Network Location Problems with Multiple Types of Facilities

by

Rong Bing Huang

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Joseph L. Rotman School of Management
University of Toronto

Network Location Problems with Multiple Types of Facilities

Rong Bing Huang, Ph.D. 2005

Joseph L. Rotman School of Management

University of Toronto

Abstract

This dissertation focuses mainly on facility location problems on networks with multiple types of facilities and multiple types of customers. There has been limited research on these problems in the literature.

Chapter 2 focuses on the minisum Collection Depots Location problem. In this problem, a server has to visit the node requesting service as well as one of several collection depots. We prove that there exists a dominating location set for the problem on a general network. The properties of the solution on some simple network topologies are discussed. To solve the problem on a general network, we suggest a Lagrangian Relaxation embedded in a branch-and-bound algorithm.

In Chapter 3, we discuss the minisum multi-purpose trip location problem with two types of facilities and three types of customers. For example, in the setting of simultaneously locating apparel and hardware stores, we can classify customers into three groups. The first

type of customers only need to buy clothes, the second type of customers only need to buy hardware, and the third type of customers need to buy both apparel and hardware items. Therefore, the first two types of customers only need one type of service, while the third one needs both types of services in a single trip. The objective is to minimize the total weighted travel distance of all trips. For this minisum problem, we prove that there exists a dominating location set on a general network. The properties of optimal solutions on networks with simple topologies are also analyzed.

In Chapter 4, we study the problem having the same setting as above, but with the objective of maximizing coverage. First, we formulate the problem as a linear integer program. Then, we propose a method which often produces a tighter bound than the LP relaxation of the linear integer program. For the simplified problem on a path, we solve the problem in polynomial time through applying a dynamic programming algorithm.

In Chapter 5, we consider a location problem with single-type facilities. The objective is to locate undesirable facilities on a network so as to minimize the total demand covered subject to the requirement that no two facilities are allowed to be closer than a pre-specified distance. We prove that there exists a dominating location set and that it is a challenging problem to determine the consistency of the distance constraints. We compare several different mathematical formulations to solve the problem.

Acknowledgements

I would like to express my gratitude to the members of my doctoral committee, Professors Oded Berman, Dmitry Krass, Joseph Milner at the Joseph L. Rotman School of Management, University of Toronto, and Professor Vedat Verter from McGill University for their guidance and support during my research and elaboration of this dissertation.

I would like to express my profound and sincere gratitude and admiration to Professor Oded Berman for his invaluable support, patience, and encouragement throughout my academic and research journey. Thank you for sharing your wisdom, knowledge, and experience with me as a researcher, a teacher, and an advisor.

I have to thank my wife Mei Tian for all her support and patience during the long duration of this research. I am very grateful to my father-in-law Zuoling Tian for his help during the most difficult time in this process. I also have to say thanks to my son Andrew. He is the biggest motivation that keeps me going.

Finally, I would like to thank two of my best friends graduated from Rotman School of Management: Seokjin Kim and Mozart Menezes for sharing their academic and nonacademic experience with me.

Contents

List of Figures

Chapter 1

Introduction

1.1 Facility location problems

Facility location problems have occupied an important place in operations research since the early 1960's. They investigate where to physically locate a set of facilities so as to optimize a given function subject to a set of constraints.

Facility location models are used in a wide variety of applications. Examples include locating warehouses within a supply chain to minimize the average travel time to the markets, locating hazardous material sites to minimize exposure to the public, locating railroad stations to minimize the variability of delivery schedules, locating automatic teller machines to best serve the bank's customers, and locating a coastal search and rescue station to minimize the maximum response time to maritime accidents (Hale and Moberg, 2003).

There are different types of facility location problems. Some basic classes of facility location problems are listed below (Berman and Krass, 2002).

- *Discrete facility location problem:* location problem where the sets of demand points and potential facility locations are finite.
- *Continuous facility location problem:* location problem in a general space endowed with some metric, e.g., l_p norm. Facilities can be located anywhere in the given space.
- *Network facility location problem:* location problem which is confined to the links and nodes of an underlying network.
- *Stochastic facility location problem:* location problem where some parameters, e.g., demand or travel time, are uncertain.

We can furthermore classify a model as *capacitated* as opposed to *uncapacitated* where the former term refers to the upper bound on the number of clients (or demand) that a facility can serve. Models are called *dynamic* (as opposed to *static*) if the time element is explicitly represented (Wesolowsky, 1973).

The problems on which we focus our attention in this dissertation can be characterized as discrete, deterministic, uncapacitated, and static network models. Current *et al.* (2002) listed several basic discrete network location models: covering (including set-covering and maximal covering), p-center, p-dispersion, p-median, fixed charge, hub, and maxisum. Distances or some related measures (e.g., travel time or cost) are fundamental to such problems.

Consequently, we classify them according to their consideration of distance. The first four are based on maximum distance and the last four are based on total (or average) distance.

In this dissertation, we discuss problems related to the p-median and covering problems. As Marianov and Serra (2002) pointed out, both the p-median and covering problems can be considered benchmarks in the development of facility location models. While the p-center problem is also an important location model, the location set covering problem[1] can be used as a subproblem in solving the classical p-center problem(Handler and Mirchandani, 1979; Handler, 1990). Daskin (2000) showed how the maximal covering model can be used effectively in place of the location set covering model as a sub-problem in solving the unweighted vertex p-center problem.

Since all problems studied in this dissertation are either median-type or covering-type problems, in the next two sections, we will introduce them briefly. More detailed literature reviews will be given in the following four chapters for each investigated problem separately.

1.2 The p-median problem

The p-median problem belongs to a class of formulations called minisum location models. The problem can be stated as:

[1] We will introduce this problem in Section 3 of this chapter

Find the location of a fixed number of p facilities so as to minimize the weighted average distance of the system.

The first explicit formulation of the p-median problem is attributed to Hakimi (1964). Hakimi not only stated the formulation of the problem, but also proved that in a connected network optimal locations can always be found on nodes. Consequently, it is only necessary to consider the nodes of the network as potential locations. Goldman (1971) provided simple algorithms for locating a single facility for both an acyclic network (a tree) and a network containing exactly one cycle. The p-median problem on a plane (continuous feasible space) is also known as the Weber problem. We refer the reader to Drezner *et al.* (2002) for a detailed discussion of the Weber problem. For the polynomial time algorithm of the p-median problem on a tree network, the interested reader is directed to Kariv and Hakimi (1979). For a discussion of formulations and solution approaches of the p-median problem, we refer the readers to Mirchandani (1990). Recently, Marionov and Serra (2002) gave a state-of-the-art review of the p-median problem and its extensions.

1.3 Covering problems

Unlike the p-median problem which seeks to minimize the weighted travel distance, covering models are based on the concept of acceptable proximity. A customer is considered covered, if she has a facility sited within a preset distance. Covering models can be classified according to several criteria. One of such criteria is the type of objective, which allows us to distinguish between two types of formulations. The first type belongs to the Location Set Covering Problem (LSCP), which seeks to find the minimum number of facilities that cover all customers' demand. This problem was originally stated in Toregas *et al.* (1971). The second type can be classified as the Maximal Covering Location Problem (MCLP), which maximizes covered customer demand, given a limited number of facilities. The MCLP was first introduced in Church and ReVelle (1974).

Church and Meadows (1979) provided a pseudo-Hakimi property for the MCLP. This property states that for any network, there exists a finite set of points that will contain at least one of the optimal solutions to the MCLP. Daskin and Stern (1981), Hogan and ReVelle (1986), and Batta and Mannur (1990) developed the MCLP that contains a secondary "backup" coverage objective. Berman and Krass (2002) showed that the MCLP with a step coverage function is equivalent to the uncapacitated facility location problem (Cornuéjols et al, 1990). They developed two IP formulations for the problem and showed an interesting result that the LP relaxations of both formulations provide the same value of the upper bound. In a recent paper, Berman *et al.* (2003) investigated the MCLP with a coverage decay function whose value decreases from full coverage at the lowest pre-specified radius to no coverage at the highest pre-specified radius. Daskin (1983) provided a probabilistic formulation of the problem in which the probability of an arbitrary server being busy is specified exogenously. The objective, then, is to locate facilities so as to maximize the expected number of demand that a facility can cover. Daskin's formulation is sometimes referred to as the Maximal Expected Covering Location Problem. A review of covering models and their applications can be found in ReVelle and Williams (2002).

1.4 Location Models with Multiple Facility and/or Customer Types

Among the extensive research on location theory, there are only a few models considering multiple types of facilities and/or multiple types of customers. In practice, systems that provide products/services generally consist of more than one type of facilities.

For example, the health care delivery system cited in Narula (1984) consists of health centers, hospitals, and medical centers. In this system, medical centers provide specialized care in addition to the services available at hospitals and health centers. Similarly, hospitals provide more services than health centers. Therefore, a patient who is not accepted by a health center may be referred to a hospital or a medical center; and if necessary, a patient who first visits a hospital may be referred to a medical center. A medical center is equipped to treat all cases.

The system described in the above example is sometimes termed as ”successively inclusive hierarchical system”. The objective of a k-hierarchical facility location problem is to minimize the total weighted travel distance through locating p_i type-i facilities, $i = 1, \cdots, k$, on the network. Narula (1984) gave a classification of the various hierarchical facility location problems based on successively inclusive and successively exclusive hierarchies. Informally, in a successively inclusive hierarchy, a type-m facility ($m = 1, \cdots, k$) can serve demands of types $1, \cdots, m$, while in a successively exclusive hierarchy, a type-m facility serves only demands of type-m. See Calvo and Marks (1979), Narula and Ogbu (1979, 1985) for the formulation of the successively inclusive hierarchical location problem and the Lagrangian Relaxation algorithm for the system with two hierarchies. Mirchandani (1987) generalized different hierarchical systems and provided an integer programming formulation.

Berlin *et al.* (1976) considered the ambulance-hospital location problem with the objective of minimizing the total weighted round trip distance through locating p ambulance stations. They assumed that an ambulance will always bring the patient from the demand node to the nearest hospital. They showed that the ambulance-hospital location problem could be reduced into the p-median problem. Note that this is not a clear hierarchical model since relations among levels differ from the traditional hierarchical models.

Goldman (1969), Hakimi and Maheshwari (1972) and Wendell and Hurter (1973) studied a multi-commodity facility location problem, in which the nodes of the network include sources (supply points) and destinations (demand points). There are multiple types of demands (commodities) between source-destination pairs and each commodity may require several stages of processing. The objective is to minimize the total transportation cost through locating n center facilities. However, they assumed that each facility is capable of performing any of the processing stages on all commodities. In other words, only a single type of facility is located on the network.

Another version of multi-commodity facility location problem (Geoffrion and Graves, 1974) is an extension of the uncapacitated facility location problem. There are several commodities produced at several plants and there is a known demand for each commodity at each demand node. The objective is to minimize the total distribution cost and facility costs (fixed facility open cost and variable facility running cost) through locating distribution center facilities.

The previous model is one extension of the uncapacitated facility location problem, which involves in multiple types of demands and one type of facility to be located. Another extension of the uncapacitated facility location problem is the k-level uncapacitated facility location problem. For example, in the 2-level uncapacitated facility location problem (2-UFLP), demands must be satisfied from a plant through a warehouse (Kaufman, Eede and Hansen, 1977). The 2-UFLP seeks to determine where to open plants and warehouses and how to assign clients to paths along open facilities such that the total cost of opening facilities and serving clients is minimized (See Zhang, 2004 for the discussion and literature review). Unlike the previous multi-commodity facility location problem, there is only one type of demand in this problem.

Drezner and Wesolowsky (2001), and Berman, Drezner and Wesolowsky (2002) considered the Collection Depots Location Problem in the plane and on the network, respectively. In this problem, a single facility needs to be located to serve a set of customers. Each service consists of a trip to the customer, collecting materials, dropping the materials at one of the available collection depots and returning to the facility to wait for the next call. Therefore, there exist two type of facilities in this system: sever facility and depot facility.

All previous models mainly use the minisum criterion (Berman, Drezner and Wesolowsky, 2002, also discussed the problem with the maximin objective). In terms of the objective of maximum coverage, it seems that the FLEET (Facility Location Equipment Emplacement Technique), due to Schilling *et al.* (1979), is the first model considering the simultaneous location of several types of fire emergency facilities. The FLEET model locates a limited number of engine companies, truck companies and stations in such a way so as to maximize customers covered by both types of companies within appropriate distance standards. Both types of companies have to be sited in a single station. This model requires only a single engine and a singe truck to cover customer demand. Later, Marionov and ReVelle (1991) extended the FLEET model to fit the situation of requiring multiple servers for each type of facility.

Moore and ReVelle (1982) investigated a hierarchical service location problem utilizing a covering objective function. They considered a two-level system in which a customer is covered only when he or she can access *both* the higher and the lower level services. Since people may be willing to travel farther to a higher level facility to obtain a lower level service, coverage is defined in terms of access to services, not in terms of access to facilities. Similar to our problem, as we will see, this two-level system incurs three coverage radii, but there exists only one type of customers in this hierarchical service location problem.

Recently, Berman *et al.* (2004a, 2004b, 2004c) studied a type of hub location problem, in which a hub serves as a transfer point for customers who need the services of a facility. The objective is to minimize the total travel time for all customers on the network through locating the facility and/or a set of p hubs. They also investigated the minimax version of the hub location problem. In this hub location problem, there exist two types of facilities on the network.

In the hierarchical location model, there are at most as many types of customers as the hierarchical levels, while in the FLEET model, the hierarchical service location problem (Moore and ReVelle, 1982), and the hub location problem (Berman *et al.*, 2004a, 2004b, 2004c), there is only a single type of customers. However, as we mentioned in Abstract, multi-purpose trips are quite common in practice. It is easy to see that in the setting where multi-purpose trips exist, the number of customer types is more than that of facility types. Suzuki and Hodgson (2002) investigated the multi-purpose trip location problem for two different types of facilities and three different types of customers, however only an integer programming formulation for the minisum problem was presented.

1.5 Models Studied in This Dissertation

In this section, we briefly introduce four location models which are studied in this dissertation. Further literature review and applications are given in each chapter.

We first study the minisum collection depots location problem with multiple facilities on a network in Chapter 2. A septic tank cleaning service is an application of the collection depots location problem. The same location model arises when a depot must be included on the way to the customer. For example , consider a case where orders are received by phone, and a driver drives to one of avaliable warehouses to pick the required order, drives to the customer to deliver, and then drives back to wait at the facility for the next order (Drezner and Wesolowsky, 2001). As we mentioned above, both Drezner and Wesolowsky (2001), and Berman, Drezner and Wesolowsky (2002) considered the collection depots location problem with a single facility either in the plane or on the network. We extended this problem to the setting of multiple facilities on the network with a median-type objective. We prove that there exists a dominating location set for the problem on a general network. The properties of the solution on some simple network topologies are discussed. To solve the problem on a general network, we suggest a Lagrangian Relaxation imbedded in a branch-and-bound algorithm. We note that the collection depots location problem with multiple facilities employing a center-type objective is studied by Tamir and Halman (2003).

In Chapters 3 and 4, we consider multi-purpose trip location problems with the same setting as of Suzuki and Hodgson (2002). There are two different types of facilities to be located on the network. Customers can classified into three groups: two of them only need one type of service; the third one needs service from both types of facilities in a single trip. The objective is to minimize the total travel distance for all three types of customers through locating two different types of facilities. The model we present can be applicable to a number of settings. For example, Sears Canada operates several apparel and hardware stores simultaneously in most cities of Canada. Obviously, some customers only want to buy clothes while others only need some hardware. When planning where to locate the two types of facilities Sears Canada can take into account those customers that may need both hardware and clothes at the same time. Since customers can earn some reward points whenever they shop at Sears, the third type of customers is quite common.

The multi-purpose trip location model can be of great benefit for companies that are not in competition with each other and may want to coordinate their location decisions. An example is book stores and coffee shops. By planning their location decisions together book store companies and coffee shop companies can take advantage of many customers that may wish to visit both types of facilities on the same trip. Co-location of these type of facilities that occur in our model can explain the co-location of coffee shops and book stores that is becoming very popular.

In Chapter 3, we discuss the minisum multi-purpose trip location problem with two types of facilities and three types of customers. The objective is to minimize the total weighted travel distance of all trips. For this minisum problem, we prove that there exists a dominating location set on a general network. The properties of optimal solutions on networks with simple topologies are analyzed. We also investigate several heuristic approaches to the problem on general networks.

In Chapter 4, we study the multi-purpose trip location problem with the objective of maximizing coverage. We propose a method which sometimes produces a tighter bound than the LP relaxation of our formulation. A dynamic programming algorithm is applied to the problem with a general number of facilities of one type and a single facility of the other type on a path. Although the objective function of the problem is not submodular, computational experiments show that the greedy heuristic performs very well.

According to the desirability of facilities, covering models can be classified into desirable and undesirable location problems. In Chapter 5, we study a location problem with a minimum covering criterion. The objective is to locate a number of undesirable facilities on a network so as to minimize the total demand covered subject to the requirement that no two facilities are allowed to be closer than a pre-specified distance. This problem is an extension to the Expropriation Location Problem studied by Berman, Drezner and Wesolowsky (2003), in which only a single facility is considered. It is easy to see that if there is no distance constraint between facilities, all facilities will be concentrated in one point. On the other hand, for sensitivity and safety reasons, sometimes facilities should be separated. For example, if several nuclear reactors are clustered in the same region, they may all be attacked by an aggressor. We prove that there exists a dominating location set if there is some feasible solution to the problem. However, given the distance constraints, it is NP-hard to determine whether there exists a feasible solution or not. We compare several different mathematical formulations to solve the problem.

Chapter 2

The Minisum Collection Depots Location Problem with Multiple Facilities

2.1 Introduction

In the collection depots location problem, which is introduced in Drezner and Wesolowsky (2001), the objective is to find an optimal home location of a facility which needs to service a given set of customers. A set of collection depots is also given. Each service consists of a trip to the customer, collecting materials, dropping the materials at one of the available collection depots and returning to the facility to wait for the next call. A service trip can also take the reverse form: a server leaves the facility for a collection depot to pick up some material, then deliver this material to the customer and returns to the home location. The server is assumed to have complete information on traffic conditions and always selects the collection depot such that the total travel distance is minimized. We refer the reader to Drezner and Wesolowsky (2001), and Berman, Drezner and Wesolowsky (2002) for discussion of applications of this problem.

A typical objective is to minimize the weighted travel distance of all trips. Notice that the two types of service trips discussed above have the same expression of the objective function because of the property of round trips. Drezner and Wesolowsky (2001) address the problem in the plane. Berman, Drezner and Wesolowsky(2002) study the problem on a general network and on a tree network. Both papers only consider the problem of locating a single facility.

The objective of this chapter is to extend the collection depots location problem to siting a general number of facilities on networks. Note that the problem presented in this chapter is different from the ambulance-hospital location problem introduced in Berlin, ReVelle and Elzinga (1976). In Berlin, ReVelle and Elzinga (1976), the objective is also to minimize the total weighted round trip distance through locating p ambulance stations, but they assume that an ambulance will always bring the patient from the demand node to the nearest hospital. This assumption is quite normal for emergency service. With this assumption, Berlin, ReVelle and Elzinga (1976) reduced the problem to the p-median problem. The problem presented in the current chapter emphasizes the efficiency of server utilization, rather than minimizing customer travel time, thus in our setting the collection depot selected for serving a particular customer will generally not be the closest one to the customer.

A related problem to the median-type problem in this chapter is the center-type minimax collection depots location problem on networks, where the objective is to minimize the maximum weighted service time through locating a given number of facilities on the network. The problem of locating one facility was studied by Berman, Drezner and Wesolowsky(2002) and for general number of facilities by Tamir and Halman (2003).

The rest of the chapter is organized as follows. In the next section, we present the problem and prove that there exists an optimal nodal solution. In Section 3, we study the problem on some simple networks. In Section 4, a Lagrangian relaxation heuristic is proposed for the problem. In case when the Lagrangian relaxation doesn't converge, a branch-and-bound algorithm is called to close the duality gap. Section 5 contains an extension of our problem in which service facilities and collection depots are located simultaneously. Computational experiments are given in Section 6. Concluding remarks are provided in the last section.

2.2 Problem Formulation and Structural Results

Let $G = (N, L)$ be a network, where N is the set of nodes with $|N| = n$ and L is the set of links. Customers are located at nodes of the network. Let ω_i be the amount of material that needs to be collected in a typical period at node i. Let $X = \{x_1, x_2, \cdots, x_m\}$ be a given set of collection depots that are located on the network at m different points (note that $X \in N$ is not necessary). Let $Y = \{y_1, y_2, \cdots, y_p\}$ be the set of facilities to be located on the network.

Each service trip is composed of three parts: travelling from the facility to a customer to collect some material, travelling from the customer to a collection depot to dispose of the material, and travelling from the collection depot back to the facility's home location. The minisum problem is to find a set of locations Y that minimizes the weighted length of trips. Let $d(u, v)$ be the shortest distance between any two points u and v on the network. Assume that distances satisfy the triangle inequality. For a customer at node i, the total travel distance is

$$g_i(Y) \equiv \min_{1 \le j \le p} \min_{1 \le k \le m} \{d(y_j, i) + d(i, x_k) + d(x_k, y_j)\}$$

The minisum problem can be written as following:

$$\min_{Y \in G} f(Y) = \sum_{i=1}^{n} \omega_i g_i(Y)$$

A natural idea is to co-locate facilities with depots (assuming that there are at least as many facilities as depots). This is indeed optimal as shown in the following result.

Lemma 2.1 *When $m \le p$, the optimal solution for the minisum problem is the set of depots' locations.*

Proof. Suppose $Y^\star$ is an optimal solution. Let $y_j^\star(i)$ and $x_k(i)$ be the pair of facility and depot serving node i in an optimal assignment determined by $Y^\star$. Since $g_i(Y^\star) =$

$d(y_j^\star(i), i) + d(i, x_k(i)) + d(x_k(i), y_j^\star(i)) \geq 2d(i, x_k(i))$ (using triangle inequality),

$$f(Y^\star) \geq 2\sum_{i=1}^{n} \omega_i d(i, x_k(i)) \geq f(x_1, \cdots, x_m).$$

■

It follows that when $m \leq p$ there is no advantage to locating more than m facilities. In the example given on page 11, when orders are not that frequent, there is no incentive to locate too many facilities. We will assume from now on that $p < m$.

If we consider a non-nodal depot as a node with weight 0, we can replace G with a new graph $G' = (N', L')$, where $N' = N \cup X$ and $\omega_i = 0$ for $i \in X$. As an example consider the 3-node network in Figure 5.2 with 2 non-nodal depots x_1, x_2, where $N = \{1, 2, 3\}$ and $L = \{(1, 2), (2, 3), (1, 3)\}$. The new graph is $G' = (N', L')$, where $L' = \{(1, x_1), (x_1, x_2), (x_2, 2), (2, 3), (1, 3)\}$ and $N' = \{1, 2, 3, x_1, x_2\}$. It is easy to verify that the objective functions of the problem in G and G' are same.

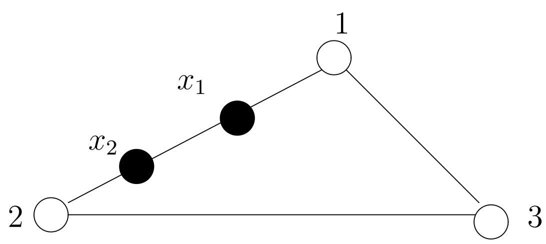


Figure 2.1: A simple 3-node network with two non-nodal depots

The following result generalizes a similar result for the 1-facility case from Berman, Drezner and Wesolowsky (2002).

Theorem 2.1 *There exists an optimal set of location $Y^\star$ on the nodes of G'.*

Proof. Suppose that in an optimal solution $Y^\star = (y_1^\star, \cdots, y_p^\star)$, $y_p^\star$ is an internal point on link (a, b) between nodes a and b. We will show that by changing the location $y_p^\star$ to one of the two nodes a and b, the weighted travel distance cannot increase. Let $N'(y_p^\star, x_k)$ ($k = 1, \cdots, m$) be the set of nodes that are serviced by the facility located at $y_p^\star$ and depot x_k jointly. Let $\bar{N}'(y_p^\star)$ be the set of nodes not serviced by $y_p^\star$, i.e., $\bar{N}'(y_p^\star) = N' - \cup_{k=1}^m N'(y_p^\star, x_k)$. Ties can be broken arbitrarily. The weighted travel distance $f(Y^\star)$ can be expressed as

$$\begin{aligned} &\sum_{k=1}^{m} \sum_{i \in N'(y_p^\star, x_k)} \omega_i [d(y_p^\star, i) + d(i, x_k) + d(x_k, y_p^\star)] \\ &\quad + \sum_{i \in \bar{N}'(y_p^\star)} \omega_i \min_{1 \le j \le p-1} \min_{1 \le k \le m} \{d(y_j^\star, i) + d(i, x_k) + d(x_k, y_j^\star)\} \end{aligned} \tag{2.1}$$

Let $AA(y_p^\star, x_k)$ be the set of nodes in $N'(y_p^\star, x_k)$ for which the server from $y_p^\star$ travels most efficiently first through a and finally returns to its home location through a, i.e., if $i \in AA(y_p^\star, x_k)$, then the most efficient trip for the demand at node i is $y_p^\star - a - \cdots - i - \cdots - x_k - \cdots - a - y_p^\star$. Note that to finish a service for a customer(node) in $AA(y_p^\star, x_k)$, the server located at $y_p^\star$ may pass through a two or more times. Let $AB(y_p^\star, x_k)$ be the set of nodes in $N'(y_p^\star, x_k)$ for which the server from $y_p^\star$ travels first through node a and finally returns to its home location through node b, i.e., if $i \in AB(y_p^\star, x_k)$, then the most efficient trip for the demand at node i is $y_p^\star - a - \cdots - i - \cdots - x_k - \cdots - b - y_p^\star$. Again, to provide service to $i \in AB(y^\star, x_k)$, the server pass through a at least once. Similarly, we define set $BB(y_p^\star, x_k)$, $BA(y_p^\star, x_k)$. We break ties arbitrarily so that $AA(y_p^\star, x_k) \cup BB(y_p^\star, x_k) \cup AB(y_p^\star, x_k) \cup BA(y_p^\star, x_k) = N'(y_p^\star, x_k)$ and $AA(y_p^\star, x_k), BB(y_p^\star, x_k), AB(y_p^\star, x_k), BA(y_p^\star, x_k)$ are pairwise disjoint.

We denote by l the length of link (a, b). Assume the distance from a to $y_p^\star$ is y. For any $i \in AA(y_p^\star, x_k)$, the shortest travel distance is $2y + d(a, i) + d(i, x_k) + d(x_k, a)$ whereas for any $i \in BB(y_p^\star, x_k)$, the shortest travel distance is $2l - 2y + d(b, i) + d(i, x_k) + d(x_k, b)$. For any $i \in AB(y_p^\star, x_k)$, the shortest travel distance is $l + d(a, i) + d(i, x_k) + d(x_k, b)$, and for any $i \in BA(y_p^\star, x_k)$, the shortest travel distance is $l + d(b, i) + d(i, x_k) + d(x_k, a)$. Therefore the first term of expression (2.1) can be written as

$$\begin{aligned}
&\sum_{k=1}^{m} \sum_{i \in AA(y_p^\star, x_k)} \omega_i [2y + d(a, i) + d(i, x_k) + d(x_k, a)] \\
&\quad + \sum_{k=1}^{m} \sum_{i \in BB(y_p^\star, x_k)} \omega_i [2l - 2y + d(b, i) + d(i, x_k) + d(x_k, b)] \\
&\quad + \sum_{k=1}^{m} \sum_{i \in AB(y_p^\star, x_k)} \omega_i [l + d(a, i) + d(i, x_k) + d(x_k, b)] \\
&\quad + \sum_{k=1}^{m} \sum_{i \in BA(y_p^\star, x_k)} \omega_i [l + d(b, i) + d(i, x_k) + d(x_k, a)] \\
=&2\Big(\sum_{k=1}^{m} \sum_{i \in AA(y_p^\star, x_k)} \omega_i - \sum_{k=1}^{m} \sum_{i \in BB(y_p^\star, x_k)} \omega_i\Big) y + C,
\end{aligned} \tag{2.2}$$

where

$$\begin{aligned}
C =& \sum_{k=1}^{m} \sum_{i \in AA(y_p^\star, x_k)} \omega_i [d(a, i) + d(i, x_k) + d(x_k, a)] \\
&+ \sum_{k=1}^{m} \sum_{i \in BB(y_p^\star, x_k)} \omega_i [2l + d(b, i) + d(i, x_k) + d(x_k, b)] \\
&+ \sum_{k=1}^{m} \sum_{i \in AB(y_p^\star, x_k)} \omega_i [l + d(a, i) + d(i, x_k) + d(x_k, b)] \\
&+ \sum_{k=1}^{m} \sum_{i \in BA(y_p^\star, x_k)} \omega_i [l + d(b, i) + d(i, x_k) + d(x_k, a)]
\end{aligned}$$

Without loss of generality, we assume that

$$\sum_{k=1}^{m} \sum_{i \in AA(y_p^\star, x_k)} \omega_i \geq \sum_{k=1}^{m} \sum_{i \in BB(y_p^\star, x_k)} \omega_i$$

Substituting (2.2) in (2.1), we can conclude that by taking $y_p^\star$ to a, the weighted travel distance can't increase even without changing the assignment of nodes to facilities or depots. Therefore, the location $(y_1^\star, \cdots, y_{p-1}^\star, a)$ is at least as good as $Y^\star$. The same process can be performed for any other non-nodal location, which concludes the theorem. ■

Note that the optimal solution is not necessarily in X. As an example consider the tree with $2(v+1)$ nodes, where v is a positive integer (e.g., $v = 6$ in Figure 2.2 where the length of link $(1, 2)$ is 5 and the length of all other links is 1). Suppose that there are $2v$ depots located on the leaves of the tree and that we want to locate 2 facilities. Assume $\omega_i = 1$ for all i. It is easy to verify that the optimal locations are node 1 and 2 with optimal objective value of $4(v + 1)$. If we force facility locations in X, then it is easy to see that the best objective function value is $2[0 + 2 + 4(v - 1)] = 8v - 4$. As v goes to infinity, relative error approaches 100%. Therefore, restricting potential sites in X may produce a very bad solution.

To complete this section, we formulate the problem described above as a mathematical program. The decision variables in the model are:

$$z_{ijk} = \begin{cases} 1 & \text{If node } i \text{ is assigned to the facility at } j \text{ and depot } k \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

and

$$\pi_j = \begin{cases} 1 & \text{If there is a facility located at } j \in N' \\ 0 & \text{otherwise} \end{cases}$$

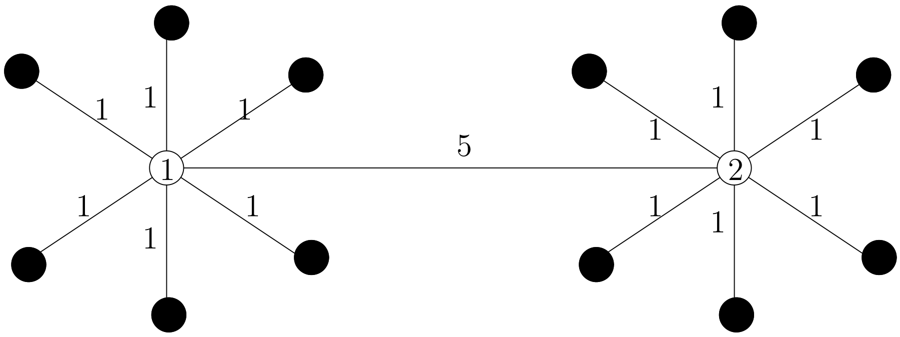


Figure 2.2: A network with the optimal solution not in X

Now according to Theorem 2.1, the problem can be formulated as:

$$\textit{problem } P \qquad \min \sum_{i \in N} \sum_{j \in N'} \sum_{k \in X} \omega_i [d(j,i) + d(i,k) + d(k,j)] z_{ijk}$$

$$s.t. \qquad \sum_{j \in N'} \pi_j = p \tag{2.3}$$

$$\sum_{j \in N'} \sum_{k \in X} z_{ijk} = 1 \qquad i \in N \tag{2.4}$$

$$z_{ijk} \leq \pi_j \qquad i \in N,\ j \in N',\ k \in X \tag{2.5}$$

$$z_{ijk}, \pi_j \in \{0,1\} \qquad i \in N,\ j \in N',\ k \in X \tag{2.6}$$

Constraint (2.3) specifies that exactly p facilities are to be located. Constraint set (2.4) states that each demand node can only be assigned to one pair of facility and collection depot. Constraint set (2.5) forbids the assignment of a demand node except to an open facility. Finally, constraint set (2.6) establishes a binary restriction on the decision variables. To solve this problem, we develop a Lagrangian relaxation embedded in a branch-and-bound algorithm in Section 4. In the next section, we study the location problem on some special topology networks.

2.3 Multiple facilities location on some simple network topologies

In this section we analyze the multi-facility collection depots location problem for three types of networks: trees, cycles, and paths.

Suppose $T = (N, L)$ is a tree. Considering each depot as a new node with weight 0, we obtain a new tree $T' = (N', L')$ as we did in the previous section. The following result extending the corresponding result in Berman, Drezner and Wesolowsky(2002) to the multiple facilities case allows us to restrict the search for optimal locations to a subtree of T'.

