Journal of the Operations Research
Society of Japan
Vol. 43, No. 2, June 2000

MULTI-FACILITY LOCATION PROBLEM WITH NONINCREASING PIECEWISE LINEAR DEMAND ON A TREE

Masashi Umezawa   Hisakazu Nishino
Keio University

(Received May 18, 1999; Final January 26, 2000)

Abstract This paper deals with a multi-facility location problem on a tree. Given the number of facilities and the tree structure, the problem is to find the optimal locations of facilities so as to maximize the service provider's gain obtained from customers accessing the nearest facility. Customers are located only at vertices of the tree. For each vertex, customers' demand function is given, which is nonincreasing piecewise linear in the distance from the vertex to the nearest facility location. We modify the algorithm proposed by Megiddo-Zemel-Hakimi (1983), and show that it yields the exact optimum within a polynomial time.

1. Introduction

The problem of facility location arises in many different contexts such as information networks, logistic systems, and retail chain stores. The importance of efficient optimization methods for this problem can never be overemphasized. Unfortunately, the multi-facility location problem on a network including cycles is $\mathcal{NP}$-hard.

Multi-facility location problems on a network have been studied since the appearance of Hakimi [2] (see also Hakimi [3]). They are known as a p-median problem and p-center problem (see Tancel, Francis, and Lowe [9]). Matula and Kolde [6] and Kariv and Hakimi [5] presented polynomial time algorithms for the problems on a tree network. Megiddo, Zemel, and Hakimi [7] developed a polynomial time algorithm for the location problem of multiple facilities on a tree, given the constant demand function with a finite support for each vertex.

In this paper, the p-median problem and the problem by Megiddo et al. [7] are extended by introducing a nonincreasing piecewise linear demand function at each vertex. Customers are located only at vertices of a tree. Associated with each vertex is a demand function, which is a nonincreasing piecewise linear function of the distance from the vertex to the nearest facility. When the trip distance exceeds a certain limit (the maximum trip distance), the demand vanishes. For a given number p of facilities, a service provider aims at finding locations of facilities to maximize the total gain obtained from the customers accessing the facilities.

The facilities may be established anywhere on the network. We prove, however, that the locations of facilities can be restricted to a polynomial number of points on the network. Then we develop an algorithm for the problem with nonincreasing piecewise linear demand functions that yields an optimal solution within a polynomial time. Our algorithm is based on the method proposed by Megiddo et al. [7].

Megiddo et al. [7] state that the optimal gain function is concave in the number of facilities. This statement, however, is not valid in general. We show a simple example exhibiting that their algorithm does not yield an optimal solution. The assumption of the concavity appears to bring about this failure. We make a modification to their algorithm, which enables us to attain an optimal solution in their model as well.

In Section 2, we formally describe a location problem on a tree with multiple facilities, and propose a new algorithm to compute an optimal solution. In Section 3, we show that the gain function is not necessarily concave through a simple example. Finally, in Section 4, some concluding remarks are given.

2. Multi-Facility Location Problem

2.1. Model

Consider a tree $T = (V, E)$ ($|V| = n, |E| = n - 1$), where V denotes the set of vertices and E the set of edges. Tree T can be embedded in Euclidean plane. For notational simplicity, this embedded set is also denoted by T. Let $V = \{1, \ldots, n\}$, and $e \in E$ connecting $i, j \in V$, $1 \leq i < j \leq n$, be represented by the closed interval $[i, j]$. The length of an edge $[i, j] \in E$ is represented by d_{ij}. For every pair of points $x, y \in T$, let $d(x, y)$ $(= d(y, x))$ be the length of the path connecting x with y. We associate a set C_i of customers with each $i \in V$. For each C_i, the customers' demand is given by a nonincreasing piecewise linear function ϕ_i of trip distance from i to a facility. It may be reasonable to assume that for each ϕ_i there is a positive number r_i such that $\phi_i(d) = 0$ whenever $d > r_i$. ϕ_i is supposed to have q_i intervals $[s_1^i, s_2^i], [s_2^i, s_3^i], \ldots, [s_{q_i}^i, +\infty)$ where $s_1^i = 0$, $s_{q_i}^i = r_i$, on each of which ϕ_i is linear. We can collect some of the consecutive intervals whenever ϕ_i is convex over them since our algorithm described later works as long as ϕ_i is composed of a finite number of convex pieces. Let $[\sigma_1^i, \sigma_2^i], [\sigma_2^i, \sigma_3^i], \ldots, [\sigma_{t_i}^i, +\infty)$ be the resulting unified intervals. Note that $t_i \leq q_i$, $\sigma_1^i = 0$, and $\sigma_{t_i}^i \leq r_i$.

