Journal of the Operations Research
Society of Japan
Vol. 28, No. 4, December 1985

A SHORTEST PATH APPROACH TO A MULTIFACILITY MINIMAX LOCATION PROBLEM WITH RECTILINEAR DISTANCES

Tetsuo Ichimori
Osaka Institute of Technology

(Received July 19, 1985)

Abstract The multifacility minimax location problem with rectilinear distances is considered. It is reduced to a parametric shortest path problem in a network with no negative length arcs. The reduction scheme contributes to this location problem and yields an efficient algorithm with time complexity $0(n \max(m \log m, n^3))$ where n and m denote the numbers of the new and existing facilities in the plane, respectively. For a special case the time bound is further reducible to $0(n \max(m, n^2))$.

1. Introduction

There are m old facilities already located at points (a_i, b_i) for i=1,2,...,m in the plane, and n new facilities are to be located at points (x_j, y_j) for j=1,2,...,n in the plane. The travel distance between two points (X_1, Y_1) and (X_2, Y_2) in the plane is measured by the rectilinear distance, i.e., $|X_1 - X_2| + |Y_1 - Y_2|$. In urban situations, rectilinear distances are typically used. Define the travel cost between an old facility at (a_i, b_i) and a new facility at (x_j, y_j) as

$$w_{ij} (|a_i - x_j| + |b_i - y_j|) + g_{ij}$$

and the travel cost between a new facility at (x_j, y_j) and another new facility at (x_k, y_k) as

$$v_{jk} (|x_j - x_k| + |y_j - y_k|) + h_{jk}.$$

The constants w_{ij}, v_{jk} are considered as nonnegative costs per unit of distance, while the constants g_{ij}, h_{jk} are considered as nonnegative fixed costs.

The problem of interest, denoted by (P1), is to minimize

$$\max \{w_{ij}(|a_i - x_j| + |b_i - y_j|) + g_{ij} \text{ for } i=1,2,\ldots,m,\ j=1,2,\ldots,n,$$

$$(1.1) \qquad v_{jk}(|x_j - x_k| + |y_j - y_k|) + h_{jk} \text{ for } j=1,2,\ldots,n-1$$

$$k=j+1,\ j+2,\ldots,\ n\}$$

subject to

$$|a_i - x_j| + |b_i - y_j| \le d_{ij} \text{ for all } i \text{ and } j$$

$$(1.2) \qquad |x_j - x_k| + |y_j - y_k| \le c_{jk} \text{ for all } j \text{ and } k>j.$$

The constraints (1.2) give the upper bounds, $d_{ij} \geqq 0$ and $c_{jk} \geqq 0$, on how far apart facilities may be, (which may be $+\infty$). The distance constraints such as (1.2) may be important to facilities of some kind, as mentioned by Schaefer and Hurter [12] and Francis et al. [8]. As an example, a fire station may be required to be within a specified driving distance of any point that it serves. For simplicity, assume that (P1) has a feasible solution that satisfies the constraints (1.2) in this paper.

The minimax location problem such as (P1) may be important to the poor to whom the travel costs are the most significant factors, or may be important to emergency service facilities such as fire, police, and hospital stations, as pointed out by Hakimi [9]. For another application see [13].

Special cases of problem (P1) have been studied by some authors. The single new facility case, i.e. $n = 1$, without constraints (1.2) has been considered by Francis [5]. And then, Elzinga and Hearn [3] and Francis [6] have independently given a closed-form solution to problem (P1) where $n = 1$, $w_{1j} = 1$, and $g_{1j} \ge 0$ for all j, and constraints (1.2) are deleted.

Problem (P1) with constraints (1.2) deleted and with $g_{ij} = 0$, $h_{jk} = 0$ for all i, j, and k, has been studied by Wesolowsky [19], Elzinga and Hearn [4], and Morris [17]. Morris [16] has considered problem (P1) with constraints (1.2) and $g_{ij} = 0$, $h_{jk} = 0$ for all i, j, and k, in the context of linear programming. Dearing and Francis [2] have solved (P1) with $h_{jk} = 0$ for all j and k as a parametric shortest path problem. However, their solution procedure needs a shortest path algorithm which must permit negative length arcs, and it cannot solve (P1) if $h_{jk} \neq 0$. Moreover the time bound of it depends on the input data such as costs and upper bounds on travel distances. (The statement [2] that their algorithm can run in time $O(n^3 \log n)$ is incorrect in this sence.)

