

## LAYOUT OF FACILITIES WITH SOME FIXED POINTS

ZVI DREZNER†

School of Management, University of Michigan, Dearborn, MI 48128, U.S.A.

and

GEORGE O. WESOLOWSKY‡

Dept. of Manage Science and Information Systems, McMaster University, Hamilton, Ontario, Canada

**Scope and purpose**—Location problems involve finding the best locations for facilities on some surface in accordance with some given objective. We restrict ourselves to finding the best facility locations on the plane. The objective in this paper is to minimize the greatest (weighted) distance between these facilities and simultaneously between the facilities and some fixed points. Applications include: layout of circuits on a board (minimizing the length of the longest wire), location of radio or TV stations (minimizing the distance between the stations and the furthest customer), and the location of emergency facilities such as hospitals and fire stations.

**Abstract**—This paper deals with the location of facilities or “movable” points on a planar area, on which there already exist fixed points. The minimax criterion for optimality is used and distances among points are assumed to be rectilinear. Two very efficient algorithms for the solution of the problem are presented. One is based on a univariate search, and the other on a steepest descent method. Some computational results are presented.

### INTRODUCTION

We deal in this paper with the location of points on a plane. When we have freedom of choosing the location of all points, then the problem is known as the layout problem, or the quadratic assignment problem [5]. When all points but one are fixed in place and when the *maximum* distance to all fixed points is to be minimized, then the problem is known as the one-center problem; when the sum of all distances to the fixed points is to be minimized it is known as the one-median problem [13]. Here we assume that most points are fixed in place. Problems with this assumption include the multifacility minimax problem, when the greatest weighted distance in the system is to be minimized [6], and the multifacility minimum problem where the weighted sum of distances is minimized [13]. We investigate a version of the final problem. In this paper, each distance is rectilinear (also known as the  $\ell_1$  distance). This is the distance measure when connections or travel are allowed only along horizontal or vertical directions.

### PROBLEM DEFINITION

We must locate  $n$  points on a “board.” We assume that  $r$  points are fixed, and  $s = n - r$  points are “free points” in that their sites need to be determined. Also, a nonnegative weight is associated with any pair of points. Typically this weight is zero when there are no connections between the two points. The connection can be weighted by any appropriate positive multiplier for flexibility in the model.

Let  $(x_i, y_i)$  be the location of point  $i$  for  $i = 1, \dots, n$ . For simplicity of notation we assume that the *last*  $r$  points are fixed (their locations given). Let  $w_{ij}$  be the weight between points  $i$  and  $j$  for  $i, j = 1, \dots, n$ . Note that  $w_{ij} = 0$  when both  $i$  and  $j$  are fixed. The weighted distance

†Zvi Drezner is an associate Professor of Business Administration at The University of Michigan–Dearborn. He has a B.Sc. in Mathematics and a D.Sc. in Computer Science from the Technion, Haifa, Israel. He has recently published over 30 papers in Operations Research, Management Science, IIE Transactions, Naval Research Logistics Quarterly, and other journals.

‡George O. Wesolowsky is a Professor of Management Science and Chairman of the Management Science and Information Systems Area of McMaster University in Hamilton, Ontario, Canada. He has a B.Sc. from the University of Toronto, an MBA from the University of Western Ontario and a Ph.D. in Business administration from the University of Wisconsin. His research interests include location models, inventory problems, and microcomputer graphics algorithms.

between points  $i$  and  $j$  is  $F_{ij}(\mathbf{X}, Y)$ :

$$F_{ij}(\mathbf{X}, Y) = w_{ij} [|x_i - x_j| + |y_i - y_j|]. \quad (1)$$

The minimax problem is

$$\text{Minimize}_{\mathbf{X}, Y} \{F(\mathbf{X}, Y)\},$$

where

$$F(\mathbf{X}, Y) = \max_{i,j} \{F_{ij}\}. \quad (2)$$

As is known, problem (2) can be decomposed into two one-dimensional problems.