We do not assume the continuity of ϕ_i, but the finitely many number of points of discontinuity. Thus, $\{\sigma_j^i | j = 2, \ldots, t_i\}$ includes all the discontinuous points of ϕ_i. Even if ϕ_i is not continuous, the convexity of ϕ_i over each unified interval holds true (see Figure 1).[1]

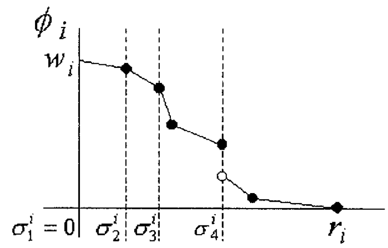

Figure 1: Piecewise Linear Demand Function

Let p be the number of facilities which the service provider can put on T. By the weight w_i, we mean the demand of C_i at zero trip distance, i.e., $w_i = \phi_i(0) = \max_{d \geq 0} \phi_i(d)$. The service provider obtains the gain $\phi_i(d(i, x))$ from i if and only if the facility on $x \in T$ is the nearest from i among p facilities. Any point on T is feasible to locate facilities. The service provider's object is to maximize the total gain from the customers. Let $f(p)$ be the maximum gain which the service provider can obtain, and call it the *optimal gain function*

[1] Every nonincreasing function with a finite support can be approximated by a piecewise linear function at an arbitrary given level of accuracy. Thus, even if ϕ_i is not piecewise linear, our scheme is applicable to an approximated problem.

in p.[2] Let $X = \{x_1, \ldots, x_p\}$ be the set of p points on T. The *multi-facility location problem* can be formulated as follows:

$$f(p) = \max_{X \ on \ T} \sum_{i \in V} \phi_i(d(i, X)), \tag{1}$$

where $d(i, X) = \min_{1 \leq j \leq p}\{d(i, x_j)\}$. On the other hand, the *p-median problem* is defined as follows:

$$\min_{X \ on \ T} \sum_{i \in V} w_i \cdot d(i, X). \tag{2}$$

(2) is equivalent to the following problem for any constant L.

$$\max_{X \ on \ T} \sum_{i \in V} \{-w_i \cdot d(i, X) + L\}.$$

If L is taken to be larger than or equal to $\max_{i \in V}\{w_i \cdot r_i\}$, $-w_i \cdot r_i + L$ is nonnegative for any $i \in V$. Moreover, $-w_i \cdot d(i, X) + L$ is linear and nonincreasing in trip distance $d(i, X)$. Accordingly, our multi-facility location problem is a generalization of the p-median problem.

Megiddo et al. [7] treat the following special case on demand function: for any $i \in V$, $q_i = t_i = 2$, $\phi_i(d) = w_i$ if $d \in [0, r_i]$ and $\phi_i(d) = 0$ if $d \in (r_i, +\infty)$, which is convex on each of $[0, r_i]$ and $[r_i, +\infty)$.

Megiddo et al. [7] show that the multi-facility location problem on a network including cycles is $\mathcal{NP}$-hard in their scheme (see also Kariv and Hakimi [5]). It is a direct consequence of Megiddo et al. [7] that our model is also $\mathcal{NP}$-hard on a network including cycles (see Garey and Johnson [1]).

2.2. Potential locations

The facilities can be established at any point of T, as we stated in Section 2.1. We can, however, select in advance the potential locations to maximize the service provider's gain.