Theorem 2.2 *Let $\hat{T} = (\hat{N}, \hat{L})$ be the smallest subtree of T' that includes X. An optimal solution to the problem on a tree exists in $\hat{N}$.*

Proof. Without loss of generality, suppose that in an optimal solution $Y^\star = (y_1^\star, \cdots, y_p^\star)$, $y_p^\star \notin \hat{N}$. Let v be the closest node in $\hat{N}$ to $y_p^\star$. Let T_v be the largest subtree of T' that includes node $y_p^\star$ if node v is deleted (the subtree that includes node $y_p^\star$ and all points in $N' - \hat{N}$ such that the path from them to node $y_p^\star$ does not go through node v). Since the network we consider is a tree, we know that there exists a unique path from $y_p^\star$ to v. Let the node adjacent to $y_p^\star$ on this path be u. Let T_u be the largest subtree that includes node $y_p^\star$

if node u is deleted.

As before, let $N'(y_p^\star, x_k)$ $(k = 1, \cdots, m)$ be the set of nodes that are served by the facility located at $y_p^\star$ and collection depot x_k jointly and let $\bar{N}'(y_p^\star) = N' - \cup_{k=1}^m N'(y_p^\star, x_k)$ be the set that includes the rest of nodes. Therefore, $f(Y^\star)$ can be expressed as

$$\sum_{k=1}^{m} \sum_{i \in N'(y_p^\star, x_k)} \omega_i [d(y_p^\star, i) + d(i, x_k) + d(x_k, y_p^\star)] + C, \tag{2.7}$$

where

$$C = \sum_{i \in \bar{N}'(y_p^\star)} \omega_i \min_{1 \le j \le p-1} \min_{1 \le k \le m} \{d(y_j^\star, i) + d(i, x_k) + d(x_k, y_j^\star\}$$

According to the definition, it is easy to see that T_u is a subtree of T_v, i.e., $T_u \cup (T_v - T_u) \cup (T - T_v) = T$. Therefore, the first term of (2.7) can be written as

$$\begin{aligned}
& \sum_{k=1}^{m} \sum_{i \in N'(y_p^\star, x_k) \cap T_u} \omega_i [d(y_p^\star, i) + d(i, x_k) + d(x_k, y_p^\star)] \\
& \quad + \sum_{k=1}^{m} \sum_{i \in N'(y_p^\star, x_k) \cap (T_v - T_u)} \omega_i [d(y_p^\star, i) + d(i, x_k) + d(x_k, y_p^\star)] \\
& \quad + \sum_{k=1}^{m} \sum_{i \in N'(y_p^\star, x_k) \cap (T - T_v)} \omega_i [d(y_p^\star, i) + d(i, x_k) + d(x_k, y_p^\star)] \\
= & \sum_{k=1}^{m} \sum_{i \in N'(y_p^\star, x_k) \cap T_u} \omega_i [d(i, v) + d(v, x_k) + d(x_k, i)] \\
& \quad + \sum_{k=1}^{m} \sum_{i \in N'(y_p^\star, x_k) \cap (T_v - T_u)} \omega_i [2d(y_p^\star, i) + d(i, x_k) + d(x_k, v) + d(v, i)] \\
& \quad + \sum_{k=1}^{m} \sum_{i \in N'(y_p^\star, x_k) \cap (T - T_v)} \omega_i [2d(y_p^\star, v) + d(v, i) + d(i, x_k) + d(x_k, v)] \\
\ge & \sum_{k=1}^{m} \sum_{i \in N'(y_p^\star, x_k)} \omega_i [d(v, i) + d(i, x_k) + d(x_k, v)]
\end{aligned}$$

Even if there is no change of the assignment of nodes to facilities or depots, then

$$\begin{aligned}
& f((y_1^\star, \cdots, y_{p-1}^\star, v)) \\
= & \sum_{k=1}^{m} \sum_{i \in N'(y_p^\star, x_k)} \omega_i [d(v,i) + d(i,x_k) + d(x_k, v)] \\
& + \sum_{i \in \bar{N}'(y_p^\star)} \omega_i \min_{1 \le j \le p-1} \min_{1 \le k \le m} \{d(y_j^\star, i) + d(i, x_k) + d(x_k, y_j^\star\}
\end{aligned}$$

Therefore, $f(Y^\star)$ cannot be better than $f(y_1^\star, \cdots, y_{p-1}^\star, v)$. The same process can be performed for any other location not in $\hat{N}$, which proves the theorem. ■

Next we consider a network which is a cycle. The following results shows that in this case, it is always optimal to co-locate facilities with depots.

Theorem 2.3 *Suppose G is a cycle, then there must exist an optimal solution in X.*

Proof. Suppose that $Y^\star = (y_1^\star, \cdots, y_p^\star)$ is an optimal solution. From Theorem 2.1, we know that $y_i^\star \in N \cup X$ for all $i \in \{1, \cdots, p\}$. Without loss of generality, we assume that $y_p^\star$ is a non-depot node. Let x_1 and x_2 be two depots adjacent to $y_p^\star$ (see Figure 2.3). Assume that $y_p^\star$ is closer to x_2 than to x_1. Let t be the point on the path $x_1 - y_p^\star - x_2$ such that $d(x_1, t) = d(x_2, y_p^\star)$. Denote by v the antipode of node $y_p^\star$ on the cycle, i.e., $d(y_p^\star, x_1) + d(x_1, v) = d(y_p^\star, x_2) + d(x_2, v)$. Let the set of nodes on the path $t - x_1 - v$ be N_1 and let the set of nodes on the path $y_p^\star - x_2 - v$ be N_2. Denote $N \backslash (N_1 \cup N_2)$ by N_3. Ties can be broken arbitrarily, so $N_1 \cup N_2 \cup N_3 = N$ and N_1, N_2, N_3 are pairwise disjoint.

Let $N(y_p^\star)$ be the set of nodes that are served by $y_p^\star$ most efficiently and let $\bar{N}(y_p^\star)$ be the set that includes the rest of nodes. It is obvious that all nodes in $N(y_p^\star)$ will utilize collection

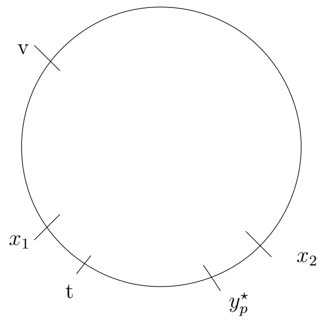


Figure 2.3: A cycle network

depots x_1 or x_2. The weighted travel distance $f(Y^\star)$ can be written as

$$\begin{aligned}
&\sum_{i\in N(y_p^\star)\cap N_1} \omega_i[d(y_p^\star,i)+d(i,x_1)+d(x_1,y_p^\star)] \\
&+\sum_{i\in N(y_p^\star)\cap N_2} \omega_i[d(y_p^\star,i)+d(i,x_2)+d(x_2,y_p^\star)] \\
&+\sum_{i\in N(y_p^\star)\cap N_3} \omega_i[d(y_p^\star,i)+d(i,x_2)+d(x_2,y_p^\star)] \\
&+\sum_{i\in \bar{N}(y_p^\star)} \omega_i \min_{1\le j\le p-1}\min_{1\le k\le m}\{d(y_j^\star,i)+d(i,x_k)+d(x_k,y_j^\star)\}
\end{aligned}$$

Let $y_p^{\star-}$ be the point at a distance ϵ from $y_p^\star$ in the clockwise direction from $y_p^\star$ to x_1 and let $y_p^{\star+}$ be the point at a distance ϵ from $y_p^\star$ in the counter-clockwise direction from $y_p^\star$ to x_2, where ϵ is a small positive number such that there is no other demand node in $(y_p^{\star-}, y_p^{\star+})$. Similarly to $N(y_p^\star)$ and $\bar{N}(y_p^\star)$, we define $N(y_p^{\star-})$, $N(y_p^{\star+})$, $\bar{N}(y_p^{\star-})$, and $\bar{N}(y_p^{\star+})$. We have

$$\begin{aligned}
&f(y_1^\star,\cdots,y_{p-1}^\star,y_p^{\star-}) \\
\le &\sum_{i\in N(y_p^{\star-})\cap N_1} \omega_i[d(y_p^\star,i)+d(i,x_1)+d(x_1,y_p^\star)-2\epsilon]
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{i \in N(y_p^{\star -}) \cap N_2} \omega_i [d(y_p^\star, i) + d(i, x_2) + d(x_2, y_p^\star) + 2\epsilon] \\
&+ \sum_{i \in N(y_p^{\star -}) \cap N_3} \omega_i [d(y_p^\star, i) + d(i, x_2) + d(x_2, y_p^\star)] \\
&+ \sum_{i \in \bar{N}(y_p^{\star -})} \omega_i \min_{1 \le j \le p-1} \min_{1 \le k \le m} \{d(y_j^\star, i) + d(i, x_k) + d(x_k, y_j^\star)\}
\end{aligned}
$$

The inequality is due to the fact that the assignment of depots to nodes can be changed after we move the facility $y_p^\star$. If ϵ is sufficiently small, we have $N(y_p^\star) = N(y_p^{\star -})$. So, we can obtain

$$
f(y_1^\star, \cdots, y_{p-1}^\star, y_p^{\star -}) \le \; f(Y^\star) - 2\Big(\sum_{i \in N(y_p^{\star -}) \cap N_1} \omega_i - \sum_{i \in N(y_p^{\star -}) \cap N_2} \omega_i\Big)\epsilon \tag{2.8}
$$

Similarly, we have

$$
\begin{aligned}
& f(y_1^\star, \cdots, y_{p-1}^\star, y_p^{\star +}) \\
\le & \sum_{i \in N(y_p^{\star +}) \cap N_1} \omega_i [d(y_p^\star, i) + d(i, x_1) + d(x_1, y_p^\star) + 2\epsilon] \\
&+ \sum_{i \in N(y_p^{\star +}) \cap N_2} \omega_i [d(y_p^\star, i) + d(i, x_2) + d(x_2, y_p^\star) - 2\epsilon] \\
&+ \sum_{i \in N(y_p^{\star +}) \cap N_3} \omega_i [d(y_p^\star, i) + d(i, x_2) + d(x_2, y_p^\star)] \\
&+ \sum_{i \in \bar{N}(y_p^{\star +})} \omega_i \min_{1 \le j \le p-1} \min_{1 \le k \le m} \{d(y_j^\star, i) + d(i, x_k) + d(x_k, y_j^\star)\} \\
= & \; f(Y^\star) + 2\Big(\sum_{i \in N(y_p^{\star +}) \cap N_1} \omega_i - \sum_{i \in N(y_p^{\star +}) \cap N_2} \omega_i\Big)\epsilon
\end{aligned} \tag{2.9}
$$

Note that $N(y_p^{\star -})$ and $N(y_p^{\star +})$ may be different since we can break ties as we need. It is easy to see that $N(y_p^{\star -}) \cap N_1 \supseteq N(y_p^{\star +}) \cap N_1$ and $N(y_p^{\star +}) \cap N_2 \supseteq N(y_p^{\star -}) \cap N_2$. Define

$$
NN^- = [N(y_p^{\star -}) \cap N_1] \backslash [N(y_p^{\star +}) \cap N_1]
$$

$$NN^+ = [N(y_p^{\star+}) \cap N_2] \backslash [N(y_p^{\star-}) \cap N_2]$$

NN^- and NN^+ can be empty. Expression (2.8) and (2.9) can be written as:

$$\begin{aligned} & f(y_1^\star, \cdots, y_{p-1}^\star, y_p^{\star-}) \\ & \leq f(Y^\star) - 2(\sum_{i \in N(y_p^{\star+}) \cap N_1} \omega_i - \sum_{i \in N(y_p^{\star-}) \cap N_2} \omega_i)\epsilon - 2\sum_{i \in NN^-} \omega_i \epsilon \end{aligned} \tag{2.10}$$

and

$$\begin{aligned} & f(y_1^\star, \cdots, y_{p-1}^\star, y_p^{\star+}) \\ & \leq f(Y^\star) + 2(\sum_{i \in N(y_p^{\star+}) \cap N_1} \omega_i - \sum_{i \in N(y_p^{\star-}) \cap N_2} \omega_i)\epsilon - 2\sum_{i \in NN^+} \omega_i \epsilon \end{aligned} \tag{2.11}$$

$Y^\star$ can't be a local optimum except when

$$\sum_{i \in N(y_p^{\star+}) \cap N_1} \omega_i - \sum_{i \in N(y_p^{\star-}) \cap N_2} \omega_i = 0 \text{ and } NN^- = NN^+ = \emptyset. \tag{2.12}$$

In that case, when we move $y_p^\star$ further we will reach some point for sure either that some demand node will change its server or depot or that (2.12) won't hold. Therefore, $Y^\star$ can't be global optimal. We conclude that either $(y_1^\star, \cdots y_{p-1}^\star, x_1)$ or $(y_1^\star, \cdots y_{p-1}^\star, x_2)$ is at least as good as $Y^\star$. The same process can be performed for any other non-depot node, which proves the theorem. ■

Corollary 2.1 *On a path, there must exist an optimal solution in X.*

Proof. Denote by 1 and n the two extreme nodes of the path. Add a link $(1, n)$ with $d(1, n) = +\infty$, we create a cycle. Since the length of link $(1, n)$ is infinity, it will never appear in any service trip. Therefore, the optimal solution to the original path will be as same as that to the new cycle. Following the result on a cycle, there exists an optimal solution in X if the underlying network is a path. ■

2.4 Solution to problem P on a general network

If $X = N$, our problem is a p-median problem. Therefore, the Collection Depots Location Problem for multiple facilities belongs to the notably difficult class of $\mathcal{NP}$-complete problems. There are no available algorithms specifically designed to solve problem P. We will describe a Lagrangian relaxation based solution procedure. The Lagrangian scheme has been successfully applied to the p-median problem (Narula, Ogbu and Samuelsson, 1977; Daskin, 1995) and other integer programming problems (Cornuejols, Fisher and Nemhauser, 1977; Fisher, 1981; Pirkul and Schilling, 1991).

Define $d_{ijk} = d(j,i) + d(i,k) + d(k,j)$. By relaxing constraint (2.4) of Problem P defined in Section 2, we have the Lagrangian dual of problem P:

$$\textit{problem } L \qquad L = \max_{\lambda} \min_{z,\pi} \sum_{i \in N} \sum_{j \in N'} \sum_{k \in X} [\omega_i d_{ijk} - \lambda_i] z_{ijk} + \sum_{i \in N} \lambda_i$$

$$s.t. \qquad \sum_{j \in N'} \pi_j = p$$

$$z_{ijk} \leq \pi_j \qquad i \in N,\ j \in N',\ k \in X$$

$$z_{ijk}, \pi_j \in \{0,1\} \qquad i \in N,\ j \in N',\ k \in X$$

As an alternative, we can also relax constraint (2.5) of Problem P. However, there are far more constraints being relaxed when (2.5) is relaxed than when (2.4) is relaxed. Based on the experience on the Lagrangian relaxation of the p-median problem (See page 231 of Daskin (1995) for the discussion), we use the relaxation of (2.4).

For fixed values of the Lagrangian multipliers, λ_i, we have a minimization problem, denoted by $L(\lambda)$, where $\lambda = (\lambda_1, \cdots, \lambda_n)$. Let

$$V_j = \sum_{i \in N} \sum_{k \in X} \min\{0, \omega_i d_{ijk} - \lambda_i\}. \tag{2.13}$$

Suppose the V_js are sorted in ascending order, we have $V_{j_1}, \cdots, V_{j_{n'}}$, where n' is the cardinality of N'. Since in the relaxed problem a demand node can be assigned to many facilities/depots, it is easy to verify that the objective value for fixed values of the Lagrangian multipliers is given by the summation of all λ_i and p smallest values of V_j. To obtain a feasible solution of $L(\lambda)$, we set $\pi_{j_1} = \cdots = \pi_{j_p} = 1$ and set all other values of $\pi_j = 0$. We then set

$$z_{ijk} = \begin{cases} 1 & \text{if } \pi_j = 1 \text{ and } \omega_i d_{ijk} - \lambda_i < 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

We denote by $LB(\lambda)$ the objective value of $L(\lambda)$, i.e.,

$$LB(\lambda) = \sum_{s=1}^{p} V_{j_s} + \sum_{i=1}^{n} \lambda_i. \tag{2.15}$$

Suppose that we locate facilities at those points for which $\pi_j = 1$. We can easily generate a feasible solution (z^0_{ijk}) to Problem P as follows: for each $i \in N$, select any one pair of j^* and

k^* (if not unique) such that

$$d_{ij^*k^*} = \min_{j,\,k}\{d_{ijk} : j \in N'\ , \pi_j = 1, k \in X\},$$

set $z^0_{ij^*k^*} = 1$ and $z^0_{ijk} = 0$ for all other combinations of j and k. Substituting the feasible solution in the objective function of Problem P, we can obtain an upper bound, which we denote by $UB(\lambda)$. It is well known that $LB(\lambda) \leq L \leq Z^\star \leq UB(\lambda)$, where $Z^\star$ is the optimal objective function value of Problem P. Therefore if z_{ijk} satisfy constraint (2.4), then (z_{ijk}, π_j) is optimal in problem P. Of course, we should not, in general, expect that z_{ijk} will be feasible in constraint (2.4).

To derive bounds using Problem L, we employ the subgradient optimization method (Crowder, 1976; Goffin, 1977; Fisher, 1981) which can be described as follows: given initial multipliers λ_i^o $(i = 1, \cdots, n)$, a sequence of multipliers is generated by

$$\lambda_i^{n+1} = \max\{0, \lambda_i^n - t^n(\sum_j \sum_k z_{ijk} - 1)\}, \tag{2.16}$$

where z_{ijk} is the optimal solution to Problem $L(\lambda^n)$ and t^n is a positive scalar step size defined as follows:

$$t^n = \frac{\alpha^n(UB(\lambda^n) - LB(\lambda^n))}{\sum_i(\sum_j \sum_k z_{ijk} - 1)^2} \tag{2.17}$$

where α^n is a scalar satisfying $0 < \alpha^n \leq 2$. We start with $\alpha^0 = 2$ and cut it by half every time $LB(\lambda)$ fails to increase after a fixed number of iterations. The algorithm is terminated either when the upper and lower bounds are sufficiently close to each other or when the iteration limit is reached. The initial trial multipliers are given by

$$\lambda_i^0 = \min_{(j,k)\neq(i,i)} \omega_i d_{ijk} \tag{2.18}$$

The algorithm can be stated as follows:

Procedure-RELAX

1. If $\binom{m}{p}$ is small, use complete enumeration to obtain best upper bound UB_X through locating only on X. Otherwise, set the upper bound $UB_X = \infty$. Let $n = 0$.

2. For λ^n, calculate the value of $LB(\lambda^n)$, $UB(\lambda^n)$ and a feasible solution $z_{ijk},\ \pi_i$ for $L(\lambda)$.

3. Terminate the procedure if either z_{ijk} satisfy constraint set (2.4) or the relative error between $LB(\lambda^n)$ and $\min\{UB_X, UB(\lambda^n\}$ is less than 0.1 percent or the iteration limit is reached, or the lower bound fails to increase after some consecutive iterations. Otherwise, calculate λ^{n+1} using (5.25) and the step size t^n using (11). Go to step 2.

As an example consider the simple network in Figure 2.4 where the length of links are the numbers next to the links and the demand weights are the numbers next to the nodes. Suppose there are three depots located at nodes 1, 2, and 3 and we wish to locate two facilities on the network.

If we locate facilities only on $X = \{1, 2, 3\}$, we can obtain an upper bound 2.6 when $Y = \{1, 3\}$. The values of $\omega_i d_{ijk}$s are calculated in Table 2.1.

Now using (2.18), we obtain $\lambda^0 = (1.2, 1.2, 1.6, 1.2)$. Applying (2.13), we obtain $V^0 = (-1.2, -1.2, -1.6, 0)$. Thus π^0 can be $(1, 0, 1, 0)$ or $(0, 1, 1, 0)$. We choose $\pi^0 = (1, 0, 1, 0)$ and get a lower bound $-1.2 - 1.6 + 1.2 + 1.2 + 1.6 + 1.2 = 2.4$. If we locate at nodes 1 and 3,

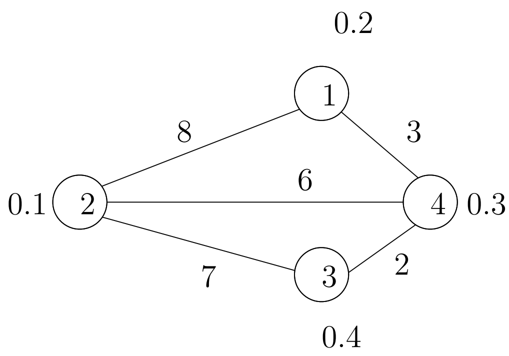


Figure 2.4: A simple network with 3 depots located at nodes 1,2, and 3

	j\k	1	2	3
i=1	1	0	3.2	2
	2	3.2	3.2	4
	3	2	4	2
	4	1.2	3.4	2
i=2	1	1.6	1.6	2
	2	1.6	0	1.4
	3	2	1.4	1.4
	4	1.7	1.2	1.5

a

	j\k	1	2	3
i=3	1	4	8	4
	2	8	5.6	5.6
	3	4	5.6	0
	4	4	6	1.6
i=4	1	1.8	5.1	3
	2	5.1	3.6	4.5
	3	3	4.5	1.2
	4	1.8	3.6	1.2

b

Table 2.1: values of $\omega_i d_{ijk}$

we obtain an upper bound 2.6. According to (2.14), we can get a feasible solution z_{ijk} of the relaxed problem for λ^0 depicted in Table 2.2.

It is obvious that z_{ijk} violates constraint (2.4) when $i = 2, 4$. Using (5.26), we obtain a step size of

$$t^0 = \frac{2(2.6 - 2.4)}{0 + (-1)^2 + 0 + (-1)^2} = 0.2$$

Next, using (5.25), we obtain $\lambda^1 = (1.2, 1.4, 1.6, 1.4)$. This ends the first iteration.

At the second iteration, we have $V^1 = (-1.2, -1.4, -1.8, -0.4)$ and $\pi^1 = (0, 1, 1, 0)$. So

	j\k	1	2	3
i=1	1	1	0	0
	2	0	0	0
	3	0	0	0
	4	0	0	0
i=2	1	0	0	0
	2	0	0	0
	3	0	0	0
	4	0	0	0

a

	j\k	1	2	3
i=3	1	0	0	0
	2	0	0	0
	3	0	0	1
	4	0	0	0
i=4	1	0	0	0
	2	0	0	0
	3	0	0	0
	4	0	0	0

b

Table 2.2: Solution (z_{ijk}) to the relaxed problem after the first iteration

the lower bound is still $2.4 = -1.4 - 1.8 + 1.2 + 1.4 + 1.6 + 1.4$. Locating at nodes 2 and 3, we obtain an upper bound 3.2. Since it is worse than the previous upper bound 2.6, the best upper bound remains at 2.6. According to (8), we obtain a feasible solution z_{ijk} of the relaxed problem for λ^1 depicted in Table 2.3.

	j\k	1	2	3
i=1	1	0	0	0
	2	0	0	0
	3	0	0	0
	4	0	0	0
i=2	1	0	0	0
	2	0	1	0
	3	0	0	0
	4	0	0	0

a

	j\k	1	2	3
i=3	1	0	0	0
	2	0	0	0
	3	0	0	1
	4	0	0	0
i=4	1	0	0	0
	2	0	0	0
	3	0	0	1
	4	0	0	0

b

Table 2.3: Solution (z_{ijk}) to the relaxed problem after the second iteration

Using (5.26), we obtain a step size of

$$t^1 = \frac{2(2.6 - 2.4)}{(-1)^2 + 0 + 0 + 0} = 0.4$$

Next, using (5.25), we obtain $\lambda^2 = (1.6, 1.4, 1.6, 1.4)$. This ends the second iteration.

At the third iteration, we have $V^2 = (-1.6, -1.4, -1, 8, -0.8)$. So the lower bound is $2.6 = -1.6 - 1.8 + 1.6 + 1.4 + 1.6 + 1.4$. Since the lower bound equals the best upper bound that we have found so far, we stop here and conclude the optimal solution is $(1, 3)$.

In this example, it was guaranteed that the solution we obtained from the Lagrangian relaxation is optimal since the upper bound equals the lower bound. Unfortunately, Lagrangian relaxation doesn't always converge to the optimal solution. In that case we use branch-and-bound to close the duality gap.

In the branch-and-bound scheme we branch on the values of π_j. The branch-and-bound procedure can be stated as follows:

Procedure-B&B

1. Separate Problem P into two subproblems according to setting $\pi_j = 0$ and $\pi_j = 1$ for some $j \in N'$.
2. Select an unfathomed subproblem.
3. Call *Procedure*-RELAX to solve the subproblem. If the lower bound is greater than the best upper bound or if the lower and upper bounds are sufficiently close, go to Step 5.
4. Separate a subproblem into two subproblems.
5. Update the upper bound if an improved upper bound is found. If all subproblems have been fathomed, stop; Otherwise, go to Step 2.

In Step 4, we choose to branch on a node i for which $|\sum_j \sum_k z_{ijk} - 1|$ is the largest in the current subproblem. When using the subgradient method in *Procedure*-RELAX, we take λ^0 equal to the terminal value of λ at the previous node (Fisher, 1981).

To illustrate the branch-and-bound algorithm, let's consider the simple network in Figure 2.5 where the length of links are the numbers next to the links and the demand weights of nodes are the numbers next to the nodes. Suppose there are 4 depots located at nodes 1, 2, 3 and 4. We want to locate 2 facilities on the network.

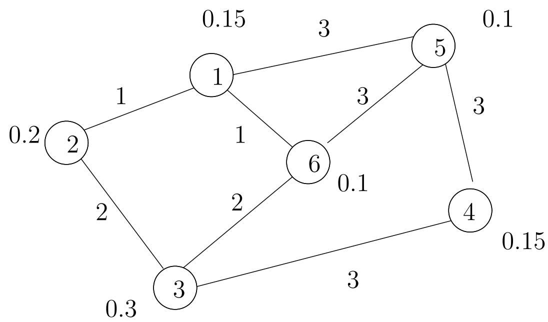


Figure 2.5: A network on which the Lagrangian Relaxation doesn't converge

The upper and lower bounds given by *Procedure*-RELAX for Problem P are 2.1 (corresponding to the feasible solution to problem P of locating at node 1 and 3) and 1.9, respectively. The values of $\sum_j \sum_k z_{ijk} - 1$ $(i = 1, \cdots, 6)$ are 1, 1, 0, -1, 0, 2, so we choose node 6 as the entering node to branch. For the subproblem with constraint $\pi_6 = 1$, *Procedure*-RELAX returns a lower bound 2.59 at the first iteration. Since 2.59 is greater than the best upper bound 2.1, this subproblem has been fathomed. The complete branch-and-bound tree is depicted in Figure 2.6. The best upper bound, 2.1, is the optimal objective value and nodes 1 and 3 are the optimal locations for the two facilities.

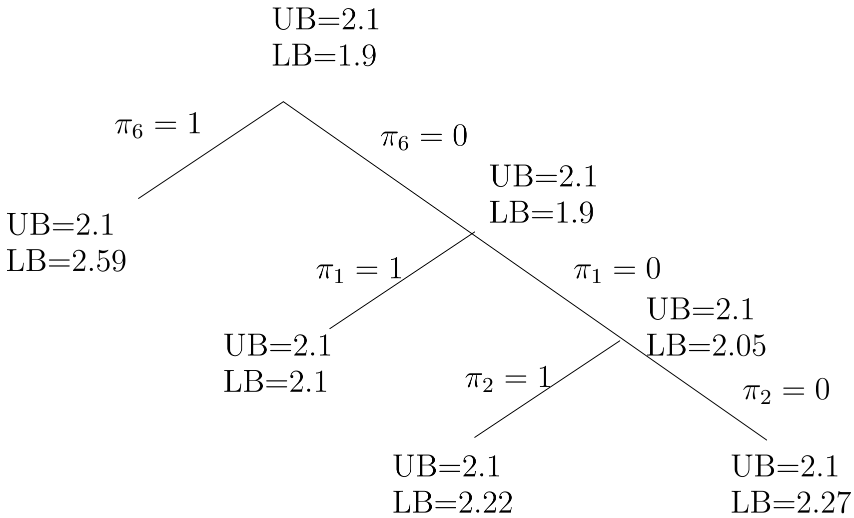


Figure 2.6: B& B tree for the network of Figure 2.5

2.5 Finding optimal locations of facilities and depots

If the set of facilities is given, the problem of optimally locating collection depots is the same as the problem we studied in Section 2. It is so since when we exchange the role of facilities and collection depots, the two problems are exactly the same.

In this section, we consider the case when both facilities and collection depots are decision variables, so we need to find the optimal location of m collection depots and p facilities simultaneously. The minisum problem can be written as

$$\min_{X,Y\in G} f(X,Y) = \sum_{i=1}^{n} \omega_i \min_{1\le j\le p} \min_{1\le k\le m} \{d(y_j,i) + d(i,x_k) + d(x_k,y_j)\}$$

Lemma 2.2 *If $m \ge p$, the problem is a p-median problem.*

Proof. Suppose $(X^\star, Y^\star) = (x_1^\star, \cdots, x_m^\star, y_1^\star, \cdots, y_p^\star)$ is an optimal solution. We will show that $(Y^\star, Y^\star)$ is always at least as good as $(X^\star, Y^\star)$.

$$\begin{aligned} f(X^\star, Y^\star) =& \sum_{i=1}^{n} \omega_i \min_{1\le j\le p} \min_{1\le k\le m} \{d(i,y_j^\star) + d(y_j^\star, x_k^\star) + d(x_k^\star, i)\} \\ \ge& \sum_{i=1}^{n} \omega_i \min_{1\le j\le p} \{2d(i,y_j^\star)\} \\ =& 2\sum_{i=1}^{n} \omega_i \min_{1\le j\le p} \{d(i,y_j^\star)\} = f(Y^\star, Y^\star), \end{aligned}$$

which is the objective function of the p-median problem. ■

Lemma 2.3 *If $m < p$, the problem is an m-median problem.*

Proof.

$$\begin{aligned} \min_{X,Y\in G} f(X,Y) =& \min_{X\in G} \min_{Y\in G} f(X,Y) \\ =& \min_{X\in G} \sum_{i=1}^{n} \omega_i \min_{1\le j\le m} \{2d(x_j,i)\} \\ =& \min_{X\in G} f(X,X) \end{aligned}$$

■

Combining Lemma 2.2 and Lemma 2.3, we have following theorem:

Theorem 2.4 *When both collection depots and facilities are decision variables, the minisum problem is equivalent to* $\boldsymbol{\min\{m,p\}}$*-median problem and the optimal solution is to co-locate* $\boldsymbol{\min\{m,p\}}$ *facilities and depots.*

There has been an extensive research on the p-median problem. We refer the reader to Mirchandani (1990).

2.6 Computational experiments

Procedure-RELAX and *Procedure*-B&B, were coded in ANSI C. The problem data used in the experiments were generated randomly as follows. The Cartesian coordinates of the nodes were generated over the interval $(0, 100)$ according to the uniform distribution. Then nodes were connected randomly until a tree was formed. Finally, a random number of links were added to the generated tree. Demand weights were generated over the interval (0,1) randomly. Let the first m nodes be collection depots. The length of each link was calculated using the Euclidean distance formula.

We ran the experiments on a Dell Optiplex GX110 machine equipped with 677Mhz processor and 128M RAM. Before calling *Procedure*-B&B, iteration limit of *Procedure*-RELAX was set at 400 and the scalar α in (2.15) was halved whenever the lower bound did not improve in 20 consecutive iterations. If the lower bound failed to improve in 50 consecutive iterations, we terminated *Procedure*-RELAX before the iteration limit was reached. If

Procedure-RELAX did not converge, we called *Procedure*-B&B, and set iteration limit of *Procedure*-RELAX at 50 and halved the scalar α when the lower bound did not improve in 5 iterations. If the lower bound failed to improve in 10 consecutive iterations, we terminated *Procedure*-RELAX before the iteration limit 50 was reached. We note that whenever we branch out from a node, we can use the lower bound of that node as a starting point for calculating the lower bounds of its children and save calculation time since the lower bounds can not be decreasing. We let the algorithm memorize the lower bound at each node of the tree. This led to a reduction in computational time since the algorithm can either find an improved lower bound, or terminate the subproblem and go further very quickly.

In Table 2.4, we generate 50 instances For each triplet of n, m and p. Each line reports average Lagrangian duality gap, average relative error of the Lagrangian solution, and average CPU time over the 50 instances for each triplet of n, m and p for our algorithm and CPLEX 8.1, where Lagrangian duality gap is the gap between the upper and lower bounds returned by Lagrangian Relaxation without calling *Procedure*-B&B.

The following conclusions can be drawn from the computational results of Table 2.4.

- Our algorithm results in a smaller CPU time in comparison with CPLEX for large p, while CPLEX is much faster for small p.
- Even though the duality gap is large for some cases, Lagrangian relaxation always gives a near-optimal solution (relative error is less than 1 percent).

In Table 2.5, we examine large networks. We set a time limit of 30 minutes for both our algorithm and CPLEX and generate 10 instances for each triplet of n, m and p. The computational results indicate that the performance of CPLEX and our algorithm are quite close, but CPLEX can not solve large dimensions problems. On the other hand, even though our branch-and-bound algorithm can handle the problems with large dimensions, the solution time is very large. In this circumstances, we suggest using Lagrangian Relaxation without branch-and-bound due to its good performance on medium-sized problems.

