LEMMA 5.30. *A box* B *in* R_i *is a* 0*-box if and only if* $W_i^* < E_i^*$ *and* $W_i^* < B < E_i^*$. □

LEMMA 5.3.1. *A box* B *in* R_i *is a* 2*-box if and only if* $W_i^* > E_i^*$ *and* $E_i^* < B < W_i^*$. □

The next two lemmas characterize horizontal and vertical leading edges belonging to the set of efficient points.

LEMMA 5.3.2. *A vertical leading edge* e *of a* 1*-box* B *in* R_i*, contained in the vertical line* V *and not an edge of a* 0*-box or* 2*-box, belongs to the set of efficient points if and only if* $W_i^* = V = E_i^*$.

PROOF. If $e \in V_1$ (the westmost vertical line), then we have two possibilities:

1. SE(B) is the only empty direction, i.e., $SW_i = SE_i = NW_i = V_1$ and $NE_i > SE_i$
2. NE(B) is the only empty direction, i.e., $SW_i = NW_i = NE_i = V_1$ and $SE_i > NE_i$.

If $e \in V_{q+1}$ (the eastmost vertical line), then we have two possibilities:

1. NW(B) is the only empty direction, i.e., $NW_i = NE_i = SE_i$ and $SW_i < SE_i$;
2. NW(B) is the only empty direction, i.e., $SW_i = SE_i = NE_i$ and $NW_i < NE_i$.

If $e \in V_j$ $(1<j<q+1)$, then e is not in a 0-box or 2-box if and only if e is a leading edge of a box B' adjacent to B(see Figure 5.33)

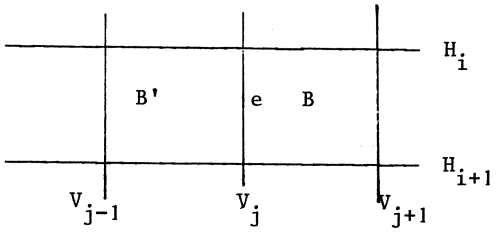

Figure 5.33. Leading edge e of B and B'.

There are two possibilities:

1. SW(B') and SE(B) are the only empty directions, i.e., $SW_i = SE_i = V_j$, $NW_i < SW_i$ and $NE_i > SE_i$;
2. NW(B') and NE(B) are the only empty directions, i.e., $NW_i = NE_i = V_j$, $SW_i < NW_i$ and $SE_i > NE_i$.

Combining all these results prove the lemma. □

LEMMA 5.3.4. *A horizontal edge* e *on* H_i *between* V_{j-1} *and* V_j, *which is not contained in a 2-box, belongs to the set of efficient solutions if and only if*

$$\max\{NW_i, SW_{i-1}\} \leq V_{j+1} \text{ and } V_j \leq \min\{NE_i, SE_{i-1}\}.$$

PROOF. If e is on H_1, then e belongs to the efficient set if and only if there exist a point on H_1 that lies W of V_{j-1} and there exist a point that lies E of V_j. If we define $SW_0 = -\infty$ and $SE_0 = \infty$, then this case can be characterized as indicated by the lemma. If e is on H_{p+1}, then when we define $NW_{p+1} = -\infty$ and $NE_{p+1} = \infty$ this can also be characterized as indicated by the lemma. If e is on H_i, $1 < i < p+1$, then let the box B and the box B' be the boxes in R_i, R_{i-1} respectively which have e in common with each other (see Figure 5.35).

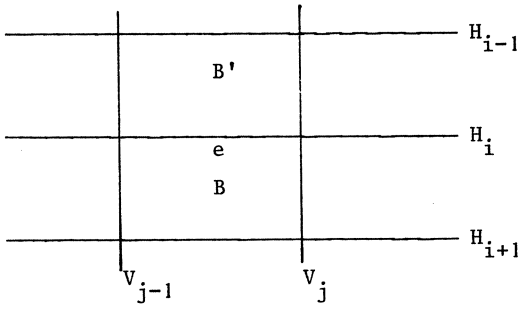

Figure 5.35. Leading edge e on H_i.

Box B' and B are not a 2-box, i.e. if $SW(B') = \emptyset$, then $NE(B') \neq \emptyset$ and e cannot contain efficient points according to result (5.28). The same holds when $SE(B') = \emptyset$, $NE(B) = \emptyset$, or $NW(B) = \emptyset$. If $NW(B) \neq \emptyset$, $NE(B) \neq \emptyset$, $SW(B') \neq \emptyset$ and $SE(B') \neq \emptyset$, then e belongs to the set of efficient points. We conclude that e belongs to the set of efficient points if and only if $NW(B) \neq \emptyset$, $NE(B) \neq \emptyset$, $SW(B') \neq \emptyset$ and $SE(B') \neq \emptyset$, i.e.,

$V_{j-1} \geq \max\{NW_i, SW_{i-1}\}$ and $V_j \leq \min\{NE_i, SE_{i-1}\}$. □

The following algorithm constructs the set of all efficient points. The correctness of the algorithm follows from the previous lemmas.

ALGORITHM. Rank the existing points by their y-coordinates to determine the lines $H_1,\ldots,H_{p+1}$. Compute W_i and E_i, $1 \leq i \leq p+1$, NW_i, NE_i, SW_i, SE_i, $1 \leq i \leq p$ and W_i^* and E_i^*, $1 \leq i \leq p$. If $W_i^* < E_i^*$, then all boxes B for which $W_i^* < B < E_i^*$ belong to the set of efficient points; if $W_i^* = E_i^*$, then the vertical edge at the line V with $W_i^* = V = E_i^*$ belongs to the set of efficient points; if $W_i^* > E_i^*$, then all boxes B for which $W_i^* > B > E_i^*$ belong to the set of efficient points. For the horizontal line H_i compute $u_i = \max\{NW_i, SW_{i-1}\}$ and $t_i = \min\{NE_i, SE_{i-1}\}$, $1 \leq i \leq p+1$. All horizontal edges lying between the vertical line $x = u_i$ and $x = t_i$ belong to the set of efficient points □

The following recursive relations can be used to compute NW_i, NE_i, SW_i and SE_i.

$$
\begin{aligned}
NW_1 &= W_1 \quad , & NE_1 &= E_1 \quad , \\
NW_i &= \min\{W_i, NW_{i-1}\} \quad , & NE_i &= \max\{E_i, NE_{i-1}\} \quad , \\
SW_i &= \min\{W_{i+1}, SW_{i+1}\} \quad , & SE_i &= \max\{E_{i+1}, SE_{i+1}\} , \\
SW_p &= W_{p+1} \quad , & SE_p &= E_{p+1} \quad ,
\end{aligned}
$$

The complexity of the algorithm is $O(n \log n)$.

CHAPTER 6

6. TOTALLY-BALANCED MATRICES AND CHORDAL GRAPHS

The covering problem on trees was introduced and solved in Chapter 2. It was formulated as

$$(6.1)\qquad \min \sum_{j=1}^{m} c_j x_j + \sum_{i=1}^{n} d_i z_i$$

$$\text{s.t. } \sum_{j=1}^{m} a_{ij} x_j + z_i \geq 1,\ i = 1,\ldots,n,$$

$$x_j \in \{0,1\},\ j = 1,\ldots,m,$$

$$z_i \in \{0,1\},\ k = 1,\ldots,n,$$

where the n×m (0,1)-matrix $A = (a_{ij})$ is in *standard greedy form*, i.e., it does not contain the 2×2 matrix (6.2) as a submatrix.

$$(6.2)\qquad \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

The algorithm of Chapter 2 for solving (6.1) did not utilize the fact that the matrix A was obtained by row and column permutations from an intersection matrix of neighborhood subtrees versus vertices of a tree. It relied only upon the information that the n×m matrix A was given in standard greedy form, and produced the optimal solution in O(mn) time. This immediately raises the following questions: what is the class of matrices that can be permuted into standard greedy form? Is there an efficient procedure to recognize whether a (0,1)-matrix is a member of this class?