Now we construct a tree $T' = (V', E')$ from $T = (V, E)$ ($|V| = n, |E| = n - 1$) in the following manner. Since each ϕ_i has t_i convex pieces, for any $e = [j, k] \in E$ the number τ_e of points $x^e \in e$ satisfying $d(i, x^e) = \sigma^i_s$ for some i and for some s $(1 \leq s \leq t_i)$ does not exceed $\sum_{i=1}^n t_i$. The points x^es are denoted by $x^e_1, \ldots, x^e_{\tau_e}$, where the subscript is arranged so as to satisfy $d(j, x^e_l) < d(j, x^e_{l+1})$, $l = 1, \ldots, \tau_e - 1$. Let $X_e = \{x^e_l | l = 1, \ldots, \tau_e\}$ and define the new vertex set as $V' = V \cup (\bigcup_{e \in E} X_e)$. All vertices i in $\bigcup_{e \in E} X_e$ are assumed to be $w_i = 0$, i.e., $\phi_i(d) = 0$, for any $d \geq 0$. Let $E'_e = \{[x^e_l, x^e_{l+1}] | l = 0, \ldots, \tau_e, \ x^e_0 = j, \ x^e_{\tau_e+1} = k\}$ and define the new edge set as $E' = \bigcup_{e \in E} E'_e$. The resulting tree T' ($|V'| = n', |E'| = n' - 1$) has $O(t(n-1))$ vertices, where $t = \sum_{i=1}^n t_i$.

Lemma 1 *There exist some optimal locations $x^*_1, \ldots, x^*_p \in T'$ such that $x^*_1, \ldots, x^*_p \in V'$.*

Proof: We redefine the demand function ϕ_i of trip distance as a demand function of a point on T'. Let each $\psi_i(x)$ be the demand function of $i \in V'$ on $x \in T'$, which is also convex on any edge of E', since for any $i \in V$ and for any $[j, k] \in E'$,

$$d(i, j) < d(i, k) \Rightarrow \left\langle \begin{array}{l} [d(i, j), d(i, k)] \subset [\sigma^i_s, \sigma^i_{s+1}] \text{ for some } s \text{ or} \\ [d(i, j), d(i, k)] \subset [\sigma^i_{t_i}, +\infty), \end{array} \right.$$

[2] It is easy to see that f is monotone nondecreasing. Indeed, for $1 \leq s \leq p$, $f(s) = f(s+1)$ if $f(s) = \sum_{j \in V} w_j$, and $f(s) < f(s+1)$ otherwise since $f(s) < \sum_{j \in V} w_j$ implies the existence of an uncovered $j^* \in V$.

and

$$d(i,k) < d(i,j) \Rightarrow \left\langle \begin{array}{l} [d(i,k), d(i,j)] \subset [\sigma_s^i, \sigma_{s+1}^i] \text{ for some } s \text{ or} \\ [d(i,k), d(i,j)] \subset [\sigma_{t_i}^i, +\infty). \end{array} \right.$$

The total demand on x is represented by $\Psi(x) = \sum_{i \in V'} \psi_i(x)$. Therefore, for any point $y = \lambda j + (1-\lambda)k$, $0 \le \lambda \le 1$, we have

$$\begin{aligned} \Psi(y) &= \sum_{i \in V'} \psi_i(y) = \sum_{i \in V'} \psi_i(\lambda j + (1-\lambda)k) \le \sum_{i \in V'} \{\lambda \psi_i(j) + (1-\lambda)\psi_i(k)\} \\ &= \lambda \sum_{i \in V'} \psi_i(j) + (1-\lambda) \sum_{i \in V'} \psi_i(k) = \lambda \Psi(j) + (1-\lambda)\Psi(k) \\ &\le \max\{\Psi(j), \Psi(k)\}. \end{aligned}$$

This implies that if $x_i^* \in [j,k]$, we can take x_i^* that coincides with j or k. **Q.E.D.**

Consequently, our problem is reduced to that of finding a subset X of V' in (1).

Figure 2 shows the demand on an edge between two adjacent vertices $i, j \in V$. If the service provider establishes a facility on $i \in V'$, he obtains the gain of $w_i + \psi_j(i)$.