Table 2.4: Computational results for medium-sized problems

n	m	p	duality gap[1]	relative error[2]	B&B time[3]	CPLEX time[4]
30	10	1	0.258828	0	0.43098	1.09922
		2	0.175361	0.00092	1.60378	1.0966
		3	0.105051	0.000017	1.9627	1.11628
		4	0.052554	0.00081	1.43392	1.00424
		5	0.040723	0.000018	1.22936	0.9811
		6	0.019763	0.001798	0.75438	0.97476
		7	0.023537	0.009719	0.41268	0.95744
		8	0.004376	0.000711	0.12782	0.94222
		9	0.00073	0	0.013	0.93902
50	10	1	0.234127	0	1.68066	4.43974
		2	0.143277	0.000084	9.14218	4.57956
		3	0.086714	0.000036	10.68602	3.78762
		4	0.043304	0.000056	9.56948	3.59054
		5	0.024895	0	5.48264	3.34728
		6	0.015159	0.001892	3.8638	3.375
		7	0.006911	0.001081	1.43532	3.11618
		8	0.002282	0.000321	0.71304	3.02228
		9	0.003354	0.001388	0.1554	3.09234
100	10	1	0.2453	0	42.2684	23.82502
		2	0.133864	0.001459	159.7633	32.85602
		3	0.069529	0.000003	158.7602	27.29954
		4	0.03588	0	90.01192	23.24306
		5	0.017922	0	36.55608	20.8197
		6	0.013258	0.001977	50.6938	19.72216
		7	0.005605	0.001751	9.72872	18.75854
		8	0.0024	0	2.7591	17.70294
		9	0.002875	0.001757	0.53568	17.05226

1=(Lagrangian solution value-lower bound)/lower bound
2=(Lagrangian solution value-optimal value)/optimal value
3,4: CPU time in seconds (I/O time excluded)

Table 2.5: Computational results for large scale problems

n	m	p	duality gap	relative error [5]	Relaxation Time [6]	Time-limited B&B time [7]	successful # [8]	Time-limited CPLEX time [9]	successful # [10]
150	10	4	0.03349	0	4.79	265.36	9	95.02	10
		5	0.01841	0	4.28	290.39	10	85.48	10
		6	0.00561	0	3.02	116.77	10	98.81	10
200	10	4	0.03448	0	8.33	265.45	6	353.82	8
		5	0.01822	0	8.58	303.79	9	300.44	9
		6	0.00474	0.0006	5.79	90.16	10	249.39	10
300	10	4	0.04239	0	19.79	511.43	3	*	0
		5	0.01873	0	20.01	41.50	6	*	0
		6	0.01886	0.0068	20.69	200.15	6	*	0
500	10	4	0.03067	0	60.59	145.82	6	*	0
		5	0.01817	0	61.07	938.19	2	*	0
		6	0.01231	0	50.44	269.10	4	*	0
1000	10	4	0.04517	0	229.78	805.43	2	*	0
		5	0.02183	0	244.97	646.65	3	*	0
		6	0.01049	0.0005	197.89	53.27	2	*	0

5: $\dfrac{\text{Lagrangian solution value} - \text{best value with the time-limited B\&B and CPLEX}}{\text{best value with the time-limited B\&B and CPLEX}}$

6,7,9: CPU time in seconds (I/O time excluded)

6: average Lagrangian Relaxation time of 10 cases without branch-and-bound

7,9: average time of successful cases

8,10: number of cases finished within 1800 seconds

*: CPLEX run out of memory

Chapter 3

The Minisum Multi-purpose Trip Location Problem

3.1 Introduction

Multi-purpose trips are quite common in practice. For example, on a typical day, a person on the way from work may buy some food from a supermarket and pick up his or her child from the day-care center.

Although there has been an extensive research on multi-purpose trips, e.g. McLafferty and Ghosh (1986), Mulligan (1987), and Thill (1992), very few studies have considered the location of facilities in the presence of multi-purpose trips. Suzuki and Hodgson (2002) addressed the problem for two different types of facilities in multi-purpose trips, however, they only presented a non-linear integer programming formulation and its linearization for the problem. The properties of locations were not given enough attention by them.

Following Suzuki and Hodgson (2002), in this chapter, we investigate the multi-purpose trip location problem on networks with two different types of facilities and three types of user groups: two of them only need one type of service; the third one needs both types of service in a single trip. The objective is to minimize the total weighted travel distance of all trips. There are three types of trips in the system corresponding to three types of customers. If all trips are multi-purpose, i.e., all customers need both types of service in a single trip, the problem investigated in this chapter is related to the Collection Depots Location Problem with multiple facilities (Berman et al 2002 and the previous chapter). However, in the Collection Depots Location Problem, locations of depots are pre-specified, while locations of both types of facilities are decision variables in this chapter.

The model we present has a number of potential applications. For example, Sears Canada operates several apparel and hardware stores in most cities of Canada. Obviously, some customers only want to buy clothes while others only need some hardware. When planning where to locate the two types of facilities Sears Canada can take into account those customers that may need both hardware and clothes at the same time. Since customers can earn some reward points whenever they shop at Sears, the third type of customers is quite common. The model can be of great benefit for companies that are not in competition and may want to plan together their location of facilities. An example is book stores and coffee shops. By planning their location decisions together book store companies and coffee shop companies can take advantage of many customers that may need to visit both types of facilities. Co-location of these type of facilities that occur in our model can explain the co-location of coffee shops and book stores that are becoming very popular.

The problem studied in this chapter is somewhat related to two other problems studied in the literature: the problem suggested in Goldman (1969) which was proved to posses nodal optimality independently by Hakimi and Maheshwari (1972) and by Wendell and Hurter (1973) and the 2-Level Uncapacitated Facility Location Problem (2-UFLP. See Zhang, 2004 for the discussion and literature review). In Goldman's problem, the nodes of the network include sources (supply points) and destinations (demand points). For each source-destination pair, there are multiple types of demands (commodities) and each demand may require several stages of processing, each taking place at a different facility. It is assumed that each facility is capable of performing any of the processing stages on any type of demand. In other words, only a single type of facility is located on the network while our problem deals with different types of facilities simultaneously.

Similar to our problem, in the 2-UFLP, there are two types of facilities to be located on the network: depots and transit stations. Demands must be routed from a depot through a transit station to clients. The 2-UFLP seeks to determine where to open facilities and how to assign clients to paths along open facilities such that the total cost of opening facilities and serving clients is minimized. The 2-UFLP is different from our problem in several important respects. First, there is only one type of demands in the 2-UFLP, while there are three types of demands in our problem. Second, the two types of facilities are hierarchical in the

2-UFLP, but hierarchical order is not necessary in our problem. Third, in our problem a round trip travel is considered whereas in the 2-UFLP travel from depots to clients through transit points is considered. We refer the reader to Zhang (2004) for the 2-UFLP and its relevant references.

Our model is formally introduced in Section 2. We prove that there exists an optimal nodal solution and develop a linear integer programming formulation, which is different from the formulation given by Suzuki and Hodgson (2002). In Sections 3 and Section 4, we study the problem of locating a single facility for each type of server on a tree and on a path. In Section 5, we turn our attention to developing efficient computational procedures for the problem on general networks. Due to the complexity of the problem, the "direct approach" of applying a commercial software to the integer linear programming formulation of Section 2 is not practical even for medium-sized problems. To overcome this difficulty, we present several heuristic procedures, ranging from the simple greedy heuristic, to a heuristic procedure based on solving two median problems to meta-heuristics (Tabu Search and Variable Neighborhood Search). Computational results are presented in Section 6. These results indicate that the Tabu Search heuristic tend to outperform other procedures in terms of solution quality, but incurring large running time. On the other hand, the Median-based heuristic, which we recommend for large scale problems, runs very fast and is only slightly inferior to the Tabu Search in terms of solution quality.

3.2 Location on a general network

Consider the situation where there are two different types of facilities, denoted by type-x and type-y facilities, in the system. Customer demand occurs only at the nodes of the network $G = (N, L)$, where N is the set of nodes with $|N| = n$ and L is the set of links. Customers in the system have to travel to a facility to obtain service, e.g., shopping or fuelling. There are three user groups in the system: type-x (type-y) customers who only need service from type-x (type-y) facilities and type-xy customers who need service from both types of facilities.

The objective is to minimize the total travel distance of the system through locating m type-x facilities and p type-y facilities on the network.

Let ω_i^x, ω_i^y and ω_i^{xy} be respectively the number (or fractions) of customers of type-x, type-y and type-xy in a typical day at node i. Let the set of locations of type-x facilities be $X = (x_1, \cdots, x_m)$ and that of type-y facilities be $Y = (y_1, \cdots, y_p)$. Let $d(u, v)$ be the shortest distance between any two points u and v on the network. Without loss of generality, we assume that each type-xy customer will always patronize type-x facility first, then type-y facility. The problem can be formulated as

$$\begin{aligned}\min_{X,Y\in G} f(X,Y) =& 2\sum_{i=1}^{n}\omega_i^x \min_{1\le j\le m}\{d(i,x_j)\} + 2\sum_{i=1}^{n}\omega_i^y \min_{i\le k\le p}\{d(i,y_k)\}\\ &+\sum_{i=1}^{n}\omega_i^{xy}\min_{1\le j\le m}\min_{1\le k\le p}\{d(i,x_j)+d(x_j,y_k)+d(y_k,i)\}\end{aligned}$$

Theorem 3.1 *An optimal location to the problem exists in the set N.*

Proof. Suppose that in an optimal solution $(X^\star, Y^\star) = (x_1^\star, \cdots, x_m^\star, y_1^\star, \cdots, y_p^\star)$ and without any loss of generality, $x_m^\star$ is an internal point on link (a, b). We will prove that by changing the location $x_m^\star$ to one of the two nodes a and b, the expected travel time cannot increase. We will show it in two steps.

If we consider each non-nodal $y_j^\star$ as a node with weights 0 for all three types of customers, we can replace G with a new graph $G' = (N', L')$, where $N' = N \cup Y^\star$. It is easy to verify that the objective functions of the problem in G and G' are the same. Now suppose that $x_m^\star$ is an internal point on link (a', b') of G'. Note that $a'(b')$ may be unequal to $a(b)$. Our first step is to show that $x_m^\star$ is dominated by either a' or b'.

Let $N'(x_m^\star)$ be the set of nodes in G' that are served by the facility located at $x_m^\star$ for those type-x customers only and let $\bar{N}'(x_m^\star)$ be the set that includes the rest of nodes for those type-x customers (ties can be broken arbitrarily). Let $N'(x_m^\star, y_k^\star)$ $(k = 1, \cdots, p)$ be the set of nodes that are served successively by the facilities located at $x_m^\star$ and $y_k^\star$ for type-xy customers. Also, let $\bar{N}'(x_m^\star, Y^\star)$ be the set that includes the rest of nodes for type-xy customers (ties can be broken arbitrarily too), i.e., $\bar{N}'(x_m^\star, Y^\star) = N' - \cup_{k=1}^p N'(x_m^\star, y_k^\star)$.

Let $X_{m-1}^\star = (x_1^\star, \cdots, x_{m-1}^\star)$. Now $f(X^\star, Y^\star)$ can be expressed as

$$\begin{aligned}
&2 \sum_{i \in N'(x_m^\star)} \omega_i^x d(i, x_m^\star) + 2 \sum_{i \in \bar{N}'(x_m^\star)} \omega_i^x d(i, X_{m-1}^\star) + 2 \sum_{i=1}^{n} \omega_i^y d(i, Y^\star) \\
&+ \sum_{i \in N'(x_m^\star, y_1^\star)} \omega_i^{xy} [d(i, x_m^\star) + d(x_m^\star, y_1^\star) + d(y_1^\star, i)] + \cdots \\
&+ \sum_{i \in N'(x_m^\star, y_p^\star)} \omega_i^{xy} [d(i, x_m^\star) + d(x_m^\star, y_p^\star) + d(y_p^\star, i)]
\end{aligned}$$

$$+\sum_{i\in \bar{N}'(x_m^\star,Y^\star)} \omega_i^{xy} \min_{1\le j\le m-1}\min_{1\le k\le p}\{d(i,x_j^\star)+d(x_j^\star,y_k^\star)+d(y_k^\star,i)\},$$

where

$$d(i,X_{m-1}^\star)=\min_{1\le j\le m-1}\{d(i,x_j^\star\},\qquad d(i,Y^\star)=\min_{1\le k\le p}\{d(i,y_k^\star\}$$

Let $A(x_m^\star)$ ($B(x_m^\star)$) be the set of nodes in $N'(x_m^\star)$ that reach $x_m^\star$ most efficiently via a' (b'). For any $k\in\{1,\cdots,p\}$, let $AA(x_m^\star,y_k^\star)$ ($BB(x_m^\star,y_k^\star)$) be the set of nodes in $N'(x_m^\star,y_k^\star)$ that reach $x_m^\star$ and $y_k^\star$ most efficiently not via $b'(a')$. Note that a customer in $AA(x_m^\star,y_k^\star)(BB(x_m^\star,y_k^\star))$ may pass through $a'(b')$ more than two times. Let $AB(x_m^\star,y_k^\star)$ ($BA(x_m^\star,y_k^\star)$) be the set of nodes in $N'(x_m^\star,y_k^\star)$ that reach $x_m^\star$ and $y_k^\star$ most efficiently via a' (b') first, then via b' (a'). Ties are broken arbitrarily. Note that a customer in $AB(x_m^\star,y_k^\star)$ ($BA(x_m^\star,y_k^\star)$) may pass through $a'(b')$ two times.

We denote by l' the length of link (a',b'). Assume the distance from a' to $x_m^\star$ is x. For those type-x customers on $i\in A(x_m^\star)$, the shortest travel distance is $2(d(i,a')+x)$, whereas for any $i\in B(x_m^\star)$, the shortest travel distance is $2(d(i,b')+l'-x)$. For those type-xy customers on $i\in AA(x_m^\star,y_k^\star)$, the shortest travel distance is $d(i,a')+2x+d(a',y_k^\star)+d(y_k^\star,i)$, whereas for any $i\in BB(x_m^\star,y_k^\star)$, the shortest travel distance is $d(i,b')+2l'-2x+d(b',y_k^\star)+d(y_k^\star,i)$. Similarly, for any $i\in AB(x_m^\star,y_k^\star)$, the shortest travel distance is $d(i,a')+l'+d(b',y_k^\star)+d(y_k^\star,i)$, whereas for any $i\in BA(x_m^\star,y_k^\star)$, the shortest travel distance is $d(i,b')+l'+d(a',y_k^\star)+d(y_k^\star,i)$. Therefore $f(X^\star,Y^\star)$ can be written as

$$2\sum_{i\in A(x_m^\star)}\omega_i^x[d(i,a')+x]+2\sum_{i\in B(x_m^\star)}\omega_i^x[d(i,a')+l'-x]$$

$$+\sum_{k=1}^{p}\sum_{i\in AA(x_m^\star,y_k^\star)}\omega_i^{xy}[d(i,a')+2x+d(a',y_k^\star)+d(y_k^\star,i)]$$

$$+\sum_{k=1}^{p}\sum_{i\in BB(x_m^\star,y_k^\star)}\omega_i^{xy}[d(i,b')+2l'-2x+d(b',y_k^\star)+d(y_k^\star,i)]+C_1$$

$$=2[\sum_{i\in A(x_m^\star)}\omega_i^x-\sum_{i\in B(x_m^\star)}\omega_i^x+\sum_{k=1}^{p}\sum_{i\in AA(x_m^\star,y_k^\star)}\omega_i^{xy}-\sum_{k=1}^{p}\sum_{i\in BB(x_m^\star,y_k^\star)}\omega_i^{xy}]x+C_1+C_2$$

where C_1 and C_2 are the terms that do not include x.

Let us assume without loss of generality that

$$\sum_{i\in A(x_m^\star)}\omega_i^x-\sum_{i\in B(x_m^\star)}\omega_i^x+\sum_{k=1}^{p}\sum_{i\in AA(x_m^\star,y_k^\star)}\omega_i^{xy}-\sum_{k=1}^{p}\sum_{i\in BB(x_m^\star,y_k^\star)}\omega_i^{xy}\geq 0$$

Now we can conclude that by taking $x_m^\star$ to node a', the weighted travel distance cannot increase even without changing the assignment of nodes to facilities. Therefore, given $Y^\star$, $N\cup Y^\star$ is a dominating location set for $X^\star$.

The second step of the proof is to show that either a or b is better than a' if $a'\neq a$. If $a'\neq a$, then point a' is a joint location of $x_m^\star$ and a type-y facility, say $y_p^\star$.

Let $N(x_m^\star)$ ($N(y_p^\star)$) be the set of nodes in G that are served by the facility located at a' for those type-x (type-y) customers only and let $\bar{N}(x_m^\star)$ ($\bar{N}(y_p^\star)$) be the set that includes the rest of nodes for those type-x (type-y) customers (ties can be broken arbitrarily). Let $N(x_m^\star,y_p^\star)$ be the set of nodes that are served by facilities $x_m^\star$ and $y_p^\star$ jointly located at a' for type-xy customers. Since $x_m^\star$ and $y_p^\star$ are located at the same point a', there is no advantage to patronize other facilities for type-xy customers if they visit a'. Also, we have $d(i,x_m^\star)+d(x_m^\star,y_p^\star)+d(y_p^\star,i)=2d(i,a')$. Let $\bar{N}(x_m^\star,y_p^\star)$ be the set that includes the rest of nodes for type-xy customers (ties can be broken arbitrarily too), i.e., $\bar{N}(x_m^\star,y_p^\star)=N-N(x_m^\star,y_p^\star)$.

Let $Y^\star_{p-1} = (y^\star_1, \cdots, y^\star_{p-1})$. Now $f(X^\star, Y^\star)$ can be expressed as

$$2\sum_{i\in N(x^\star_m)} \omega^x_i d(i, x^\star_m) + 2\sum_{i\in \bar{N}(x^\star_m)} \omega^x_i d(i, X^\star_{m-1}) + 2\sum_{i\in N(y^\star_p)} \omega^y_i d(i, y^\star_p) + 2\sum_{i\in \bar{N}(y^\star_p)} \omega^y_i d(i, Y^\star_{p-1})$$

$$+ 2\sum_{i\in N(x^\star_m, y^\star_p)} \omega^{xy}_i d(i, x^\star_m) + \sum_{i\in \bar{N}(x^\star_m, y^\star_p)} \omega^{xy}_i \min_{1\le j\le m-1} \min_{1\le k\le p-1} \{d(i, x^\star_j) + d(x^\star_j, y^\star_k) + d(y^\star_k, i)\},$$

Define $A(x^\star_m, y^\star_p)(B(x^\star_m, y^\star_p))$ as the set of nodes in $N(x^\star_m, y^\star_p)$ that reach $x^\star_m$ and $y^\star_p$ most efficiently via $a(b)$. We also define $A(x^\star_m)$, $B(x^\star_m)$, $A(y^\star_p)$ and $B(y^\star_p)$ as before. Denote by l the length of link (a, b). Assume the distance from a to a' is y. We can write $f(X^\star, Y^\star)$ as

$$2\sum_{i\in A(x^\star_m)} \omega^x_i [d(i, a) + y] + 2\sum_{i\in B(x^\star_m)} \omega^x_i [d(i, b) + l - y]$$

$$+ 2\sum_{i\in A(y^\star_p)} \omega^y_i [d(i, a) + y] + 2\sum_{i\in B(y^\star_p)} \omega^y_i [d(i, b) + l - y]$$

$$+ 2\sum_{i\in A(x^\star_m, y^\star_p)} \omega^{xy}_i [d(i, a) + y] + 2\sum_{i\in B(x^\star_m, y^\star_p)} \omega^{xy}_i [d(i, b) + l - y] + D_1$$

$$= 2[\sum_{i\in A(x^\star_m)} \omega^x_i - \sum_{i\in B(x^\star_m)} \omega^x_i + \sum_{i\in A(y^\star_p)} \omega^y_i - \sum_{i\in B(y^\star_p)} \omega^y_i + \sum_{i\in A(x^\star_m, y^\star_p)} \omega^{xy}_i - \sum_{i\in B(x^\star_m, y^\star_p)} \omega^{xy}_i]y + D_1 + D_2$$

where D_1 and D_2 are the terms that do not include y.

Now it is easy to see that if

$$\sum_{i\in A(x^\star_m)} \omega^x_i - \sum_{i\in B(x^\star_m)} \omega^x_i + \sum_{i\in A(y^\star_p)} \omega^y_i - \sum_{i\in B(y^\star_p)} \omega^y_i + \sum_{i\in A(x^\star_m, y^\star_p)} \omega^{xy}_i - \sum_{i\in B(x^\star_m, y^\star_p)} \omega^{xy}_i \ge 0,$$

the objective function value can't increase by taking $x^\star_m$ and $y^\star_p$ from a' to a; Otherwise, moving $x^\star_m$ and $y^\star_p$ to b simultaneously.

The same process can be performed for any other non-nodal location, which proves the theorem. ■

We are now ready to formulate the Minisum Multi-purpose Trip Location problem on general networks as an integer linear program. The decision variables are:

$$x_{ij} = \begin{cases} 1 & \text{If type-}x\text{ customers at node } i \text{ are served by a type-}x\text{ facility at node } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{If type-}y\text{ customers at node } i \text{ are served by a type-}y\text{ facility at node } k, \\ 0 & \text{otherwise,} \end{cases}$$

$$z_{ijk} = \begin{cases} 1 & \text{If type-}xy\text{ customers at node } i \text{ are served by a type-}x\text{ facility at node } j \\ & \text{and a type-}y\text{ facility at node } k, \\ 0 & \text{otherwise.} \end{cases}$$

$$\pi_j^x = \begin{cases} 1 & \text{If a type-}x\text{ facility is located at } j \in N, \\ 0 & \text{otherwise.} \end{cases}$$

$$\pi_k^y = \begin{cases} 1 & \text{If a type-}y\text{ facility is located at } k \in N, \\ 0 & \text{otherwise.} \end{cases}$$

Let $d_{ijk} = d(i,j) + d(j,k) + d(k,i)$. According to Theorem 3.1, the problem can be formulated as:

$$\min Z = 2\sum_{i \in N}\sum_{j \in N}\omega_i^x d_{ij} x_{ij} + 2\sum_{i \in N}\sum_{k \in N}\omega_i^y d_{ik} y_{ik} + \sum_{i \in N}\sum_{j \in N}\sum_{k \in N}\omega_i^{xy} d_{ijk} z_{ijk}$$

subject to

$$x_{ij} \leq \pi_j^x \quad i \in N,\ j \in N \tag{3.1}$$

$$\sum_{j \in N} x_{ij} = 1 \quad i \in N \tag{3.2}$$

$$y_{ik} \leq \pi_k^y \quad i \in N,\ k \in N \tag{3.3}$$

$$\sum_{k \in N} y_{ik} = 1 \quad i \in N \tag{3.4}$$

$$z_{ijk} \leq \pi_j^x \quad i \in N,\ j \in N,\ k \in N \tag{3.5}$$

$$z_{ijk} \leq \pi_k^y \quad i \in N,\ j \in N,\ k \in N \tag{3.6}$$

$$\sum_{j \in N} \sum_{k \in N} z_{ijk} = 1 \qquad i \in N \tag{3.7}$$

$$\sum_{j \in N} \pi_j^x = m \tag{3.8}$$

$$\sum_{k \in N} \pi_k^y = p \tag{3.9}$$

$$x_{ij},\ y_{ik},\ z_{ijk}, \pi_j^x,\ \pi_k^y \in \{0, 1\} \qquad i \in N,\ j \in N,\ k \in N \tag{3.10}$$

Constraint set (3.1) ensures the assignment of type-x demand (customers at a node) to an open type-x facility. Constraint set (3.2) states that each type-x demand is assigned to one open type-x facility. Constraint sets (3.3) and (3.4) have the same meaning for type-y facility as Constraint sets (3.1) and (3.2) respectively. Constraint sets (3.5) and (3.6) assign type-xy demand only to a pair of an open type-x facility and an open type-y facility. Constraint set (3.7) states that each type-xy demand node can only be assigned to one pair of type-x and type-y facilities. Constraints (3.8) and (3.9) specify exactly m type-x facilities and p type-y facilities respectively. Finally, Constraint (3.10) establishes a binary restriction on the decision variables.

Suzuki and Hodgson (2002) performed experiments to show that unequal number of type-x and type-y facilities is problematic. They also showed that when all demands are multi-purpose and that when $m = p$, jointly located facilities will always be better than separate ones. In fact, when all trips are multi-purpose, facilities are jointly located and there is no advantage to locate unequal number of type-x and type-y facilities as proved by following theorem.

Theorem 3.2 *If there are only type-xy customers in the network, i.e., $\omega_i^x = \omega_i^y = 0$ for all $i \in N$, there is no advantage to locate unequal number of type-x and type-y facilities, and our problem is reduced to a* $\min\{m, p\}$*-median problem; all type-x and type-y facilities are co-located in the optimal solution.*

Proof. Refer to Theorem 2.4. ■

In this section, we have proved that there exists an optimal nodal solution set. Due to the $\mathcal{NP}$-completeness of the p-median problem (Kariv and Hakimi, 1979), the minisum multi-purpose trip location problem is $\mathcal{NP}$-complete as well. We therefore focus on heuristics (presented later) to solve our problem. In the next two sections we study the problem on some simple networks (tree and path) when $m = p = 1$.

3.3 Location on a tree when $m = p = 1$

In this section, we restrict our attention to siting a single facility for each type of server on tree networks. When $m = p = 1$, the expected travel distance can be written as

$$\begin{aligned} & f(x_1, y_1) \\ &= 2\sum_{i=1}^{n} \omega_i^x d(i, x_1) + 2\sum_{i=1}^{n} \omega_i^y d(i, y_1) + \sum_{i=1}^{n} \omega_i^{xy}[d(i, x_1) + d(x_1, y_1) + d(y_1, i)] \\ &= \sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i, x_1) + \sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy})d(i, y_1) + \sum_{i=1}^{n}\omega_i^{xy}d(x_1, y_1) \end{aligned} \tag{3.11}$$

Using the following algorithm, which we refer to as the Dominating Subtree Algorithm, we can generate a new tree, which we call $T_x(N_x, L_x)$ (or T_x for simplicity). This new tree will be used in solving the problem.

Dominating Subtree Algorithm

Step 1. Set $F := \emptyset$.

Step 2. Create a new tree, T'_x, which has the same nodes and links as the original tree T. Demand rate at node i of T'_x is $\omega'_i = 2\omega_i^x + \omega_i^{xy}$. Set $T_x := T'_x$.

Step 3. Find a leaf, say node i, in tree T_x such that $i \notin F$. If no such i exists, stop. If $\omega'_i \leq \sum_{k=1}^{n} \omega_k^x$, go to Step 4. Otherwise go to Step 5.

Step 4. Modify the tree T_x by deleting leaf i and link (i, j), where node j is the unique node that is incident to i in T_x, and by increasing the weight of node j, ω'_j by ω'_i. Go back to Step 3.

Step 5. Set $F := F \cup \{i\}$. Go back to Step 3.

Similarly, we can create a new tree, $T_y(N_y, L_y)$, by replacing every ω_i^x with ω_i^y in the previous algorithm. As an example consider the 9-node tree depicted in Figure 3.1 where the numbers next to the nodes are ω_i^x, ω_i^y and ω_i^{xy} respectively. According to Step 1 of the proceeding algorithm, we have T_x' and T_y', which are depicted in Figure 3.2 and Figure 3.3. Now let us consider the tree T_x, which was set as T_x' at the end of Step 2. We have $\sum \omega_i^x = 1+5+3+4+1+4+5+1+2 = 26$. At the first iteration of Step 3, we choose, for example, node 4. Since the demand weight at node 4 is 10, which is less than 26, we delete node 4 and link $(4, 2)$, and change the weight of node 2 to $21(11+10)$. After several iterations, we can obtain T_x depicted in the left part of Figure 3.4. Similarly, we can obtain T_y, which is depicted in the right part of Figure 3.4.

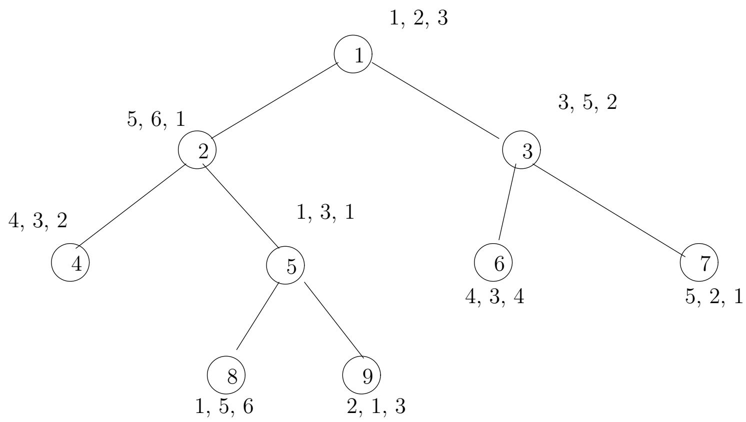


Figure 3.1: A 9-node tree

Lemma 3.1 *Suppose $T_x(N_x, L_x)$ and $T_y(N_y, L_y)$ are the two subtrees of $T(N, L)$ generated*

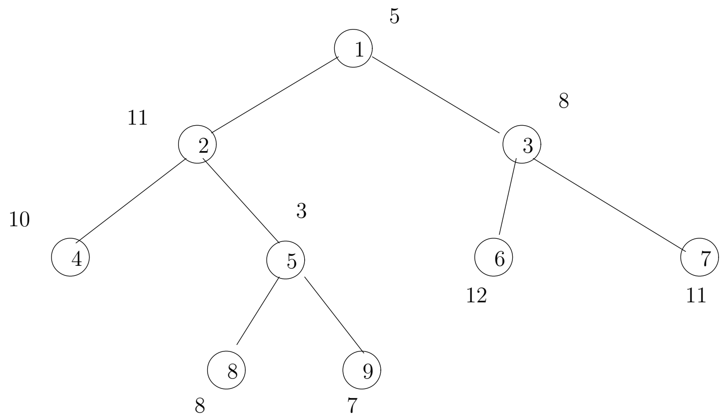


Figure 3.2: T'_x of Figure 3.1

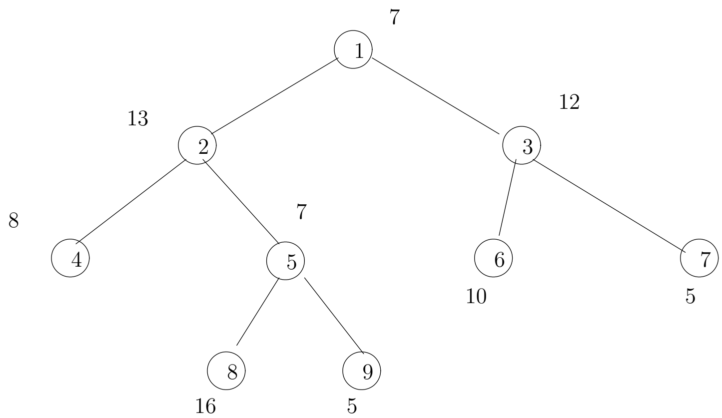


Figure 3.3: T'_y of Figure 3.1

by the Dominating Subtree Algorithm. The optimal solution $(x_1^\star, y_1^\star)$ *satisfies the condition that* $x_1^\star \in N_x$ *and* $y_1^\star \in N_y$*.*

Proof. Suppose that (u, y_1) is a solution such that $u \notin N_x$. Let node s be the closest node to u in N_x. Denote by t the node which is adjacent to node s on the path from u to s (it is possible that $t = u$). It is easy to verify that t is unique and that $t \notin N_x$. Now we partition T into two mutually exclusive subtrees, $T_s(N_s, L_s)$ and $T_t(N_t, L_t)$ as depicted in

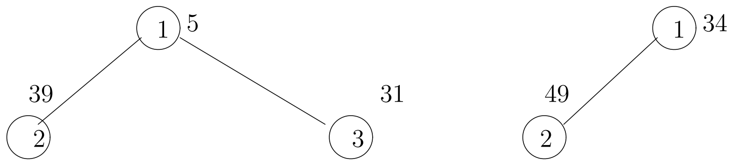


Figure 3.4: T_x(left part) and T_y(right part) of Figure 3.1

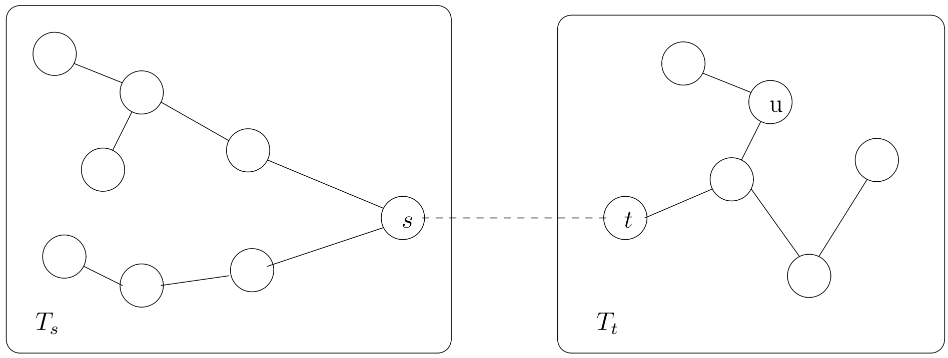


Figure 3.5: A tree network

Figure 5.3 ($N_s \cup N_t = N$). Obviously, we have that $T_x \cap T_t = \emptyset$ and that $T_x \subseteq T_s$.