*Lemma 1:*

$$|x_1 - x_2| + |y_1 - y_2| = \max \{|x_1 + y_1 - x_2 - y_2|, |x_1 - y_1 - x_2 + y_2|\}.$$

*Proof:*

If the signs of  $x_1 - x_2$  and  $y_1 - y_2$  are the same, then  $|x_1 - x_2| + |y_1 - y_2| = |x_1 - x_2 + y_1 - y_2|$  and  $|x_1 - y_1 - x_2 + y_2|$  is smaller because it is equal to  $||x_1 - x_2| - |y_1 - y_2||$ . A similar argument holds for the case when the two signs are opposite to each other, and the lemma is easily verified.

The following transformation decomposes problem (2) into two independent one-dimensional problems:

$$u_i = x_i + y_i,$$

$$v_i = x_i - y_i.$$

By Lemma 1:

$$F_{ij}(\mathbf{X}, Y) = w_{ij} \max \{|u_i - u_j|, |v_i - v_j|\},$$

and therefore,

$$F(\mathbf{X}, Y) = \max \{F(U), F(V)\},$$

where

$$F(U) = \max_{ij} \{w_{ij} |u_i - u_j|\},$$

$$F(V) = \max_{ij} \{w_{ij} |v_i - v_j|\}.$$

The best solution is obtained by solving for the minimum of  $F(U)$  and  $F(V)$  independently. For simplicity of notation let us rename the  $u_i$ 's as  $x_i$ 's, and define  $F_{ij}(\mathbf{X}) = w_{ij} |x_i - x_j|$ .

#### THE UNIVARIATE SEARCH ALGORITHM

Since very efficient algorithms exist for the one-center problem [7, 10, 11], i.e.  $s = 1$ , we propose an algorithm that optimizes in turn the location of single free points.

Some additional notation is necessary:

$$F_i(\mathbf{X}) = \max_{1 \leq j \leq n} \{F_{ij}(\mathbf{X})\} \text{ for } i = 1, \dots, s. \quad (3)$$

Then

$$F(\mathbf{X}) = \max_{1 \leq i \leq s} \{F_i(\mathbf{X})\}. \quad (4)$$

Also define

$$J(\mathbf{X}) = \{i \mid F_i(\mathbf{X}) = F(\mathbf{X})\}. \tag{5}$$

Since  $F_{ij}(\mathbf{X})$  are convex,  $F_i(\mathbf{X})$  and  $F(\mathbf{X})$  are convex also.

One can choose any starting point for the following algorithm. However, we chose as starting point the solution to the one-center problem when all free points are forced to be at a single point. The weight between this “conglomerate” center and fixed point  $j$  is  $\max_{1 \leq i \leq s} \{w_{ij}\}$ .

*Algorithm 1:* (univariate search)

- (1) Set  $k = 0$ , where  $k$  is the iteration number. Find a starting point  $\mathbf{X}^{(0)}$ .
- (2) Find  $J(\mathbf{X}^{(k)})$ .
- (3) Treat all the members of  $J(\mathbf{X}^{(k)})$  in turn as single free points and solve the resulting one-center problems; relocate the members of  $J(\mathbf{X}^{(k)})$  accordingly. Let  $\bar{\mathbf{X}}^{(k)}$  be these revised locations of the points in  $J(\mathbf{X}^{(k)})$ .
- (4) If  $F(\mathbf{X}^{(k)}) - F(\bar{\mathbf{X}}^{(k)}) > \epsilon$  (where  $\epsilon$  is a prespecified small positive “stopping criterion”), then set  $k = k + 1$ ,  $\mathbf{X}^{(k+1)} = \bar{\mathbf{X}}^{(k)}$  and go to step (2).
- (5) If  $J(\bar{\mathbf{X}}^{(k)}) \neq J(\mathbf{X}^{(k)})$  update  $\mathbf{X}^{(k)} = \bar{\mathbf{X}}^{(k)}$  and go to step (2).
- (6) If  $J(\bar{\mathbf{X}}^{(k)}) = J(\mathbf{X}^{(k)})$  stop with  $\bar{\mathbf{X}}^{(k)}$  as the solution.

Note that the one-center problem possesses a unique solution point, and therefore, all the values in the algorithm are well defined.

In the following two theorems we prove that the algorithm converges to the optimal solution. First, we prove a lemma. Let  $\phi(\mathbf{X})$  be the transformed vector  $\mathbf{X}$  after one iteration of Algorithm 1, i.e.  $\phi(\mathbf{X}^{(k)}) = \mathbf{X}^{(k+1)}$ .