To this end define a (0,1)-matrix to be *totally-balanced* if it does not contain a square submatrix with no identical columns and its row and columns sums equal to two. Figure 6.3 shows some forbidden submatrices

for a totally-balanced matrix.

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Figure 6.3. Examples of forbidden submatrices.

Since any of the forbidden submatrices contains the matrix given by (6.2) as a submatrix it follows that each matrix in standard greedy form is totally balanced. We will prove that conversely each totally-balanced matrix can be transformed into standard greedy form. This will characterize totally-balanced matrices as the class of matrices which can be transformed into standard greedy form. (Note that the totally-balancedness property is not affected by row or column permutations.)

We will present two different proofs of the result that totally-balanced matrices can be transformed into standard greedy form. The first proof is an algorithmic proof which will also provide us with an efficient procedure to recognize totally-balanced matrix. The second proof is a theoretical proof using properties of chordal graphs which are of independent interest by themselves.

The algorithmic proof is based upon the concept of a *lexical matrix* which was introduced in HOFFMAN, KOLEN and SAKAROVITCH (1984).

Let us call a (0,1)-matrix *lexical* if the following two properties hold.

(6.4). If rows i(1), i(2) (i(1) < i(2)) are different, then the last column in which they differ has a zero in row i(1) and a one in row i(2),

(6.5) If columns j(1), j(2) (j(1) < j(2)) are different, then the last row in which they differ has a zero in column j(1) and a one in column j(2).

We will present an algorithm which transforms any (0,1)-matrix into a lexical matrix by a permutation of the rows and a permutation of the columns. The next theorem states that a lexical totally-balanced matrix is in standard greedy form so that we can use the algorithm to transform a totally-balanced matrix into standard greedy form.

THEOREM 6.6. *If a totally-balanced matrix* $A = (a_{ij})$ *is lexical, then it is in standard greedy form.*

PROOF. Suppose A is not in standard greedy form. Then there exist rows i(1), i(2) (i(1) < i(2)) and columns j(1), j(2) (j(1) < j(2)) such that $a_{i(1),j(1)} = a_{i(1),j(2)} = a_{i(2),j(1)} = 1$ and $a_{i(2),j(2)} = 0$ (see Figure 6.7). Let i(3) be the last row in which column j(1) and j(2) differ, and let j(3) be the last column in which row, i(1) and i(2) differ. Since A is lexical we have $a_{i(1),j(3)} = 0$, $a_{i(2),j(3)} = 1$ and $a_{i(3),j(1)} = 0$, $a_{i(3),j(2)} = 1$. Since A does not contain a 3×3 submatrix with row and column sums equal to two we know that $a_{i(3),j(3)} = 0$.

In general we have the submatrix of A given by Figure 6.7 with ones on the lower and upper diagonal and the first element of the diagonal and zeros everywhere else. The rows and columns have the following property. (1) i(p) is the last row in which columns j(p-2) and j(p-1) differ (3≤p≤k). (2) j(p) is the last column in which rows i(p-2) and i(p-1) differ (3≤p≤k). We shall prove that we can extend this k×k submatrix to a (k+1)x(k+1) submatrix with the same properties. So we can extend this submatrix infinitely. This leads to a contradiction and therefore A must be in standard greedy form.

Let i(k+1) be the last row in which j(k-1) and j(k) differ, and let j(k+1) be the last column in which i(k-1) and i(k) differ. Since A is lexical we have $a_{i(k+1),j(k-1)} = 0$, $a_{i(k+1),j(k)} = 1$ and $a_{i(k-1),j(k+1)} = 0$, $a_{i(k),j(k+1)} = 1$. By definition of i(p) and j(p) (3≤p≤k) we know that $a_{i(k+1),j(p-2)} = a_{i(k+1),j(p-1)}$ and $a_{i(p-2),j(k+1)} = a_{i(p-1),j(k+1)}$. Using this for p = k,...,3 respectively we get $a_{i(k+1),j(q)} = a_{i(q),j(k+1)} = 0$ for q = 1,2,...,k-1. Since A does not contain a (k+1)×(k+1) submatrix with row and column sums equal to two we have $a_{i(k+1),j(k+1)} = 0$. □

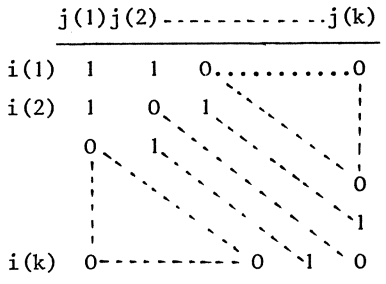

Figure 6.7. Submatrix of Theorem 6.6.

Let us now describe the algorithm which transform any (0,1)-matrix into a lexical matrix. Let $A = (a_{ij})$ be an $n \times m$ totally-balanced matrix without zero rows and columns. Let us denote column j by E_j, i.e., $E_j = \{i \mid a_{ij} = 1\}$. We assume that the matrix A is given by its columns $E_1, E_2, \ldots, E_m$. The algorithm produces a 1-1 mapping $\sigma: \{1,2,\ldots,n\} \to \{1,2,\ldots,n\}$ corresponding to a transformation of the rows of A ($\sigma(i) = j$ indicates that row i becomes row j in the transformed matrix) and a 1-1 mapping $\tau: \{E_1,\ldots,E_m\} \to \{1,\ldots,m\}$ corresponding to a transformation of the columns of A ($\tau(E_i) = j$ indicates that column i becomes column j in the transformed matrix). We present the algorithm in an informal way and give an example to demonstrate it.

The algorithm consists of m iterations. At iteration i we determine the column E for which $\tau(E) = m-i+1$ $(1<i<m)$. At the beginning of each iteration the rows are partitioned into a number of groups, say $G_r,\ldots,G_1$. If $i < j$, then rows in G_i will precede rows in G_j in the transformed matrix. Rows j and k belong to the same group G at the beginning of iteration i if and only if for all columns E we have determined so far, i.e., all columns E for which $\tau(E) \geq m-i+2$, we cannot distinguish between rows j and k, i.e., $j \in E$ if and only if $k \in E$. At the beginning of iteration 1 all rows belong to the same group. Let $G_r,\ldots,G_1$ be the partitioning into groups at the beginning of iteration i $(1<i<m)$. For each column E not yet determined we calculate the vector d_E of length r, where $d_E(j) = |G_{r-j+1} \cap E|$ $(j = 1,2,\ldots,r)$. A column E for which d_E is a lexicographically largest vector is the column determined at iteration i and $\tau(E) = m-i+1$. After we have determined E we can distinguish between some rows in the same group G if $1 < |G \cap E| < G$. If this is the case we shall take rows in $G \setminus E$ to precede rows in $G \cap E$ in the transformed matrix. This can be expressed by adjusting the partitioning into groups in the following way. For $j = r, r-1,\ldots,1$ respectively we check if the intersection of G_j with E is not empty and not equal to G_j. If this is the case we increase the index of all groups with index greater than j by one and partition the group G_j into two groups called G_j and G_{j+1}, where $G_{j+1} = G_j \cap E$ and $G_j = G_j \setminus E$. The algorithm ends after m interations with a partitioning into groups, say $G_r,\ldots,G_1$. The permutation σ is defined by assigning for $i = 1,2,\ldots,r$ the values $\sum_{j=1}^{i-1} |G_j| + 1,\ldots,\sum_{j=1}^{i} |G_j|$ in an arbitrary way to the elements in group G_i. The number of computation we have to do at each iteration is $O(nm)$. Therefore the time complexity of this algorithm is $O(nm^2)$.