We will show that (s, y_1) is at least as good as (u, y_1) for any $y_1 \in N$. According to (3.11), we have

$$\begin{aligned}
& f(s,y_1) - f(u,y_1) \\
&= \sum_{i=1}^{n} (2\omega_i^x + \omega_i^{xy})[d(i,s) - d(i,u)] + \sum_{i=1}^{n} \omega_i^{xy}[d(s,y_1) - d(u,y_1)] \\
&= \sum_{i \in N_t \cup N_s} (2\omega_i^x + \omega_i^{xy})[d(i,s) - d(i,u)] + \sum_{i=1}^{n} \omega_i^{xy}[d(s,y_1) - d(u,y_1)] \\
&= \sum_{i \in N_t} (2\omega_i^x + \omega_i^{xy})[d(i,s) - d(i,u)] - \sum_{i \in N_s} (2\omega_i^x + \omega_i^{xy}) d(s,u) + \sum_{i=1}^{n} \omega_i^{xy}[d(s,y_1) - d(u,y_1)]
\end{aligned}$$

$$\leq \sum_{i \in N_t} (2\omega_i^x + \omega_i^{xy}) d(s,u) - \sum_{i \in N_s} (2\omega_i^x + \omega_i^{xy}) d(s,u) + \sum_{i=1}^{n} \omega_i^{xy} d(s,u) \qquad \text{(triangle inequality)}$$

Subtracting and adding the first term simultaneously we obtain:

$$\begin{aligned} & f(s,y_1) - f(u,y_1) \\ \leq & 2\sum_{i\in N_t}(2\omega_i^x + \omega_i^{xy})d(s,u) - \sum_{i\in N_t\cup N_s}(2\omega_i^x + \omega_i^{xy})d(s,u) + \sum_{i=1}^{n}\omega_i^{xy}d(s,u) \\ = & 2[\sum_{i\in N_t}(2\omega_i^x + \omega_i^{xy}) - \sum_{i=1}^{n}\omega_i^x]d(s,u) \end{aligned}$$

Now from from the construction of T_x we know that $\sum_{i\in N_t}(2\omega_i^x + \omega_i^{xy}) \leq \sum_{i=1}^{n}\omega_i^x$. Therefore, $f(s,y_1) - f(u,y_1) \leq 0$, i.e., $x_1^\star \in N_x$. The same process can be performed to prove that $y_1^\star \in N_y$, which concludes the theorem. ■

Lemma 3.2 *Let*

$$\hat{x}_1 = \arg\min_{x_1\in G}\sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i,x_1)$$

and

$$\hat{y}_1 = \arg\min_{y_1\in G}\sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy})d(i,y_1),$$

i.e., $\hat{x}_1$ and $\hat{y}_1$ are 1-median respectively of T'_x and T'_y (defined in Step 1 of the Dominating Subtree Algorithm) respectively. If $(x_1^\star, y_1^\star)$ is an optimal solution, then $d(x_1^\star, y_1^\star) \leq d(\hat{x}_1, \hat{y}_1)$.

Proof. Based on the assumption of the lemma, we have

$$\sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i,\hat{x}_1) \leq \sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i,x_1^\star)$$

and

$$\sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy})d(i,\hat{y}_1) \leq \sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy})d(i,y_1^\star)$$

However, we have $f(\hat{x}_1, \hat{y}_1) \geq f(x_1^\star, y_1^\star)$, i.e.,

$$\sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i, \hat{x}_1) + \sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy})d(i, \hat{y}_1) + \sum_{i=1}^{n}\omega_i^{xy}d(\hat{x}_1, \hat{y}_1)$$
$$\geq \sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i, x_1^\star) + \sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy})d(i, y_1^\star) + \sum_{i=1}^{n}\omega_i^{xy}d(x_1^\star, y_1^\star)$$

Therefore, $d(x_1^\star, y_1^\star) \leq d(\hat{x}_1, \hat{y}_1)$. ■

Corollary 3.1 *If $d(\hat{x}_1, \hat{y}_1) = 0$, then $(x_1^\star, y_1^\star) = (\hat{x}_1, \hat{y}_1)$ and $x_1^\star = y_1^\star = \hat{x}_1 = \hat{y}_1$.*

Now we are ready to state the following theorem.

Theorem 3.3 *The optimal solution $(x_1^\star, y_1^\star)$ must be in the set $\{(a, b)|a \in N_x, b \in N_y, d(a, b) \leq d(\hat{x}_1, \hat{y}_1)\}$, where $\hat{x}_1 = \arg\min_{x_1 \in G} \sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i, x_1)$ and $\hat{y}_1 = \arg\min_{y_1 \in G} \sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy})d(i, y_1)$.*

To find $\hat{x}_1$ and $\hat{y}_1$, Goldman (1971) offers an $O(n)$ algorithm for a tree.

Consider the example of Figure 3.1 again. Applying Goldman's algorithm, it is easy to verify that $\hat{x}_1 = 2$, $\hat{y}_1 = 2$. According to Corollary 3.1, it is also the optimal solution of our problem.

It requires $O(n)$ effort to calculate the objective function value using (3.11). To find $\hat{x}_1$ and $\hat{y}_1$ requires $O(n)$ effort. Therefore, if we solve the problem through complete enumeration, the total complexity is $O(n^3)$. With the method we presented the worst case complexity is also $O(n^3)$, however, in most case it is more efficient. For example as we showed earlier, if $\hat{x}_1 = \hat{y}_1$, we can solve the problem in $O(n)$. If $\hat{x}_1$ and $\hat{y}_1$ are adjacent and all links are of unit length, it requires $O(n^2)$ effort to solve the problem using our method.

3.4 Location on a path when $m = p = 1$

A path is a special tree, so all the properties we presented in the last section hold also for a path network. We have shown that the optimal solution on a tree can be confined to a small subset of nodes when $m = p = 1$. In this section, we will show that we can decrease this subset even further when the tree is a path. We index the demand nodes by $1, 2, \cdots, n$ from left to right as depicted in Figure 4.2.

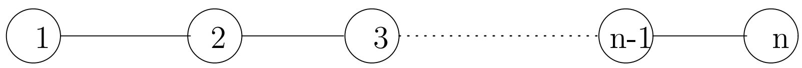


Figure 3.6: A path network

First, let us consider the path depicted in Figure 3.7 where the numbers next to the nodes are ω_i^x, ω_i^y and ω_i^{xy} respectively. All links are of unit length. Using Step 2 of the Dominating Subtree Algorithm and replacing each notation T(tree) with P(path), we construct P'_x and P'_y, which are depicted in Figure 3.8 and Figure 3.9. Path T_x, which we call P_x at the end of the Dominating Subtree Algorithm, is depicted in the left part of Figure 3.10, and P_y is depicted in the right part of Figure 3.10. It is easy to verify that $\hat{x}_1 = 6$ and $\hat{y}_1 = 7$. From Theorem 3.3, the optimal solution must be in $\{(4,5), (5,5), (5,6), (6,5), (6,6), (6,7)\}$.

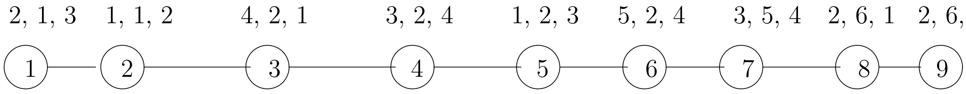


Figure 3.7: A 9-node path

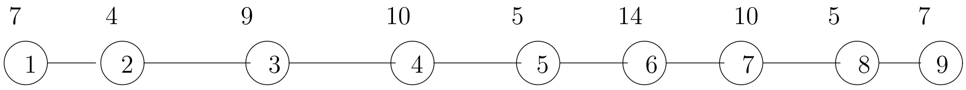


Figure 3.8: P'_x of Figure 3.7

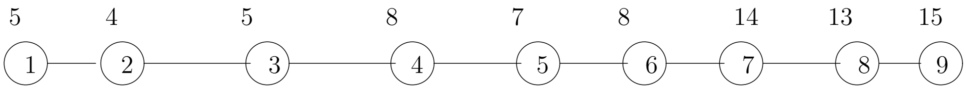


Figure 3.9: P'_y of Figure 3.7

Figure 3.10: P_x(left part) and P_y(right part) of Figure 3.7

To decrease the search further, we need results as follows:

Lemma 3.3 *The optimal solution* $(x_1^\star, y_1^\star)$ *satisfies the condition that* $x_1^\star \in [x_L, x_U] \cap N$ *and* $y_1^\star \in [y_L, y_U] \cap N$*, where*

$$x_L = \max\{x_1 | \sum_{i<x_1}(2\omega_i^x + \omega_i^{xy}) \leq \sum_{i=1}^{n} \omega_i^x\}, x_U = \min\{x_1 | \sum_{i<x_1+1}(2\omega_i^x + \omega_i^{xy}) \geq \sum_{i=1}^{n}(\omega_i^x + \omega_i^{xy})\}$$

$$y_L = \max\{y_1 | \sum_{i<y_1}(2\omega_i^y + \omega_i^{xy}) \leq \sum_{i=1}^{n} \omega_i^y\},\ y_U = \min\{y_1 | \sum_{i<y_1+1}(2\omega_i^y + \omega_i^{xy}) \geq \sum_{i=1}^{n}(\omega_i^y + \omega_i^{xy})\}$$

Proof.

$$
\begin{aligned}
& f(x_1, y_1) - f(x_1 - 1, y_1) \\
= & \sum_{i=1}^{n} (2\omega_i^x + \omega_i^{xy})[d(i, x_1) - d(i, x_1 - 1)] + \sum_{i=1}^{n} \omega_i^{xy}[d(x_1, y_1) - d(x_1 - 1, y_1)] \\
= & \sum_{i<x_1} (2\omega_i^x + \omega_i^{xy})[d(i, x_1) - d(i, x_1 - 1)] + \sum_{i \geq x_1} (2\omega_i^x + \omega_i^{xy})[d(i, x_1) - d(i, x_1 - 1)] \\
& + \sum_{i=1}^{n} \omega_i^{xy}[d(x_1, y_1) - d(x_1 - 1, y_1)] \\
= & \sum_{i<x_1} (2\omega_i^x + \omega_i^{xy}) d(x_1, x_1 - 1) - \sum_{i \geq x_1} (2\omega_i^x + \omega_i^{xy}) d(x_1, x_1 - 1) \\
& + \sum_{i=1}^{n} \omega_i^{xy}[d(x_1, y_1) - d(x_1 - 1, y_1)]
\end{aligned}
\tag{3.12}
$$

Using triangle inequality in the third term and subtracting and adding the first term simultaneously we get:

$$
\begin{aligned}
& f(x_1, y_1) - f(x_1 - 1, y_1) \\
\leq & \, 2 \sum_{i<x_1} (2\omega_i^x + \omega_i^{xy}) d(x_1, x_1 - 1) - \sum_{i=1}^{n} (2\omega_i^x + \omega_i^{xy}) d(x_1, x_1 - 1) + \sum_{i=1}^{n} \omega_i^{xy} d(x_1, x_1 - 1) \\
= & \, 2d(x_1, x_1 - 1)[\sum_{i<x_1} (2\omega_i^x + \omega_i^{xy}) - \sum_{i=1}^{n} \omega_i^x]
\end{aligned}
\tag{3.13}
$$

Now it is easy to see that $f(x_1, y_1)$ is non-increasing when $x_1 \leq x_L$. Similarly, we have

$$
\begin{aligned}
& f(x_1, y_1) - f(x_1 + 1, y_1) \\
= & \sum_{i=1}^{n} (2\omega_i^x + \omega_i^{xy})[d(i, x_1) - d(i, x_1 + 1)] + \sum_{i=1}^{n} \omega_i^{xy}[d(x_1, y_1) - d(x_1 + 1, y_1)] \\
= & \sum_{i \geq x_1 + 1} (2\omega_i^x + \omega_i^{xy}) d(x_1, x_1 + 1) - \sum_{i < x_1 + 1} (2\omega_i^x + \omega_i^{xy}) d(x_1, x_1 + 1) \\
& + \sum_{i=1}^{n} \omega_i^{xy}[d(x_1, y_1) - d(x_1 + 1, y_1)]
\end{aligned}
\tag{3.14}
$$

Using triangle inequality in the third term and subtracting and adding the second term simultaneously we have:

$$\begin{aligned} & f(x_1, y_1) - f(x_1+1, y_1) \\ \leq & -2 \sum_{i<x_1+1} (2\omega_i^x + \omega_i^{xy}) d(x_1, x_1+1) + [\sum_{i=1}^{n} (2\omega_i^x + \omega_i^{xy}) + \sum_{i=1}^{n} \omega_i^{xy}] d(x_1, x_1+1) \\ = & -2d(x_1, x_1+1)[\sum_{i<x_1+1} (2\omega_i^x + \omega_i^{xy}) - \sum_{i=1}^{n} (\omega_i^x + \omega_i^{xy})] \end{aligned} \tag{3.15}$$

According to the the definition of x_U, $f(x_1, y_1)$ is non-decreasing when $x_1 \geq x_U$. In analogous way we can prove that $y_1^\star \in [y_L, y_U] \cap N$. ■

Observing Lemma 3.3, we can realize that the path between x_L and x_U (y_L and y_U) is exactly the subtree T_x (T_y) we introduced in Section 3. We interpret it in a different way because its results are useful for the following Lemma.

Lemma 3.4 *If we fix $y_1 \geq x_L$, then $f(x_1, y_1)$ is non-increasing for $x_1 \in [x_L, \min\{x_U, y_1\}]$. Also, if we fix $y_1 \leq x_U$, then $f(x_1, y_1)$ is non-decreasing for $x_1 \in [\max\{x_L, y_1\}, x_U]$. The same is true if we replace y_1 by x_1, x_1 by y_1, x_L by y_L and x_U by y_U.*

Proof. If $y_1 \geq x_L$ and $x_1 < \min\{x_U, y_1\}$, it is easy to verify that $d(x_1, y_1) - d(x_1+1, y_1) = d(x_1, x_1+1)$. Therefore, we can change the inequality in (3.15) to equality and have

$$\begin{aligned} & f(x_1, y_1) - f(x_1+1, y_1) \\ = & -2d(x_1, x_1+1)[\sum_{i<x_1+1} (2\omega_i^x + \omega_i^{xy}) - \sum_{i=1}^{n} (\omega_i^x + \omega_i^{xy})] \\ \geq & 0 \qquad \text{by the definition of } x_U \end{aligned}$$

If $y_1 \leq x_U$ and $x_1 > \max\{x_L, y_1\}$, $d(x_1, y_1) - d(x_1 - 1, y_1) = d(x_1, x_1 - 1)$. Similarly, we can change the inequality in (3.13) to equality and have

$$
\begin{aligned}
& f(x_1, y_1) - f(x_1 - 1, y_1) \\
= & 2d(x_1, x_1 - 1)[\sum_{i<x_1}(2\omega_i^x + \omega_i^{xy}) - \sum_{i=1}^{n}\omega_i^x] \\
\geq & 0 \qquad \text{by the definition of } x_L
\end{aligned}
$$

■

Lemma 3.5 *If $x_L \leq y_L \leq x_U \leq y_U$, then $f(x_U, y_1)$ is non-decreasing for $y_1 \in [x_U, y_U]$.*

Proof.

$$
\begin{aligned}
& f(x_U, y_1) - f(x_U, y_1 + 1) \\
= & \sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy})[d(i, y_1) - d(i, y_1 + 1)] - \sum_{i=1}^{n}\omega_i^{xy} d(y_1, y_1 + 1) \\
= & [-\sum_{i\leq y_1}(2\omega_i^y + \omega_i^{xy}) + \sum_{i>y_1}(2\omega_i^y + \omega_i^{xy}) - \sum_{i=1}^{n}\omega_i^{xy}]d(y_1, y_1 + 1) \\
= & -2[\sum_{i\leq y_1}(2\omega_i^y + \omega_i^{xy}) - \sum_{i=1}^{n}\omega_i^y]d(y_1, y_1 + 1)
\end{aligned}
$$

If $y_1 \in [x_U, y_U)$ and $y_L \leq x_U$, we have $\sum_{i\leq y_1}(2\omega_i^y + \omega_i^{xy}) - \sum_{i=1}^{n}\omega_i^y \geq 0$. Therefore, $f(x_U, y_1) - f(x_U, y_1 + 1) \leq 0$. ■

According to Lemmas 3.3, 3.4 and 3.5, we can obtain the following theorem easily.

Theorem 3.4 *Without loss of generality, we assume $x_L \leq y_L$. If $x_U \leq y_L$, then $(x_1^\star, y_1^\star) = (x_U, y_L)$; otherwise, the optimal solution must be in the set $\{(a, a) | a \in [x_L, x_U] \cap [y_L, y_U] \cap N, \}$.*

Proof. If $x_L \le x_U \le y_L$, then $y_1^\star \ge x_U$. Therefore from Lemma 3.4, $x_1^\star = x_U$. Then also $x_1^\star \le y_L \le y_U$. Again from Lemma 3.4, $y_1^\star = y_L$.

From Lemma 3.4, if $x_L \le y_L \le y_U \le x_U$, then joint locations in $[y_L, y_U]$ are local optimal. Again from Lemma 3.4, if $x_L \le y_L \le x_U \le y_U$ and $y_1 \in [x_U, y_U]$, x_U is local optimal. Further, from Lemma 3.5, we know that (x_U, x_U) is better than (x_U, y_1) for $y_1 \in [x_U, y_U]$. Therefore, the optimal solution is to locate facilities jointly in $[x_L, x_U] \cap [y_L, y_U] \cap N$. ■

Lemma 3.6 *We have $x_L \le \hat{x}_1 \le x_U$ and $y_L \le \hat{y}_1 \le y_U$, where $\hat{x}_1$ and $\hat{y}_1$ are 1-median respectively of P'_x and P'_y defined in Lemma 3.2.*

Proof. According to the 1-median algorithm of Goldman (1971) for a path (special tree), we have

$$\hat{x}_1 = \min\{x_1 | \sum_{i<x_1+1} (2\omega_i^x + \omega_i^{xy}) \ge \frac{1}{2} \sum_{i=1}^{n} (2\omega_i^x + \omega_i^{xy})\},$$

and also

$$\hat{x}_1 = \max\{x_1 | \sum_{i<x_1} (2\omega_i^x + \omega_i^{xy}) \le \frac{1}{2} \sum_{i=1}^{n} (2\omega_i^x + \omega_i^{xy})\}.$$

From the definitions of x_L and x_U in Lemma 3.3, we have $x_L \le \hat{x}_1 \le x_U$. The same is true for $\hat{y}_1$. ■

Lemma 3.7 *Define functions $\phi(x_1) = \sum_{i=1}^{n} (2\omega_i^x + \omega_i^{xy}) d(i, x_1)$ and $\psi(y_1) = \sum_{i=1}^{n} (2\omega_i^y + \omega_i^{xy}) d(i, y_1)$, where $x_1,\ y_1 = 1, \cdots, n$. The function $\phi(x_1)$ ($\psi(y_1)$) is non-increasing when $x_1 \le \hat{x}_1$ ($y_1 \le \hat{y}_1$), and non-decreasing when $x_1 \ge \hat{x}_1$ ($y_1 \ge \hat{y}_1$).*

Proof.

$$
\begin{aligned}
&\phi(x_1) - \phi(x_1+1) \\
&= \sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i,x_1) - \sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i,x_1+1) \\
&= [\sum_{i\geq x_1+1}(2\omega_i^x + \omega_i^{xy}) - \sum_{i<x_1+1}(2\omega_i^x + \omega_i^{xy}]d(x_1,x_1+1)
\end{aligned}
$$

From the two different forms of $\hat{x}_1$ shown in the proof of Lemma 3.6, it is obvious that $\phi(x_1)$ is non-increasing when $x_1 \leq \hat{x}_1$ and non-decreasing when $x_1 \geq \hat{x}_1$. Similarly, we can prove that $\psi(y_1)$ is non-increasing when $y_1 \leq \hat{y}_1$ and non-decreasing when $y_1 \geq \hat{y}_1$. ■

Theorem 3.5 *Without loss of generality, we assume that $\hat{x}_1 \leq \hat{y}_1$. The optimal solution $(x_1^\star, y_1^\star)$ must satisfy the condition: if $[x_L, x_U] \cap [y_L, y_U] = \emptyset$, then $\hat{x}_1 \leq x_1^\star < y_1^\star \leq \hat{y}_1$, otherwise, $\hat{x}_1 \leq x_1^\star = y_1^\star \leq \hat{y}_1$.*

Proof. From Theorem 3.4, if $[x_L, x_U] \cap [y_L, y_U] = \emptyset$, then $(x_1^\star, y_1^\star) = (x_U, y_L)$. According to Lemma 3.6, it is obvious that $\hat{x}_1 \leq x_1^\star < y_1^\star \leq \hat{y}_1$.

Again from Theorem 3.4, if $[x_L, x_U] \cap [y_L, y_U] \neq \emptyset$, then co-location is optimal. Therefore formula (3.11) is reduced to $f(x_1, y_1) = f(x_1, x_1) = \phi(x_1) + \psi(x_1)$. From Lemma 3.7, $\phi(x_1) + \psi(x_1)$ is non-increasing when $x_1 \leq \hat{x}_1$ and non-decreasing when $x_1 \geq \hat{y}_1$. Therefore $\hat{x}_1 \leq x_1^\star = y_1^\star \leq \hat{y}_1$. ■

Obviously, Corollary 3.1 still holds here. If we find $\hat{x}_1 = \hat{y}_1$, we solve the problem in $O(n)$. If $[x_L, x_U] \cap [y_L, y_U] = \emptyset$, then from Theorem 3.4, we also solve the problem in $O(n)$. If $[x_L, x_U] \cap [y_L, y_U] \neq \emptyset$, co-location, say (s, s), is optimal. From (3.11), the optimal objective function value is

$$f(s,s) = \sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i,s) + \sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy})d(i,s) \tag{3.16}$$

Note that given adjacent distance matrix, $d(1,i)$ can be computed in $O(n)$ time for all $i = 1, \cdots, n$. Define

$$W^x(j) = \sum_{i=1}^{j}(2\omega_i^x + \omega_i^{xy}), \qquad V^x(j) = \sum_{i=1}^{j}(2\omega_i^x + \omega_i^{xy})d(1,i)$$

It is easy to see that $W^x(j)$ and $V^x(j)$ can be obtained in $O(n)$ time for all $j = 1, \cdots, n$. Now,

$$\begin{aligned}
&\sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy})d(i,j) \\
=&\sum_{i=1}^{j}(2\omega_i^x + \omega_i^{xy})[d(1,j) - d(1,i)] + \sum_{i=j+1}^{n}(2\omega_i^x + \omega_i^{xy})[d(1,i) - d(1,j)] \\
=&W^x(j)d(1,j) - V^x(j) + \sum_{i=j+1}^{n}(2\omega_i^x + \omega_i^{xy})[d(1,i) - d(1,j)] \\
=&W^x(j)d(1,j) - V^x(j) + V^x(n) - V^x(j) + [W^x(n) - W^x(j)]d(1,j)
\end{aligned}$$

It follows from the previous formula that $f(j,j)$ in (3.16) can be obtained in constant time for each j. Therefore, the problem can be solved in a total of $O(n)$ time.

Now let us reconsider the example of Figure 3.7. According to Theorem 3.4, since $[4,6] \cap [5,8] = [5,6]$, we can say the optimal solution must be in the set of $\{(5,5),(6,6)\}$. Also, it is easy to obtain that $\hat{x}_1 = 6$, $\hat{y}_1 = 7$. From Theorem **??**, we know the optimal solution must be $(6,6)$.

3.5 Heuristics

In this section, we describe several heuristic algorithms developed to solve our problem. These algorithms are: the greedy adding with substitution heuristic, the median-based heuristic, the variable neighborhood search heuristic and the tabu search heuristic. Because all of them will use a local search algorithm, we first present an interchange procedure originally proposed by Teitz and Bart (1968) for the p-median problem. This procedure starts with a feasible solution (X, Y).

Procedure INTERCHANGE(X,Y):

Step 1. Fix Y and swap $a \in X$ and $b \in N - X$. Calculate the objective value using Y and the new X.

Step 2. Swap a and b back, and repeat Step 1 with another pair until all possible swaps are finished.

Step 3. Choose X according to the swap that yields the largest reduction in the objective function value. If there is no improvement, go to Step 4. Otherwise go to step 1.

Step 4. Fix X and swap $a \in Y$ and $b \in N - Y$. Calculate the objective value using X and the new Y.

Step 5. Swap a and b back, and repeat Step 4 until all possible swaps are finished.

Step 6. Choose Y according to the swap that yields the largest reduction in the objective function value. If there is no improvement, go to Step 7. Otherwise go to Step 4.

Step 7. If there is no improvement in a full cycle of Step 1 to Step 6, stop, otherwise go to Step 1.

3.5.1 The Greedy Adding with Substitution Heuristic

Our first heuristic, the Greedy Adding with Substitution (GAS) algorithm, is a greedy type procedure (See, for example, Church and ReVelle 1974; ReVelle, Williams and Boland, 2002). This algorithm starts with an empty solution set and adds one pair of facilities at a time until m or p facilities are reached, then add additional facilities one each time until all facilities have been selected. At each iteration, it chooses the pair (or facility) whose addition causes the greatest reduction in the objective function value. The GAS algorithm calls *procedure* INTERCHANGE at the end of each iteration.

3.5.2 The Median-based Heuristic and a Lower Bound

Our second heuristic, which we call the median-based (MB) heuristic, is to solve two median problems. This heuristic is motivated by the study of the problem on a tree when $m = p = 1$ (formula (3.11)). Let $X^\star$, $Y^\star$ be an optimal solution to our problem. Let $\hat{X} = (\hat{x}_1, \cdots, \hat{x}_m)$ and $\hat{Y} = (\hat{y}_1, \cdots, \hat{y}_p)$ be the m-median and the p-median of G with demand weights $2\omega_i^x + \omega_i^{xy}$

and $2\omega_i^y + \omega_i^{xy}$ at node i respectively. We have

$$
\begin{aligned}
& f(X^\star, Y^\star) \\
= \; & 2\sum_{i=1}^{n} \omega_i^x \min_{1\le j\le m}\{d(i,x_j^\star)\} + 2\sum_{i=1}^{n} \omega_i^y \min_{i\le k\le p}\{d(i,y_k^\star)\} \\
& + \sum_{i=1}^{n} \omega_i^{xy} \min_{1\le j\le m}\min_{1\le k\le p}\{d(i,x_j^\star) + d(x_j^\star, y_k^\star) + d(y_k^\star, i)\} \\
\ge \; & 2\sum_{i=1}^{n} \omega_i^x \min_{1\le j\le m}\{d(i,x_j^\star)\} + 2\sum_{i=1}^{n} \omega_i^y \min_{1\le k\le p}\{d(i,y_k^\star)\} \\
& + \sum_{i=1}^{n} \omega_i^{xy} [\min_{1\le j\le m}\{d(i,x_j^\star)\} + \min_{1\le j\le m}\min_{1\le k\le p}\{d(x_j^\star, y_k^\star)\} + \min_{1\le k\le p}\{d(y_k^\star, i)\}] \\
\ge \; & \sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy}) \min_{1\le j\le m}\{d(i,x_j^\star)\} + \sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy}) \min_{1\le k\le p}\{d(i,y_k^\star)\} \\
\ge \; & \sum_{i=1}^{n}(2\omega_i^x + \omega_i^{xy}) \min_{1\le j\le m}\{d(i,\hat{x}_j)\} + \sum_{i=1}^{n}(2\omega_i^y + \omega_i^{xy}) \min_{1\le k\le p}\{d(i,\hat{y}_k)\} \\
\stackrel{\text{def}}{=} \; & M(\hat{X}) + M(\hat{Y})
\end{aligned}
$$

The p-median problem is $\mathcal{NP}$-complete. Many heuristics have been proposed for solving it during the last decades. A classical heuristic for the p-median problem is vertex substitution method (Teitz and Bart, 1968). Erlenkotter (1978) suggested the well-known Dual Ascent heuristic. Narula, Ogbu, and Samuelsson (1977) proposed a lagrangian dual relaxation method. We refer the reader to Mirchandani (1990) and Daskin (1995) for additional discussion on the p-median problem. Recently, some meta-heuristics have been proposed for solving the large scale p-median problem, e.g., tabu search (Voss, 1996; Rolland, Schilling, and Current, 1996), variable neighborhood search (Hansen and Mladenovic, 1997), heuristic concentration (Rosing and ReVelle, 1997; Rosing, ReVelle and Schilling, 1999) and greedy randomized adaptive search procedure (Resende and Werneck, 2002). Among all of these methods, Lagrangian Relaxation is one of the most computationally attractive heuristic. Lagrangian Relaxation using Subgradient optimization method gives a very tight lower bound most of the time (less than 0.15%, see Daskin, 1995). We use the procedure given by Narula, Ogbu, and Samuelsson(1977), which can also be found in Daskin(1995), but employ a *discounted direction sequence* (See Crowder, 1976) with discount rate $\theta = 0.3$ to reduce the oscillation during the Subgradient method.

It is easy to construct a feasible solution based on the Lagrangian dual solution (we refer the reader to Daskin, 1995 for details). Let X' (Y') be the best constructed feasible solution returned by Lagrangian Relaxation to the m-median (p-median) problem with demand weight $2\omega_i^x + \omega_i^{xy}$ ($2\omega_i^y + \omega_i^{xy}$) at node i. Let $LB(\hat{X})$ and $LB(\hat{Y})$ be the lower bounds of $M(\hat{X})$ and $M(\hat{Y})$ returned by the Lagrangian Relaxation procedure respectively. Obviously, we have

$$f(X', Y') \geq f(X^\star, Y^\star) \geq M(\hat{X}) + M(\hat{Y}) \geq LB(\hat{X}) + LB(\hat{Y}).$$

We will use the lower bound $LB(\hat{X}) + LB(\hat{Y})$ to evaluate the performance of our heuristics in the computational experiments. At the end of the algorithm, we call *procedure* INTERCHANGE(X', Y') to find a local optimal solution in the neighborhood of (X', Y').

3.5.3 The Tabu Search Heuristic

Tabu Search (TS) is a meta-heuristic that guides a local heuristic search procedure to explore the solution space beyond local optimality and has been applied to a wide range of problems(Glover and Laguna, 1997). Tabu search has also been applied to location problems successfully. For example, Rolland, Schilling, and Current (1996) provides a tabu search algorithm for the p-median problem; Al-Sultan and Al-Fawzan (1999) , and Michel and Hentenryck (2002) present two different tabu search methods for solving the uncapacitated facility location problem. A comprehensive tutorial on tabu search can be found in Glover and Laguna (1997).

The basic elements of simple tabu search are *neighborhood, tabu restrictions, aspiration criteria, and short term memory.* The neighborhood of the current solution is the set of solutions which can be obtained by a single move from the current one (the concept of the neighborhood used here is different from that of the next section). A move is a procedure that describes how a new solution can be generated from the current one. Note that the new solution is not necessarily feasible. The *strategic oscillation* approach of tabu search allows moves to encompass infeasible solutions. We adopt two types of moves: ADDs and REMOVEs. We add (delete) a pair of nodes simultaneously, one for each type of facility, at an ADD (REMOVE) move. After an ADD move, we obtain an infeasible solution. Among all possible ADD moves, we choose the best (infeasible) solution as the starting solution for the successive REMOVE moves. After the successive REMOVE moves, we construct again feasible solutions. Among all feasible solutions, we either use the best feasible solution as the start point of the next iteration or stop the algorithm if some termination conditions are satisfied. In the heuristic developed in this chapter, we terminate the algorithm either when the iteration limit is reached or when the objective function value has not improved in a number of consecutive iterations.

Tabu restrictions are used to forbid moving back to previously checked solutions. We employ two tabu restriction lists corresponding to the two types of facilities. Once two nodes are added to the set of open facilities after an ADD move, they are classified as tabu. According to the short term memory and the two tabu restriction lists we employed, we use two circular arrays in the code to store the nodes entered the tabu lists. If these two arrays are full, the heads of the two circular arrays will be removed from the tabu lists.

Aspiration criteria are used to make a tabu solution free to be selected if this solution is of sufficient quality. In this chapter, we employ the aspiration criteria when ADD moves and the successive REMOVE moves generate a feasible solution better than the best known solution. Otherwise, we choose the best generated solution which is not tabu.

In summary, the procedures described previously can be stated as follows:

Procedure ADD&REMOVE(X,Y):

step 1. Add a node from $N-X$ to X and a node from $N-Y$ to Y. Calculate the objective function value(infeasible).

step 2. Remove back the added nodes and repeat step 1 for another pair of nodes until all possible pairs are checked. Set $i := 1$.

step 3. Choose the ith best solution, and denote it by $(X \cup \{x_{m+1}\}, Y \cup \{y_{p+1}\})$.

step 4. Remove a node from X and a node from Y respectively (not from $X \cup \{x_{m+1}\}$ and $Y \cup \{y_{p+1}\}$). Calculate the objective function value.

step 5. Add back the removed nodes and repeat step 4 for another pair of nodes until all possible pairs are checked.

step 6. Choose the best solution as current (X, Y). If the pairs (x_{m+1}, y_{p+1}) is not tabu, put them into tabu lists and stop. If (x_{m+1}, y_{p+1}) is tabu, but (X, Y) is better than the old solution, stop. If (x_{m+1}, y_{p+1}) is tabu, and (X, Y) is worse than the old solution, set $i := i + 1$, go to step 3.