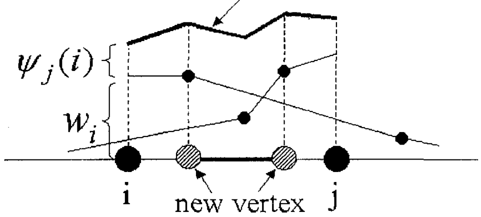

Figure 2: Total Demand on an Edge

2.3. Algorithm

The algorithm proposed here fundamentally consists of three routines INT(H, π, r), EXT(H, π, r), and ALLOC$'(f_1, \dots, f_k; \pi)$, which are based on the algorithm proposed by Megiddo et al. [7]. Instead of ALLOC$'(f_1, \dots, f_k; \pi)$, they use ALLOC$(f_1, \dots, f_k; \pi)$, where f_is are restricted to concave functions. By the concavity of f_i we mean that $f_i(s)$ satisfies $f_i(s) - f_i(s-1) \ge f_i(s+1) - f_i(s)$ for any s. Recall that their model is a special case of our model. We will show through a counterexample in the next section that f_i is not necessarily concave. This implies that their algorithm may not produce the optimal solution. On the other hand, our algorithm of ALLOC$'$ does not require the concavity of f_is, and generates the optimal solution.

First, we select an arbitrary vertex $u_0 \in V'$ as a root of T'. For any pair of vertices i, j, let $P(i,j) \subset V' \cup E'$ denote the path between i and j. For each $i \in V'$ we define V_i', E_i' as $V_i' = \{j | i \in P(u_0, j) \cap V'\}$, $E_i' = \{[a,b] | [a,b] \in P(i,j) \cap E', j \in V_i'\}$ respectively. We call $T_i' = (V_i', E_i')$ the *subtree rooted* at i. The multi-facility location problem on T' can be solved by accumulating the solutions of location problems on subtrees. Indeed, if $i \in V'$ is a leaf, V_i' is a singleton and $E_i' = \emptyset$. The solution of location problem on the leaf is w_i if $p = 1$ and 0 if $p = 0$. Thus, if all sons of $i \in V'$ are leaves, the location problem on T_i' can be solved by using the solutions given on leaves, where by the son j of i we mean that j is adjacent to i and $j \notin P(u_0, i)$. The iteration of the same procedure makes possible to solve the multi-facility location problem on subtree rooted at an arbitrary $i \in V'$.

We now formally describe the algorithm. The following notations are used in the algorithm. H denotes a subtree rooted at vertex $u \in V'$, and let the sons of u be $u_1, \ldots, u_k$. H_i represents the subtree rooted at vertex u_i. Let n_i be the number of vertices in H_i and $d_1^i \leq \ldots \leq d_{n_i}^i$ the distances between u_i and vertices of H_i.

$\mathrm{INT}(H, \pi, r)$ returns the maximum gain from H with π facilities under the restriction that at least one of the π facilities is located at a distance less than or equal to r from u. This is a routine for the problem with π internal facilities on H. On the other hand, $\mathrm{EXT}(H, \pi, r)$ returns the maximum gain from H with π internal facilities when an additional facility is located outside of H at a distance r from u. This is a routine for the problem with one external facility and π internal facilities on H. ALLOC$'$ is a routine to construct a solution of the problem on H by using optimal solutions of subproblems $\mathrm{INT}(H_i, \pi, r)$ and/or $\mathrm{EXT}(H_i, \pi, r)$ $(i = 1, \ldots, k)$. Given the solutions $f_1, \ldots, f_k$ of the problems on k subtrees, $\mathrm{ALLOC}'(f_1, \ldots, f_k; \pi)$ returns the optimal allocation of π facilities among subtrees to maximize the sum of f_i $(i = 1, \ldots, k)$.

The formal descriptions of INT, EXT, and ALLOC$'$ are presented as follows:

INT(H, π, r)

case 1. [A facility is located on the root u of H.]

Let $f_i(p_i) = \mathrm{EXT}(H_i, p_i, d(u_i, u))$, for $i = 1, \ldots, k$. The total gain is obtained by w_u + $\mathrm{ALLOC}'(f_1, \ldots, f_k; \pi - 1)$.

case 2. [A facility is not located on the root u of H.]