EXAMPLE 6 8. The 9×7 (0 1)-matrix A is given by its columns
$E_1 = \{1,2,3\}$, $E_2 = \{1,2,3,5\}$, $E_3 = \{4,5\}$, $E_4 = \{3,4,5,9\}$, $E_5 = \{5,8,9\}$
$E_6 = \{6,7,8,9\}$, $E_7 = \{6,7,8\}$.

Iteration 1: $G_1 = (1,2,3,4,5,6,7,8,9)$.

$d_{E_i} = (|E_i|)$, choose E_4, $\tau(E_4) = 7$.

Iteration 2: $G_2 = (3,4,5,9)$, $G_1 = (1,2,6,7,8)$.

E	E_1	E_2	E_3	E_5	E_6	E_7
d_E	(1,2)	(2,2)	(2,0)	(2,1)	(1,3)	(0,3)

Choose E_2, $\tau(E_2) = 6$.

Iteration 3: $G_4 = (3,5)$, $G_3 = (4,9)$, $G_2 = (1,2)$, $G_1 = (6,7,8)$.

E	E_1	E_3	E_5	E_6	E_7
d_E	(1,0,2,0)	(1,1,0,0)	(1,1,0,1)	(0,1,0,3)	(0,0,0,3)

Choose E_5, $\tau(E_5) = 5$.

Iteration 4: $G_7 = (5)$, $G_6 = (3)$, $G_5 = (9)$, $G_4 = (4)$, $G_3 = (1,2)$,
$G_2 = (8)$, $G_1 = (6,7)$.

E	d_E
E_1	(0,1,0,0,2,0,0)
E_3	(1,0,0,1,0,0,0)
E_6	(0,0,1,0,0,1,2)
E_7	(0,0,0,0,0,1,2)

Choose E_3, $\tau(E_3) = 4$.

From now on the groups do not change.

Therefore $\tau(E_1) = 3$, $\tau(E_6) = 2$, $\tau(E_7) = 1$. A mapping σ is given by σ: $(6,7,8,1,2,4,9,3,5) \rightarrow (1,2,3,4,5,6,7,8,9)$. The mapping τ is given by τ: $(E_7,E_6,E_1,E_3,E_5,E_2,E_4) \rightarrow (1,2,3,4,5,6,7)$. □

Let us now prove that a matrix transformed by the algorithm is a lexical matrix. When we say that row i is the largest row with respect to σ satisfying a property we mean that there is no row j with $\sigma(j) > \sigma(i)$ satisfying the same property. The same terminology is also used for columns with respect to τ.

LEMMA 6.9. *If rows* i *and* j $\sigma(i) < \sigma(j)$) *are different, then for the largest column* E *with respect to* τ *in which they differ we have* $i \notin E$, $j \in E$.

PROOF. Consider the last iteration in which i and j are in the same group G and let E be the column determined at this iteration. Since i and j were in the same group during all previous iterations we know that rows i and j are identical when restricted to columns which are larger than E with respect to τ. Since $\sigma(i) < \sigma(j)$ we have that after this iteration row j is in a group with larger index than the group containing row i. This implies that $j \in G \cap E$ and $i \in G \backslash E$, i.e., $i \notin E$ and $j \in E$. □

LEMMA 6.10. *If columns* E_k and E_ℓ ($\tau(E_k) < \tau(E_\ell)$) *are different, then for the largest row* i *with respect to* σ *in which they differ we have* $i \notin E_k$ *and* $i \in E_\ell$.

PROOF. If E_i is strictly contained in E_j for some i,j, then we always have $\tau(E_i) < \tau(E_j)$. If $E_k \subseteq E_\ell$, then the lemma holds. So we may assume that $E_k \not\subseteq E_\ell$ and $E_\ell \not\subseteq E_k$. Let i be the largest row with respect to σ in $E_\ell \backslash E_k$, and let j be the largest row with respect to σ in $E_k \backslash E_\ell$. We have to prove that $\sigma(i) > \sigma(j)$. Consider the iteration in which E_ℓ was determined. Let p be the largest index for which $G_p \cap E_k \neq G_p \cap E_\ell$. Since E_ℓ was determined before E_k we know that $|G_p \cap E_\ell| > |G_p \cap E_k|$. We conclude that $i \in G_p$. If $j \in G_f$ with $f < p$, then $\sigma(j) < \sigma(i)$. If $j \in G_p$, then after this iteration G_p is partitioned into two groups $G_p \cap E_\ell$ and $G_p \backslash E_\ell$ where $G_p \backslash E_\ell$ precedes $G_p \cap E_\ell$. Since $j \in G_p \backslash E_\ell$ and $i \in G_p \cap E$ we have $\sigma(j) < \sigma(i)$. □

It follows from Lemma 6.9 and Lemma 6.10 that the transformed matrix is lexical. This completes the constructive proof of the result that any totally-balanced matrix can be transformed into standard greedy form. The algorithm we gave produces a lexical matrix in standard greedy form.

This is important if we consider the following result. Let A be a n×m (0,1)-matrix. The row intersection matrix $B = (b_{ij})$ of A is a n×n (0,1)-matrix defined by $b_{ij} = 1$ if and only if these exists a column of A which covers both row i and j. It is an easy excercise to show that if A is a lexical matrix in standard greedy form, then the row intersection matrix is in standard greedy form. This is not true for any (0,1)-matrix A in standard greedy form as is shown by the following example.

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

Using this result we have proved the following theorem which was first proved by LUBIN (1982) by showing that the row intersection matrix of a totally-balanced matrix does not contain one of the forbidden submatrices.

THEOREM 6.11 (LUBIN 1982). *The row intersection matrix of a totally-balanced matrix is totally-balanced.* □

We proved in Chapter 2 that the intersection matrix of neighborhood subtrees versus vertices of a tree was totally-balanced. This result was first proved by GILES (1978). We can use Theorem 6.11 to prove that the intersection matrix of neighborhood subtrees versus neighborhood subtrees is totally-balanced. This generalization of Giles result was first obtained by TAMIR (1983).

In the last part of this chapter we discuss the relationship between totally-balanced matrices and chordal graphs. First of all let us consider the relationship with chordal bipartite graphs. A *chordal bipartite graph* is a bipartite graph for which every cycle of length strictly greater than four has a *chord*, i.e., an edge connecting two vertices which are not adjacent in the cycle. Chordal bipartite graphs were discussed by GOLUMBIC (1981) in relation with perfect Gaussian elimination for nonsymmetric matrices. Chordal bipartite graphs and totally-balanced matrices are equivalent in the following sense:

(6.12). Given a chordal bipartite graph $H = (\{1,2,\ldots,n\}, \{1,2,\ldots,m\},E)$ define the n×m (0,1)-matrix $A = (a_{ij})$ by $a_{ij} = 1$ if and only if $[i,j] \in E$. Then A is totally-balanced.

Given an n×m totally-balanced matrix $A = (a_{ij})$ define the bipartite graph $H = (\{1,2,\ldots,n\}, \{1,2,\ldots,m\},E)$ by $E = \{[i,j] \mid a_{ij} = 1\}$. Then H is a chordal bipartite graph.

An edge (i,j) of a bipartite graph is *bisimplicial* if the subgraph induced by all vertices adjacent to i and j is a complete bipartite graph. Let $M = (m_{ij})$ be a nonsingular nonsymmetric matrix. We can construct a bipartite graph from M equivalent to (6.12) where edges correspond to nonzero elements m_{ij}. If (i,j) is a simplicial edge in the bipartite graph, then using m_{ij} as a pivot in the matrix M to make m_{ij} to one and all other entries in the ith row and jth column equal to zero does not change any zero element into a nonzero element. This is important since sparse matrices are represented in computers by its nonzero elements. GOLUMBIC (1981) proved that a chordal bipartite graph has a bisimplicial edge. This result immediately follows from our result. The first one in the first row corresponds to a bisimplicial edge.