Procedure TS:

step 1. Generate a feasible solution (X, Y) and use (X, Y) as a starting point of tabu search. Label this solution as the best solution.

step 2. Call *procedure* ADD&REMOVE(X, Y) and obtain a new solution.

step 3. If the new solution generated in step 2 is better than the best solution, update the best solution. Repeat step 2 until the iteration limit is reached or the best solution has not been changed for a number of consecutive iterations.

Because we change two facilities (one for each type of facility) at each iteration, the solution returned by tabu search may not be local optimal. To guarantee that we obtain a local minimum, we call *procedure* INTERCHANGE before we terminate the algorithm.

3.5.4 The Variable Neighborhood Search Heuristic

Variable Neighborhood Search (VNS) is a recently proposed meta-heuristic for solving combinatorial problems (Mladenovic and Hansen, 1997). It changes neighborhoods within a local search algorithm systematically. The basic VNS algorithm can be described as follows (Mladenovic and Hansen, 1997).

Initialization. Find an initial solution x;

Main Step. (1) Set $k := 1$. (2) Until $k = k_{max}$, where k_{max} is a pre-specified upper bound of k, repeat the following steps: (a) generate a point x' at random from the k^{th} neighborhood of x; (b) apply some local search method with x' as the initial solution; denote with x'' the obtained local optimum; (c) if the solution thus obtained is better than the incumbent, move there ($x := x'$), and continue the search with $k := 1$; otherwise, set $k := k + 1$;

To apply the VNS algorithm to the minisum multi-purpose trip location problem, we first define a function $\rho(X_1, X_2)$ as the number of different locations in X_1 and X_2, where $|X_1| = |X_2|$. For example, if $X_1 = (1, 2, 4)$ and $X_2 = (3, 4, 5)$, then $\rho(X_1, X_2) = 2$. Denote by $\mathcal{N}_k(X, Y)$ the set of solutions in the k^{th} neighborhood of (X, Y), i.e., $\rho(X, X') + \rho(Y, Y') = k$ for any $(X', Y') \in \mathcal{N}_k(X, Y)$. Comparing to local search heuristics, which are usually "1-opt" procedures, VNS doesn't stop at the $\mathcal{N}_1(X, Y)$. Suppose (X, Y) is the current solution, VNS chooses a solution, say (X', Y'), from $\mathcal{N}_1(X, Y)$ randomly and runs a local search in $\mathcal{N}_1(X', Y')$. If there is no improvement, VNS looks for another starting solution from $\mathcal{N}_2(X, Y)$ randomly and runs a new local search and so on. If VNS finds an improved solution, this solution will be used as the new current solution and a new iteration will start until some stopping condition is met. In this chapter, we terminate the algorithm when the maximum number of iterations is reached. It is possible that there is no improvement after we run a local search in $\mathcal{N}_1(X', Y')$, where $(X', Y') \in \mathcal{N}_{(m+p)}(X, Y)$. In this case, we will choose a random solution from $\mathcal{N}_1(X, Y)$ and start a new iteration again. The Variable Neighborhood Search procedure can be stated as follows:

Procedure VNS:

step 1. Find an initial solution (X, Y). Set $(X^\star, Y^\star) := (X, Y)$ and $k := 1$.

Step 2. Choose a solution (X', Y') randomly from $\mathcal{N}_k(X^\star, Y^\star)$.

Step 3. Call *procedure* INTERCHANGE to apply a local search based on (X', Y').

Step 4. If $k < (m+p)$ and the returned solution of Step 3, (X'', Y''), is equal to (X', Y'), set $k := (k+1)$ and go back to Step 2 unless the iteration limit is reached. If $k = (m+p)$ and the returned solution of Step 3, (X'', Y''), is equal to (X', Y'), set $k := 1$ and go back to Step 2 unless the iteration limit is reached. Otherwise, set $(X^\star, Y^\star) := (X'', Y'')$ and $k := 1$ and go back to Step 2 unless the iteration limit is reached.

3.6 Computational Experiments

In order to test the heuristic algorithms described in the preceding section an extensive set of computational experiments was conducted. The number of nodes n was set to 20, 30, 50, 80, 100, 200, 500, 1000. For each n, eight combinations of m and p were considered. In total, we chose 64 combinations of parameter values. For each combination of n, m and p, 10 problem instances were generated, leading to a total of 640 problem instances. All runs were performed on a Pentium III PC equipped with 677MHZ processor and 128M RAM.

All procedures were coded in ANSI C. The problem data used in the experiments were generated randomly as follows. The Cartesian coordinates of the nodes were generated over the interval $(0, 100)$ uniformly. Then nodes were connected randomly until a tree was formed. Finally, a random number of links were added to the tree generated to create a network. All demand weights were generated over the interval (0,1) randomly. The length of each link was calculated using the Euclidean distance formula. For all problem instances, we ensured that no two instances among the total 640 problem cases shared a common random seed.

Due to the simplicity of the GAS algorithm, we used GAS solution as the starting point of the TS heuristic and the VNS heuristic. Therefore, the GAS heuristic is always inferior to the TS and VNS heuristics according to the solution quality. In *procedure* TS, the iteration limit was set at 400, and if the objective function value failed to improve in 50 consecutive iterations, we terminated it before the iteration limit was reached. In *procedure* VNS, we set the iteration limit at 500. The following performance measures were used in the experiments:

n	m	p	GAS RE	GAS Time	MB RE	MB Time	TS RE	TS Time	VNS RE	VNS Time	CPLEX Time	RELB
20	2	5	0.0328	0.01	0.0036	0.02	0.0011	1.26	0.0162	1.20	1.65	0.1135
20	3	4	0.0660	0.02	0.0074	0.02	0.0050	1.20	0.0319	1.22	1.38	0.0374
20	4	4	0.0278	0.03	0.0145	0.03	0.0052	1.33	0.0222	1.68	3.11	0.0199
20	5	5	0.0572	0.04	0.0066	0.04	0.0047	2.07	0.0547	2.84	2.21	0.0424
20	5	8	0.0810	0.08	0.0111	0.04	0.0000	2.23	0.0588	4.53	1.35	0.0830
20	5	10	0.0485	0.10	0.0092	0.05	0.0022	2.50	0.0287	5.41	1.61	0.1458
20	8	10	0.0705	0.18	0.0126	0.09	0.0030	3.19	0.0557	8.66	1.25	0.1080
20	10	10	0.0748	0.21	0.0238	0.10	0.0029	3.31	0.0460	10.82	1.22	0.0805
average on n=20			0.0573	0.08	0.0111	0.05	0.0030	2.14	0.0393	4.55	1.72	0.0788
30	2	5	0.0325	0.03	0.0118	0.07	0.0024	4.77	0.0211	2.52	1538.44	0.0993
30	3	4	0.0799	0.05	0.0036	0.05	0.0013	4.97	0.0345	2.69	355.46	0.0368
30	4	4	0.0673	0.08	0.0127	0.07	0.0029	6.80	0.0585	3.73	175.68	0.0131
30	5	5	0.0268	0.13	0.0166	0.10	0.0045	7.32	0.0268	6.94	228.05	0.0317
30	5	8	0.0471	0.22	0.0090	0.12	0.0071	8.50	0.0471	10.85	79.84	0.0748
30	5	10	0.0278	0.27	0.0080	0.15	0.0019	10.32	0.0228	13.66	142.05	0.1147
30	8	10	0.0505	0.54	0.0168	0.22	0.0056	11.31	0.0505	23.13	41.60	0.0564
30	10	10	0.0420	0.66	0.0118	0.31	0.0034	12.55	0.0363	29.51	23.39	0.0368
average on n=30			0.0467	0.25	0.0113	0.14	0.0036	8.32	0.0372	11.63	323.06	0.0580

Table 3.1: Computational results for problems with $n = 20, 30$

1. Relative Error (RE): $(z - z^\star)/z^\star$, where z is the objective function value obtained by the heuristic and $z^\star$ is the optimal objective function value returned by ILOG CPLEX.

2. Relative Error of the Lower Bound (RELB): $(z^\star - \underline{z})/\underline{z}$, where $\underline{z}$ is the lower bound obtained by the Median-based heuristic.

3. Heuristic Gap (HG): $(z - \underline{z})/\underline{z}$.

4. Time: average CPU running time of 10 instances in seconds. For TS and VNS, we excluded the time of obtaining the initial solution (GAS time). I/O time is also excluded. Local search time is incorporated into all heuristic procedures.

For small scale problems ($n = 20, 30$), we called ILOG CPLEX 8.1 to obtain optimal

			GAS		MB		TS		VNS	
n	m	p	HG	Time	HG	Time	HG	Time	HG	Time
50	2	5	0.1359	0.13	0.1113	0.16	0.1024	21.38	0.1244	7.20
50	3	4	0.1345	0.20	0.0510	0.15	0.0455	25.39	0.1165	7.75
50	4	4	0.0867	0.33	0.0143	0.16	0.0114	37.93	0.0683	10.81
50	5	5	0.0503	0.58	0.0313	0.27	0.0190	47.66	0.0503	20.38
50	5	8	0.1172	0.85	0.0764	0.38	0.0669	52.19	0.1118	32.91
50	5	10	0.1239	1.08	0.0981	0.48	0.0919	56.08	0.1239	42.59
50	8	10	0.0645	2.14	0.0463	0.71	0.0382	65.29	0.0564	72.99
50	10	10	0.0666	2.94	0.0353	0.95	0.0320	101.02	0.0666	95.63
average on n=50			0.0975	1.03	0.0580	0.41	0.0509	50.87	0.0898	36.28
80	2	5	0.1702	0.43	0.1154	0.40	0.1060	107.94	0.1370	18.82
80	3	4	0.1270	0.75	0.0529	0.58	0.0381	136.98	0.0891	20.31
80	4	4	0.1080	1.33	0.0061	0.51	0.0055	185.44	0.0991	28.56
80	5	5	0.0640	2.19	0.0115	0.71	0.0109	240.57	0.0572	52.04
80	5	8	0.0847	3.10	0.0657	1.05	0.0622	208.64	0.0842	88.79
80	5	10	0.1287	3.62	0.0810	1.35	0.0784	221.34	0.1260	117.67
80	8	10	0.0770	7.89	0.0321	2.15	0.0308	327.79	0.0704	202.84
80	10	10	0.0584	11.43	0.0163	2.82	0.0158	561.78	0.0553	271.22
average on n=80			0.1023	3.84	0.0476	1.20	0.0435	248.81	0.0898	100.03
100	2	5	0.1655	0.78	0.1190	0.73	0.1172	217.78	0.1496	30.77
100	3	4	0.1271	1.51	0.0470	0.85	0.0404	247.75	0.1014	33.14
100	4	4	0.0927	2.54	0.0095	0.82	0.0072	347.89	0.0709	46.16
100	5	5	0.0728	4.59	0.0120	1.36	0.0120	483.54	0.0728	89.57
100	5	8	0.0805	5.43	0.0719	1.93	0.0621	495.90	0.0805	142.62
100	5	10	0.1342	6.01	0.0821	2.13	0.0793	501.98	0.1226	188.44
100	8	10	0.0790	15.04	0.0402	3.33	0.0399	703.93	0.0754	325.78
100	10	10	0.0592	21.70	0.0203	4.52	0.0159	1153.64	0.0548	430.53
average on n=100			0.1014	7.20	0.0503	1.96	0.0468	519.05	0.0910	160.88

Table 3.2: Computational results for problems with $n = 50, 80, 100$

objective function values using the Integer Linear Programming formulation given in Section 2. Computational results for small scale problems are listed in Table 3.1, in which we report average Relative Error (RE) and average computational Time (in seconds) for all heuristic procedures. Average CPLEX running time and average Relative Error of the Lower Bound (RELB) returned by the Median-based heuristic are also presented in Table 3.1. Problems with $n \geq 50$ always ran out of memory when using CPLEX. Therefore, the optimal objective function values are not available for problems with $n \geq 50$. We report average Heuristic Gap

(HG) and average computational Time of all heuristics for medium-sized problems(n=50, 80, 100) in Table 3.2. We also report the computational results of the GAS heuristic and the Median-based heuristic for large scale problems(n=200, 500, 1000) in Table 3.3.

			GAS		MB	
n	m	p	HG	Time	HG	Time
200	2	5	0.1820	3.92	0.1229	2.34
200	3	4	0.1704	8.78	0.0568	2.53
200	4	4	0.1222	15.85	0.0030	2.67
200	5	5	0.0758	26.01	0.0072	3.61
200	5	8	0.0861	30.41	0.0767	5.62
200	5	10	0.1329	33.34	0.0946	7.37
200	8	10	0.0878	81.86	0.0364	12.45
200	10	10	0.0558	132.71	0.0150	15.62
average on n=200			0.1141	41.61	0.0516	6.53
500	2	5	0.1774	69.28	0.1246	22.18
500	3	4	0.1801	150.30	0.0657	22.28
500	4	4	0.1433	292.72	0.0018	27.24
500	5	5	0.0655	428.14	0.0055	29.10
500	5	8	0.0890	452.21	0.0817	46.77
500	5	10	0.1300	472.04	0.1048	65.56
500	8	10	0.0788	1337.95	0.0427	108.45
500	10	10	0.0666	2174.50	0.0087	123.74
average on n=500			0.1163	672.14	0.0544	55.67
1000	3	4	0.1860	1247.58	0.0710	92.72
1000	4	4	0.1453	2316.49	0.0004	92.27
1000	5	5	0.0633	3868.15	0.0017	127.22
1000	5	8	0.0904	3896.21	0.0825	261.11
1000	5	10	0.1169	4032.71	0.1059	322.15
1000	8	10	0.0744	14586.77	0.0407	665.56
1000	10	10	0.0634	26595.82	0.0086	835.51
average on n=1000			0.1057	9216.02	0.0444	342.36

Table 3.3: Computational results for problems with $n = 200, 500, 1000$

The following conclusions can be drawn from the computational results:

- In terms of solution quality, the Tabu Search heuristic is the most successful method.

Of the 16 combinations of n, m and p considered in the Table 3.1, average relative errors are always less than 1 percent. The Median-based heuristic is very good compared to the GAS and VNS heuristics, and only slightly worse than the Tabu Search procedure.

- It should be noted that the Heuristic Gap is actually a maximum percentage error, since the lower bound is seldom tight. In Table 3.1, the average relative error of the lower bounds are 7.88% and 5.80% for $n = 20$ and $n = 30$ respectively. Sometimes the relative error of the lower bound is more than 10%. Therefore, the average relative error of lower bounds are more tight than the Heuristic Gaps presented in Tables 3.2 and 3.3. Based on the instances presented in Table 3.2, we can conclude that the solution quality of the Median-based heuristic and the Tabu Search procedure are quite good.

- In terms of running time, the Median-based heuristic is clearly the best one. For small scale problems, the VNS heuristic incurred the largest running time. It is not surprising since we didn't terminate the VNS algorithm although the objective function value failed to improve in a large number of consecutive iterations, whereas we terminate *procedure* TS if there is no improvement in 50 consecutive iterations. For the medium-sized problem considered in Table 3.2, the TS heuristic always incurred the largest running time and it seems that this heuristic is impractical for large scale problems. The running time of CPLEX increased very fast with the problem dimension and it is also not stable. In Table 3.3, we can find that the running time of the GAS heuristic increase drastically. It took more than 8 hours averagely for solving the instance of $(n, m, p) = (1000, 10, 10)$. On the other hand, the Median-based heuristic terminated in less than 14 minutes.

- It is surprising that the VNS heuristic didn't perform well for our problem. In Hansen and Mladenovic (1997), they reported that VNS is better than Tabu Search for p-median problem when problem scales are large. In their paper, they compared VNS with the Tabu Search scheme given by Voss (1996), but not that of Rolland, Schilling, and Current (1996). In fact, the TS heuristic given in this chapter is similar to the latter.

Overall, combining good solution quality with fast running time, the Median-based heuristic employing Lagrangian Relaxation and subgradient optimization is our recommendation for real-life applications.

Chapter 4

Maximizing Covering of Multiple Purpose Customers

4.1 Introduction

Covering models have been widely used to locate facilities such as emergency services, school systems and postal facilities. Many of the extensions to covering models are based on the Maximal Covering Location Problem (MCLP). The MCLP seeks the location of a fixed number of facilities in such a way that the number of customers covered by the facilities is maximized, where a customer is considered covered if there exists a facility within a pre-specified coverage radius. The MCLP on the network was first introduced by Church and ReVelle (1974). The problem on the plane has been considered by Watson-Gandy (1982) and by Drezner (1986). Megiddo *et al* (1983) showed that the MCLP on the network is generally $\mathcal{NP}$-hard and proposed a polynomial time algorithm for the problem on the tree. Hassin and Tamir (1991) applied dynamic programming to the problem on the path and showed that the MCLP on the path can be solved very efficiently. Later, Tamir (1996) improved the complexity bound of the MCLP on the tree through using a "leaves to root" dynamic programming algorithm. A more general set of covering problems, of which the MCLP is a special case, are studied in Kolen and Tamir (1990) and Hochbaum and Pathria (1998).

Usually, a server can't handle more than one call at a time. This means that equipment can frequently be busy when called. When congestion is not severe, the concept of backup coverage is suggested. This approach seeks the allocation of more than one server to cover each customer within the standard criterion. We refer the readers to Daskin and Stern (1981), Hogan and ReVelle (1986), and Batta and Mannur (1990) for details. When congestion is expected to be more severe, the maximization of expected coverage was proposed by Daskin (1983). A review of these models and their applications can be found in Schilling *et al* (1993) and ReVelle and Williams (2002).

One criticism of the MCLP is the definition of coverage, that is, a customer is either fully covered, or not covered at all. However, in many applications such all or nothing coverage assumption may not be realistic. Church and Roberts (1983) extended the definition of coverage using several step functions. Berman and Krass (2002) showed that the MCLP with a step coverage function is equivalent to the uncapacitated facility location problem. They developed two IP formulations for the problem and showed an interesting result that the LP relaxations of both formulations provide the same value of the upper bound. In a recent paper, Berman *et al.* (2003) investigated the MCLP with a coverage decay function whose value decreases from full coverage at the lowest pre-specified radius to no coverage at the largest pre-specified radius.

One common thread in these models is that they only consider a single type of customers and a single type of facilities. Among the extensive research on location theory, little has been done with multiple types of facilities or multiple types of customers. In practice, systems that provide products/services generally consist of more than one type of facilities. We've already given some examples in Section 4, Chapter 1.

In this chapter we investigate the problem having the same setting as the one introduced in the previous chapter, but with a coverage objective, which will be described formally in the next section. This model could be beneficial to supply chain integration, either horizontal or vertical. Suppose two products or services of a company are supplementary in some degree, the company wants to intercept as many customers as possible. In such a case, the coverage model can be served for this purpose.

To the best of our knowledge, there appears to be no previously published work on covering models with multiple types of customers and multiple types of facilities. The formulation of the problem is developed in Section 2 below. For most combinatorial optimization problems, sometimes it is hard to obtain a bound better than the value of their LP relaxations. In Section 3 we propose a method which sometimes produces a tighter bound than the LP relaxation of our formulation. A dynamic programming algorithm is applied to the problem with a general number of type-x facility and a single type-y facility on the path in Section 4. As we can see, even for this simplified case on the path, the problem incurs high order polynomial time algorithm. A greedy type heuristic is discussed in Section 5. We report the computational experiments in Section 6 and finish with some concluding remarks in Section 7.

4.2 Problem Formulation

Let $G = (N, L)$ be a network with nodal set $N(|N| = n)$ and link set L. Let $M(|M| = m)$ be the potential facility location set. There are two types of facilities and three types of customers in the network. Type-x customers require service only from Type-x facilities on any shopping trip, Type-y customers require service only from Type-y facilities on any shopping trip. Type-xy customers require service from both facilities on any shopping trip.

Let w_i^x, w_i^y and w_i^{xy} be the demand weight of respectively type-x, type-y and type-xy customers residing at node $i \in N$. Each customer type-x (type-y) is considered intercepted if it is within a radius of $R^x(R^y)$ from a facility type-x (type-y). A customer type-xy is considered intercepted if the total service trip (from node i to facility x (y) to facility y (x) and back to i) is within distance R^{xy}.

The problem is to locate p type-x and q type-y facilities in M so as to maximize the total number of customers intercepted.

Denote by $d(v, z)$ the shortest distance between any two points $v, z \in G$. Assume that the triangle inequality always holds. Let $X = (x_1, \cdots, x_p)$ and $Y = (y_1, \cdots, y_q)$ be the sets of locations of type-x and type-y facilities respectively. Let us define the following three indicator variables as follows:

$$I^x(X, i) = \begin{cases} 1 & d(X, i) \leq R^x \\ 0 & \text{otherwise} \end{cases}$$

$$I^y(Y, i) = \begin{cases} 1 & d(Y, i) \leq R^y \\ 0 & \text{otherwise} \end{cases}$$

$$I^{xy}(X, Y, i) = \begin{cases} 1 & \exists v \in X, z \in Y \text{ such that } d(i, v) + d(v, z) + d(z, i) \leq R^{xy} \\ 0 & \text{otherwise} \end{cases}$$

where

$$d(S, i) = \begin{cases} \min\limits_{v \in S} d(v, i) & S \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$

The problem can be formulated as follows:

$$\max_{\substack{X, Y \subset M \\ |X|=p,\, |Y|=q}} \quad \sum_{i=1}^{n} \quad [w_i^x I^x(X, i) + w_i^y I^y(Y, i) + w_i^{xy} I^{xy}(X, Y, i)]$$

Proposition 4.1 *Suppose that* $w_i^x = w_i^y = 0$ *for all* $i \in N$*. If* $p = q$*, then co-locating two types of facilities is optimal.*

Proof. Suppose that (X^*, Y^*) is an optimal solution and $X^* \neq Y^*$. It is easy to see that both (X^*, X^*) and (Y^*, Y^*) are at least as good as (X^*, Y^*).

■

Proposition 4.2 *Suppose that $w_i^x = w_i^y = 0$ for all $i \in N$. There is no advantage to locate unequal number of two types of facilities and co-location is always optimal.*

Proof. Without loss of generality, we assume that $p < q$. Using the same argument as the proof of Proposition 4.1, (X^*, X^*) is at least as good as (X^*, Y^*) and all type-y facilities except for those located at X^* are redundant.

■

Based on Propositions 4.1 and 4.2, we have

Theorem 4.1 *If $w_i^x = w_i^y = 0$ for all $i \in N$, our problem can be reduced to the maximum covering location problem of siting* $\min\{p, q\}$ *facilities.*

Theorem 4.1 will be used in the next section to derive an upper bound for the problem. We can formulate the problem as an integer program. For each $i \in N$, define three sets of points and six sets of decision variables:

$X(i) = \{j | j \in M, d(i,j) \leq R^x\}$

$Y(i) = \{j | j \in M, d(i,j) \leq R^y\}$

$XY(i) = \{(j,k) | j,k \in M, d(i,j) + d(j,k) + d(k,i) \leq R^{xy}\}$

$$X_j = \begin{cases} 1 & \text{if there exists a type-x facility located at } j \in M \\ 0 & \text{otherwise} \end{cases}$$

$$Y_j = \begin{cases} 1 & \text{if there exists a type-y facility located at } j \in M \\ 0 & \text{otherwise} \end{cases}$$

$$XY_{jk} = \begin{cases} 1 & \text{if there exists a type-x facility located at } j \in M \text{ and a type-y facility} \\ & \text{located at } k \in M, \\ 0 & \text{otherwise} \end{cases}$$

$$Z_i^x = \begin{cases} 1 & \exists j \in X(i) \text{ such that } X_j = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$Z_i^y = \begin{cases} 1 & \exists j \in Y(i) \text{ such that } Y_j = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$Z_i^{xy} = \begin{cases} 1 & \exists j,k \in XY(i) \text{ such that } X_j = 1, Y_k = 1 \\ 0 & \text{otherwise} \end{cases}$$

The problem is formulated as follows:

$$\text{Problem P} \qquad \max \sum_{i \in N} (Z_i^x \omega_i^x + Z_i^y \omega_i^y + Z_i^{xy} \omega_i^{xy}) \tag{4.1}$$

$s.t.$

$$\sum_{j \in X(i)} X_j \geq Z_i^x \qquad i \in N \tag{4.2}$$

$$\sum_{j \in Y(i)} Y_j \geq Z_i^y \qquad i \in N \tag{4.3}$$

$$\sum_{(j,k) \in XY(i)} XY_{jk} \geq Z_i^{xy} \qquad i \in N \tag{4.4}$$

$$X_j \geq XY_{jk} \qquad j,k \in M \tag{4.5}$$

$$Y_k \geq XY_{jk} \qquad j,k \in M \tag{4.6}$$

$$\sum_{j \in M} X_j = p \tag{4.7}$$

$$\sum_{j \in M} Y_j = q \tag{4.8}$$

$$X_j, Y_j, XY_{jk}, Z_i^x, Z_i^y, Z_i^{xy} = 0,1 \qquad j,k \in M, i \in N \tag{4.9}$$

In (4.1) the total intercepted number of type-x customers, type-y customers and type-xy customers is maximized. In constraints (4.2) and (4.3) we make sure that type-x or type-y customers at node i can not be intercepted unless there exists at least one facility which is opened in $X(i)$ or $Y(i)$. Constraint (4.4) ensures that type-xy customers can not be intercepted unless there exists at least one pair of facilities that can intercept them. In constrains (4.5) and (4.6) we make a connection between the decision variables XY_{jk} and the decision variables X_j and Y_k. $XY_{jk} = 0$ unless both X_j and Y_k are equal to 1. Constraints (4.7) and (4.8) limit the total numbers of type-x and type-y facilities located to be p and q respectively. Finally, Constraint (4.9) establishes a binary restriction on the decision variables.

Constraints (4.5) and (4.6) can be replaced by the more compact set of constraints

$$X_j + Y_k \geq 2XY_{jk}, \qquad (j,k) \in XY(i), \quad i \in N \tag{4.10}$$

However, the two formulations are equivalent only for 0-1 values of variables. Constraint (4.10) is obtained by summing (4.5) over (4.6). Hence, any solution to Problem P satisfies constraint (4.10). But the converse is false when $(Z_i^x, Z_i^y, Z_i^{xy}, X_j, Y_k, XY_{jk}) \in [0,1]^6$. Therefore, the upper bound obtained by LP relaxation using constraints (4.5) and (4.6) is tighter than that using constraint (4.10).

4.3 Upper Bounds

The upper bound for the contribution of type-x facilities to the objective function from type-x customers can be obtained by solving the following MCLP problem.

$$\text{Problem } P_x \qquad \max W^x = \sum_{i \in N} w_i^x Z_i^x$$

s.t.

$$\sum_{j \in X(i)} X_j \geq Z_i^x \qquad i \in N$$

$$\sum_{j \in M} X_j = p$$

$$X_j, Z_i^x = 0, 1 \quad j \in M, i \in N$$

The upper bound for the contribution of type-y facilities to the objective function from type-y customers can be obtained by solving

$$\text{Problem } P_y \qquad \max W^y = \sum_{i \in N} w_i^y Z_i^y$$

s.t.

$$\sum_{j \in Y(i)} Y_j \geq Z_i^y \qquad i \in N$$

$$\sum_{j \in M} Y_j = q$$

$$Y_j, Z_i^y = 0, 1 \quad j \in M, i \in N$$

Define $Z(i) = \{j | j \in M, d(i,j) \leq R^{xy}/2\}$. From Theorem 4.1, the upper bound for the contribution of facilities to the objective function from type-xy customers can be obtained by solving the following MCLP problem.

$$\text{Problem } P_{xy} \qquad \max W^{xy} = \sum_{i \in N} w_i^{xy} Z_i^{xy}$$

s.t.

$$\sum_{j \in Z(i)} X_j \geq Z_i^{xy} \qquad i \in N$$

$$\sum_{j \in M} X_j = \min\{p, q\}$$

$$X_j, Z_i^{xy} = 0, 1 \quad j \in M, i \in N$$

Obviously $W^x + W^y + W^{xy}$ is an upper bound on any solution of our problem. Also $W_{LP}^x + W_{LP}^y + W_{LP}^{xy}$ where we use the LP-relaxations of Problems P_x, P_y and P_{xy} is an upper bound. Moreover it is well known (ReVelle, 1993) that a LP relaxed MCLP very often results in an all-integer solution or a solution that is quite close to be all-integer.

As an example consider the 5-node network where the numbers next to the links are length, and the three numbers next to each node i are respectively ω_i^x, ω_i^y and ω_i^{xy}.

It is easy to verify that

$X(1) = Y(1) = \{1\}$,

$X(2) = Y(2) = \{2\}$,

$X(3) = Y(3) = \{3\}$,

$X(4) = Y(4) = \{4, 5\}$,

$X(5) = Y(5) = \{4, 5\}$,

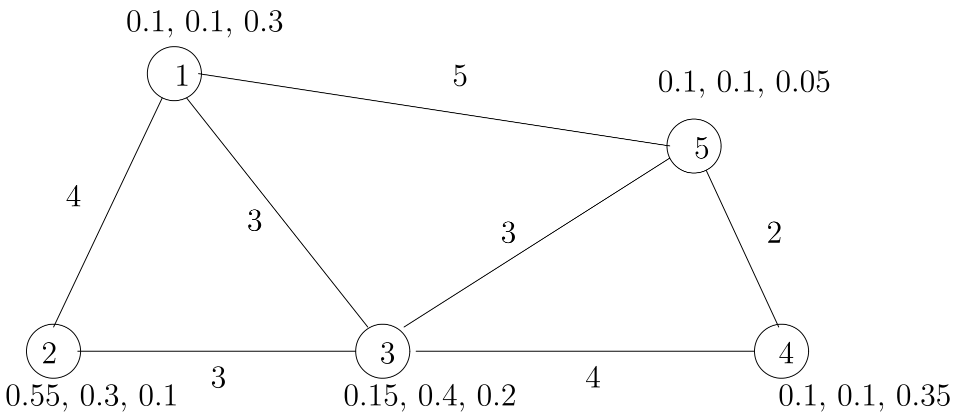


Figure 4.1: The 5-node network. Suppose that $R^x = R^y = 2$ and $R^{xy} = 6$ and for simplicity we assume that $M = N$.

$XY(1) = \{(1,1),(1,3),(3,1),(3,3)\}$,

$XY(2) = \{(2,2),(2,3),(3,2),(3,3)\}$,

$XY(3) = \{(1,1),(2,2),(1,3),(3,1),(2,3),(3,2),(3,3),(3,5),(5,3),(5,5)\}$,

$XY(4) = \{(4,5),(5,4),(4,4),(5,5)\}$,

$XY(5) = \{(3,5),(5,3),(3,3),(4,5),(5,4),(4,4),(5,5)\}$.

Solving the exact problem for $p = 1$ and $q = 1$ we obtain the optimal solution: $X_2 = Y_3 = XY_{23} = 1$ (all other decision variables are 0) and the optimal objective function value is 1.25. Solving the LP relaxed problems P_x, P_y and P_{xy} we obtain all-integer solutions for all of them and $W^x + W^y + W^{xy} = W^x_{LP} + W^y_{LP} + W^{xy}_{LP} = 0.55 + 0.4 + 0.65 = 1.6$.

Let W_{LP} be the objective function value of the LP relaxation of Problem P. Obviously, W_{LP} is also an upper bound. For the previous example, using CPLEX we solve the LP relaxation problem and obtain $W_{LP} = 1.33 < W_{LP}^x + W_{LP}^y + W_{LP}^{xy}$. We are interested in comparing W_{LP} with the previous bound $W_{LP}^x + W_{LP}^y + W_{LP}^{xy}$. There is no dominance between these two upper bounds. To see this, we reformulate Problem P_{xy} as

$$\text{Problem } P'_{xy} \qquad \max W^{xy} = \sum_{i \in N} Z_i^{xy} w_i^{xy}$$

s.t.

$$\sum_{(j,k) \in XY(i)} XY_{jk} \geq Z_i^{xy} \qquad i \in N$$

$$X_j \geq XY_{jk} \qquad j,k \in M$$

$$Y_k \geq XY_{jk} \qquad j,k \in M$$

$$\sum_{j \in M} X_j = p$$

$$\sum_{j \in M} Y_j = q$$

$$X_j, Y_j, XY_{jk}, Z_i^x, Z_i^y, Z_i^{xy} = 0,1 \qquad j,k \in M, i \in N$$

According to Theorem 4.1, Problems P_{xy} and P'_{xy} have the same optimal objective function value. Now merging Problems P_x, P_y and P'_{xy} together, we obtain:

$$\text{Problem W} \qquad \max W^x = \sum_{i \in N} w_i^x Z_i^x$$

$$\max W^y = \sum_{i \in N} w_i^y Z_i^y$$

$$\max W^{xy} = \sum_{i \in N} w_i^{xy} Z_i^{xy}$$

$s.t.$

$$(4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8)$$

$$X_j, Y_j, XY_{jk}, Z_i^x, Z_i^y, Z_i^{xy} = 0, 1 \qquad j, k \in M, i \in N$$

Problem W has the same constraint set as Problem P, but here we maximize each component of the objective function of Problem P. Therefore, we again prove that $W^x + W^y + W^{xy}$ is an upper bound. However, Problem P'_{xy} is not integer friendly any more. Obviously, the objective function value of the LP relaxation of Problem P, W_{LP}, is better than the summation of the objective function values of LP relaxed Problem W. But after we replace Problem P'_{xy} by the integer friendly formulation Problem P_{xy} , we can separate Problem W into three independent problems: P_x, P_y and P_{xy}. Since these three problems are all integer friendly, their LP relaxation often results in all-integer solutions or their objective function values are very close to those of LP relaxation problems. Therefore, it is not necessary that $W_{LP} \leq W_{LP}^x + W_{LP}^y + W_{LP}^{xy}$.