Choose a subtree H_j $(1 \leq j \leq k)$. For $\rho \in \{d_1^j, \ldots, d_{n_j}^j\}$ such that $\rho + d(u_j, u) \leq r$, let $f_i(p_i) = \mathrm{EXT}(H_i, p_i, \rho + d(u_j, u_i))$ for $i \neq j$ $(i = 1, \ldots, k)$. Let $f_j(p_j) = \mathrm{INT}(H_j, p_j, \rho)$. $A_j(\rho) \equiv \mathrm{ALLOC}'(f_1, \ldots, f_k; \pi) + \phi_u(\rho + d(u_j, u))$. $A_j = \max_\rho\{A_j(\rho)\}$, for $j = 1, \ldots, k$.

Thus, $\mathrm{INT}(H, \pi, r)$ returns a maximum value among the A_js in the case 2 and the value in the case 1.

EXT(H, π, r)

case 1. [Some facility is located on H within the distance r from u.]

This is, by definition, $\mathrm{EXT}(H, \pi, r) = \mathrm{INT}(H, \pi, r)$.

case 2. [Some facility is not located on H within the distance r from u.]

In this case, there already exists a facility at the distance r outside of H. Therefore there are no interactions between the subtrees H_is. Thus, let $f_i(p_i) = \mathrm{EXT}(H_i, p_i, d(u_i, u) + r)$, $i = 1, \ldots, k$, and $A = \mathrm{ALLOC}'(f_1, \ldots, f_k; \pi) + \phi_u(r)$.

Thus, $\mathrm{EXT}(H, \pi, r)$ returns a maximum value among the case 1 and the case 2.

ALLOC$'$ $(f_1, \ldots, f_k; \pi)$

Let each f_i $(i = 1, \ldots, k)$ be monotone nondecreasing function of a nonnegative integer variable and π be a nonnegative integer.

$$\begin{aligned} \text{Maximize} \quad & F(\pi) = \sum_{i=1}^{k} f_i(p_i) \\ \text{subject to} \quad & \sum_{i=1}^{k} p_i = \pi \\ & p_i : \text{nonnegative integer.} \end{aligned} \tag{3}$$

It is known that ALLOC$'$ can be solved by using dynamic programming.[3] The total effort to solve ALLOC$'$ is $O(k\pi^2)$. For more details of the algorithm, one is referred to Ibaraki and Katoh [4].

[3] Use $F(s) := \max_{0 \leq l \leq s}\{F(s-l) + f_i(l)\}, s = 1, \ldots \pi$ recursively for $i = 1, \ldots, k$.

Lemma 2 *The above algorithm generates an optimal solution to ALLOC* $'(f_1, \ldots, f_k; \pi)$ *within* $O(k\pi^2)$ *time.*

We see in Section 3 how INT, EXT, and ALLOC$'$ run by using an example.

The algorithm terminates when we obtain $f(p) = \mathrm{INT}(T', p, r')$, where r' represents the maximum value of the distances between u_0 and the vertices of T'. There are $O(n')$ subtrees to be considered, where n' is the number of vertices in V'. Each subproblem $\mathrm{INT}(H, \pi, r)$ or $\mathrm{EXT}(H, \pi, r)$ on subtree has at most n' values for the parameter r and π can take the values $0, 1, \ldots, p$. Therefore, the number of different subproblems is $O(n'^2 p)$. Thus, it takes $O(n'^3 p^3)$ time to solve the multi-facility location problem since ALLOC$'$ requires at most $O(n'p^2)$ time. We conclude this section by the following theorem.

Theorem 1 *Our algorithm computes an optimal solution to the problem (1) within a polynomial time.*

3. An Example

We shall abbreviate the algorithm proposed by Megiddo et al. [7] to *M-Alg*.