In the second part of this chapter we give a theoretical proof of the fact that a totally-balanced matrix can be transformed into standard greedy form.

A row i of a (0,1)-matrix is called a *nest row* if all columns covering this row can be totally ordered by inclusion, i.e., if $a_{ij} = a_{ik} = 1$, then $E_j \subseteq E_k$ or $E_k \subseteq E_j$. We present a proof that a totally-balanced matrix has a nest row. We take this row to be the first row of the transformed matrix. After deleting this row from the matrix we still have a totally-balanced matrix. The resulting matrix has a nest row which we take to be the second row of the transformed matrix. Repeating this procedure we find a transformed matrix which has the *nestordering property* for columns, i.e. for all i ($1 \le i \le n$) the following holds: all columns having a one in row i can be totally ordered by inclusion when restricted to rows $\{i,i+1,\ldots,n\}$. It was proved in Lemma 2.23 that if we order the columns in a lexical nondecreasing order, then the matrix is in standard greedy form.

So in order to prove that a totally-balanced matrix can be transformed into standard greedy form it suffices to prove the existence of a nest row. The proof we will present differs from the original proof given by Brouwer and Kolen (1980) which used induction on the sum of the number of rows and the number of columns. We will use properties of chordal graphs. As defined before a *chordal graph* is a graph with the property that any cycle of length at least four has a *chord*, i.e., an edge connecting two vertices which are not adjacent in the cycle. The link between totally-balanced matrices and

chordal graphs is due to the row intersection graph of the matrix. The *row intersection graph* $G = (\{1,2,\ldots,n\},E)$ of a $n\times m$ (0,1)-matrix $A = (a_{ij})$ is defined by $[i,j] \in E$ iff $a_{ik} = a_{jk} = 1$ for some $k(1\leq k\leq m)$. It is a trivial observation that the row intersection graph of a totally-balanced matrix is a chordal graph (any cycle in the intersection graph without a chord would correspond to a square matrix with row and column sums equal to two; rows corresponding to vertices in the cycle and columns corresponding the columns which defined the edges of the cycle). Chordal graphs are sometimes called *triangulated graphs* or *rigid circuit graphs*. A *clique* is a subset of pairwise adjacent vertices. A *simplical vertex* is a vertex with the property that all vertices adjacent to it form a clique. If b and c are two nonadjacent vertices of a graph, then a subset of vertices, with the property that after its removal from the graph b and c are in distinct component, is called a b-c *separator*. If no proper subset of the separator is a separator, then it is called a *minimal separator*. Let us denote for a graph $G = (V,E)$ by $G(S)$ the graph induced by the subset S of vertices and all edges in E connecting vertices in S.

LEMMA 6.13. *If* $G = (V,E)$ *is a chordal graph, then every minimal vertex separator induces a clique.*

PROOF. Suppose S is a minimal b-c separator. Let B and C be subsets of V inducing the components of $G(V\setminus S)$ containing b and c, respectively. Since S is minimal each $x \in S$ is adjacent to some vertex in B and to some vertex in C. Therefore for any pair $x,y \in S$ there exist paths $[x,b_1,\ldots,b_r,y]$ and $[y,c_1,\ldots,c_t,x]$ where $b_i \in B$, $c_i \in C$ such that these paths are of minimal length. If follows that the cycle $[x,b_1,\ldots,b_r,y,c_1,\ldots,c_t,x]$ must have a chord. But $[b_i,c_j] \notin E$ since b_i and c_j are in distinct components, $[b_i,b_j] \notin E$ and $[c_i,c_j] \notin E$ by the minimality of r and t. Hence the only possible chord is $[x,y] \in E$. We conclude that S is a clique. □

LEMMA 6.14. *If* $G = (V,E)$ *is a chordal graph and* S *a minimal separator, then in each component of* $G(V\setminus S)$ *there is a vertex which together with* S *induces a clique.*

PROOF. Let B be a component of $G(V\setminus S)$. Let $X \subseteq S$ be the largest subset of S for which the result holds (note $|X| \geq 1$) and assume $X \neq S$. Let $y \in S \setminus X$ and let Z be the subset of vertices with the property that they are adjacent to all vertices in X. By definition of X we know that $[y,z] \notin E$ for all $z \in Z$.

For each $z \in Z$ there is a path $[y,b_1,\ldots,b_r z]$ with $b_i \in B$. Let $z \in Z$ be the vertex for which this path has minimal length, consider the minimal y-z separator T. Besided X the set T must contain a vertex b_i $(1 \leq i \leq r)$. Since T is a clique (Lemma 6.13) b_i is adjacent to all vertices in X. This leads to a contradiction since $b_i \in Z$ and b_i is closer to y than z. We conclude that X = S. □

The next theorem due to Dirac (1961) plays a major role in the theory on chordal graphs.

THEOREM 6.15. *Every chordal graph* G = (V,E) *has a simplical vertex. Moreover, if* G *is not a clique, then it has two nonadjacent simplical vertices.*

PROOF. If G is a clique, then the theorem is trivial. Assume G has two nonadjacent vertices b and c and assume the result holds for graphs with fewer vertices than G. Let S be a minimal b-c separator. Let B,C be the components of G(V\S) containing b,c respectively. By induction, either the graph G(BUS) has two nonadjacent simplical vertices one of which must be in B (since S induces a clique), or G(BUS) is a clique and any vertex in B is a simplical vertex of G(BUS). Furthermore, since the set of vertices connecting to a vertex in B is contained in BUS, a simplical vertex of G(BUS) is a simplical vertex of G. Similarly C contains a simplical vertex. □

A *perfect scheme* of a graph G = (V,E) is a mapping $\sigma: V \to \{1,\ldots,|V|\}$ such that for all vertices $v \in V$ the set of vertices $\{w \mid [v,w] \in E, \sigma(v) < \sigma(w)\}$ induces a clique. Since an induced subgraph of a chordal graph is a chordal graph, it follows from Theorem 6.15 that a perfect scheme exists for a chordal graph. A perfect scheme can be obtained by defining $\sigma(v_i) = i$, where v_i is a simplical vertex of the subgraph of the chordal graph G = (V,E) induced by $V\setminus\{v_j \mid j = 1,\ldots,i-1\}$, $i=1,\ldots,|v|$.

In Chapter 1 we showed the existence of a perfect scheme for the intersection graph of subtrees of a tree. This intersection graph was shown to be a chordal graph. The converse is also true. Every chordal graph is the intersection graph of subtrees of a tree. This was first proved by Walter (1972, 1978), Buneman (1974) and Gavril (1974). A very nice proof of this result is due to Farber (1981). He constructs the tree and subtrees directly from the chordal graph. Assume the vertices are numbered according to the perfect scheme, say 1,..., n, i.e. for all i the vertices j(j>i) adjacent to i form a clique. Farber constructs the tree $T = (\{1,\ldots,n\},E^*)$ and

subtrees T_i, i = 1,...,n from the chordal graph G = ({1,...,n},E) as follows.

(6.16) $[i,j] \in E^*$ iff i>j and $i = \min\{k \mid k>j, [j,k] \in E\}$, and for i = 1,...,n, T_i is the subgraph of T induced by all vertices k < i which are adjacent to i in G, and vertex i itself.

EXAMPLE 6.17. Consider the chordal graph given by Figure 6.18.

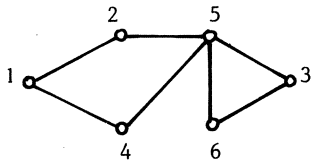

Figure 6.18. Chordal graph of Example 6.17

The tree T is given by Figure 6.18.