As an example consider the 5-node network depicted in Figure 4.1 again. Solve the problem for $p = 2$ and $q = 1$ we obtain that the optimal solution is to open 2 type-x facilities at nodes 2 and 3 and open the type-y facility at node 3. The optimal objective function value is 1.75. Using CPLEX, we solve the LP relaxation problem of Problem P and obtain $W_{LP} = 1.89$. Also, solving the LP relaxed P_x, P_y and P_{xy} we obtain all-integer solutions for all of them and $W^x + W^y + W^{xy} = W_{LP}^x + W_{LP}^y + W_{LP}^{xy} = 0.75 + 0.4 + 0.65 = 1.8$, which is tighter than W_{LP}. Computational results about comparing these two bounds will be given in Section 6.

4.4 Location on a path when $\min\{p, q\} = 1$

In this section, we investigate the problem on a path network when $\min\{p, q\} = 1$. Without loss of generality, assume that $q = 1$. For the sake of simplicity, assume that $M = N$. We index the demand nodes by $1, 2, \cdots, n$ from left to right as depicted in Figure 4.2.

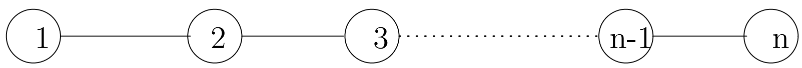


Figure 4.2: A path network

Let $P^r(j, k)$, where $j \le k$, denote the subproblem of locating r type-x facilities in $\{j, j + 1, \cdots, n\}$ with restrictions that one type-x facility has to be located at j and that the type-y facility has to be established at k. Let $P^r(j, \emptyset)$ be the same subproblem as $P^r(j, k)$ except that there are no type-y facilities located for it. Denote by $V^r(j, k)$ and $V^r(j, \emptyset)$ the optimal objective function value of subproblems $P^r(j, k)$ and $P^r(j, \emptyset)$, respectively.

It is easy to see that $P^r(j, \emptyset)$ is the classic maximal covering location problem on path $\{j, j + 1, \cdots, n\}$ with the restriction that a facility must be located at j. If the number of nodes in a subproblem is less than that of type-x facilities, then this subproblem can not be optimal for the original problem. Therefore, let $V^r(j, k) = V^r(j, \emptyset) = -\infty$ for all $j + r \ge n$.

We now have the following recursion for $V^r(j,k)$.

$$V^1(j,\emptyset) = \sum_{i=j}^{n} \omega_i^x I^x(\{j\},i), \qquad 1 \le j \le n$$

$$V^r(j,\emptyset) = \max_{j<l\le n}\{\sum_{i=j}^{l-1} \omega_i^x I^x(\{j,l\},i) + V^{r-1}(l,\emptyset)\}, \quad 1 \le j \le n,\ 1 < r \le p$$

$$V^1(j,k) = \sum_{i=j}^{n} \omega_i^x I^x(\{j\},i) + \sum_{i=j}^{n} \omega_i^y I^y(\{k\},i) + \sum_{i=j}^{n} \omega_i^{xy} I^{xy}(\{j\},\{k\},i), \quad n \ge k \ge j \ge 1$$

$$\begin{aligned} V^r(j,k) \quad &= \max\left\{ \begin{array}{l} \max\limits_{j<l\le k}\{A(j,k;l) + V^{r-1}(l,k)\}, \\ \max\limits_{k<l\le n}\{B(j,k;l) + V^{r-1}(l,\emptyset)\} \end{array} \right\}, \quad 1 < r \le p, n \ge k \ge j \ge 1, \end{aligned}$$

where

$$A(j,k;l) = \sum_{i=j}^{l-1} \omega_i^x I^x(\{j,l\},i) + \sum_{i=j}^{l-1} \omega_i^y I^y(\{k\},i) + \sum_{i=j}^{l-1} \omega_i^{xy} I^{xy}(\{l\},\{k\},i), \quad j < l \le k$$

$$B(j,k;l) = \sum_{i=j}^{l-1} \omega_i^x I^x(\{j,l\},i) + \sum_{i=j}^{n} \omega_i^y I^y(\{k\},i) + \sum_{i=j}^{n} \omega_i^{xy} I^{xy}(\{j,l\},\{k\},i), \quad j \le k < l$$

The second part of $V^r(j,k)$ is due to the fact that if the index of the next type-x facility is greater than that of the type-y facility, the remaining problem is reduced to the classic maximal covering location problem. The optimal objective function value is given by

$$\max_{1 \le k \le n} \Bigg\{ V^p(1,k),$$
$$\max_{1<j\le k} \Big\{ \sum_{i=1}^{j-1} \omega_i^x I^x(\{j\},i)\} + \sum_{i=1}^{j-1} \omega_i^y I^y(\{k\},i)\} + \sum_{i=1}^{j-1} \omega_i^{xy} I^{xy}(\{j\},\{k\},i)\} + V^p(j,k) \Big\},$$
$$\max_{k<j\le n} \Big\{ \sum_{i=1}^{j-1} \omega_i^x I^x(\{j\},i)\} + \sum_{i=1}^{n} \omega_i^y I^y(\{k\},i)\} + \sum_{i=1}^{n} \omega_i^{xy} I^{xy}(\{j\},\{k\},i)\} + V^p(j,\emptyset) \Big\}$$
$$\Bigg\}$$

To illustrate the algorithm, consider the 4-node path depicted in Figure 5.5 where the numbers next to the nodes are ω_i^x, ω_i^y, and ω_i^{xy}, respectively, and the numbers next to the links are the distance between adjacent nodes. Suppose that $R^x = R^y = 2$ and $R^{xy} = 5$ and $p = 2$, $q = 1$. First, solve the recursion of the classic maximal covering location problem for type-x customers and obtain $V^1(1,\emptyset) = 1$, $V^1(2,\emptyset) = 6$, $V^1(3,\emptyset) = 4$, $V^1(4,\emptyset) = 3$, and

$$V^2(1,\emptyset) = \max\{1 + V^1(2,\emptyset), 1 + 2 + V^1(3,\emptyset), 1 + V^1(4,\emptyset)\} = \max\{7,7,4\} = 7$$

$$V^2(2,\emptyset) = \max\{2 + V^1(3,\emptyset), 2 + 4 + V^1(4,\emptyset)\} = \max\{6,9\} = 9$$

$$V^2(3,\emptyset) = 7$$

$$V^2(4,\emptyset) = -\infty$$

It is easy to obtain $V^1(j,k)$ for $k \ge j \ge 1$ as follows.

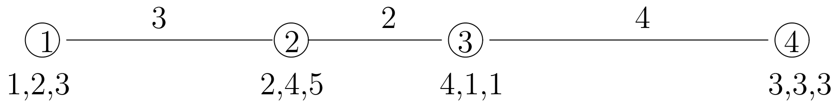


Figure 4.3: A simple 4-node path

$$
\begin{array}{llll}
V^1(1,1) = 6, & V^1(1,2) = 6, & V^1(1,3) = 6, & V^1(1,4) = 4 \\
 & V^1(2,2) = 17, & V^1(2,3) = 17, & V^1(2,4) = 9 \\
 & & V^1(3,3) = 6, & V^1(3,4) = 7 \\
 & & & V^1(4,4) = 9
\end{array}
$$

$$V^2(1,1) = \max\{6 + V^1(2,\emptyset), 8 + V^1(3,\emptyset), 6 + V^1(4,\emptyset)\} = \max\{12, 12, 9\} = 12$$

$$V^2(1,2) = \max\{1 + V^1(2,2), 14 + V^1(3,\emptyset), 6 + V^1(4,\emptyset)\} = \max\{18, 18, 9\} = 18$$

$$V^2(1,3) = \max\{1 + V^1(2,3), 12 + V^1(3,3), 6 + V^1(4,\emptyset)\} = \max\{18, 18, 9\} = 18$$

$$V^2(1,4) = \max\{1 + V^1(2,4), 3 + V^1(3,4), 1 + V^1(4,4)\} = \max\{10, 10, 10\} = 10$$

$$V^2(2,2) = \max\{13 + V^1(3,\emptyset), 17 + V^1(4,\emptyset)\} = \max\{17, 20\} = 20$$

$$V^2(2,3) = \max\{11 + V^1(3,3), 17 + V^1(4,\emptyset)\} = \max\{17, 20\} = 20$$

$$V^2(2,4) = \max\{2 + V^1(3,4), 6 + V^1(4,4)\} = \max\{9, 15\} = 15$$

$$V^2(3,3) = 9$$

$$V^2(3,4) = 13$$

$$V^2(4,4) = -\infty$$

So the optimal objective function value is

$$\max \left\{ \begin{array}{llll} V^2(1,1), & 2+V^2(2,\emptyset), & 4+V^2(3,\emptyset), & 2+V^2(4,\emptyset), \\ V^2(1,2), & 0+V^2(2,2), & 13+V^2(3,\emptyset) & 5+V^2(4,\emptyset), \\ V^2(1,3), & 0+V^2(2,3), & 11+V^2(3,3), & 1+V^2(4,\emptyset), \\ V^2(1,4), & 0+V^2(2,4), & 2+V^2(3,4), & 0+V^2(4,4) \end{array} \right\}$$

$$=\max\{12,11,11,-\infty,18,20,20,-\infty,18,20,20,-\infty,10,15,15,-\infty\}$$

$$=20$$

There are several optimal solutions to this problem. One of them is to locate type-x facilities at nodes 2 and 4 and type-y facility at node 2.

Hassin and Tamir (1991) showed that $V^r(j,\emptyset)$, for all $r \leq p$ and $j = 1,\cdots,n$, can be obtained in $O(pn)$ total effort. Note that for each triple $(j,k;l)$, it takes $O(n)$ time to compute $A(j,k;l)$ and $B(j,k;l)$. Therefore, in general $V^r(j,k)$ can be computed in $O(n^2)$ for fixed r, j, and k. The problem on a path network can be solved in a total of $O(pn^4)$ time.

To improve the complexity of the algorithm, we define $u^x(j)$ and $v^x(j)$ as

$$u^x(j) = \min_k\{k|d(k,j) \leq R^x; k \in N\}, \qquad v^x(j) = \max_k\{k|d(k,j) \leq R^x; k \in N\},$$

for each $j \in N$, respectively (See Figure 4.4).

It is easy to see that $u^x(1) = 1$, $v^x(n) = n$ and that $u^x(j)$ and $v^x(j)$ are both monotone in j. Therefore, they can be computed in $O(n)$ time. Similarly, the following sequences can also be obtained in $O(n)$ time.

$$u^y(j) = \min_k\{k|d(k,j) \leq R^y; k \in N\}, \qquad v^y(j) = \max_k\{k|d(k,j) \leq R^y; k \in N\},$$

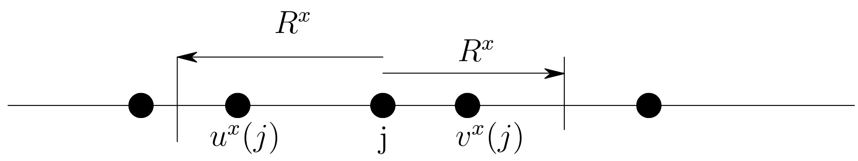


Figure 4.4: $u^x(j)$ and $v^x(j)$

$$u^{xy}(j) = \min_k \{k | d(k,j) \leq \frac{R^{xy}}{2}; k \in N\}, \qquad v^{xy}(j) = \max_k \{k | d(k,j) \leq \frac{R^{xy}}{2}; k \in N\},$$

Define

$$W^x(j) = \sum_{i=1}^{j} w_i^x, \quad W^y(j) = \sum_{i=1}^{j} w_i^y, \quad W^{xy}(j) = \sum_{i=1}^{j} w_i^{xy}$$
$$W^x(0) = 0, \qquad W^y(0) = 0, \qquad W^{xy}(0) = 0$$

The total effort to compute $W^x(j)$, $W^y(j)$, and $W^{xy}(j)$, $j \in N$, is $O(n)$. Now given $j < l \leq k$, the components of $A(j,k;l)$ can be computed as follows.

$$\sum_{i=j}^{l-1} \omega_i^x I^x(\{j,l\},i) = \begin{cases} W^x(v^x(j)) - W^x(j-1) + W^x(l-1) - W^x(u^x(l)-1) & \text{if } v^x(j) < u^x(l) \\ W^x(l-1) - W^x(j-1) & \text{otherwise} \end{cases}$$

$$\sum_{i=j}^{l-1} \omega_i^y I^y(\{k\},i) = \begin{cases} 0 & \text{if } u^y(k) \geq l \\ W^y(l-1) - W^y(u^y(k)-1) & \text{if } l > u^y(k) > j \\ W^y(l-1) - W^y(j-1) & \text{otherwise} \end{cases}$$

$$\sum_{i=j}^{l-1} \omega_i^{xy} I^{xy}(\{l\},\{k\},i) = \begin{cases} 0 & \text{if } u^{xy}(k) \geq l \\ W^{xy}(l-1) - W^{xy}(u^{xy}(k)-1) & \text{if } l > u^{xy}(k) > j \\ W^{xy}(l-1) - W^{xy}(j-1) & \text{otherwise} \end{cases}$$

So $A(j,k;l)$ can be computed in constant time for any given triple $(j,k;l)$ such that $j < l \leq k$.

Given $j \leq k < l \leq n$, we obtain

$$\sum_{i=j}^{n} \omega_i^y I^y(\{k\}, i) = \begin{cases} W^y(v^y(k)) - W^y(u^y(k)-1) & \text{if } u^y(k) \geq j \\ W^y(v^y(k)) - W^y(j-1) & \text{otherwise} \end{cases}$$

$$\sum_{i=j}^{n} \omega_i^{xy} I^{xy}(\{j,l\}, \{k\}, i) = \begin{cases} 0 & \text{if } u^{xy}(k) > j \text{ and } v^{xy}(k) < l \\ W^{xy}(v^{xy}(k)) - W^{xy}(j-1) & \text{if } u^{xy}(k) \leq j \text{ and } v^{xy}(k) \geq l \\ W^{xy}(v^{xy}(j)) - W^{xy}(j-1) & \text{if } u^{xy}(k) \leq j \text{ and } v^{xy}(k) < l \\ W^{xy}(v^{xy}(k)) - W^{xy}(u^{xy}(l)-1) & \text{if } u^{xy}(k) > j \text{ and } v^{xy}(k) \geq l \end{cases}$$

So $B(j,k;l)$ can be computed in constant time for any given triple $(j,k;l)$ such that $j \leq k < l \leq n$. Therefore, $V^r(j,k)$ can be computed in $O(n)$ for fixed r, j, and k and the problem on a path network can be solved in a total of $O(pn^3)$ time. It is easy to check that the problem on a path can be solved in $O(n^2)$ time using complete enumeration if $p = q = 1$.

4.5 The Greedy Adding with Substitution Heuristic

Due to the complexity of the problem (Even for the problem with $\min\{p,q\} = 1$ on a path, it incurs high degree polynomial time algorithm), the "direct approach" of applying a commercial software to the integer linear programming formulation of Section 2 is not practical even for medium-sized problems. To overcome this difficulty, we present the Greedy Adding with Substitution (GAS) heuristic, which has been successfully used on the MCLP (See Church and ReVelle, 1974).

The Greedy Adding with Substitution (GAS) heuristic starts with an empty solution set and adds one pair of facilities at a time until p or q facilities are reached, then add additional facilities one each time until all facilities have been selected. At each iteration, it chooses the pair (or facility) whose addition causes the greatest increase in the objective function value. The GAS algorithm calls a local search algorithm, *procedure* INTERCHANGE, which was originally proposed by Teitz and Bart (1968) for the p-median problem, at the end of each iteration. *Procedure* INTERCHANGE starts with a feasible solution (X, Y).

Procedure INTERCHANGE(X,Y):

Step 1. Fix Y and swap $a \in X$ and $b \in N - X$. Calculate the objective function value using Y and the new X.

Step 2. Swap a and b back, and repeat Step 1 with another pair until all possible swaps are finished.

Step 3. Choose X according to the swap that yields the largest reduction in the objective function value. If there is no improvement, go to Step 4. Otherwise go to step 1.

Step 4. Fix X and swap $a \in Y$ and $b \in N - Y$. Calculate the objective function value using X and the new Y.

Step 5. Swap a and b back, and repeat Step 4 until all possible swaps are finished.

Step 6. Choose Y according to the swap that yields the largest increase in the objective function value. If there is no improvement, go to Step 7. Otherwise go to Step 4.

Step 7. If there is no improvement in a full cycle of Step 1 to Step 6, stop, otherwise go to Step 1.

It is well known that the greedy heuristic has a worst case guaranteed performance for nondecreasing submodular functions (Cornuéjols *et al*, 1990). The MCLP is known to be nondecreasing and submodular (Berman and Krass, 2002), and so is the classic Uncapacitated Facility Location Problem with maximization form. However, the Multi-level Uncapacitated Facility Location Problem with maximization form is not submodular (Barros and Labbé, 1994). The same is true for our problem. To see this, recall the definition of submodularity (Nemhauser and Wolsey, 1990).

> *Let T be a finite set and let Z be a real-valued function defined on the subsets of T. The function Z is submodular if*
>
> $$Z(S' \cup \{t\}) - Z(S') \geq Z(S \cup \{t\}) - Z(S)$$
>
> *for all $S' \subseteq S \subseteq T$ and $t \notin S$.*

Consider now the example depicted in Figure 4.5 where $R^x = R^y = 2$ and $R^{xy} = 5$. For simplicity, we assume that $M = N$. Here we denote each set S by $X \times Y$. Take $S' = \{2\} \times \{4\}$ and $S = \{2\} \times \{3, 4\}$. Hence, $Z(S') = 2$ and $Z(S) = 4$. Assume now a type-x facility is opened at node 1, we have $Z(\{1, 2\} \times \{4\}) = 4$ and $Z(\{1, 2\} \times \{3, 4\}) = 206$, which contradicts the definition of submodularity.

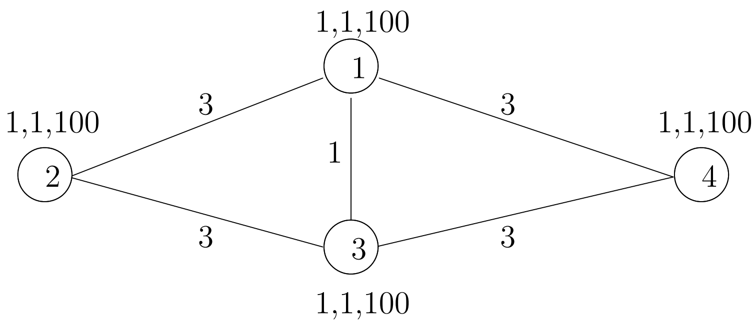


Figure 4.5: The 4-node network showing that submodularity doesn't hold

4.6 Computational experiments

In order to test the greedy algorithms described in the preceding section and compare the two bounds described in Section 3, we conducted some computational experiments. The number of nodes n was set to 50, 80, 100, 150, 200, 300. For each n, four combinations of p and q were considered. In total, we chose 24 combinations of parameter values. For each combination of n, p and q, 10 problem instances were generated, leading to a total of 240 problem instances. All runs were performed on a Pentium IV PC equipped with 2.8GHZ processor and 512M RAM. For all cases, we set $R^{xy} = 2.4R^x = 2.4R^y$.

All procedures were coded in ANSI C. The problem data used in the experiments were generated randomly as follows. The Cartesian coordinates of the nodes were generated over the interval $(0, 100)$ uniformly. Then nodes were connected randomly until a tree was formed.

Finally, a random number of links were added to the tree generated to create a network. All demand weights were generated over the interval (0,1) randomly. The length of each link was calculated using the Euclidean distance formula. For all problem instances, we ensured that no two instances among the total 240 problem cases shared a common random seed.

Computational results are reported in Table 4.1, in which each entry is the average value of 10 random cases. For all cases, CPLEX is called to obtain the optimal objective function value. The following conclusions can be drawn from the computational results:

- The upper bound returned by solving LP-relaxations of three MCLP problems is better than that of Problem P. Although our experiments are not extensive, we can obtain a tighter bound than LP-relaxation quite often.
- Although our problem is not submodular, the greedy heuristic performs very well.
- It is much faster to obtain the upper bound by solving the LP-relaxations of three MCLPs than by solving the LP-relaxation of Problem P.
- When the cardinality of the underlying network is greater than 500, CPLEX runs out of memory on the testing computer.

Table 4.1: Computational results for random graphs

			Relative Error			Running time (in seconds)			
n	p	q	R_{LP1} [a]	R_{LP2} [b]	R_{GAS} [c]	T_{IP} [d]	T_{LP1} [e]	T_{LP2} [f]	T_{GAS} [g]
50	2	5	0.104	0.064	0.0032	0.04	0.01	0.00	0.02
50	3	4	0.090	0.077	0.0028	0.04	0.02	0.00	0.05
50	4	4	0.109	0.140	0.0000	0.04	0.02	0.00	0.08
50	5	5	0.072	0.113	0.0000	0.04	0.01	0.00	0.14
80	2	5	0.130	0.043	0.0000	0.12	0.04	0.00	0.10
80	3	4	0.128	0.082	0.0002	0.16	0.04	0.01	0.19
80	4	4	0.138	0.127	0.0000	0.20	0.05	0.01	0.35
80	5	5	0.138	0.122	0.0000	0.14	0.04	0.00	0.54
100	2	5	0.173	0.082	0.0049	0.27	0.06	0.01	0.18
100	3	4	0.118	0.077	0.0000	0.25	0.06	0.01	0.37
100	4	4	0.139	0.101	0.0000	0.30	0.07	0.00	0.65
100	5	5	0.141	0.119	0.0010	0.35	0.06	0.01	1.03
150	5	5	0.128	0.112	0.0000	0.86	0.16	0.01	3.50
150	5	8	0.139	0.096	0.0016	1.73	0.16	0.00	4.06
150	8	10	0.098	0.081	0.0022	0.96	0.16	0.01	10.38
150	10	10	0.086	0.090	0.0000	0.90	0.16	0.01	16.31
200	5	5	0.170	0.115	0.0000	2.68	0.30	0.02	8.71
200	5	8	0.161	0.088	0.0034	3.84	0.30	0.01	9.55
200	8	10	0.129	0.087	0.0017	3.70	0.31	0.01	24.99
200	10	10	0.113	0.109	0.0001	4.74	0.31	0.00	41.66
300	5	5	0.211	0.092	0.0000	25.69	1.12	0.01	30.09
300	5	8	0.214	0.068	0.0012	134.86	0.80	0.02	31.77
300	8	10	0.166	0.069	0.0034	320.90	0.84	0.01	83.85
300	10	10	0.155	0.102	0.0010	155.42	0.83	0.02	143.15

a: $R_{LP1} = \frac{W_{LP} - W^*}{W^*}$, where W^* is the optimal objective function value of Problem P

b: $R_{LP2} = \frac{W^x_{LP} + W^y_{LP} + W^{xy}_{LP} - W^*}{W^*}$

c: $R_{GAS} = \frac{W_{LP} - GAS}{W^*}$, where GAS is the objective function value by the GAS heuristic

d: running time of Problem P by CPLEX

e: running time of LP-relaxation of Problem P by CPLEX

f: total running times of LP-relaxations of Problems P_x, P_y, and P_{xy} by CPLEX

g: running time of the GAS heuristic

Chapter 5

The Minimum Weighted Covering Location Problem with Distance Constraints

5.1 Introduction

A primary objective in siting service facilities is to cover as much of the potential customer demand as possible. Many types of location models have been developed using covering objectives. One of the most important models is the Maximal Covering Location Problem (MCLP) of Church and ReVelle (1974). A node is covered if there exists a facility within a pre-specified coverage radius. The objective of MCLP is to locate a fixed number of facilities so as to maximize the total coverage. We refer the readers to ReVelle and Williams (2002) for a discussion on the problem and its application.

Maximum covering models are suitable for siting desirable facilities. However, some facilities such as garbage dumps, nuclear reactors and prisons are undesirable or obnoxious. Although these "undesirable" facilities provide service to the society, they may have an adverse effect on the people living nearby. In such instances, maximin or maxisum objectives may be appropriate. For the single facility case, the maximin objective is to find the location of the undesirable facility such that the least (weighted) distance to all nodes is maximized (See Tamir, 1991 and Berman and Drezner, 2000), while the maxisum objective is to site a facility so as to maximize the (weighted) sum of the distances from the facility to all customers located at the nodes of the network (See Church and Garfinkel, 1978). For the multiple facilities case, there are many well-motivated problems, depending on how one defines the objective function. For example, one version of maximin problems is the p-dispersion problem, in which there are p facilities to be located on the network such that the minimum distance between any two facilities is as large as possible (Moon and Chaudhry, 1984). In the maxisum dispersion problem, the objective is to maximize the summation of all distances between the p facilities (Kuby, 1987). The p-dispersion problem and the maxisum dispersion problem on general networks are both $\mathcal{NP}$-hard (see Tamir, 1991). We refer the readers to Erkut and Neuman (1989) and Cappanera (1999) for literature review of undesirable facility location problems.

In this chapter, the problem of locating undesirable facilities on networks employing a coverage type objective function is considered. We call this problem the Minimum Covering Location Problem with Distance Constraints (MCLPDC). Through locating a fixed number of facilities, we wish to minimize the number of covered customers (where a customer is considered covered if her distance to the closest facility is less than a pre-determined radius). When the number of facilities is greater than one, minimal distance constraints are imposed to prevent all facilities to be located at the same point. To motivate the MCLPDC consider the problem of locating nuclear reactors. Nuclear reactors may pose a serious danger to the individuals living nearby. The fewer people covered, the better. Sometimes, for sensitivity and safety reasons, they should be also separated (e.g., if several reactors are clustered in the same region, they may all be attacked by an aggressor). Separation between nuclear reactors can be constrained by a pre-specified minimum distance.

It is easy to see that the MCLPDC is related to the p-dispersion problem. If the solution of the p-dispersion problem is not less than the pre-determined radius of the distance constraint, the MCLPDC is feasible; otherwise, there are no feasible solutions to the MCLPDC.

Another related problem is the Expropriation Location Problem (ELP), which was introduced by Berman, Drezner and Wesolowsky (2003). Each demand point is associated with a given expropriation price. Demand points within a pre-specified distance from a facility will be expropriated. The ELP seeks the location set of a fixed number of obnoxious facilities such that the total cost of expropriation is minimized.

Drezner and Wesolowsky (1994) considered the Minimum Covering Location problem on the plane. Berman, Drezner and Wesolowsky (1996) studied the problem on a network and presented an algorithm to solve the problem. They also analyzed the sensitivity of the coverage radius. Berman, Drezner and Wesolowsky (2003) investigated the Minimum Covering Location problem which they call Problem 2 of ELP on a network and generalized the search for the optimal solution to a dominant set of points. They defined all demand nodes whose *weighted* distance from the facility is less than a pre-specified radius as covered. All three papers only considered the problem of locating a single facility. Therefore, there was no need to incorporate distance constraints between facilities.

Of particular relevance to the current chapter is the Location Set Covering Problem (LSCP)(Toregas *et al*, 1971) where the objective of the LSCP is to find the minimum number of facilities that cover all customer demand. If there is a cost associated with each demand node, the objective of the LSCP with variable weights minimizes the total cost of siting facilities so as to cover the whole network (Schilling, Jayaraman and Barkhi, 1993). As mentioned before, facilities are not always completely desirable. Orloff (1977) recognized this fact and extended the basic LSCP to include distance constraints, which restrict that facilities can be no closer than some specified distances from demand nodes. He finally showed that his model has the same structure as the LSCP.

The problem studied in this chapter is different from the Anti-Covering Location Problem (ACLP) introduced by Moon and Chaudhry (1984). The ACLP attempts to maximize the set of selected location sites so that no two selected sites are within a pre-specified distances. Firstly, in the MCLPDC the total weighted coverage is minimized, while the ACLP employs a maximization objective. Secondly, we fix the number of facilities to be located in the MCLPDC, in contrast to the ACLP, where as many facilities as possible can be sited as long as they don't violate the distance constraint. Thirdly, in the ACLP, the set of nodes is assumed to be the set of potential locations and a node is covered only if there is an open facility at it, i.e., there are no coverage radiuses associated with facilities in the ACLP. In the MCLPDC, a facility covers all demand nodes which are within a pre-determined coverage radius. Finally, the ACLP is a discrete location problem, i.e., the set of potential locations is given. The MCLPDC is a network location problem, which means that facilities can be located at any point of the network (we will show that the MCLPDC can be reduced to a discrete location problem in the next section).

Our problem is very similar to the Limited Impact Location Problem (LILP) addressed by Murray *et al* (1998). The LILP considers equity of impact in locating undesirable facilities on networks since it is sometimes not equitable to cluster facilities. The objective of the LILP is to minimize the total covered demand through locating a fixed number of facilities such that each demand can be covered by at most one facility. Although both LILP and MCLPDC employ the same objective through siting a fixed number of facilities, we can see the difference in the example depicted in Figure 5.1, where the number next to each link is the link length. Suppose that δ is a very small positive number and the coverage radius is 1. Obviously, it is feasible to open both facilities by the LILP and the total demand covered by these two open facilities is zero, but it is unallowable to open both simultaneously by the MCLPDC if the facilities are required to be at a distance of more than δ from each other.

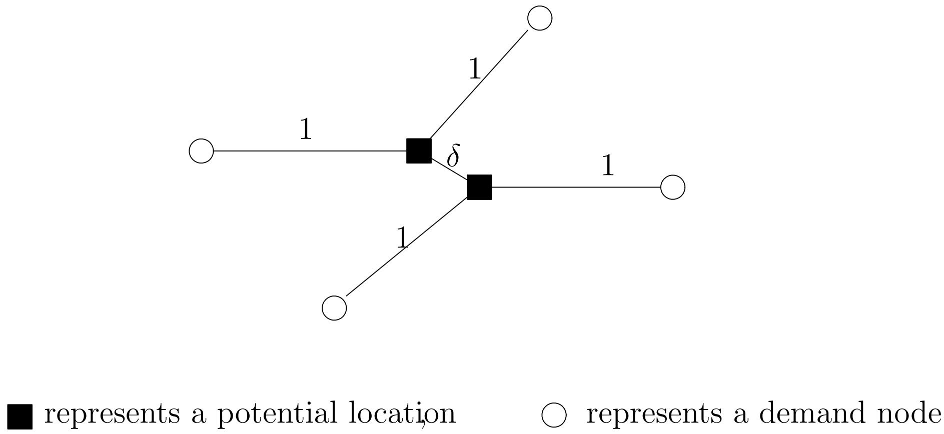


Figure 5.1: A simple network on which both facilities can be opened by the LILP

The objective of this chapter is to extend the Minimum Covering Location Problem to siting general number of facilities on networks. The remainder of this chapter is organized as follows. In the next section, we study the MCLPDC on a general network and show that it is $\mathcal{NP}$-hard to determine the feasibility of the problem. Section 3 contains several mathematical program formulations for the MCLPDC. Several heuristics are presented in Section 4. Computational experiments and concluding remarks are included in the last two sections respectively.

5.2 The MCLPDC on a General Network

Let $G = (N, L)$ be a network, where N is the set of nodes with $|N| = n$ and L is the set of links. Customers are located at nodes of the network. Let $c_i (c_i \geq 0)$ be the weight associated with node i. The weight c_i can represent the population at node i. Denote by (a, x, b) a non-nodal point at a distance of x from node a on link (a, b). Assume that $0 \leq x \leq l(a, b)$, where $l(a, b)$ is the length of link (a, b).

Let $Y = (y_1, \cdots, y_p)$ be the location set of facilities. Let r and s be two given positive numbers. We say that node i is covered if $d(i, Y) < r$, where $d(i, Y) = \min_{1 \leq j \leq p}\{d(i, y_j)\}$. Distance constraints imposed between facilities are $d(y_j, y_k) \geq s$ for all $j \neq k \in \{1, \cdots, p\}$. The objective of the MCLPDC is to site p facilities so as to minimize the total weighted coverage subject to the requirement that no two facilities are allowed to be located less than a pre-specified distance s from each other.

For the single facility location problem, i.e. $p = 1$, Berman, Drezner and Wesolowsky (2003) showed that there exists at least one optimal solution belonging to the Network Intersect Point Set(NIPS). Any point on the network that is r distance away from demand point $i \in N$ is a NIP. The NIPS is the set of all NIPs plus all demand points (see also Church and Meadows, 1979). When $p > 1$, the claim doesn't hold any more. As an example consider the 3-node network depicted in Figure 5.2, where the numbers next to the links are link lengths. When $r = 4$, the NIPS is $\{1, 2, 3, (1, 2, 2), (1, 4, 2),\ (2, 4, 3), (2, 6, 3), (1, 2, 3), (1, 4, 3)\}$. Two facilities are to be located on the network. When $s = 11$, there is no feasible solution in NIPS, but it is obvious that any two points with a distance of 11 from each other on the network are feasible. Let's consider another example in which there are feasible solutions in NIPS, but none of them is optimal. In Figure 5.3, the numbers next to the nodes are the node weights, $r = 1$ and $s = 2.5$. Three facilities are to be located on the network. The NIPS is $\{1, 2, 3, 4, 5, (2, 1, 3), (2, 2, 3), (3, 0.5, 5), (3, 1, 5)\}$. It is easy to verify that the unique optimal solution is $\{1, 4, (2, 1.5, 3)\}$, which is not in the NIPS, and that $\{1, 4, 5\}$ is a feasible solution in the NIPS.