The purpose of this paper is to develop an algorithm for (P1) bounded by a polynomial in n and m only. This implies that our result obtained covers all previous results, considering the computational complexities and capabilities of them. To our aim, we also reduce (P1) to a parametric shortest path problem in a network. However, our network does not contain any negative length arc while Dearing and Francis's network contains such arcs. This point is critical in constructing an efficient, polynomial algorithm because

(i) any shortest path algorithm which permits negative length arcs runs much more slowly than Dijkstra's algorithm (or some variant);

(ii) no algorithm can find a shortest path in a network with negative cycles in polynomial time.

Hence we describe the reduction scheme in detail. After that, we apply the parametric approach [15] to the reduced shortest path problem. Then we show that this leads to an $O(n \max(m \log m, n^3))$ algorithm. Furthermore, we show that the computational complexity can further be reduced to $O(n \max(m, n^2))$ for a special case.

A selected bibliography of location literature appears in [7]. For location problems involving generalized distances, see, e.g., [18].

The organization of this paper is as follows. In Section 2 the reduction of (P1) to a parametric shortest path problem is discussed and an efficient algorithm for (P1) is presented. In Section 3 a special case of (P1) is mentioned. Finally a numerical example is worked out in Section 4.

2. Reduction of (P1) and Solution Procedure

In this section we will reduce problem (P1) to a parametric shortest path problem in a network in which no negative length arcs exist.

Lemma 2.1. For two real numbers X and Y, $|X| + |Y| = \max(|X + Y|, |X - Y|)$.

Proof: Obvious. Q. E. D.

The changes of variables:

$$\alpha_i = a_i + b_i \qquad \text{for all } i$$

$$\beta_i = a_i - b_i \qquad \text{for all } i$$

$$s_j = x_j + y_j \qquad \text{for all } j$$

$$t_j = x_j - y_j \qquad \text{for all } j$$

together with Lemma 2.1 yield a problem, to be denoted by (P2), which is equivalent to (P1). Problem (P2) is to minimize

$$\max \{A_{ij} \text{ for all } i \text{ and } j,\ B_{jk} \text{ for all } j \text{ and } k > j\}$$

where

$$A_{ij} = \max (w_{ij}|\alpha_i - s_j| + g_{ij},\ w_{ij}|\beta_i - t_j| + g_{ij}),$$

$$B_{jk} = \max (v_{jk}|s_j - s_k| + h_{jk},\ v_{jk}|t_j - t_k| + h_{jk})$$

subject to

$$\max (|\alpha_i - s_j|, |\beta_i - t_j|) \le d_{ij} \qquad \text{for all } i \text{ and } j,$$

$$\max (|s_j - s_k|, |t_j - t_k|) \le c_{jk} \qquad \text{for all } j \text{ and } k > j.$$

Since (P2) is separable, it suffices to consider the following two problems (P3) and (P4):

(P3) $$\text{min-max } (w_{ij}|\alpha_i - s_j| + g_{ij} \quad \text{for all } i \text{ and } j,\quad v_{jk}|s_j - s_k| + h_{jk} \quad \text{for all } j \text{ and } k > j)$$

subject to

$$|\alpha_i - s_j| \le d_{ij} \qquad \text{for all } i \text{ and } j,$$

$$|s_j - s_k| \le c_{jk} \qquad \text{for all } j \text{ and } k > j.$$

For simplicity assume $\alpha_i \ge 0$ for all i. If the assumption fails, replace α_i by $\alpha_i + M$ for all i and s_j by $s_j + M$ for all j, where M is a suitably large positive constant. Note that this replacement makes (P3) remain as it is.

(P4) $$\text{min-max } (w_{ij}|\beta_i - t_j| + g_{ij} \quad \text{for all } i \text{ and } j,\quad v_{jk}|t_j - t_k| + h_{jk} \quad \text{for all } j \text{ and } k > j)$$

subject to

$$|\beta_i - t_j| \le d_{ij} \qquad \text{for all } i \text{ and } j,$$

$$|t_j - t_k| \le c_{jk} \qquad \text{for all } j \text{ and } k > j.$$

Similarly, assume $\beta_i \ge 0$ for all i.
Note that (P3) and (P4) can be considered on a line. Since they are of the same type, we concentrate on (P3) only. Problem (P3) can be restated as follows:

(P5) minimize λ

s.t. $|\alpha_i - s_j| \le \min((\lambda - g_{ij})/w_{ij}, d_{ij})$ for all i and j,

(2.1)

$|s_j - s_k| \le \min((\lambda - h_{jk})/v_{jk}, c_{jk})$ for all j and k > j,

where we define $(\lambda - g_{ij})/w_{ij} = \infty$ and $(\lambda - h_{jk})/v_{jk} = \infty$ if $w_{ij} = 0$ and $v_{jk} = 0$, respectively.