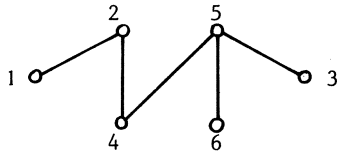

Figure 6.18. Tree of Example 6.17

The subtrees are given by their vertex set. We have $T_1 = \{1\}$, $T_2 = \{1,2\}$, $T_3 = \{3\}$, $T_4 = \{1,2,4\}$, $T_5 = \{2,3,4,5\}$, $T_6 = \{3,5,6\}$. □

After this short intermetzo on chordal graphs let us return to totally-balanced matrices and their row intersection graphs.

LEMMA 6.19. *For every clique in the row intersection graph of a totally-balanced matrix* A *there exists a column covering all rows in the clique.*

PROOF. The proof procees by induction on the size of the clique. In case of only two rows in the clique the result holds by definition of the row intersection graph. Assume the result holds for all cliques of size at most k-1 (k≥3). Let $C = \{i_1,i_2,\ldots,i_k\}$ be a clique of size k. By the induction hypothesis there is a column E_j covering all rows i_r, r = 1,2,...,k, r ≠ j, for j = 1,2,...,k. If $i_j \in E_j$, then E_j covers all rows in C. So we may assume that $i_j \notin E_j$ for all j = 1,2,...,k. But then the 3×3 submatrix A defined by rows i_1,i_2,i_3 and columns E_1,E_2 and E_3 has row and column sums equal to

two, which leads to a contradiction. We conclude that there is a column E_j ($1 \leq j \leq k$) covering all rows in C. □

The next theorem is a generalization of the theorem of Dirac (Theorem 6.15).

THEOREM 6.20. *Each totally-balanced matrix with more than one row has at least two nest rows. Moreover if the row intersection graph is not complete it has two nest rows which are nonadjacent in this graph.*

PROOF. The proof proceeds by induction on the number of rows of the totally-balanced matrix. In case of two rows the theorem is trivial. Assume the result holds for all totally-balanced matrices with less than n rows and let A be a totally-balanced matrix with n rows ($n \geq 3$). Let G be the intersection graph of A. If G is complete, then it follows from Lemma 6.19 that there is a column which covers all rows. We delete this column from the matrix. Nest rows of the remaining matrix are also nest rows of A. Therefore assume without loss of generality that G is not a clique. Let b,c be two nonadjacent vertices and let S be a minimal vertex separator of b and c with B and C the components of $G(\{1,\ldots,n\}\backslash S)$ containing b and c respectively.

Consider the submatrix A(BUS) of A defined by the rows of BUS and all columns covering at least one of those rows. According to Lemma 6.14 there is a vertex in G which together with S induces a clique. Let E be a column which covers the row corresponding to this vertex and the rows corresponding to S. (According to Lemma 6.19 such a column exists). Since S is a minimal vertex separator no vertex in C is adjacent to a vertex in B. Hence column E does not cover a row belonging to B. Consider the row intersection graph of A(BUS). If this graph is not complete, then A(BUS) contains two nonadjacent nest rows. At least one of these rows belongs to B(S is a clique since column E covers all rows in S). Since B is a component of $G(\{1,\ldots,n\}\backslash S)$ any column covering a row in B does not cover a row outside B U S. Therefore any nest row of A(BUS) is also a nest row of A. If the row intersection graph of A(BUS) is a clique, then we delete all columns which cover the rows belonging to B U S(Lemma 6.19). The remaining matrix has a nest row belonging to B. This nest row is also a nest row of A(BUS) and of A. Similarly one proves that there exists a nest row in C. □

This completes the theoretical proof of the fact that totally-balanced matrices form the class of matrices which can be transformed into standard greedy form.

We proved that a graph is chordal iff it is the intersection graph of subtrees of a tree. We also proved that the intersection graph of neighborhood subtrees versus neighborhood subtrees of a tree is a totally-balanced matrix. The converse of this result is not true. It was proved by Broin and Lowe (1984) that the matrix given by (6.21) is not the intersection matrix of neighborhood subtrees versus neighborhood subtrees of a tree.

(6.21)
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Let us finish this chapter by considering an extension of problem (6.1)

(6.22)
$$\min \sum_{j=1}^{m} c_j x_j + \sum_{i=1}^{n} d_i z_i$$
$$\text{s.t.} \sum_{j=1}^{m} a_{ij} x_j + z_i \geq 1, \quad i = 1,\ldots,n,$$
$$\sum_{j=1}^{m} x_j \leq p,$$
$$x_j \in \{0,1\}, \quad j = 1,\ldots,m,$$
$$z_i \in \{0,1\}, \quad i = 1,\ldots,n,$$

where $A = (a_{ij})$ is in standard greedy form.

Application of this problem are discussed in Kolen en Tamir (1985), they include the p-median problem on the tree (i.e., we locate p facilities on the tree so as to minimize the sum of the linear transportation cost, linear with respect to the distance between a client and the closest facility.)

In contrast to problem (6.1) the LP-relaxation of (6.22) does not always have a (0,1)-optimal solution. However there still is an $O(n^2m^2)$ algorithm to solve this problem. This algorithm by Broin and Lowe (1984) is based on dynamic programming.

The *neighborhood matrix* of a graph $G = (\{v_1,\ldots,v_n\},E$ is the $n \times n$ (0,1)-matrix $A = (a_{ij})$ defined by $a_{ij} = 1$ iff $i = j$ or $[v_i,v_j] \in E$. A graph for which the neighborhood matrix is totally-balanced is called a *strongly*

chordal graph. For further information on chordal graphs, strongly chordal graphs and totally-balanced matrices we refer to Anstee and Farber (1982), Farber (1982), 1983), Chang (1982), Lubin (1982) and Golumbic (1980).

REFERENCES

AHO, A.V., J.E. HOPCROFT & J.D. ULLMAN (1974), *The Design and Analysis of Computer Algorithms* (Addison-Wesley, Reading, MA).

BROIN, M.W. & T.J. LOWE (1984), *A dynamic programming algorithm for covering problems with (greedy) totally-balanced constraint matrices*, SIAM J. Algebraic and Discrete Methods (to appear).

BROUWER, A.E. & A. KOLEN (1980), *A super-balanced hypergraph has a nest-point*. Report ZW 146, Mathematisch Centrum, Amsterdam.

BUNEMAN, P. (1974), *A characterization of rigid circuit graphs*. Discrete Math. 9, 205-212.

CHALMET, L., R.L. FRANCIS & A. KOLEN (1981), *Finding efficient solutions for rectilinear distance location problems efficiently*. European J. Oper.Res. 6, 117-124.

CHAN, A. & R.L. FRANCIS (1976), *A round-trip location problem on a tree graph*. Transportation Sci. 10, 35-51.

CHAN, A. & D.W. HEARN (1977), *A rectilinear distance minimax round-trip location problem*. Transportation Sci. 11, 107-123.

CHANG, G.J. (1982), K-*domination and graph covering problems*. Ph.D. dissertation, Cornell University, Ithaca, N.Y.

DEARING, P.M., R.L. FRANCIS & T.J. LOWE (1976), *Convex location problems on tree networks*. Oper. Res. 24, 628-642.

DIRAC, G.A. (1961), *On rigid circuit graphs*, Abh. Math. Sem. Univ. Hamburg 25, 71-76.

FARBER, M. (1981), *Applications of linear programming duality to problems involving independence and domination*. Ph. D. dissertation, Rutgers University TR 81-13.

FARKAS, J. (1902), *Ueber die Theorie der einfachen Ungleichungen*. J. rein. angew. Math. 124, 1-27.

FOURIER, J.B.J. (1826), *Solution d'une question particulière du calcul des inéqualités*. Oeuvres II, 317-328.