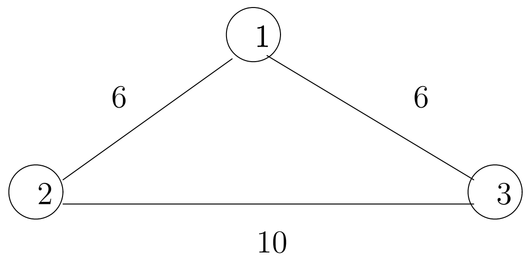


Figure 5.2: A simple network without feasible solution in the NIPS when $r = 4$, $s = 11$.

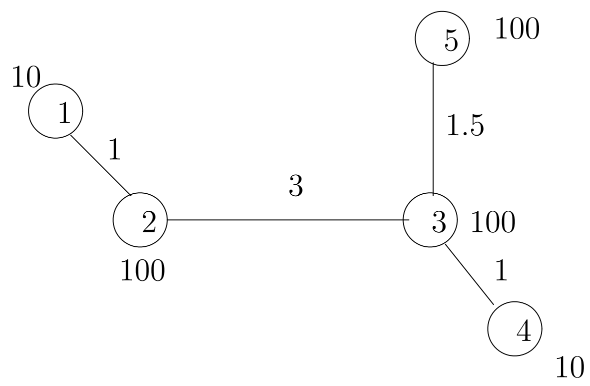


Figure 5.3: A tree network with optimal solution not in the NIPS when $r = 1$, $s = 2.5$.

Define any point on the network with a distance of $k * s$ away from point $j \in$NIPS as a Multiple-constraint-distance Intersect Point (MIP), where k is an integer. The MIPS is the set of all Multiple-constraint-distance Intersect Points. To obtain all MIPs relating to point $j \in$ NIPS, we can find the longest paths from j to all nodes, then intersect these paths with distance s from j repeatedly. To show it, consider the example depicted in Figure 5.3. The longest path from node 1 to node 5 is $1 - 2 - 3 - 5$, so points $(2, 1.5, 3)$ and $(3, 1, 5)$ are in the MIPS. The longest path from node 1 to node 4 is $1-2-3-4$, so point $(2, 1.5, 3)$ and node 4 are in the MIPS, and $(2, 1.5, 3)$ has already been counted. Since the distance between node 1 to node 2 is less than 2.5 and the longest path from node 1 to node 3 is a part of paths from node 1 to nodes 4 and 5, we don't need to consider these paths. Therefore, the elements in the MIPS corresponding to node 1 are $(2, 1.5, 3), (3, 1, 5)$ and node 4.

We say that distance constraints between facilities are consistent if there exists a location set that satisfies them.

Theorem 5.1 *If distance constraints are consistent for the MCLPDC, then there must exist an optimal solution that consists entirely of points belonging to the $MIPS \cup NIPS$.*

Proof. Suppose that $Y^\star = (y_1^\star, \cdots, y_p^\star)$ is an optimal solution. Without loss of generality, we assume that $y_1^\star, \cdots, y_q^\star$ $(q \leq p)$ are not in the $MIPS \cup NIPS$. Suppose that $y_j^\star$ $(1 \leq j \leq q)$ is located in interval (a_j, b_j), where $a_j, b_j \in MIPS \cup NIPS$, and there is no other point in (a_j, b_j) belonging to the $MIPS \cup NIPS$. Denote by d_j^l (d_j^r) the distance from $y_j^\star$ to a_j (b_j). We will show that moving the facility $y_j^\star$ to a_j or b_j won't make the objective function worse.

Suppose that d_k^l $(1 \leq k \leq q)$ is the smallest distance among all d_j^ls and d_j^rs $(j = 1, \cdots, q)$. In Figure 5.4, A and B are two points with a distance of s from a_k and $y_k^\star$ respectively. Therefore, $A \in MIPS \cup NIPS$ and $d(A, B) = d_k^l$. If d_k^l is strictly the smallest among d_j^l, then any facilities on the left of $y_k^\star$ must be on the left of point A. Therefore, $y_k^\star$ can be moved to a_k without violating any distance constraint.

Now consider the case that d_k^l is not strictly the smallest and there is a facility u on the left of $y_k^\star$ with $d_u^l = d_k^l$. If $d(y_u^\star, y_k^\star) \neq s$, $y_u^\star$ must be on the left of point A because if $y_u^\star$ is in the interval (A, B), it will violate the assumption of the smallest d_k^l. Therefore, $y_k^\star$ can be moved to a_k without violating any distance constraint. If $d(y_u^\star, y_k^\star) = s$, then $y_u^\star$ must be located at point B, so we can change $y_k^\star$ to a_k and $y_u^\star$ to A simultaneously.

Using the same argument, any point not in the $MIPS \cup NIPS$ can be replaced iteratively with its closest point in the $MIPS \cup NIPS$ without increasing the objective function value.

■

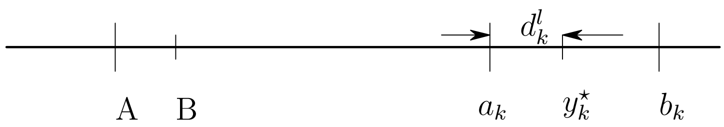


Figure 5.4: Moving $y_k^\star$ to a_k

It is easy to verify that the previous two examples conform to Theorem 5.1. Although the $MIPS \cup NIPS$ is finite, it may be too huge to be tractable. Moreover, column reduction techniques of the zero-one covering matrix (rows corresponding to noes and columns to points of $MIPS \cup NIPS$) that can be used to reduce NIPS in the maximal covering location problem cannot be used in our problem because of the distance constraints. In the remainder of the chapter it is assumed that the triangle inequality always holds. Denote by M ($|M| = m$) the set of potential locations. Since we can make any point in $MIPS \cup NIPS$ as a demand node with weight equal to zero, without loss of generality, we can assume that M is the set of nodes in the new network.

Proposition 5.1 *It is $\mathcal{NP}$-complete to determine the consistency of distance constraints.*

Proof. It is well-known that the maximum independent set problem is $\mathcal{NP}$-complete (Gary and Johnson, 1979). An independent set is a set of nodes such that for any pair of nodes, there is no link between them. Given a graph $H = (V, E)$, the maximum independent set problem is to determine a subset $I \subseteq V$ of largest possible size such that no two nodes in I are joined by a link in E. Suppose that we want to know whether the graph $H = (V, E)$ contains an independent set of size p. Construct the graph $G = (N, L)$ as follows: set $N = V$ and $L = E$, and $d_{ij} = 1$ for all $(i, j) \in L$. Consider the MCLPDC for graph G with potential set of facilities $M = N$ with $s = 2r = 2$. Since any two location must be at a distance of at least 2 from each other, distance constraints of the MCLPDC are consistent if and only if there exists an independent set of size p in H. Hence, the existence of a polynomial time algorithm to determine the consistency of distance constraints for the MCLPDC would contradict the $\mathcal{NP}$-completeness of the maximum independent set problem. ■

To check the consistency of distance constraints, we can solve the unweighted Anti-Covering Location Problem on G with potential locations M so that no two located facilities are within a distance of s. We can also solve the maximum independent set problem on G', where $G' = (N', L')$ such that $N' = M$ and $L' = \{(j, k)|d(j, k) < s; j, k \in M, j \neq k\}$. It is easy to see that these two approaches are equivalent. If the size of the maximum independent set of G' is less than p, the distance constraints are inconsistent. Although the problem of checking the consistency of distance constraints is $\mathcal{NP}$-complete, some heuristics can be used to solve the maximum independent set problem. If they return an independent set of size larger than or equal p, the distance constraints are consistent for sure. We refer the reader to Chaudhry, McCormick and Moon (1986) and Murray and Church (1997) for heuristic approaches to the Anti-Covering Location Problem, and Nemhauser and Sigismondi (1992) for the cutting plane/branch-and-bound algorithm for the maximum independent set problem.

5.3 Mathematical formulations for the MCLPDC

In this section, we present several mathematical formulations for the MCLPDC. The first formulation of the MCLPDC is expressed as follows:

$$MCLPDC1 \quad \min \sum_{i=1}^{n} c_i x_i$$

$$s.t. \quad y_j + y_k \leq 1 \qquad \text{for all } j,k \in M \ni d(j,k) < s, \tag{5.1}$$

$$x_i \geq y_j \qquad i \in N, j \in M_i \tag{5.2}$$

$$\sum_{j=1}^{m} y_j = p \tag{5.3}$$

$$x_i, y_j \in \{0,1\} \qquad i \in N, j \in M \tag{5.4}$$

where

$$x_i = \begin{cases} 1 & \text{If } i \in N \text{ is covered by a facility with a distance of less than } r \text{ ,} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{If there is a facility established at point } j \in M \text{ ,} \\ 0 & \text{otherwise.} \end{cases}$$

$M_i = \{j \in M | d(j,i) < r\}$

Constraints (5.1) state that two facilities can't be established if the distance between them is less than s. Constraints (5.2) ensure that node i will be covered ($x_i = 1$) if some facility is opened in M_i. If there is no open facility in M_i, x_i will equal zero due to the minimization property. Constraint (5.3) specifies exactly p open facilities on the network. Finally, Constraints (5.4) establish binary restrictions on the decision variables.

It is well known that the pairwise constraints of (5.1) are actually a special case of clique constraints. A clique is a fully connected subgraph. To introduce the clique formulation of the MCLPDC, we construct again graph $G' = (N', L')$ such that $N' = M$ and $L' = \{(j,k) | d(j,k) < s; j,k \in M, j \neq k\}$. Note that G' may not be a connected graph. A maximal clique is a clique which is not a subset of any other larger cliques. Denote by MC the set of all maximal cliques of G'. Note that "maximal" is different from "maximum". As adjective, "maximum" means "maximum-sized", and "maximal" means "no larger one contains this one" (see page 29 of West, 2001). As an example consider G' which is the simple 4-node network depicted in Figure 5.5. $\{4,3\}$ is a maximal clique, but the maximum clique is $\{1,2,3\}$. In this example, $MC = \{\{1,2,3\},\{3,4\}\}$. Now we are ready to give the second formulation of the MCLPDC.

$$MCLPDC2 \quad \min \sum_{i=1}^{n} c_i x_i$$

$$s.t. \quad (5.2), (5.3), \text{and } (5.4)$$

$$\sum_{k \in C} y_k \leq 1 \qquad \text{for all } C \in MC \tag{5.5}$$

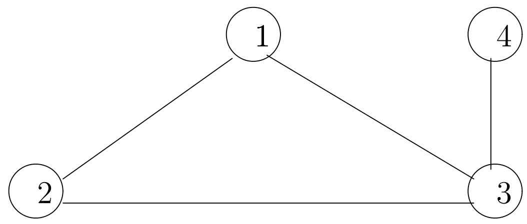


Figure 5.5: A 4-node network

The LP relaxation of MCLPDC2 provides at least as good and usually a tighter bound than the LP relaxation of MCLPDC1 since each $(j,k) \in L'$ (i.e., $d(j,k) < s$) is contained in some maximal clique of G'. As an example consider the example depicted in Figure 5.6, where $c_i = 1$ and all links are of unit length. When $r = 1$, $s = 1.5$, $p = 2$, and $M = N$, it is easy to verify that the unique feasible solution is to locate 2 facilities at nodes 1 and 4, and that the objective function value is 2. Although the objective function value of the LP relaxation of MCLPDC1 is also 2, each variable is 1/2 in the optimal solution of the LP relaxation. Notice that $MC = \{\{1,2,3\},\{2,3,4\}\}$. Using MCLPDC2, at most one facility can be located in $\{1,2,3\}$ since the distance between any two points is less than 1.5. The same is true for $\{2,3,4\}$. Now replace constraint set (5.1) with

$$y_1 + y_2 + y_3 \leq 1,$$

$$y_2 + y_3 + y_4 \leq 1,$$

and solve the LP relaxation of MCLPDC2 again and an all-integer solution is obtained, which is obviously optimal.

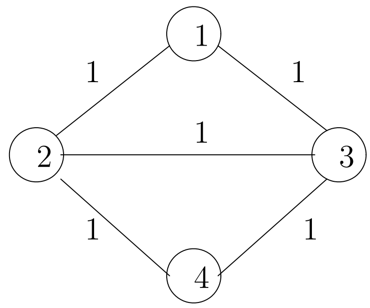


Figure 5.6: A simple network where LP relaxation of MCLPDC1 generates fractional solution

As shown in the preceding example, we observed that in general the LP relaxation of MCLPDC2 was usually more integer friendly than that of MCLPDC1. However, it is possible that there exists an exponential number of maximal cliques. This means that it is generally impossible to impose all necessary constraints for the MCLPDC2.

The third integer program formulation is given as follows:

$$MCLPDC3 \quad \min \sum_{i=1}^{n} c_i x_i$$

$$s.t. \quad (5.2), (5.3), \text{and } (5.4)$$

$$p(1 - y_j) \geq \sum_{k \in Q_j} y_k \qquad j \in M \tag{5.6}$$

where for $j \in M$, $Q_j = \{k \in M | d(k,j) < s, k \neq j\}$.

Constraints (5.6) also forbid two established facilities closer than s. If a facility is established at $j \in M$, y_k must take a value 0 for all $k \in Q_j$. When $y_j = 0$, at most p facilities can be sited in Q_j. This constraint set was originally used by Moon and Chaudhry (1984) to formulate the Anti-Covering Location Problem. Obviously, MCLPDC3 has fewer constraints than MCLPDC1. As reported for the Anti-Covering Location Problem, the LP relaxation approach using constraint set (5.6) is somewhat disappointing. Murray and Church (1997) improved Moon and Chaudhry (1984)'s formulation, and applied the Lagrangian relaxation heuristic successfully to the Anti-Covering Location Problem. The main idea of the following two formulations is based on their work for the Anti-Covering Location Problem.

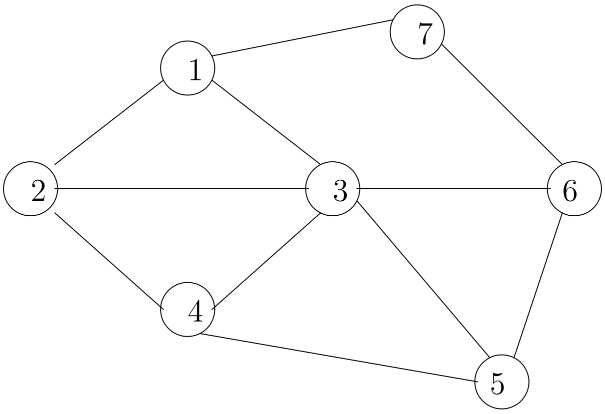
Our fourth formulation of the MCLPDC is as follows:

$$MCLPDC4 \quad \min \sum_{i=1}^{n} c_i x_i$$

$$s.t. \quad (5.2), (5.3), \text{and } (5.4)$$

$$n_j(1 - y_j) \geq \sum_{k \in Q_j} y_k \qquad j \in M \tag{5.7}$$

where n_j is the minimum coefficient necessary to impose spatial restrictions between members of the set Q_j. Recall the graph $G' = (N', L')$ which we constructed before, where $N' = M$ and $L' = \{(j,k)|d(j,k) < s; j,k \in M, j \neq k\}$. Let G'_j be the subgraph of G' induced by $Q_j \cup \{j\}$ and q_j is the cardinality of the maximum independent set of G'_j. The maximum independent set problem is $\mathcal{NP}$-complete, but G'_j is usually a small graph, so q_j can be obtained with relative ease by complete enumeration. Consider the example of G' depicted in Figure 5.7 with $p = 3$ and $M = N$. G'_2 is depicted in the right part of Figure 5.7. It is easy to verify the maximum independent set of G'_2 is $\{1, 4\}$. Therefore, $q_2 = 2$. Once q_j is obtained, the value of n_j can be determined by $n_j = \min\{p, q_j\}$ easily. Obviously, constraint (5.7) of MCLPDC4 is tighter than constraint (5.6) of MCLPDC3.



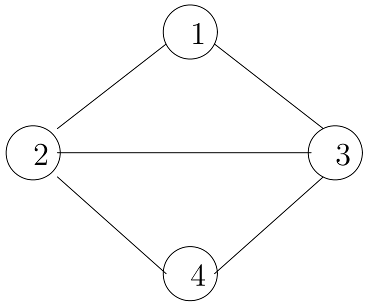


Figure 5.7: G' (left part) and G'_2 (right part)

Let C_j be a maximal (not necessarily maximum) clique which includes node j of G'. As an example consider again the example depicted in Figure 5.7. There are 2 maximal cliques containing node 1, $\{1, 2, 3\}$ and $\{1, 7\}$. We greedily generate C_j for each j. Therefore, C_1 is either $\{1, 2, 3\}$ or $\{1, 7\}$. Denote by Q'_j the subset of $Q_j \cup \{j\}$ after removing all elements in C_j, i.e., $Q'_j = \{k|k \in Q_j \cup \{j\}, k \notin C_j\}$. Note that Q'_j can be an empty set. Similar to n_j, we can obtain the coefficient n'_j imposed on Q'_j. Consider the example of G' depicted in Figure 5.7 with $p = 3$ and $M = N$ again. $Q_6 \cup \{6\} = \{3, 5, 6, 7\}$ with $n_6 = 2$ and if C_6 is set as $\{3, 5, 6\}$, $Q'_6 = \{7\}$ and it is obvious that $n'_6 = 1$. Since $Q_j = Q'_j \cup (C_j \backslash \{j\})$ and C_j is a clique, we have $n'_j \leq n_j \leq n'_j + 1$ for all $j \in M$, but it is not necessary that $n_j = n'_j + 1$. For example, in the example depicted in the following figure, if $\{1, 2, 3\}$ is chosen as C_1, then $Q'_1 = \{4, 5\}$. Therefore, in this example $n_1 = n'_1 = 2$.

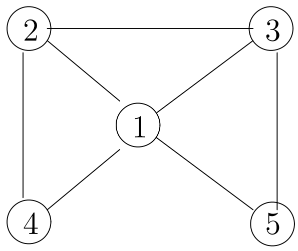


Figure 5.8: $n_1 = n_1'$

Denote by L_1 the set of all C_j $(j \in M)$ and by L_2 the set of all nonempty Q_j' $(j \in M)$. Note that the cardinality of L_1 and L_2 may be less than $|M|$ since some points in M may share the same maximal clique and that some Q_j' may be empty.

$$MCLPDC5 \quad \min \sum_{i=1}^{n} c_i x_i$$

$$s.t. \quad (5.2), (5.3), \text{and } (5.4)$$

$$\sum_{k \in C_j} y_k \leq 1 \qquad \text{for all } C_j \in L_1 \tag{5.8}$$

$$n_j'(1 - y_j) \geq \sum_{k \in Q_j'} y_k \qquad \text{for all } Q_j' \in L_2 \tag{5.9}$$

When $s \geq 2r$, constraint (5.2) in the previous mathematical formulations can be replaced with the following more compact one:

$$\sum_{j \in M_i} y_j = x_i \qquad i \in N \tag{5.10}$$

It is guaranteed that no two facilities are located in M_i due to triangle inequality property and the fact that $s \geq 2r$. Therefore, constraint (5.10) also ensures that node i will not be covered ($x_i = 0$) if there is no open facilities in M_i.

Substituting x_i in the objective function with the left hand side of (5.10), we can drop constraint (5.10) and obtain the new objective function

$$\min \sum_{i=1}^{n} c_i \sum_{j \in M_i} y_j = \min \sum_{j=1}^{m} (\sum_{k \in E_j} c_k) y_j,$$

where $E_j = \{k \in N | d(j,k) < r\}$ for each $j \in M$. Let $K = \max_{j=1,\cdots,m}\{\sum_{k \in E_j} c_k\}$. Since $\sum_{j=1}^{m} y_j = p$, the MCLPDC is equivalent to the following problem:

$$\begin{aligned} &\max \sum_{j=1}^{m} (K - \sum_{k \in E_j} c_k) y_j \\ s.t. \quad &(5.1), \text{ or } (5.5), \text{ or } (5.6), \text{ or } (5.7), \text{ or } (5.8) \text{ and } (5.9) \\ &\sum_{j=1}^{m} y_j = p \\ &y_j \in \{0,1\} \qquad\qquad j \in M \end{aligned}$$

If the constraint $\sum_{j=1}^{m} y_j = p$ is dropped, the problem is exactly the Anti-Covering Location Problem, which is a well-studied problem. Unfortunately, we can't get too much information from the solution to the Anti-Covering Location Problem. Suppose that there are q open facilities in the solution to the Anti-Covering Location Problem. If $q < p$, it is not certain whether or not the distance constraints are inconsistent. For example, in the following figure where the number next to each node is $K - \sum_{k \in E_j} c_k$, suppose that $M = N$ and all links are of unit length. If the distance constraint is set as $s = 1.5$, it is easy to see that the solution to the Anti-Covering Location Problem is to locate a single facility at the central node ($q = 1$), but the distance constraints are consistent for the MCLPDC unless

$p > 4$.

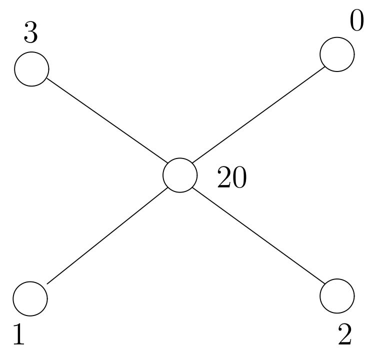


Figure 5.9: An example where the number of facilities located by the ACLP is less than p

If $q > p$, the consistency of the distance constraints is guaranteed, but generally we can't obtain the optimal solution to the MCLPDC from the solution to the Anti-Covering Location Problem. For example, in the following example suppose that all links are of unit length, $M = N$, $s = 1.5$ and $p = 3$. It is easy to verify that the optimal solution to the Anti-Covering Location Problem is to open 4 facilities at nodes 1, 2, 4, and 5, but the optimal solution to the MCLPDC is to open facilities at nodes 1, 2, and 6.

5.4 Solution approaches to the Problem

5.4.1 The Greedy Heuristic (GH)

The Greedy Heuristic tries to choose potential locations that are the "best" at each iteration. This algorithm starts with an empty solution set Y and adds facilities to Y until p facilities

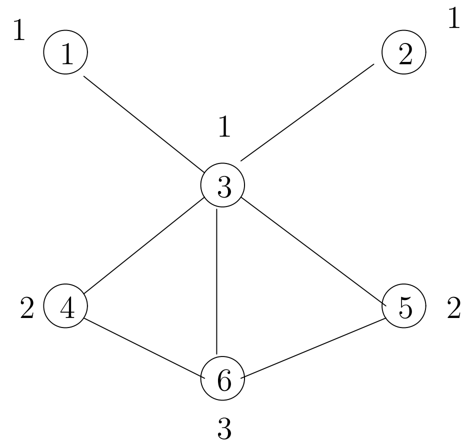


Figure 5.10: An example where the solution to the MCLPDC is not in which to the ACLP

have been selected or there are no more facilities that can be added to Y. At each iteration, it chooses the facility whose addition causes the least increase in the objective function value without violating distance constraints. If distance constraints are consistent, but the cardinality of the set Y is less than p, the Greedy Heuristic fails to find a feasible solution. As an example, let us consider the simple network depicted in Figure 5.2, where $c_1 = 1$, and $c_2 = c_3 = 2$. When $M = N$, $r = 4$, $s = 9$, and $p = 2$, it is obviously that distance constraints are consistent. The set returned by the Greedy Heuristic is $\{1\}$, so this algorithm fails to find a feasible solution.

Note that when the Greedy Heuristic returns a feasible solution, in contrast to other location problems where the worst case performance ratio (relative error) with the Greedy Heuristic is bounded, the solution quality may be very bad. As an example, consider the path network depicted in Figure 5.11, where the lengths of links are the numbers next to the links and the weights are the numbers next to the nodes. If $M = N$, $p = 2$, $r = 2$ and $s = 5$, the objective function value returned by the Greedy Heuristic is $K + 1$, while the optimal objective function value is 4. The worst case performance ratio tends to infinity as K goes to infinity. However, as it will be shown later, the performance of this heuristic in general is good.

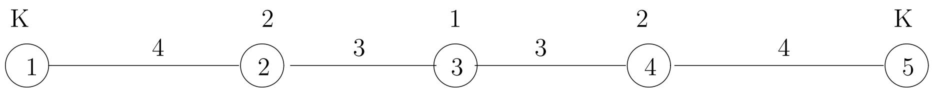


Figure 5.11: a 5-node path network where K is a large number

5.4.2 The Tabu Search Heuristic

Tabu search has been applied to location problems successfully. For example, Rolland, Schilling, and Current (1996) provides a tabu search algorithm for the p-median problem; Al-Sultan and Al-Fawzan (1999) , and Michel and Van Hentenryck (2002) present two different tabu search methods for solving the uncapacitated facility location problem. A comprehensive tutorial on tabu search can be found in Glover and Laguna (1997).

The tabu search proceeds from the terminal solution of the Greedy Heuristic algorithm. Suppose Y is the starting solution, which doesn't have to be feasible. Define a swap as one-for-one exchange of points in Y and $M - Y$. The heuristic chooses the best solution among all swaps as the starting solution of the next iteration, given that this solution is not tabu or that it is tabu but it has a better objective function value than the incumbent solution.

After one iteration of swaps, if there is no feasible solution, the algorithm stops; otherwise, add the selected solution to the tabu list and repeat the iteration until a fixed number of iterations have passed. A circular array is employed to store the potential locations entered the tabu list. If the list is full, the head of the array will be removed from tabu.

The Tabu Search Heuristic algorithm:

step 1. Select the solution of the Greedy Heuristic algorithm as the starting solution and as the best found solution. The tabu list is empty.

step 2. set i:=1.

step 3. Consider all swaps. If no solution is feasible, stop; otherwise, choose the ith best solution as the starting solution of the next iteration. If this solution is not in the tabu list, add it to the tabu list; if this solution is in the tabu list but it is better than the best found solution, update the best found solution so far and empty the tabu list; otherwise set i:=i+1 and repeat step 3.

step 4. Repeat steps 2-3 until the prespecified number of iterations is reached.

5.4.3 The Lagrangian Relaxation Heuristic

In this subsection we describe two Lagrangian relaxation approaches to solve the MCLPDC. One is for the case of $s < 2r$, the other is for $s \geq 2r$. The Lagrangian scheme has been successfully applied to the p-median problem (Narula, Ogbu and Samuelsson, 1977; Daskin, 1995) and other integer programming problems (Cornuejols, Fisher and Nemhauser, 1977; Fisher, 1981; Pirkul and Schilling, 1991). Computational experiments by Murray and Church (1997) showed that constraints (5.8) and (5.9) are better than constraint (5.7) for the Anti-Covering Location Problem when the Lagrangian relaxation method is applied. Therefore, our algorithm is based on MCLPDC5.

If $s \geq 2r$, as we discussed at the end of the previous section, the MCLPDC5 can be reduced to the following integer program.

$$\max \sum_{j=1}^{m} (K - \sum_{k \in E_j} c_k) y_j$$

$$s.t. \quad \sum_{k \in C_j} y_k \leq 1 \qquad \text{for all } C_j \in L_1 \tag{5.11}$$

$$n'_j (1 - y_j) \geq \sum_{k \in Q'_j} y_k \qquad \text{for all } Q'_j \in L_2 \tag{5.12}$$

$$\sum_{j=1}^{m} y_j = p$$

$$y_j \in \{0, 1\} \qquad j \in M$$

By relaxing constraints (5.11) and (5.12), we have the Lagrangian dual.

$$\min_{\lambda,\, \theta} \max_{y_j} \sum_{j=1}^{m} (K - \sum_{k \in E_j} c_k) y_j + \sum_{j=1}^{|L_1|} \lambda_j (1 - \sum_{k \in C_j} y_k) + \sum_{j=1}^{|L_2|} \theta_j (n'_j - n'_j y_j - \sum_{k \in Q'_j} y_k)$$

$$s.t. \qquad \sum_{j=1}^{m} y_j = p$$

$$y_j \in \{0, 1\} \qquad j \in M$$

For fixed values of the Lagrangian multipliers, the dual problem is easy to solve. The objective function can be rewritten in the format $\sum_j a_j y_j + b$, where a_j and b are constant numbers for fixed multipliers and a_j are sorted in descending order. The solution to the dual problem is $y_j = 1$ for the first p y_j's and $y_j = 0$ for other y_j's. The subgradient optimization algorithm suggested is similar to the method by Murray and Church (1997). Readers can refer to their chapter for details.

If $s < 2r$, we consider the initial version of MCLPDC5.

$$
\begin{aligned}
& \min \sum_{i=1}^{n} c_i x_i \\
s.t. \quad & \sum_{k \in C_j} y_k \leq 1 && \text{for all } C_j \in L_1 && (5.13) \\
& n'_j(1 - y_j) \geq \sum_{k \in Q'_j} y_k && \text{for all } Q'_j \in L_2 && (5.14) \\
& x_i \geq y_j && i \in N, j \in M_i && (5.15) \\
& \sum_{j=1}^{m} y_j = p && && (5.16) \\
& x_i, y_j \in \{0, 1\} && i \in N, j \in M && (5.17)
\end{aligned}
$$

By relaxing constraints (5.13), (5.14) and (5.16), we have the Lagrangian dual of the MCLPDC, which we call Problem L.

$$
\max_{\lambda,\, \theta} \min_{x_i,\, y_j} \sum_{i=1}^{n} c_i x_i + \sum_{j=1}^{|L_1|} \lambda_j \Big(\sum_{k \in C_j} y_k - 1\Big) + \sum_{j=1}^{|L_2|} \theta_j \Big(\sum_{k \in Q'_j} y_k - n'_j + n'_j y_j\Big) + \eta\Big(p - \sum_{j=1}^{m} y_j\Big)
$$

$$
\begin{aligned}
s.t. \quad & x_i \geq y_j && i \in N, j \in M_i && (5.18) \\
& x_i, y_j \in \{0, 1\} && i \in N, j \in M && (5.19)
\end{aligned}
$$

For fixed values of the Lagrangian multipliers, λ_j $(j = 1, \cdots, |L_1|)$, θ_j $(j = 1, \cdots, |L_2|)$ and η, we have a minimization problem, denoted by $L(\lambda,\ \theta,\ \eta)$ (λ and θ are vectors).

A matrix is said to be Totally Unimodular, termed TU, if the determinant of each square submatrix is 0, 1 or -1. It is well known that if matrix A is a TU matrix and b is a integral vector, then the polyhedron $P(A) = \{x | Ax \leq b\}$ is integral (Schrijver, 1986). Therefore, if the constraint matrix of Problem $L(\lambda,\ \theta,\ \eta)$ is TU, we can simply relax the binary constraints and solve it as a linear program. To prove that constraint matrix of Problem $L(\lambda,\ \theta,\ \eta)$ is TU, we need following theorem which can be found in Chapter 19 of Schrijver (1986).

Theorem 5.2 *Let A be a matrix whose entries is 0, 1 or -1. Let S be the set of square submatrices of A with each row sum and column sum even. If the sum of the entries of any $B \in S$ is divisible by four, then A is TU.*

Now we are ready to have the following theorem.

Theorem 5.3 *The constraint matrix of Problem $L(\lambda,\ \theta,\ \eta)$ is TU.*

Proof. Let B be any square submatrix of the constraint matrix of Problem $L(\lambda,\ \theta,\ \eta)$ with even row and column sums. It is easy to see since that each row sum of B is zero, the sum of the entries of B must be zero, which is obviously divisible by 4. According to Theorem 5.2, the constraint matrix of Problem $L(\lambda,\ \theta,\ \eta)$ is TU. ■

It is well-known that in the uncapacitated facility location problem

$$\min \sum_i \sum_j d_{ij} x_{ij} + \sum_j f_j y_j$$

$$s.t. \qquad \sum_j x_{ij} = 1 \qquad \text{all } i \tag{5.20}$$

$$x_{ij} \leq y_j \qquad \text{all } i, j \tag{5.21}$$

$$x_{ij}, y_j \in \{0, 1\} \qquad \text{all } i, j \tag{5.22}$$

if constraint set (5.20) is relaxed, the Lagrangian dual problem satisfies the condition of total unimodularity (Cornuejols, Fisher and Nemhauser, 1977). Note that in the uncapacitated facility location problem, if the constraint $\sum_j y_j = p$ is added to the dual problem, total unimodularity still holds (see Cornuejols, Fisher and Nemhauser, 1977), but if only constraints (5.13), (5.14) in the MCLPDC are relaxed when $s < 2r$, the Lagrangian dual problem is not totally unimodular any more.

It follows from Geoffrion (1974) and Theorem 5.3 that the optimal objective function value of Problem L is equal to the optimal value of the linear relaxation of MCLPDC5. Although the bound returned by the Lagrangian relaxation is not satisfactory, a good solution may be obtained in iterations of the algorithm. Now constraints (5.19) can be replaced with $0 \leq x_i \leq 1,\ 0 \leq y_j \leq 1$ for all $i,\ j$.