*Lemma 2:*

If  $F[\phi(\mathbf{X})] = F(\mathbf{X})$ , then either  $J[\phi(\mathbf{X})] \subset J(\mathbf{X})$  or  $\phi(\mathbf{X}) = \mathbf{X}$ .

*Proof:*

Each one-center problem has a unique solution. Consider solving the one-center problem involving point  $i \in J(\mathbf{X})$ . One possibility is that the new location is different from the previous one, and thus  $F_i(\mathbf{X})$  decreased and hence  $J(\mathbf{X})$  is reduced by the removal of  $i$ . If for all  $i \in J(\mathbf{X})$  the solution remains the same, then  $\phi(\mathbf{X}) = \mathbf{X}$ , and the lemma is proven.

*Theorem 1:*

Algorithm 1 converges.

*Proof:*

Since  $F(\mathbf{X}^{(k+1)}) \leq F(\mathbf{X}^{(k)})$  and since  $F(\mathbf{X}^{(k)}) \geq 0$ , there must be some  $F^0$  such that  $\lim_{k \rightarrow \infty} F(\mathbf{X}^{(k)}) = F^0$ . Since  $\mathbf{X}^{(k)}$  are in the convex hull of the points and therefore are bounded, there must exist a subsequence  $\mathbf{X}^{(k_i)}$  of  $\mathbf{X}^{(k)}$  such that  $\lim_{i \rightarrow \infty} \mathbf{X}^{(k_i)} = \mathbf{X}^0$ . Let  $\mathbf{X}^1 = \phi(\mathbf{X}^0)$ ,  $\mathbf{X}^2 = \phi(\mathbf{X}^1)$ , and so on. As  $\phi(\mathbf{X})$  is a continuous function of  $\mathbf{X}$  (a solution to a one-center problem is continuous as a function of the locations of the points):

$$\lim_{i \rightarrow \infty} \mathbf{X}^{(k_i+1)} = \lim_{i \rightarrow \infty} \phi(\mathbf{X}^{(k_i)}) = \phi(\lim_{i \rightarrow \infty} \mathbf{X}^{(k_i)}) = \phi(\mathbf{X}^0) = \mathbf{X}^1.$$

In the same way:  $\lim_{i \rightarrow \infty} \mathbf{X}^{(k_i+t)} = \mathbf{X}^t$ , for  $t = 0, 1, 2, \dots$ . Since  $F(\mathbf{X}^t) = F^0$  for  $t = 0, 1, 2, \dots$ , then by Lemma 2, if  $\phi(\mathbf{X}^t) \neq \mathbf{X}^t$  then  $J(\mathbf{X}^t)$  shrinks; however after some iterations we must have  $J(\mathbf{X}^{t+1}) = J(\mathbf{X}^t)$ , and therefore, for that  $t$ ,  $\phi(\mathbf{X}^t) = \mathbf{X}^t$ . This means that  $\lim_{k \rightarrow \infty} \mathbf{X}^{(k)} = \mathbf{X}^t$ , and the theorem is proved.

*Theorem 2:*

Algorithm 1 converges to the optimum.

*Proof:*

By Theorem 1, Algorithm 1 converges to some point  $\mathbf{X}^0$ . Let

$$I = \{i, j \mid F_{ij}(\mathbf{X}^0) = F(\mathbf{X}^0)\}, \quad (6)$$

and let  $I$  have  $t$  members. Also, let  $i_1, \dots, i_t$  be the members of  $I$  ordered so that  $x_{i_1} \leq \dots \leq x_{i_t}$ . Every free point,  $j \in I$ , must have two points, one to its left,  $j_1$ , and one to its right,  $j_2$ , such that  $x_{j_1} < x_j < x_{j_2}$ , and  $F_{j_1 j}(\mathbf{X}^0) = F_{j j_2}(\mathbf{X}^0) = F(\mathbf{X}^0)$ . If this were not so (for example there is no such  $j_1$ ), then the solution to the one-center problem for  $j$  could have been improved by moving point  $j$  to the right. It follows that  $i_1$  and  $i_t$  must be fixed points.