In the model by Megiddo et al. [7], it is assumed that every customer in C_i has a common maximum trip distance r_i to access a facility, as we stated in Section 2.1. They state that f_i is concave in the number of facilities. The proof, however, is not presented in Megiddo et al. [7]. Moreover, M-Alg proceeds by utilizing the concavity in ALLOC. For each $S \subset V'$, define the gain function $W(S) = \sum_{i \in V'} \phi_i(d(i, S))$.[4] It is known that $W(S)$ is *submodular*[5] on the powerset of V'(see Tamir [8]). Submodularity of $W(S)$, however, doesn't necessarily imply that $f(p)$ is concave in p. The following example exhibits that the optimal gain function on a subtree is not concave, which implies that M-Alg may not produce an optimal solution.

Example

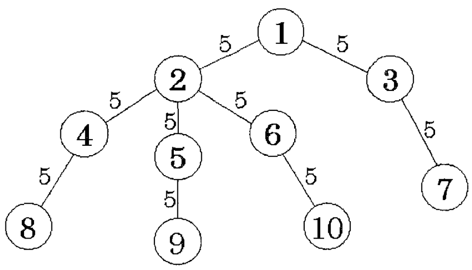

Figure 3: Tree T

Table 1: Properties of Vertices in Tree T

Vertex(i)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	8	9	10
Maximum trip distance(r_i)	1	2	1	6	6	6	1	6	6	6
Weight(w_i)	0	1	4	3	3	3	1	3	3	3

[4] When $S = X$, we can rewrite (1) as follows: $f(p) = \max_{X on T} W(X)$.

[5] $W(S)$ is said to be *submodular* on the powerset of V' if for any $S \subset R \subset V'$, $R \neq V'$, and any $i \notin R$, $W(S \cup \{i\}) - W(S) \geq W(R \cup \{i\}) - W(R)$ holds.

Consider a tree T described by Figure 3 and Table 1, where the length of each edge is assumed to be 5. For any vertex i, $\phi_i(d) = w_i$ if $d \in [0, r_i]$ and $\phi_i(d) = 0$ if $d \in (r_i, +\infty)$. A service provider is to establish 3 facilities on T.

Now, let U_i be the r_i-neighborhood of $i \in V$, i.e., $U_i = \{x | d(i, x) \leq r_i\}$. For every set $S \subset V$, let $U_S = \cap_{i \in S} U_i$. S is said to be *maximal*, if $U_S \neq \emptyset$ and $U_R = \emptyset$ for every $R \supset S$ and $R \neq S$. Without loss of generality, we may assume that every facility belongs to U_S for some maximal S. For every maximal S each $x \in U_S$ brings the same gain to the service provider. Thus, for each maximal S, we can select a representative point $y_S \in U_S$ as a facility location in the following manner: If U_S includes some vertices, let $y_S = i \in U_S \cap V$. Otherwise, choose any $y_S \in U_S$ and regard it as a new vertex possessing no customer and having zero weight, i.e., $w_{y_S} = 0$. The maximal sets are $S_1 = \{\mathbf{1}\}$, $S_2 = \{\mathbf{2}, 4, 5, 6\}$, $S_3 = \{\mathbf{3}\}$, $S_4 = \{\mathbf{4}, 8\}$, $S_5 = \{\mathbf{5}, 9\}$, $S_6 = \{\mathbf{6}, 10\}$, and $S_7 = \{\mathbf{7}\}$. Since $i \in U_{S_i} \cap V$, $i = 1, \ldots, 7$, we can select each i $(1 \leq i \leq 7)$ as the facility location of S_i.

The optimal gain $f(p)$ is obtained by $\mathrm{INT}(T, p, r')$, where T is rooted at vertex 1, $p = 3$, and $r' = 15$. There are no interactions among vertex 1, subtree T_2, and T_3, where T_2 and T_3 represent the subtrees rooted at vertices 2, 3 respectively. Moreover, locating a facility at vertex 1 doesn't make sense because the service provider gains nothing. It implies that it is sufficient to consider the case 2 of routine $\mathrm{INT}(T, p, r')$ to solve this example. The case 2 first requires $\mathrm{INT}(T_2, s, \rho_1)$ and $\mathrm{EXT}(T_3, s, \rho_1 + 10)$ for $s = 0, 1, \ldots, p$ and each $\rho_1 \in \{0, 5, 10\}$, and secondly $\mathrm{EXT}(T_2, s, \rho_2 + 10)$ and $\mathrm{INT}(T_3, s, \rho_2)$ for $s = 0, 1, \ldots, p$ and each $\rho_2 \in \{0, 5\}$. Thus, if we run M-Alg, the algorithm solves $\mathrm{ALLOC}(f_1, f_2; 3)$ at the final step, where $f_1(s)$ and $f_2(s)$ are the optimal gains on T_2 and T_3 as follows:

Table 2: Optimal Gains on T_2

s	$f_1(s)$	Location points	Covered maximal sets	$f_1(s) - f_1(s-1)$
1	10	2	S_2	10
2	13	2, 4	S_2, S_4	3
3	18	4, 5, 6	S_4, S_5, S_6	5

Table 3: Optimal Gains on T_3

s	$f_2(s)$	Location points	Covered maximal sets	$f_2(s) - f_2(s-1)$
1	4	3	S_3	4
2	5	3, 7	S_3, S_7	1

Since $f_1(2) - f_1(1) = 3 < f_1(3) - f_1(2) = 5$, the optimal gain function f_1 is not concave. Thus, ALLOC is not applicable to this example.

If we apply ALLOC$'$, the function F is constructed from $f_1(s)$ and $f_2(s)$ as follows: $F(0) = 0$, $F(1) = \max_{0 \leq l \leq 1}\{f_1(1-l) + f_2(l)\} = f_1(1) + f_2(0) = 10$, $F(2) = \max_{0 \leq l \leq 2}\{f_1(2-l) + f_2(l)\} = f_1(1) + f_2(1) = 14$, $F(3) = \max_{0 \leq l \leq 3}\{f_1(3-l) + f_2(l)\} = f_1(3) + f_2(0) = 18$. Thus, we have $\mathrm{INT}(T, 3, 15) = F(3) = 18$. This is the optimal solution to this example, that locates facilities at vertices 4, 5, and 6.

4. Concluding Remarks

In this section, we explore some possible extensions of this paper as concluding remarks.

This paper describes the multi-facility location problem assuming that there is no existing facility. In practice, one may have to consider the problem of locating new facilities in addition to existing ones. This problem can be easily accommodated by setting demand function $\phi_i(d)$ to be zero for d larger than the distance between i and the nearest existing facility.

Implicitly assumed in this paper is the uniformity of price. In some cases, it may be reasonable to consider that the commodity price at each facility is a control variable. This will bring another dimension to the optimization problem.

Acknowledgements

We would like to thank two anonymous referees of JORSJ for their invaluable comments on the original version of this manuscript. Especially, we appreciate one of them for his suggestion on the relationship between our problem and the p-median problem. We are also grateful to Prof. Yasushi Masuda of Keio University for his encouragement and helpful discussions.

References

[1] M. R. Garey and D. S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman, San Francisco, 1979).

[2] S. L. Hakimi: Optimal locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, **12** (1964) 450-459.

[3] S. L. Hakimi: Optimal locations of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, **13** (1965) 462-475.

[4] T. Ibaraki and N. Katoh: *Resource Allocation Problems: Algorithmic Approaches* (MIT Press, Cambridge, Massachusetts, 1988).

[5] O. Kariv and S. L. Hakimi: An algorithmic approach to network location problems. *SIAM Journal on Applied Mathematics*, **37** (1979) 513-560.

[6] D. W. Matula and R. Kolde: Efficient multi-median location in acyclic networks. *ORSA / TIMS Bulletin*, No. 2 (1976).

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

[8] A. Tamir: A unifying location model on tree graphs based on submodurarity properties. *Discrete Applied Mathematics*, **47** (1993) 275-293.

[9] B. C. Tancel, R. L. Francis, and T. J. Lowe: Location on networks: A survey. *Management Science*, **29** (1983) 482-511.

Masashi Umezawa
Department of Administration Engineering
Faculty of Science and Technology
Keio University
Kohoku-ku, Yokohama 223-8522, Japan
E-mail: ume@ae.keio.ac.jp