FRANCIS, R.L., T.J. LOWE & H.D. RATLIFF (1978), *Distance constraints for tree network multifacility location problems*. Oper. Res. 26, 570-596.

FREDERICKSON, G.N. & D.B. JOHNSON (1983), *Finding k-th paths and p-centers by generating and searching good data structures*. J. Algorithms 4, 61-80.

GAREY, M.R. & D.S. JOHNSON (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Franciso

GARFINKEL, R.S. & G.L. NEMHAUSER (1972), *Integer Programming*. Wiley, New York.

GAVRIL, F. (1974), *The intersection graphs of subtrees are exactly the chordal graphs*. J. Combin. Theory Ser. B 16, 47-56.

GILES, R. (1978), *A balanced hypergraph defined by subtrees of a tree*. Ars Combin. 6, 179-183.

GOLUMBIC, M.C. (1980), *Algorithmic Graph Theory and Perfect Graphs* (Academic Press, New York).

HAKIMI, S.L. & O. KARIV (1979), *An algorithmic approach to network location problems; Part I: The p-centers*. SIAM J. Appl. Math. 37, 539-560.

HOFFMAN, A.J., A. KOLEN & M. SAKAROVITCH (1984), *Totally balanced and greedy matrices*, SIAM J. Algebraic and Discrete Methods (to appear).

JUEL, H. & R.F. LOVE (1976), *An efficient computational procedure for solving the multifacility rectilinear facilities location problem*. Oper. Res. Quart. 27, 697-703.

KARZANOV, A.V. (1974), *Determining the maximal flow in a network by the method of preflows*. Soviet Math. Dokl 15, 434-437.

KOLEN, A. (1981), *Equivalence between the direct search approach and the cut approach to the rectilinear distance location problem*. Oper. Res. 29, 616-620.

KOLEN, A.W.J. (1983), *Solving covering problems and the uncapacitated plant location problem on trees*. European J. of Operational Research 12, 266-278.

KOLEN, A.W.J. & A. TAMIR (1985), *Covering problems*, Chapter 6 in *Discrete Location Theory*, edited by Francis and Mirchandani; Wiley, New York.

KUHN, H.W. (1956), *Solvability and consistency for linear equations and inequalities*. Amer. Math. Monthly 63, 217-232.

LAWLER, E.L. & J.K. LENSTRA (1982), *Machine scheduling with precedence constraints*. I.Rival (ed.). *Ordered Sets*, Reidel.

LUBIW, A. (1982), *Γ-free matrices*. Master Thesis, Faculty of Mathematics, University of Waterloo, Ontario, Canada.

MEGIDDO, A. & A. TAMIR (1983), *New results on p-center problems*. SIAM J. Computing 12, 751-758.

MOTZKIN, T.S. (1936), *Beiträge zur Theorie der linearen Ungleichungen*. Inaugural Dissertation Basel, Azriel, Jerusalem.

PICARD, J.C. & H.D. RATLIFF (1978), *A cut approach to the rectilinear distance location problem*. Oper. Res. 26, 422-434.

PRITSKER, A.A.B. & P.M. GHARE (1970), *Locating new facilities with respect to existing facilities*. AIIE Trans. 2, 290-297.

RAO, M.R. (1973), *On the direct search approach to the recilinear facilities location problem*. AIIE Trans. 5, 256-264.

SHERALI, H.D. & C.M SHETTY (1978), *A primal simplex based solution procedure for the rectilinear distance multifacility location problem*. J. Oper. Res. Soc. 29, 373-381.

TAMIR, A. (1983), *A class of balanced matrices arising from location problems*. SIAM J. Algebraic and Discrete Methods 4, 363-370.

STOER, J. & C. WITZGALL (1970), *Convexity and Optimization in Finite Dimensions* I (Springer, Berlin).

TANSEL, B.C., R.L. FRANCIS, T.J. LOWE & M.L. CHEN (1982), *Duality and distance constraints for the non-linear p-center problem and covering problem on a tree network*. Operations Research 30, 725-744.

WALTER, J.R. (1972), *Representations of rigid cycle graphs*. Ph.D. dissertation, Wayne State University, Detroit, Michigan.

WALTER, J.R. (1978), *Representations of chordal graphs as subtrees of a tree*. J. Graph Theory 2, 265-267.

WENDEL, R.E. A.P. HURTER & T.J. LOWE (1977), *Efficient points in location problems*. AIIE Trans. 9, 238-246.

MATHEMATICAL CENTRE TRACTS

1 T. van der Walt. *Fixed and almost fixed points.* 1963.

2 A.R. Bloemena. *Sampling from a graph.* 1964.

3 G. de Leve. *Generalized Markovian decision processes, part I: model and method.* 1964.

4 G. de Leve. *Generalized Markovian decision processes, part II: probabilistic background.* 1964.

5 G. de Leve, H.C. Tijms, P.J. Weeda. *Generalized Markovian decision processes, applications.* 1970.

6 M.A. Maurice. *Compact ordered spaces.* 1964.

7 W.R. van Zwet. *Convex transformations of random variables.* 1964.

8 J.A. Zonneveld. *Automatic numerical integration.* 1964.

9 P.C. Baayen. *Universal morphisms.* 1964.

10 E.M. de Jager. *Applications of distributions in mathematical physics.* 1964.

11 A.B. Paalman-de Miranda. *Topological semigroups.* 1964.

12 J.A.Th.M. van Berckel, H. Brandt Corstius, R.J. Mokken, A. van Wijngaarden. *Formal properties of newspaper Dutch.* 1965.

13 H.A. Lauwerier. *Asymptotic expansions.* 1966, out of print; replaced by MCT 54.

14 H.A. Lauwerier. *Calculus of variations in mathematical physics.* 1966.

15 R. Doornbos. *Slippage tests.* 1966.

16 J.W. de Bakker. *Formal definition of programming languages with an application to the definition of ALGOL 60.* 1967.

17 R.P. van de Riet. *Formula manipulation in ALGOL 60, part 1.* 1968.

18 R.P. van de Riet. *Formula manipulation in ALGOL 60, part 2.* 1968.

19 J. van der Slot. *Some properties related to compactness.* 1968.

20 P.J. van der Houwen. *Finite difference methods for solving partial differential equations.* 1968.

21 E. Wattel. *The compactness operator in set theory and topology.* 1968.

22 T.J. Dekker. *ALGOL 60 procedures in numerical algebra, part 1.* 1968.

23 T.J. Dekker, W. Hoffmann. *ALGOL 60 procedures in numerical algebra, part 2.* 1968.

24 J.W. de Bakker. *Recursive procedures.* 1971.

25 E.R. Paërl. *Representations of the Lorentz group and projective geometry.* 1969.

26 European Meeting 1968. *Selected statistical papers, part I.* 1968.

27 European Meeting 1968. *Selected statistical papers, part II.* 1968.

28 J. Oosterhoff. *Combination of one-sided statistical tests.* 1969.

29 J. Verhoeff. *Error detecting decimal codes.* 1969.

30 H. Brandt Corstius. *Exercises in computational linguistics.* 1970.

31 W. Molenaar. *Approximations to the Poisson, binomial and hypergeometric distribution functions.* 1970.

32 L. de Haan. *On regular variation and its application to the weak convergence of sample extremes.* 1970.

33 F.W. Steutel. *Preservation of infinite divisibility under mixing and related topics.* 1970.

34 I. Juhász, A. Verbeek, N.S. Kroonenberg. *Cardinal functions in topology.* 1971.

35 M.H. van Emden. *An analysis of complexity.* 1971.

36 J. Grasman. *On the birth of boundary layers.* 1971.

37 J.W. de Bakker, G.A. Blaauw, A.J.W. Duijvestijn, E.W. Dijkstra, P.J. van der Houwen, G.A.M. Kamsteeg-Kemper, F.E.J. Kruseman Aretz, W.L. van der Poel, J.P. Schaap-Kruseman, M.V. Wilkes, G. Zoutendijk. *MC-25 Informatica Symposium.* 1971.