As a notational convenience, the minimum value of the objective function of (P5) will be denoted by λ^*. Let $\underline{\lambda} = \max(g_{ij}$ for all i and j, h_{jk} for all j and k > j). Then it is obvious that $\lambda^* \ge \underline{\lambda}$. So in the sequel restrict $\lambda \ge \underline{\lambda}$. Note that all the constraints (2.1) are not satisfied for $\lambda < \lambda^*$.

Let

(2.2) $\delta_{ij} = \min(\lambda - g_{ij})/wij, d_{ij})$ for all i and j

(2.3) $F_{0j} = \min_{i=1,2,\ldots,m} (\alpha_i + \delta_{ij})$ for all j

(2.4) $F_j = \max_{i=1,2,\ldots,m} (\alpha_i - \delta_{ij})$ for all j

(2.5) $F_{jk} = \min(\lambda - h_{jk})/v_{jk}, c_{jk})$ for all j and k > j.

Since F_{0j}, F_j, and F_{jk} are functions of λ, we may sometimes denote them by $F_{0j}(\lambda)$, $F_j(\lambda)$, and $F_{jk}(\lambda)$, respectively.

The constraints (2.1) are expressed as (2.6)-(2.8):

(2.6) $s_j \le s_0 + F_{0j}$ for all j

(2.7) $s_j \ge F_j$ for all j

(2.8) $|s_j - s_k| \le F_{jk}$ for all j and k > j

where $s_0 = 0$. From the theory of shortest paths, it follows that linear inequalities (2.6) and (2.8) imply the network G with node set {0,1,2,...,n} and a directed arc from node 0 to node j with length F_{0j} for each inequality of (2.6) and an undirected arc between nodes j and k with length F_{jk} for each inequality of (2.8). The network G for n = 3 is illustrated in Fig. 2.1.

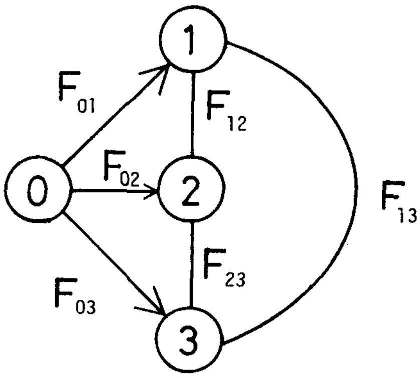

Fig. 2.1 G for n = 3

It is well known that linear inequalities (2.6) and (2.8) can be satisfied by the shortest path lengths from node 0 to the other nodes. For any $\lambda \geq \underline{\lambda}$, the shortest path lengths can be computed by Dijkstra's shortest path algorithm [1] since each arc length F_{0j} or F_{jk} is nonnegative, recalling $\alpha_i \geq 0$, $\lambda \geq g_{ij}$, and $\lambda \geq h_{jk}$. Let $P_j(\lambda)$ be the shortest path length from node 0 to node j in G for λ. Then, from the above, we have the equivalent problem (P6).

(P6) minimize λ

s.t. $P_j(\lambda) \geq F_j$ for j = 1,2,...,n.

We will solve (P6) by computing the shortest path lengths from node 0 to the other nodes parametrically in the parameter λ, noting that λ^* is the smallest value of λ such that

$$P_j(\lambda) \geq F_j \qquad \text{for all } j.$$

Obviously, $F_{0j}(\lambda)$ is concave and piecewise linear with at most (m + 1) linear pieces, $F_j(\lambda)$ is convex and piecewise linear with at most (m + 1) linear peices, and $F_{jk}(\lambda)$ is concave and piecewise linear with at most two linear pieces. Hence F_{0j} and F_j each can be expressed as the form:

$$\begin{array}{lll} & A_1\lambda + B_1 & \underline{\lambda} \leq \lambda \leq \omega_1 \\ (2.9) & A_2\lambda + B_2 & \omega_1 \leq \lambda \leq \omega_2 \\ & \cdots\cdots\cdots\cdots\cdots & \end{array}$$

$$A_{r+1}\lambda + B_{r+1} \qquad \omega_r \le \lambda < \infty$$

where A_j and B_j for j = 1, 2,..., r+1 are real numbers, $r \le m$, and $\underline{\lambda} < \omega_1 < \omega_2 < \dots < \omega_r < \infty$. It is shown in [15] that the expression entails O(m log m) time for each F_{0j}, F_j, and hence O(nm log m) time for all F_{0j} and F_j for j=1,2,...,n. (The algorithm due to Megiddo for the minimum of m linear functions is different from the "parametric" algorithm due to him.) Each F_{jk} can be also expressed as the form (2.9) with $r \le 1$. This takes $O(n^2)$ time for all j and k > j.

For any fixed value of λ, let the feasibility test for problem (P6), i.e., whether $P_j(\lambda) \geqq F_j$ for all j are satisfied or not, be denoted by (FT). For any fixed value of λ, the computation time for determining the value of each $F_{0j}(\lambda)$ or F_j is O(log m) by finding an interval including this λ with binary search, while that for each $F_{jk}(\lambda)$ is O(1). After the computation of F_{0j}, F_j, and F_{jk} for all j and k > j, which takes $O(n \log m + n^2)$ time in all, $P_j(\lambda)$ for all j are determinable in $O(n^2)$ time by Dijkstra's algorithm. And then, the comparisons between $P_j(\lambda)$ and F_j for j=1,2,...,n need O(n) time. Hence the time complexity of (FT) for any fixed value of λ is $O(n \log m + n^2)$.

Since each function F_{0j} or F_j has at most m breaking points and each F_{jk} has at most one breaking point, there are at most 2mn + n(n-1)/2 breaking points in all. Let the distinct λ-coordinates of the breaking points no less than $\underline{\lambda}$ be $\underline{\lambda} = \Omega_0 < \Omega_1 < \Omega_2 < \dots$. Because each function $P_j(\lambda)$ is strictly increasing until its slope becomes zero while each function $F_j(\lambda)$ is strictly decreasing until its slope becomes zero, there exists an interval $[\Omega_r, \Omega_{r+1}]$ satisfying the following (2.10) and (2.11):

$$(2.10) \qquad P_j(\Omega_r) < F_j(\Omega_r) \qquad \text{for some } j$$

and

$$(2.11) \qquad P_j(\Omega_{r+1}) \geqq F_j(\Omega_{r+1}) \qquad \text{for all } j,$$

i.e., $\Omega_r < \lambda^* \leqq \Omega_{r+1}$. Such an interval can be found by performing binary search over intervals $[\Omega_0, \Omega_1]$, $[\Omega_1, \Omega_2]$, $[\Omega_2, \Omega_3]$,..., and using feasibility test (FT). Then we can obtain the quantities of the form:

$$(2.12) \qquad F_{ij}(\lambda) = e_{ij}\lambda + f_{ij} \qquad \Omega_r < \lambda \le \Omega_{r+1}$$

for each directed or undirected arc (i,j) in G, and of the form:

$$(2.13) \qquad F_j(\lambda) = e_j\lambda + f_j \qquad \Omega_r < \lambda \le \Omega_{r+1}$$

for all nodes j in G.

To derive the quantities of the form (2.12) or (2.13), it takes time $O(\log(2mn + n(n-1)/2))O(n \log m + n)$

$$= O(n^2 \log (n+m)) + O(n \log m \log (n+m))$$
$$= O(n \log (n+m) \max(n, \log m)).$$

At this point all data (i.e. arc lengths) for computing shortest path lengths are of the form (2.12) or (2.13). Hence Megiddo's parametric approach is applicable to our purpose, see [15] for details. (A good illustration of Megiddo's approach is given in [10], although in the context of maximum flow.)

The parametric computation for the shortest path lengths in G needs $O(n^2)$ comparisons since Dijkstra's algorithm does so. The compoarisons each involve a feasibility test (FT), which needs $O(n^2)$ time since each F_{0j} or F_j is now of the form (2.12) or (2.13) and hence the value of $F_{0j}(\lambda)$ or $F_j(\lambda)$ can be determined in $O(1)$ time for any fixed value of λ. Therefore the shortest path lengths, $P_j(\lambda)$ for all j can be obtained in time

$$O(n^2)\ O(n^2) = O(n^4).$$

This is exactly the result given in [15] for this problem.