To adjust the Lagrangian multiplier values, the subgradient optimization method is utilized. Given initial multipliers λ_j^0 $(j = 1, \cdots, |L_1|)$, θ_j^0 $(j = 1, \cdots, |L_2|)$ and η^0, a sequence of multipliers is generated by

$$\lambda_j^{n+1} = \max\{0, \lambda_j^n + t^n(\sum_{k\in C_j} y_k - 1)\}, \tag{5.23}$$

$$\theta_j^{n+1} = \max\{0, \theta_j^n + t^n(\sum_{k\in Q'_j} y_k - n'_j + n'_j y_j)\}, \tag{5.24}$$

$$\eta^{n+1} = \max\{0, \eta^n + t^n(p - \sum_{j=1}^{m} y_j)\}, \tag{5.25}$$

where y_j is the optimal solution to Problem $L(\lambda^n,\ \theta^n,\ \eta^n)$ and t^n is a positive scalar step size defined as follows:

$$t^n = \frac{\alpha^n(Z^u - Z^l)}{\sum_{j=1}^{|L_1|}(\sum_{k\in C_j} y_k - 1)^2 + \sum_{j=1}^{|L_2|}(\sum_{k\in Q'_j} y_k - n'_j + n'_j y_j)^2 + (\sum_{j=1}^{m} y_j - p)^2} \tag{5.26}$$

where α^n is a scalar satisfying $0 < \alpha^n \leq 2$. Z^u is the best-known upper bound and Z^l is the best-known lower bound. We start with $\alpha^0 = 2$ and cut it by half every time the objective function value of Problem $L(\lambda^n,\ \theta^n,\ \eta^n)$, $Z^l(\lambda^n,\ \theta^n,\ \eta^n)$, fails to increase after a fixed number of iterations. The algorithm is terminated either when the upper and lower bounds are sufficiently close to each other or when the iteration limit is reached. The algorithm starts with $\lambda_j^0 = 0$, $\theta_j^0 = 0$ for all j and $\eta^0 = 0$. The algorithm can be stated as follows:

Lagrangian relaxation algorithm when $s < 2r$

step 1. Set $n = 0$. Solve Problem $L(\lambda^n,\ \theta^n,\ \eta^n)$ with $\lambda_j^n = 0$, $\theta_j^n = 0$ for all j and $\eta^n = 0$ and initialize the best lower bound Z^l (zero obviously). Initialize the best upper bound Z^u as the summation of all c_i.

step 2. Set $n := n + 1$ and update the λ_j^n, θ_j^n and η^n as described above.

step 3. Solve Problem $L(\lambda^n,\ \theta^n,\ \eta^n)$ and update Z^l if necessary. Greedily drop open facilities to eliminate constraints (5.13) and (5.14) violations. If the number of open facilities is still greater than p, greedily drop again until there are exactly p open facilities; otherwise, greedily add open facilities as allowed by constraints (5.13), (5.14) until there are p open facilities or no additional facilities can be added. Update Z^u if necessary.

step 4. If n reaches the iteration limit or the lower bound fails to increase after some predetermined number of consecutive iterations, stop. If the relative error between Z^u and Z^l is less than 0.1 percent, stop. Otherwise, go to step 2.

5.5 Computational Experiments

All test problems were generated by a random graph generator written in the C language using the standard Visual C++ random number function. We first generated the Cartesian coordinates of the nodes in the rectangle $(0, 100)^2$ uniformly. Then nodes were connected randomly until a tree was formed. Finally, a random number of links were added to the tree generated. Demand weights were chosen randomly among the integers between 1 and the cardinality of the graph. The length of each link was calculated using the Euclidean distance formula.

The experiments were run on a Dell Optiplex GX270 machine equipped with 2.793Ghz processor and 512M RAM. The number of nodes n was set to 100, 200, 300, 400 and 500. For simplicity, we assumed $M = N$ in the experiments. The number of facilities p was set to 5, 10 and 15. The combination (r, s) was chosen from $(10, 15), (15, 25), (20, 30)$. This led to 45 combinations of parameters. For each combination of n, p, r and s, 10 problem instances were generated, leading to a total of 450 problem instances.

To obtain the set L_1 in MCLPDC5, we greedily generated C_j for each j. Redundancy was eliminated by checking the set step by step. To obtain n'_j, ILOG CPLEX 8.1 was used to solve the following integer programming formulation of the maximum independent set problem.

$$\max \sum_{v \in Q'_j} x_v$$

$$\begin{aligned} s.t. \quad & x_u + x_v \leq 1 && \text{for all } u, v \in Q'_j \text{ and } (u, v) \in G' \\ & x_v \in \{0, 1\} && \text{for all } v \in Q'_j \end{aligned}$$

Since Q'_j is usually small, this step didn't take too much time for any test problem.

First, we compare the running time of different formulations of the MCLPDC. Due to the possibility of exponential number of maximal cliques, we only compare MCLPDC1, MCLPDC3, MCLPDC4 and MCLPDC5 using ILOG CPLEX 8.1. For each formulation, we set the time limit at 30 minutes. Running time (I/O time excluded) of computational results are listed in Tables 5.1-5.9. According to the computational experiments, it seems that

MCLPDC4 doesn't outperform MCLPDC3 by much. MCLPDC1 and MCLPDC5 run much faster than MCLPDC3 and MCLPDC4. In general, MCLPDC1 outperforms MCLPDC5. But as Table 5.9 shows, when $p = 15$, $r = 20$, $s = 30$ and $n = 300$ or 400, MCLPDC5 runs faster than MCLPDC1. We noticed that for most instances where MCLPDC5 outperforms MCLPDC1, the distance constraints are inconsistent.

Table 5.1: Running time of different formulations when $p = 5$, $r = 10$ and $s = 15$

n	MCLPDC1	MCLPDC3	MCLPDC4	MCLPDC5
100	0.01	0.01	0.01	0.01
200	0.02	0.51	0.35	0.02
300	0.06	11.91	10.71	0.06
400	0.11	313.44	281.59	0.16
500	0.21	1278.69	1246.66	0.34

Table 5.2: Running time of different formulations when $p = 10$, $r = 10$ and $s = 15$

n	MCLPDC1	MCLPDC3	MCLPDC4	MCLPDC5
100	0.01	0.02	0.02	0.00
200	0.02	1.13	0.74	0.03
300	0.06	33.51	33.60	0.08
400	0.12	1800	1800	0.19
500	0.26	1800	1800	0.46

Table 5.3: Running time of different formulations when $p = 15$, $r = 10$ and $s = 15$

n	MCLPDC1	MCLPDC3	MCLPDC4	MCLPDC5
100	0.01	0.05	0.04	0.01
200	0.02	0.83	0.90	0.04
300	0.07	1057.83	574.84	0.10
400	0.14	1800	1800	0.28
500	0.28	1800	1800	0.49

Table 5.4: Running time of different formulations when $p = 5$, $r = 15$ and $s = 25$

n	MCLPDC1	MCLPDC3	MCLPDC4	MCLPDC5
100	0.01	0.48	0.37	0.02
200	0.05	379.63	351.88	0.08
300	0.28	1800	1800	0.52
400	238.16	1800	1800	881.95
500	1800	1800	1800	1800

Table 5.5: Running time of different formulations when $p = 10$, $r = 15$ and $s = 25$

n	MCLPDC1	MCLPDC3	MCLPDC4	MCLPDC5
100	0.02	0.38	0.52	0.02
200	0.07	1669.19	1689.21	0.12
300	0.99	1800	1800	3.00
400	757.22	1800	1800	1493.10
500	1800	1800	1800	1800

Table 5.6: Running time of different formulations when $p = 15$, $r = 15$ and $s = 25$

n	MCLPDC1	MCLPDC3	MCLPDC4	MCLPDC5
100	0.01	0.45	0.84	0.02
200	0.07	958.07	1800	0.17
300	0.42	1800	1800	1.92
400	21.19	1800	1800	1446.96
500	1368.62	1800	1800	1800

Table 5.7: Running time of different formulations when $p = 5$, $r = 20$ and $s = 30$

n	MCLPDC1	MCLPDC3	MCLPDC4	MCLPDC5
100	0.02	10.51	8.64	0.02
200	0.28	1800	1800	0.65
300	1800	1800	1800	1800
400	1800	1800	1800	1800
500	1800	1800	1800	1800

Table 5.8: Running time of different formulations when $p = 10$, $r = 20$ and $s = 30$

n	MCLPDC1	MCLPDC3	MCLPDC4	MCLPDC5
100	0.02	1.54	13.25	0.05
200	0.59	1800	1800	1.34
300	1273.33	1800	1800	1800
400	1800	1800	1800	1800
500	1800	1800	1800	1800

Table 5.9: Running time of different formulations when $p = 15$, $r = 20$ and $s = 30$

n	MCLPDC1	MCLPDC3	MCLPDC4	MCLPDC5
100	0.04	1.13	2.35	0.11
200	0.04	1.13	2.35	0.11
300	212.41	1800	973.92	99.77
400	1800	1800	1800	1463.89
500	1800	1800	1800	1800

To test our heuristics, the iteration limit of the Tabu Search heuristic was set at 50 and the length of the tabu list was set at 20. The iteration limit of the Lagrangian relaxation was set at 100 and the scalar α was halved whenever the lower bound did not improve in 10 consecutive iterations. Table 5.10 contains computational results of heuristics for the same problem set as in Tables 5.1, 5.2, 5.4 and 5.5. If any heuristic obtained a feasible solution, we calculated the gap between that solution value and the best solution value among the three heuristics and the time-limited CPLEX. The average gap in Table 5.10 is the average value of the successful cases.

Contrary to our expectation, the Greedy heuristic outperforms the Lagrangian relaxation heuristic in terms of the number of times feasible solutions are found successfully. We find that the Lagrangian relaxation fails to find a feasible solution quite often for cases of $s = 25$. It is not a surprise that the Tabu Search is superior to the Greedy heuristic since the Tabu search starts from the terminal solution returned by the Greedy heuristic. When both Greedy heuristics and Lagrangian relaxation heuristic return a feasible solution, it is hard to say which one is superior. In general, the performance of the Greedy Heuristic is good, except for the instance with $n = 200$, $p = 5$, $r = 15$ and $s = 25$. The average gap of the Greedy Heuristic is 7.78% for that case, which is much higher than the gap of the Lagrangian relaxation. This is because there is one instance for which the Greedy Heuristic returns a value 48.5% higher than the optimal value. Comparing Table 5.10 with Tables 5.1, 5.2, 5.4 and 5.5, we can conclude that for large networks solving the problem by the Tabu Search heuristic can be recommended. We only report the case of $s < 2r$. Actually, when $s \geq 2r$, the computational results are quite similar to the case of $s < 2r$. The Lagrangian Relaxation in the MCLPDC is not as satisfactory as that in the Anti-Covering Location Problem (Murray and Church, 1997). After we fix the number of facilities, Step 3 in the Lagrangian Relaxation Heuristic may fail to find a feasible solution. However, since the Anti-Covering Location Problem doesn't have any restriction on the number of facilities, through adding and dropping in the Lagrangian Relaxation, a feasible solution can be obtained for sure. Therefore, the MCLPDC becomes even harder than the Anti-Covering Location Problem in this sense.

Table 5.10: Computational results for heuristics

				Greedy			Tabu Search			Lagrangian		
n	p	r	s	successful # [1]	average gap [2]	Time[3]	successful # [4]	average gap [5]	Time[6]	successful # [7]	average gap [8]	Time[9]
100	5	10	15	10	0	0.0	10	0.0	2.17	8	0	3.35
100	10	10	15	10	0.0018	0.0	10	0.0	4.66	9	0.0020	6.81
200	5	10	15	10	0	0.0	10	0.0	13.51	10	0	14.15
200	10	10	15	10	0.0024	0.1	10	0.0	25.95	5	0.0094	22.63
300	5	10	15	10	0.0067	0.1	10	0.0	45.54	10	0.0103	65.97
300	10	10	15	10	0.0166	0.3	10	0.0	93.72	9	0.0026	192.76
400	5	10	15	10	0.0075	0.27	10	0.0	91.74	10	0.0075	137.67
400	10	10	15	10	0.0022	0.5	10	0.0	233.78	8	0.0098	239.83
500	5	10	15	10	0.0020	0.5	10	0.0	171.62	10	0	162.75
500	10	10	15	10	0.0137	0.94	10	0.0040	362.26	9	0.0152	507.98
100	5	15	25	10	0	0.0	10	0.0	1.7	8	0	3.17
100	10	15	25	10	0.0128	0.0	10	0.0001	2.29	4	0.0052	3.43
200	5	15	25	10	0.0778	0.0	10	0.0	10.68	8	0.0390	20.22
200	10	15	25	10	0.0096	0.03	10	0.0018	15.89	2	0.0034	17.26
300	5	15	25	10	0.0033	0.1	10	0.0	44.93	5	0	59.99
300	10	15	25	10	0.0035	0.2	10	0.0035	63.80	9	0.0035	75.04
400	5	15	25	10	0	0.26	10	0.0	96.14	7	0	132.37
400	10	15	25	10	0.0029	0.46	10	0.0007	125.19	1	0.0127	59.44
500	5	15	25	10	0.0020	0.49	10	0	190.93	7	0	143.52
500	10	15	25	10	0.0008	0.86	10	0.0080	301.09	2	0.0656	189.88

1,4,7: number of cases out of 10 returned a feasible solution

2,5,8: average gap of the successful cases where

$$\text{gap}=\frac{\text{Heuristic value} - \text{ best value among three heuristics and time-limited CPLEX}}{\text{best value among three heuristics and time-limited CPLEX}}$$

3,6,9: in seconds

Chapter 6

Conclusions and Future Research

Facility location problems are among the most well-studied problems in optimization literature. In this dissertation we have examined several network facility location problems.

In Chapter 2 we extended the collection depots location problem to siting general number of facilities and/or depots on networks. We discussed the minisum problem where the objective is to minimize the total weighted travel distance. Several conclusions can be drawn from the results of Chapter 2:

- The set of nodes and collection depots is the dominating set for the collection depots location problem.
- For a tree network, there exists an optimal solution on the smallest subtree generated by the set of collection depots.

- For a cycle network, there exists an optimal solution in the set of collection depots.
- Models for the simultaneous location of collection depots and facilities can be reduced to the p-median problems.
- CPLEX cannot solve problems on large networks while our B&B algorithm can.
- The Lagrangian Relaxation can be used as a good heuristic for the collection depots location problem.

We discussed the multiple purpose trip location problem in Chapters 3 and 4. For the median-type problem, the objective is to minimize the total travel time of the system through locating two types of facilities on the network. We showed that there always exists an optimal nodal solution. For the problem of locating a single facility for each type of server on tree and path networks, we analyzed the properties of the optimal solutions. Using the Dominating Subtree Algorithm and the theorems we presented, we can reduce the search range dramatically. Sometimes we can even pinpoint the optimal solution very quickly. To solve the problem on a general network, we proposed several heuristic algorithms. Among them, the Median-based heuristic, which can be used to solve large scale problems, provides good solutions and incurs smallest running time.

In Chapter 4, we investigated the multi-purpose trip location problem employing a covering objective. The objective is to maximize the total intercepted customers through locating two types of facilities on the network. For most combinatorial optimization problems, sometimes it is hard to obtain a bound better than the value of their LP relaxations. We proposed a method which sometimes produces a tighter bound than the LP relaxation of our formulation. For the problem with a general number of type-x facility and a single type-y facility on the path, we developed a dynamic programming algorithm. Computational experiments showed that the greedy heuristic performed well even though our problem is not submodular. However, when the underlying network becomes large, the greedy heuristic is also slow. We believe that some fast heuristic should be encouraged for large networks.

Future research can be done in a number of ways for the multiple purpose trip location problem.

- Analyzing the minisum multi-purpose trip location problem when there are n types of facilities.
- Suzuki and Hodgson (2002) suggested a non-linear formulation to solve the problem which they converted to a linear integer programming formulation that is different from ours. The formulation in Suzuki and Hodgson (2002) has less decision variables but appears to be less user friendly. Comparing the two formulations could be quite interesting.
- As we described in Chapter 3, the multi-purpose trip location model could be beneficial for companies that are not in competition and might want to plan together their facility locations. The model could be relevant in practice when both companies are monopolists in their own markets. Question such as how much extra gain they can obtain together comparing to the non-cooperation case and how to allocate this extra

gain could be very interesting to investigate. Thus, introducing cooperative game theory to the multi-purpose trip location problem could be a direction for future research when there are two different decision makers for each facility type.

- According to Theorem 3.5, when we locate a single facility for both types of facilities on a path, and if the two subpaths obtained by the Dominating Subtree Algorithm overlap, then the optimal solution is to co-locate. This is probably due to the effect of the economies of scope and the correlations among the three types of demands. We suggest to study how to measure the economies of scope and the correlations and how these two measures can affect the location decision.

- Some researchers have tried to explain the shopping mall phenomenon from the angle of anchor stores (See Konishi and Sandfort, 2003, for discussion and literature review). The co-location of our model discussed above provides a starting point to study the shopping mall phenomenon from a different angle.

- For the multi-purpose trip covering problem, a customer who needs both types of services is covered only when the total service trip is within a pre-specified distance. The discussion in Chapter 4 shows that this problem is very hard. We can relax this coverage constraint for type-xy customers, say, if x_1 is the closest type-x facility and y_1 is the closest type-y facility to a customer, and both x_1 and y_1 are within a pre-specified radius, then this customer is covered. Actually, it is easy to show that under this new condition, Theorem 4.1 also holds, but the IP formulation is less complicated. We believe that investigating the problem with the new assumption could be promising.

In Chapter 5 we investigated the undesirable facility location problem. The objective is to minimize the total demand covered through locating a fix number of facilities with distance constraints on a network. We showed that there exists a dominating location set for the problem and it is $\mathcal{NP}$-hard even to check feasibility. To solve the problem, we presented several mathematical programming formulations. We compared the running time of four formulations and found that MCLPDC1 and MCLPDC5 incurred significant less running time than MCLPDC3 and MCLPDC4. As the number of facilities increased, the problem became infeasible and it seemed that MCLPDC5 verified infeasibility faster than MCLPDC1 for large networks. We proposed respectively two Lagrangian relaxation heuristics for two possible types of distance constraints and analyzed one of them. The Tabu Search in general performs better than the Lagrangian and Greedy Heuristics. For large networks, solving the problem by the Tabu Search heuristic can be recommended. We believe that further research to find a better heuristic should be encouraged.

Bibliography

[1] Al-Sultan, K. S., and Al-Fawzan, M. A. A tabu search approach to the uncapacitated facility location problem. *Annals of Operations Research*, 86(1999), 91-103.

[2] Barros, A. I. and Labbé, M. The multi-level uncapacitated facility location problem is not submodular. *European Journal of Operational Research*, 72(1994), 607-609.

[3] Batta, R. and Mannur, N. R. Covering-location models for emergency situations that require multiple response units. *Management Science*, 36(1990), 16-23.

[4] Berman, O., Drezner, Z. A note on the location of an obnoxious facility on a network. *European Journal of Operational Research*, 120(2000), 215-217.

[5] Berman, O., Drezner, Z. and Wesolowsky, G. O. Minimum covering criterion for obnoxious facility location on a network. *Networks*, 28(1996), 1-5.

[6] Berman, O., Drezner, Z. and Wesolowsky, G. O. The collection depots location problem on networks. *Naval Research Logistics*, 49(2002), 15-24.

[7] Berman, O., Drezner, Z. and Wesolowsky, G. O. The expropriation location problem. *Journal of the Operational Research Society*, 54(2003), 769-776.

[8] Berman, O., Drezner, Z. and Wesolowsky, G. O. The hub location problem. *working paper*, 2004a.

[9] Berman, O., Drezner, Z. and Wesolowsky, G. O. The hub location-allocation problem. *working paper*, 2004b.

[10] Berman, O., Drezner, Z. and Wesolowsky, G. O. The facility and hub location-allocation problem. *working paper*, 2004c.

[11] Berman, O. and Krass, D. The generalized maximal covering location problem. *Computers and Operations Research*, 29(2002), 563-581.

[12] Berman, O. and Krass, D. Recent developments in the theory and applications of location models: a preview. *Annals of Operations Research*, 111(2002), 15-16.

[13] Berman, O., Krass, D. and Drezner, Z. The gradual covering decay location problem on a network. *European Journal of Operational Research*, 151(2003), 474-480.

[14] Berlin, G. N., ReVelle, C. S. and Elzinga, D. J. Determining ambulance-hospital locations for on-scene and hospital services. *Environment and Planning A*, 8(1976), 553-561.

[15] Calvo, A. B. and Marks, D. H. Location of health care facilities: an analytical approach. *Socio-Economic Planning Science*, 7(1973), 407-422.

[16] Cappanera, P. A survey on obnoxious facility location problems. Technical Report TR-99-11, University of Pisa.

[17] Chaudhry, S. S., McCormick, S. T. and Moon, I. D. Locating independent facilities with maximum weight: greedy heuristics. *Omega*, 14(1986), 383-389.

[18] Church, R. L. and Garfinkel, R. S. Locating an obnoxious facility on a network. *Transportation Science*, 12(1978), 107-118.

[19] Church, R. L. and Meandows, M. E. Locaiton modeling utilizing maximum service distance criteria. *Geographical Analysis*, 11(1979), 358-373.

[20] Church, R. L. and ReVelle, C. The maximal covering location problem. *Papers of Regional Science Association*, 32(1974), 101-118.

[21] Church, R. L. and Roberts, K. L. Generalized coverage models and public facility location. *Papers of the Regional Science Association*, 53(1983), 117-135.

[22] Cornuéjols, G., Fisher, M. L. and Nemhauser, G. L. Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Management Science*, 23(1977), 789-810.

[23] Cornuéjols, G., Nemhauser, G. L., and Wolsey, L. A. The uncapacitated facility location problem, in *Discrete Location Theory*, Mirchandani P.B., Francis R.L., (Eds.), John Wiley & Sons Inc., New York, (1990), 263-304.

[24] Crowder, H. Computational improvements for subgradient optimization. *Symposia Mathematica*, 19(1976), 357-372.

[25] Current, J., Daskin, M. and Schilling, D. Discrete network location models, in *Facility location: applications and theory*, Z. Drezner and H. Hamacher (Eds), Springer-Verlag, Berlin, (2002), 81-118.

[26] Daskin, M. S. A maximum expected covering location model: Formulation, properties and heuristic solution. *Transportation Science*, 17(1983), 48-70.

[27] Daskin, M. S. Network and discrete location: models, algorithms and applications, John Wiley & Sons, New York, 1995.

[28] Daskin, M. S. A new approach to solving the vertex *p*-center problem to optimality: Algorithm and computational results. *Communications of the Japanese OR Society*, 9(2000), 428-436.

[29] Daskin, M. S. and Stern, E. H. A hierarchical objective set covering model for emergency medical service vehicle deployment. *Transportation Science*, 15(1981), 137-152.

[30] Drezner, Z. The p-cover problem, *European Journal of Operational Research*, 26(1986),312-313.

[31] Drezner, Z., Klamroth, K., Schobel, A. and Wesolowsky, G. The Weber problem, in *Facility location: applications and theory*, Drezner, Z. and Hamacher, H. (Eds), Springer-Verlag, Berlin, (2002), 1-36.

[32] Drezner, Z. and Wesolowsky, G. Finding the circle or rectangle containing the minimum weight of points. *Location Science*, 2(1994), 83-90.

[33] Drezner, Z. and G. O. Wesolowsky On the collection depots location problem. *European Journal of Operational Research*, 130(3), (2001), 510-518.

[34] Erkut, E. and Neuman, S. Analytical models for locating undesirable facilities. *European Journal of Operational Research*, 40(1989), 275-291.

[35] Erlenkoter, D. A dual-based procedure for uncapacitated facility location. *Operations Research*, 26(1978), 992-1009.

[36] Fisher, M. L. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1981), 1-18.

[37] Geoffrion, A. M. Lagrangian relaxation for integer programming". *Mathematical Programming Study*, 2(1974), 82-114.

[38] Geoffrion, A. and Graves, G. Multicommodity distribution system design by bender decomposition. *Management Science*, 20(1974), 822-844.

[39] Glover, F. and Laguna, M. Tabu Search. Kluwer Academic Publishers, Norwell, Massachusetts, 1997.

[40] Goffin, J. L. On convergence of rates of subgradient optimization methods. *Mathematical Programming*, 13(1977), 329-347.

[41] Goldman, A. J. Optimal Locations for Centers in a Network, *Transportation Science*, 3(1969), 352-360.

[42] Goldman, A. J. Optimal center location in simple networks. *Transportation Science*, 5(1971), 212-221.

[43] Hakimi, S. L. Optimum locations of switching centres and the absolute centres and medians of a graph. *Operations Research*, 12(1964), 450-459.

[44] Hakimi, S. L. and Maheshwari, S. N. Optimum Locations of Centers in Network, *Operations Research*, 20(1972), 967-973.

[45] Hale, T. S. and Moberg, C. R. Location science research: a review. *Annals of Operations Research*, 123(2003), 21-35.

[46] Handler, G. Y. p-center problems, in *Discrete Location Theory*, Mirchandani, P. B. and Francis, R. L. (Eds), Wiley, New York, 1990.

[47] Handler, G. Y. and Mirchandani, P. B. Location on Networks, M.I.T. Press, Cambridge, MA, 1979.

[48] Hansen, P. and N. Mladenovic Variable neighborhood search for the p-median. *Location Science*, 5(1997), 207-226.

[49] Hassin, R. and Tamir, A. Improved complexity bounds for location problems on the real line. *Operations Research Letters*, 10(1991), 395-402.

[50] Hochbaum, D.S. and Pathria A. Analysis of the greedy approach in the problems of maximum k-coverage, *Naval Research Logistics,* 45(1998), 615-627.

[51] Hogan, K. and ReVelle, C. Concepts and applications of backup coverage. *Management Science*, 31(1986), 1434-1444.

[52] Kariv, O. and Hakimi, S. L. An algorithm approach to network location problems: part 2, the p-medians. *SIAM Journal on Applied Mathematics*, 37(1979), 539-560.

[53] Kaufman, L., Eede, M., and Hansen, P. A plant and warehouse location problem. *Operational Research Quarterly*, 28(1977), 547-554.

[54] Kolen, A. and Tamir, A. Covering problems, in *Discrete Location Theory*, Mirchandani, P.B., Francis, R.L., (Eds.), John Wiley & Sons Inc., New York, (1990), 263-304.

[55] Kolish, H. and Sandfort, M. Anchor Stores. *Journal of Urban Economics,* 53(2003), 413-435.

[56] Kuby, M. J. Programming models for facility dispersion: The p-dispersion and maxisum dispersion problem. *Geographical Analysis*, 19(1987), 315-329.

[57] marianov, V. and ReVelle, C. The standard response fire protection siting problem. *INFOR*, 29(1991), 116-129.

[58] Marianov, V. and Serra, D. Location problems in the public sector, in *Facility location: applications and theory*, Drezner, Z. and Hamacher, H. (Eds), Springer-Verlag, Berlin, (2002), 119-150.

[59] McLafferty, S. L. and Ghosh, A. Multi-purpose shopping and the location of retail firms. *Geographical Analysis*, 18(1986), 215-226.

[60] Megiddo, N., Zemel, E., and Hakimi, S. L. The maximum coverage location problem, *SIAM Journal on Algebraic and Discrete Methods*, 4(1983), 253-261.

[61] Michel, L. and Hentenryck, P. V. A simple Tabu Search for warehouse location." *Technical Report*, CS-02-05, Brown University, April 2002

[62] Mirchandani, P. B. Generalized hierarchical facility locations. *Transportation Science*, 21(1987), 123-125.

[63] Mirchandani, P. B. The p-median problem and generalizations. in *Discrete Location Theory*, Mirchandani, P. B. and Francis, R. L. (Eds), Wiley, New York, 1990.

[64] Mladenovic, N. and Hansen, P. Variable Neighborhood Search. *Computers and Operations Research*, 24(1997), 1097-1100.

[65] Moon, I. D. and Chaudhry, S. An analysis of network location problems with distance constraints. *Management Science*, 30(1984), 290-307.

[66] Moore, G. C. and ReVelle, C. The hierarchical service location problem. *Management Science*, 28(1982), 775-780.

[67] Mulligan, G. F. Consumer travel behavior: extensions of a multi-purpose shopping model. *Geographical Analysis*, 19(1987), 364-375.

[68] Murray, A. T. and Church, R. L. Solving the anti-covering location problem using Lagrangian relaxation. *Computers and Operations Research*, 24(1997), 127-140.

[69] Murray, A. T., Church, R. L., Gerrard, R. A. and Tsui, Wing-Sing impact models for siting undesirable facilities. *Papers in Regional Science*, 77(1998), 19-36.

[70] Narula, S. C. Hierarchical location-allocation problems: A classification scheme. *European Journal of Operational Research*, 15(1984), 93-99.

[71] Narula, S. C. and Ogbu, U. I. An hierarchical location-allocation problem. *OMEGA*, 7(1979), 137-143.

[72] Narula, S. C. and Ogbu, U. I. Lagrangian relaxation and decomposition in an uncapacitated 2-hierarchical location-allocation problem. *Computers and Operations Research*, 12(1985), 169-180.

[73] Narula, S. C., Ogbu, U. I. and Samuelsson, H. M. An algorithm for the p-medain problem. *Operations Research*, 25(1977), 709-713.

[74] Nemhauser, G. L. and Sigismondi, G. A strong cutting plane/branch-and-bound algorithm for node packing. *Journal of the Operational Research Society*, 43(1992), 443-457.

[75] Nemhauser, G. L. and Wolsey, L. A. *Integer and Combinatorial Optimization*, Wiley/Interscience, New York, 1990.

[76] Orloff, C. S. A theoretical model of net accessibility in public facility location. *Geographical Analysis*, 9(1977), 244-256.

[77] Pirkul, H. and Schilling, D. A. The maximal covering location problem with capacities on total workload. *Management Science*, 37(1991), 233-248.

[78] Resende, M. G. C. and Werneck, R. F. A GRASP with path-relinking for the p-median problem. *Technical Report TD-5E53XL, AT&T Labs Research*, 2002.

[79] ReVelle, C. Facility siting and integer-friendly programming. *European Journal of Operational Research*, 65(1993), 147-158.

[80] ReVelle, C. S., Williams, J. C. and Boland, J. J. Counterpart models in facility location science and reserve selection science. *Environmental Modeling and Assessment*, 7(2002),71-80.

[81] ReVelle, C. and Williams, J. C. Reserve design and facility siting, in *Facility location: Applications and Theory*. Drezner, Z. and Hamacher, H. W. (Ed.), Springer-Verlag, Berlin, 2002, 307-328.

[82] Rolland, E., Schilling, D. A. and Current, J. R. An efficient tabu search procedure for the p-Median Problem. *European Journal of Operational Research*, 96(1996), 329-342.

[83] Rosing, K. E. and ReVelle, C. S. Heuristic concentration; two stage solution construction. *European Journal of Operational Research*, 97(1997), 75-86.

[84] Rosing, K. E., ReVelle, C. S. and Schilling, D. A. A gamma heuristic for the p-median problem. *European Journal of Operational Research*, 117(1999), 522-532.

[85] Schilling, D., Elzinga, D. J., Cohon, J., Church, R. and ReVelle, C. The Team/Fleet models for simultaneous facility and equipment siting. *Transportation Science*, 13(1979), 163-175.

[86] Schilling, D. A., Jayaraman, V., and Barkhi, R. A review of covering problems in facility location. *Location Science*, 1(1993), 25-55.

[87] Schrijver, A. Theory of linear and integer programming. Wiley, New York, 1986.

[88] Suzuki, T. and Hodgson, M. J. Optimal facility location with multi-purpose trip making. *Working paper,* 2002

[89] Suzuki, T. and Hodgson, M. J. Multi-service facility location models. *Annals of Operations Research*, 123(2003), 223-240.

[90] Tamir, A. Obnoxious facility location on graphs. *SIAM J. Discrete Math.* 4(1991), 550-567.

[91] Tamir, A. An $O(pn^2)$ algorithm for the p-median and related problems on tree graphs. *Operations Research Letters*, 19(1996), 59-64.

[92] Tamir, A. and Halman, N. One-way and round-trip center location problem. *Working Paper*, 2003.

[93] Teitz, M. B. and Bart, P. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16(1968), 955-961.

[94] Thill, J. C. Spatial duopolistic competition with multi-purpose and multi-shop shopping. *Annals of Regional Science*, 26, (1992), 287-304.

[95] Toregas, C., Swain, R., ReVelle, C., and Bergman, L. The location of emergency service facilities. *Operations Research*, 19(1971), 1363-1373.

[96] Voss, S. A reverse elimination approach for the p-median problem. *Studies in Locational Analysis*, 8(1996), 49-58.

[97] Watson-Gandy, C. Heuristic procedures for the m-partial cover problem on a plane, *European Journal of Operational Research*, 11(1982), 149-157.

[98] Wendell, R. E. and Hurter, A. P. Optimal location on a network, *Transportation Science*, 7(1973), 18-33.

[99] Wesolowsky, G. Dynamic facility location. *Management Science*, 19(1973), 1241-1248.

[100] West, D. B. Introduction to graph theory, Prentice Hall, 2001.

[101] Zhang, J. Approximating the two-Level facility location problem via a quasi-greedy approach. *Proceedings of The 15th ACM-SIAM Symposium on Discrete Algorithms (SODA), New Orleans, Louisiana*, 1(2004), 808-817.