38 W.A. Verloren van Themaat. *Automatic analysis of Dutch compound words.* 1972.

39 H. Bavinck. *Jacobi series and approximation.* 1972.

40 H.C. Tijms. *Analysis of (s,S) inventory models.* 1972.

41 A. Verbeek. *Superextensions of topological spaces.* 1972.

42 W. Vervaat. *Success epochs in Bernoulli trials (with applications in number theory).* 1972.

43 F.H. Ruymgaart. *Asymptotic theory of rank tests for independence.* 1973.

44 H. Bart. *Meromorphic operator valued functions.* 1973.

45 A.A. Balkema. *Monotone transformations and limit laws.* 1973.

46 R.P. van de Riet. *ABC ALGOL, a portable language for formula manipulation systems, part 1: the language.* 1973.

47 R.P. van de Riet. *ABC ALGOL, a portable language for formula manipulation systems, part 2: the compiler.* 1973.

48 F.E.J. Kruseman Aretz, P.J.W. ten Hagen, H.L. Oudshoorn. *An ALGOL 60 compiler in ALGOL 60, text of the MC-compiler for the EL-X8.* 1973.

49 H. Kok. *Connected orderable spaces.* 1974.

50 A. van Wijngaarden, B.J. Mailloux, J.E.L. Peck, C.H.A. Koster, M. Sintzoff, C.H. Lindsey, L.G.L.T. Meertens, R.G. Fisker (eds.). *Revised report on the algorithmic language ALGOL 68.* 1976.

51 A. Hordijk. *Dynamic programming and Markov potential theory.* 1974.

52 P.C. Baayen (ed.). *Topological structures.* 1974.

53 M.J. Faber. *Metrizability in generalized ordered spaces.* 1974.

54 H.A. Lauwerier. *Asymptotic analysis, part 1.* 1974.

55 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 1: theory of designs, finite geometry and coding theory.* 1974.

56 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 2: graph theory, foundations, partitions and combinatorial geometry.* 1974.

57 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 3: combinatorial group theory.* 1974.

58 W. Albers. *Asymptotic expansions and the deficiency concept in statistics.* 1975.

59 J.L. Mijnheer. *Sample path properties of stable processes.* 1975.

60 F. Göbel. *Queueing models involving buffers.* 1975.

63 J.W. de Bakker (ed.). *Foundations of computer science.* 1975.

64 W.J. de Schipper. *Symmetric closed categories.* 1975.

65 J. de Vries. *Topological transformation groups, 1: a categorical approach.* 1975.

66 H.G.J. Pijls. *Logically convex algebras in spectral theory and eigenfunction expansions.* 1976.

68 P.P.N. de Groen. *Singularly perturbed differential operators of second order.* 1976.

69 J.K. Lenstra. *Sequencing by enumerative methods.* 1977.

70 W.P. de Roever, Jr. *Recursive program schemes: semantics and proof theory.* 1976.

71 J.A.E.E. van Nunen. *Contracting Markov decision processes.* 1976.

72 J.K.M. Jansen. *Simple periodic and non-periodic Lamé functions and their applications in the theory of conical waveguides.* 1977.

73 D.M.R. Leivant. *Absoluteness of intuitionistic logic.* 1979.

74 H.J.J. te Riele. *A theoretical and computational study of generalized aliquot sequences.* 1976.

75 A.E. Brouwer. *Treelike spaces and related connected topological spaces.* 1977.

76 M. Rem. *Associons and the closure statement.* 1976.

77 W.C.M. Kallenberg. *Asymptotic optimality of likelihood ratio tests in exponential families.* 1978.

78 E. de Jonge, A.C.M. van Rooij. *Introduction to Riesz spaces.* 1977.

79 M.C.A. van Zuijlen. *Emperical distributions and rank statistics.* 1977.

80 P.W. Hemker. *A numerical study of stiff two-point boundary problems.* 1977.

81 K.R. Apt, J.W. de Bakker (eds.). *Foundations of computer science II, part 1.* 1976.

82 K.R. Apt, J.W. de Bakker (eds.). *Foundations of computer science II, part 2.* 1976.

83 L.S. van Benthem Jutting. *Checking Landau's "Grundlagen" in the AUTOMATH system.* 1979.

84 H.L.L. Busard. *The translation of the elements of Euclid from the Arabic into Latin by Hermann of Carinthia (?), books vii-xii.* 1977.

85 J. van Mill. *Supercompactness and Wallman spaces.* 1977.

86 S.G. van der Meulen, M. Veldhorst. *Torrix I, a programming system for operations on vectors and matrices over arbitrary fields and of variable size.* 1978.

88 A. Schrijver. *Matroids and linking systems.* 1977.

89 J.W. de Roever. *Complex Fourier transformation and analytic functionals with unbounded carriers.* 1978.

90 L.P.J. Groenewegen. *Characterization of optimal strategies in dynamic games.* 1981.

91 J.M. Geysel. *Transcendence in fields of positive characteristic.* 1979.

92 P.J. Weeda. *Finite generalized Markov programming.* 1979.

93 H.C. Tijms, J. Wessels (eds.). *Markov decision theory.* 1977.

94 A. Bijlsma. *Simultaneous approximations in transcendental number theory.* 1978.

95 K.M. van Hee. *Bayesian control of Markov chains.* 1978.

96 P.M.B. Vitányi. *Lindenmayer systems: structure, languages, and growth functions.* 1980.

97 A. Federgruen. *Markovian control problems; functional equations and algorithms.* 1984.

98 R. Geel. *Singular perturbations of hyperbolic type.* 1978.

99 J.K. Lenstra, A.H.G. Rinnooy Kan, P. van Emde Boas (eds.). *Interfaces between computer science and operations research.* 1978.

100 P.C. Baayen, D. van Dulst, J. Oosterhoff (eds.). *Proceedings bicentennial congress of the Wiskundig Genootschap, part 1.* 1979.

101 P.C. Baayen, D. van Dulst, J. Oosterhoff (eds.). *Proceedings bicentennial congress of the Wiskundig Genootschap, part 2.* 1979.

102 D. van Dulst. *Reflexive and superreflexive Banach spaces.* 1978.

103 K. van Harn. *Classifying infinitely divisible distributions by functional equations.* 1978.

104 J.M. van Wouwe. *Go-spaces and generalizations of metrizability.* 1979.

105 R. Helmers. *Edgeworth expansions for linear combinations of order statistics.* 1982.

106 A. Schrijver (ed.). *Packing and covering in combinatorics.* 1979.

107 C. den Heijer. *The numerical solution of nonlinear operator equations by imbedding methods.* 1979.

108 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science III, part 1.* 1979.

109 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science III, part 2.* 1979.

110 J.C. van Vliet. *ALGOL 68 transput, part I: historical review and discussion of the implementation model.* 1979.

111 J.C. van Vliet. *ALGOL 68 transput, part II: an implementation model.* 1979.

112 H.C.P. Berbee. *Random walks with stationary increments and renewal theory.* 1979.

113 T.A.B. Snijders. *Asymptotic optimality theory for testing problems with restricted alternatives.* 1979.

114 A.J.E.M. Janssen. *Application of the Wigner distribution to harmonic analysis of generalized stochastic processes.* 1979.