After the parametric computation we obtain the following expression for $P_j(\lambda)$ in terms of certain real numbers e_j^* and f_j^*:

$$P_j(\lambda) = e_j^*\lambda + f_j^* \qquad L < \lambda \le U, \tag{2.14}$$

where L satisfies $P_j(L) < F_j(L)$ for some j and U does $P_j(U) \ge F_j(U)$ for all j. Then, note that $F_j(\lambda) = e_j\lambda + f_j$, $L < \lambda \le U$. Denote the set of all j such that $P_j(L) < F_j(L)$ by J.

Theorem 2.1. Define $\lambda_j = (f_j - f_j^*)/(e_j^* - e_j)$ $j \in J$. Then

$$\lambda^* = \max_{j \in J} \lambda_j .$$

Proof: Let $\lambda^* = \lambda_{j^*}$. Then, for $\lambda < \lambda^*$ we have $P_j(\lambda) < F_j(\lambda)$ for $j = j^*$ (see Fig. 2.2). On the other hand, for $\lambda \ge \lambda^*$ we have $P_j(\lambda) \ge F_j(\lambda)$ for all $j = 1,2,\ldots,n$. Q.E.D.

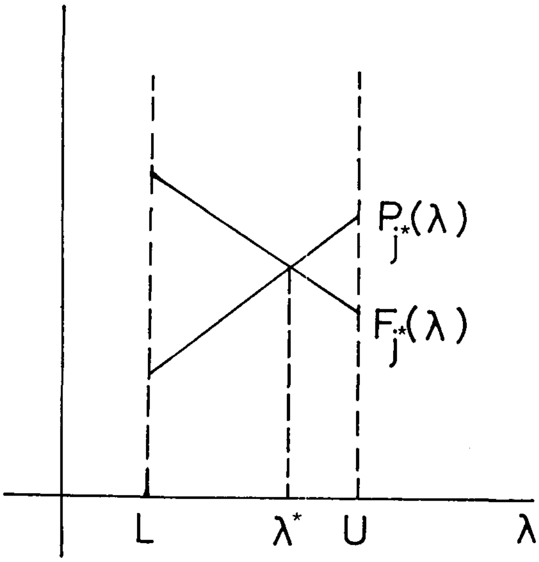

Fig. 2.2.

Summarizing what has been discussed above, the algorithm for (P6) follows.

Algorithm and time complexity

S1. Express F_{0j}, F_j, and F_{jk} for all j and k > j as the form (2.9). —— $O(nm \log m + n^2)$.

S2. Express F_{ij} for all arcs and F_j for all nodes in G as the form (2.12) and (2.11), respectively. —— $O(n \log (n + m) \max (n, \log m))$.

S3. Compute $P_j(\lambda)$ of the form (2.14) for all j. —— $O(n^4)$.

S4. Compute λ^* by Theorem 2.1. —— $O(n)$.

Remark 2.1. The optimal solution to (P3) is given by $s_j^* \leftarrow P_j(\lambda^*)$ j = 1, 2,...n.

Remark 2.2. S3 corresponds to the parametric algorithm due to Megiddo.

Theorem 2.2. The overall time complexity of the preceding algorithm is $O(n \max(m \log m, n^3))$.

Proof: It is obvious that the time bounds of Steps 1, 3, and 4 are not greater than $O(n \max(m \log m, n^3))$. Consider the case of Step 2.

(i) Let $n \geq \log m$, or $2^n \geq m$. Then

$$O(n \log (n+m) \max(n, \log m)) \leq O(n^2 \log (n + 2^n)) \leq O(n^2 \log 2^{2n}) = O(n^3) < O(n^4).$$

(ii) Let $n < \log m$, which yields $n < m$. Then

$$O(n \log (n+m) \max(n, \log m)) = O(n \log (n+m) \log m) \leq O(n \log (2m) \log m) < O(nm \log m).$$

Hence the theorem follows. Q.E.D.

Theorem 2.2 states that the preceding algorithm can solve (P6) in time $O(n \max(m \log m, n^3))$. Hence it means that the algorithm can solve problem (P1) in the same time as well, noting that the time for reducing (P1) to (P6) is negligible compared with $O(n \max(m \log m, n^3))$.

3. Special Case

In this section consider the case $w_{ij} = 0$ or 1 for all i and j, $v_{jk} = 0$ or 1 for all j and $k > j$, briefly. Of course, the problem of this case can be solved by the algorithm presented in the preceding section. However its time complexity can be reduced in this case. Step 1 needs $O(nm + n^2)$ time since the slope of each F_{0j} or F_{jk} is 0 or 1 and that of each F_j is 0 or -1. With respect to Step 2, each (FT) entails $O(n + n^2) = O(n^2)$ time and there are at most $2n + n(n - 1)/2$ breaking points in all. Hence Step 2 needs time

$$O(\log (2n + n(n - 1)/2)) \ O(n^2) = O(n^2 \log n).$$

Step 3 can be performed by Karp and Orlin's algorithm in time $O(n^3)$ (see [11]), noting that each arc length contained in G is of the form

$$A\lambda + B$$

where $A = 0$ or 1, B is a real number. Step 4 is $O(n)$. Hence the entire algorithm requires $O(n \max(m, n^2))$ time.

We conclude this section by stating that the reduction of the time bound is possible since the slopes of functions of λ, F_{0j} and F_{jk} for all j and $k > j$, are restricted to 0 and 1 (hence the slope of each $P_j(\lambda)$ is 0, 1, ..., n), while this is not true for the case of the preceding section.

4. Numerical Example

Consider the following numerical example of problem (P3) with $m = n = 3$:

$$(w_{ij}) = \begin{pmatrix} 1.2 & 1 & 0.5 \\ 1 & 1.5 & 2 \\ 1 & 1.25 & 4 \end{pmatrix} \quad (g_{ij}) = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 1 & 2 \\ 2 & 0 & 1 \end{pmatrix} \quad (d_{ij}) = \begin{pmatrix} 7 & 15 & 12 \\ 13 & 11 & 10 \\ 20 & 15 & 14 \end{pmatrix}$$

$$(v_{jk}) = \begin{pmatrix} - & 10 & 1 \\ - & - & 1.5 \\ - & - & - \end{pmatrix} \quad (h_{jk}) = \begin{pmatrix} - & 2 & 1 \\ - & - & 1 \\ - & - & - \end{pmatrix} \quad (c_{jk}) = \begin{pmatrix} - & 1 & 3 \\ - & - & 10 \\ - & - & - \end{pmatrix}$$

$(\alpha_i) = (2, 3, 6)$.

From the preceding data we have:

$\underline{\lambda} = 2$,

$$F_{01} = \begin{cases} (5/6)\lambda + 7/6 & 2 \le \lambda \le 47/5 \\ 9 & \lambda \ge 47/5 \end{cases}$$

$$F_{02} = \begin{cases} \lambda + 1 & 2 \le \lambda \le 4 \\ (2/3)\lambda + 7/3 & 4 \le \lambda \le 16 \\ (1/2)\lambda + 5 & 16 \le \lambda \le 18 \\ 14 & \lambda \ge 18 \end{cases}$$

$$F_{03} = \begin{cases} \lambda & 2 \le \lambda \le 25/3 \\ (1/4)\lambda + 25/4 & 25 \le \lambda \le 47 \\ 18 & \lambda \ge 47 \end{cases}$$

$$F_{1} = \begin{cases} 10 - 2\lambda & 2 \le \lambda \le 6 \\ 4 - \lambda & 6 \le \lambda \le 7 \\ 17/6 - (5/6)\lambda & 7 \le \lambda \le 47/5 \\ -5 & \lambda \ge 47/5 \end{cases}$$

$$F_{2} = \begin{cases} 7 - (1/2)\lambda & 2 \le \lambda \le 22 \\ -4 & \lambda \ge 22 \end{cases}$$

$$F_{3} = \begin{cases} 25/4 - (1/4)\lambda & 2 \le \lambda \le 57 \\ -8 & \lambda \ge 57 \end{cases}$$

$$F_{12} = \begin{cases} (1/10)\lambda - 1/5 & 2 \le \lambda \le 12 \\ 1 & \lambda \ge 12 \end{cases}$$

$$F_{13} = \begin{cases} \lambda - 1 & 2 \le \lambda \le 4 \\ 3 & \lambda \ge 4 \end{cases}$$

$$F_{23} = \begin{cases} (2/3)\lambda - 2/3 & 2 \le \lambda \le 16 \\ 10 & \lambda \ge 16 \end{cases}$$

The above functions have the distict λ-coordinates of the breaking points:

2, 4, 6, 7, 25/3, 47/5, 12, 16, 18, 22, 47, 57.