115 P.C. Baayen, J. van Mill (eds.). *Topological structures II, part 1.* 1979.

116 P.C. Baayen, J. van Mill (eds.). *Topological structures II, part 2.* 1979.

117 P.J.M. Kallenberg. *Branching processes with continuous state space.* 1979.

118 P. Groeneboom. *Large deviations and asymptotic efficiencies.* 1980.

119 F.J. Peters. *Sparse matrices and substructures, with a novel implementation of finite element algorithms.* 1980.

120 W.P.M. de Ruyter. *On the asymptotic analysis of large-scale ocean circulation.* 1980.

121 W.H. Haemers. *Eigenvalue techniques in design and graph theory.* 1980.

122 J.C.P. Bus. *Numerical solution of systems of nonlinear equations.* 1980.

123 I. Yuhász. *Cardinal functions in topology - ten years later.* 1980.

124 R.D. Gill. *Censoring and stochastic integrals.* 1980.

125 R. Eising. *2-D systems, an algebraic approach.* 1980.

126 G. van der Hoek. *Reduction methods in nonlinear programming.* 1980.

127 J.W. Klop. *Combinatory reduction systems.* 1980.

128 A.J.J. Talman. *Variable dimension fixed point algorithms and triangulations.* 1980.

129 G. van der Laan. *Simplicial fixed point algorithms.* 1980.

130 P.J.W. ten Hagen, T. Hagen, P. Klint, H. Noot, H.J. Sint, A.H. Veen. *ILP: intermediate language for pictures.* 1980.

131 R.J.R. Back. *Correctness preserving program refinements: proof theory and applications.* 1980.

132 H.M. Mulder. *The interval function of a graph.* 1980.

133 C.A.J. Klaassen. *Statistical performance of location estimators.* 1981.

134 J.C. van Vliet, H. Wupper (eds.). *Proceedings international conference on ALGOL 68.* 1981.

135 J.A.G. Groenendijk, T.M.V. Janssen, M.J.B. Stokhof (eds.). *Formal methods in the study of language, part I.* 1981.

136 J.A.G. Groenendijk, T.M.V. Janssen, M.J.B. Stokhof (eds.). *Formal methods in the study of language, part II.* 1981.

137 J. Telgen. *Redundancy and linear programs.* 1981.

138 H.A. Lauwerier. *Mathematical models of epidemics.* 1981.

139 J. van der Wal. *Stochastic dynamic programming, successive approximations and nearly optimal strategies for Markov decision processes and Markov games.* 1981.

140 J.H. van Geldrop. *A mathematical theory of pure exchange economies without the no-critical-point hypothesis.* 1981.

141 G.E. Welters. *Abel-Jacobi isogenies for certain types of Fano threefolds.* 1981.

142 H.R. Bennett, D.J. Lutzer (eds.). *Topology and order structures, part 1.* 1981.

143 J.M. Schumacher. *Dynamic feedback in finite- and infinite-dimensional linear systems.* 1981.

144 P. Eijgenraam. *The solution of initial value problems using interval arithmetic; formulation and analysis of an algorithm.* 1981.

145 A.J. Brentjes. *Multi-dimensional continued fraction algorithms.* 1981.

146 C.V.M. van der Mee. *Semigroup and factorization methods in transport theory.* 1981.

147 H.H. Tigelaar. *Identification and informative sample size.* 1982.

148 L.C.M. Kallenberg. *Linear programming and finite Markovian control problems.* 1983.

149 C.B. Huijsmans, M.A. Kaashoek, W.A.J. Luxemburg, W.K. Vietsch (eds.). *From A to Z, proceedings of a symposium in honour of A.C. Zaanen.* 1982.

150 M. Veldhorst. *An analysis of sparse matrix storage schemes.* 1982.

151 R.J.M.M. Does. *Higher order asymptotics for simple linear rank statistics.* 1982.

152 G.F. van der Hoeven. *Projections of lawless sequences.* 1982.

153 J.P.C. Blanc. *Application of the theory of boundary value problems in the analysis of a queueing model with paired services.* 1982.

154 H.W. Lenstra, Jr., R. Tijdeman (eds.). *Computational methods in number theory, part I.* 1982.

155 H.W. Lenstra, Jr., R. Tijdeman (eds.). *Computational methods in number theory, part II.* 1982.

156 P.M.G. Apers. *Query processing and data allocation in distributed database systems.* 1983.

157 H.A.W.M. Kneppers. *The covariant classification of two-dimensional smooth commutative formal groups over an algebraically closed field of positive characteristic.* 1983.

158 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science IV, distributed systems, part 1.* 1983.

159 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science IV, distributed systems, part 2.* 1983.

160 A. Rezus. *Abstract AUTOMATH.* 1983.

161 G.F. Helminck. *Eisenstein series on the metaplectic group, an algebraic approach.* 1983.

162 J.J. Dik. *Tests for preference.* 1983.

163 H. Schippers. *Multiple grid methods for equations of the second kind with applications in fluid mechanics.* 1983.

164 F.A. van der Duyn Schouten. *Markov decision processes with continuous time parameter.* 1983.

165 P.C.T. van der Hoeven. *On point processes.* 1983.

166 H.B.M. Jonkers. *Abstraction, specification and implementation techniques, with an application to garbage collection.* 1983.

167 W.H.M. Zijm. *Nonnegative matrices in dynamic programming.* 1983.

168 J.H. Evertse. *Upper bounds for the numbers of solutions of diophantine equations.* 1983.

169 H.R. Bennett, D.J. Lutzer (eds.). *Topology and order structures, part 2.* 1983.

CWI TRACTS

1 D.H.J. Epema. *Surfaces with canonical hyperplane sections.* 1984.

2 J.J. Dijkstra. *Fake topological Hilbert spaces and characterizations of dimension in terms of negligibility.* 1984.

3 A.J. van der Schaft. *System theoretic descriptions of physical systems.* 1984.

4 J. Koene. *Minimal cost flow in processing networks, a primal approach.* 1984.

5 B. Hoogenboom. *Intertwining functions on compact Lie groups.* 1984.

6 A.P.W. Böhm. *Dataflow computation.* 1984.

7 A. Blokhuis. *Few-distance sets.* 1984.

8 M.H. van Hoorn. *Algorithms and approximations for queueing systems.* 1984.

9 C.P.J. Koymans. *Models of the lambda calculus.* 1984.

10 C.G. van der Laan, N.M. Temme. *Calculation of special functions: the gamma function, the exponential integrals and error-like functions.* 1984.

11 N.M. van Dijk. *Controlled Markov processes; time-discretization.* 1984.

12 W.H. Hundsdorfer. *The numerical solution of nonlinear stiff initial value problems: an analysis of one step methods.* 1985.

13 D. Grune. *On the design of ALEPH.* 1985.

14 J.G.F. Thiemann. *Analytic spaces and dynamic programming: a measure theoretic approach.* 1985.

15 F.J. van der Linden. *Euclidean rings with two infinite primes.* 1985.

16 R.J.P. Groothuizen. *Mixed elliptic-hyperbolic partial differential operators: a case-study in Fourier integral operators.* 1985.

17 H.M.M. ten Eikelder. *Symmetries for dynamical and Hamiltonian systems.* 1985.

18 A.D.M. Kester. *Some large deviation results in statistics.* 1985.

19 T.M.V. Janssen. *Foundations and applications of Montague grammer, part 1: Philosophy, framework, computer science.* 1986.

20 B.F. Schriever. *Order dependence.* 1986.

21 D.P. van der Vecht. *Inequalities for stopped Brownian motion.* 1986.

22 J.C.S.P. van der Woude. *Topological dynamix.* 1986.

23 A.F. Monna. *Methods, concepts and ideas in mathematics: aspects of an evolution.* 1986.

24 J.C.M. Baeten. *Filters and ultrafilters over definable subsets of admissible ordinals.* 1986.

25 A.W.J. Kolen. *Tree network and planar rectilinear location theory.* 1986.