Since the median is 47/5 (or 12), we perform (FT) for $\lambda = 47/5$ on G, which is shown in Fig. 4.1. Here $F_1 = -5$, $F_2 = 2.3$, $F_3 = 3.9$, $P_1 = 9$, $P_2 = 8.6$, $P_3 = 8.6$. Since $P_j \geq F_j$ for j = 1,2,3, we obtain $2 \leq \lambda^* \leq 47/5$.

Since the median of 2, 4, 6, 7 and 25/3 is 6, (FT) is done for $\lambda = 6$ on G, which is shown in Fig. 4.2. Here $F_1 = -2$, $F_2 = 4$, $F_3 = 19/4$, $P_1 = 37/6$, $P_2 = 19/3$, $P_3 = 6$. Since $P_j \;\; F_j$ for j = 1,2,3, we have $2 \leq \lambda^* \leq 6$. For $\lambda = 4$, $F_1 = 2$, $F_2 = 5$, $F_3 = 5.25$, $P_1 = 4.5$, $P_2 = 4.7$, $P_3 = 4$ (see Fig. 4.3.) Since $P_2 < F_2$, we have $4 < \lambda^* \leq 6$. Since there is no breaking point strictly between 4 and 6, the functions above are expressed as follows:

$$
\begin{aligned}
F_{01} &= (5/6)\lambda + 7/6\\
F_{02} &= (2/3)\lambda + 7/3\\
F_{03} &= \lambda\\
F_1 &= 10 - 2\lambda\\
F_2 &= 7 - (1/2)\lambda\\
F_3 &= 25/4 - (1/4)\lambda\\
F_{12} &= (1/10)\lambda - 1/5\\
F_{13} &= 3\\
F_{23} &= (2/3)\lambda - 2/3\\
4 &< \lambda \leq 6 .
\end{aligned}
$$

Here we have the parametric shortest path problem with additional constraints $P_j(\lambda) \geq F_j(\lambda)$ for j = 1,2,3, which is shown in Fig. 4.4. The result is as follows:

$$
\begin{aligned}
P_1 &= (5/6)\lambda + 7/6\\
P_2 &= (14/15)\lambda + 29/30\\
P_3 &= \lambda\\
4 &< \lambda \leq 41/8 .
\end{aligned}
$$

Since $P_2(4) < F_2(4)$, $P_3(4) < F_3(4)$, we have $J = \{2, 3\}$,

$$\lambda_2 = (7 - 29/30)/(14/15 - (-1/2)) = 181/43,$$

$$\lambda_3 = (25/4 - 0)/(1 - (-1/4)) = 5,$$

hence $\lambda^* = \max\{\lambda_2, \lambda_3\} = 5$. The optimal solution to (P3) is $s_1^* = 16/3$, $s_2^* = 169/30$, $s_3^* = 5$.

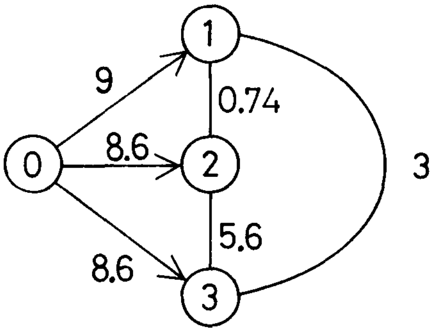

Fig. 4.1. λ = 47/5.

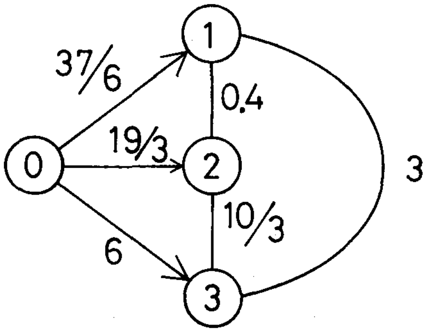

Fig. 4.2. λ = 6.

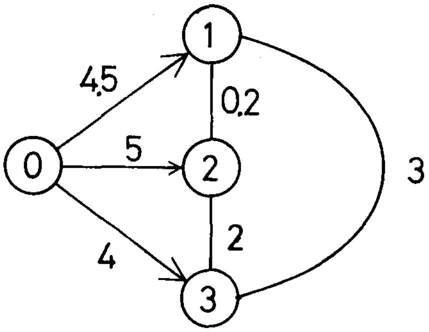

Fig. 4.3. λ = 4.

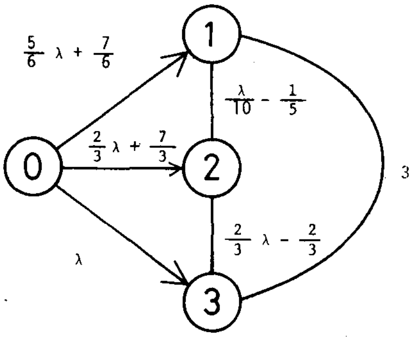

Fig. 4.4

References

[1] Dijkstra, E. W.: A Note on Two Problems in Connection with Graphs, *Numeriche Mathematik*, Vol.1 (1959), 269-271.

[2] Dearing, P. M. and Francis, R. L.: A Network Flow Solution to a Multi-facility Minimax Location Problem Involving Rectilinear Distances, *Transportation Science*, Vol.8 (1974), 126-141.

[3] Elzinga, L. and Hern, D. W.: Geometrical Solutions for Some Minimax Location Problems, *Transportation Science* Vol.4 (1972), 379-394.

[4] _________ and _________: A Note on a Minimax Location Problem, *Transportation Science*, Vol.7 (1973), 100-103.

[5] Francis, R. L.: Some Aspects of a Minimax Location Problem, *Operations Research*, Vol.15 (1967), 1163-1169.

[6] _________: A Geometrical Solution Procedure for a Rectilinear Distance Minimax Location Problem, *AIIE Transactions*, Vol.4 (1972), 328-332.

[7] Francis, R. L., and Goldstein, J. M.: Location Theory: A Selective Bibliography, *Operations Research*, Vol.22 (1974), 400-410.

[8] _________, Lowe, T. J. and Ratliff, H. D.: Distance Constraints for Tree Network Multifacility Location Problems, *Operations Research*, Vol.12 (1964), 450-459.

[9] Hakimi, S. L.: Optimum Location of Switching Centers and the Absolute Centers and Medians of a Graph, *Operations Research*, Vol.12 (1964), 450-459.

[10] Ichimori, T., Ishii, H. and Nishida, T.: Weighted Minimax Real-Valued Flows, *Journal of the Operations Research Society of Japan*, Vol.24 (1981), 52-60.

[11] Karp, R. M. and Orlin, J. B.: Parametric Shortest Path Algorithms with an Application to Cyclic Staffing, *Discrete Applied Mathematics*, Vol.3 (1981), 37-45.

[12] Schaefer, M. K. and Hurter, A. P.: An Algorithm for the Solution of a Location Problem with Metric Constraints, *Naval Research Logist. Quarterly*, Vol.4 (1974), 625-636.

[13] Smallwood, R. D.: Minimax Detection Station Placement, *Operations Research*, Vol.13 (1965), 632-646.

[14] Lawler, E. L.: *Combinatorial Optimization; Networks and Matroids*. Holt, Rinehart and Winston, New York, 1976.

[15] Megiddo, N.: Combinatorial Optimization with Rational Objective Functions, *Mathematics of Operations Research*, Vol.4 (1979), 414-424.

[16] Morris, J. G.: A Linear Programming Approach to the Solution of Constrained Multi-Facility Minimax Location Problems Where Distances are Rectangular, *Operational Research Quarterly*, Vol.24 (1973), 419-435.

[17] _________: A Further Note on a Minimax Location Problem, *Transportation Science*, Vol.10 (1976), 405-408.

[18] _________: Convergence of the Weiszfeld Algorithm for Weber Problems Using a Generalized "Distance" Function, *Operations Research*, Vol.29 (1981), 37-48.

[19] Wesolowsky, G. O.: Rectangular Distance Location under the Minimax Optimality Criterion. *Transportation Science*, Vol.6 (1972), 103-113.

Tetsuo ICHIMORI: Department of
Industrial Management,
Osaka Institute of Technology,
Omiya, Asahi-ku, Osaka 535, Japan.