Construct a sequence  $j_1, \dots, j_v$  of  $i_1, \dots, i_t$  such that it meets the conditions:

$$\begin{aligned} (1) \quad & j_1 = i_1; \\ (2) \quad & x_{j_k} < x_{j_{k+1}} \text{ for } k = 1, \dots, v-1; \\ (3) \quad & F_{j_k j_{k+1}}(\mathbf{X}^0) = F(\mathbf{X}^0) \text{ for } k = 1, \dots, v-1; \\ (4) \quad & j_v \text{ is a fixed point.} \end{aligned} \quad (7)$$

Such a sequence must exist because if  $x_{j_k}$  is not a fixed point there must be a point to serve as  $j_{k+1}$ . Note that  $v \geq 3$ . Now:

$$F(\mathbf{X}^0) = w_{j_k j_{k+1}} (x_{j_{k+1}} - x_{j_k}) \text{ for } k = 1, \dots, v-1. \quad (8)$$

If we divide each equation of (8) by the weight and then sum up the equations we get

$$x_{j_v} - x_{j_1} = WF(\mathbf{X}^0),$$

where

$$W = \sum_{k=1}^{v-1} [1/w_{j_k j_{k+1}}].$$

Therefore,

$$F(\mathbf{X}^0) = (x_{j_v} - x_{j_1})/W. \quad (9)$$

Consider now any  $\mathbf{X}$ . Then:

$$F(\mathbf{X}) \geq w_{j_k j_{k+1}} (x_{j_{k+1}} - x_{j_k}) \text{ for } k = 1, \dots, v-1.$$

Dividing by the weights and summing up the inequalities as done before for the equalities we obtain:  $F(\mathbf{X}) \geq (x_{j_v} - x_{j_1})/W$ . Since  $j_1$  and  $j_v$  are fixed points this inequality means:  $F(\mathbf{X}) \geq F(\mathbf{X}^0)$ , and  $\mathbf{X}^0$  must be optimal.

It is interesting to see what the equivalent theorems are for Euclidean distances rather than for the rectilinear distances discussed here. It is easy to verify that Theorem 1 still holds for Euclidean distances, but Theorem 2 does not. The univariate search converges to a point, but (as seen by the example in the Appendix) it does not necessarily converge to the optimum.

#### THE STEEPEST DESCENT ALGORITHM

$F(\mathbf{X})$  is convex but its gradient is not continuous at many points. Moving in the direction of steepest descent is similar to moving on an edge of a simplex [1, 4, 8, 12]. Assume we change  $\mathbf{X}$  by an infinitesimal vector  $\mathbf{H}$  such that  $x_i$  is moved to  $x_i + h_i$ . Then

$$\begin{aligned} F_{ij}(\mathbf{X} + \mathbf{H}) - F_{ij}(\mathbf{X}) &= w_{ij} \{ |x_i - x_j + h_i - h_j| - |x_i - x_j| \} \\ &= w_{ij} \delta_{ij} (h_i - h_j), \end{aligned}$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } x_i \geq x_j \\ -1 & \text{if } x_i < x_j \end{cases}.$$

For an infinitesimal  $\mathbf{H}$ ,  $F(\mathbf{X} + \mathbf{H})$  is determined only by those pairs  $(i, j)$  for which  $F_{ij}(\mathbf{X}) = F(\mathbf{X})$ . Let

$$I(\mathbf{X}) = \{(i, j) \mid F_{ij}(\mathbf{X}) = F(\mathbf{X})\}. \quad (10)$$

Then

$$F(\mathbf{X} + \mathbf{H}) = F(\mathbf{X}) + \max_{i,j \in I(\mathbf{X})} \{w_{ij} \delta_{ij} (h_i - h_j)\}.$$

The descent along direction  $\mathbf{H}$  is

$$-[F(\mathbf{X} + \mathbf{H}) - F(\mathbf{X})]/(\mathbf{H}^T \mathbf{H})^{1/2}.$$

Therefore, the steepest descent is obtained when

$$\max_{i,j \in I(\mathbf{X})} \{w_{ij} \delta_{ij} (h_i - h_j)\} / [\sum h_i^2]^{1/2}$$

is minimized. Alternatively,

$$\text{minimize } \{M\},$$

subject to (11)

$$w_{ij} \delta_{ij} (h_i - h_j) \leq M [\sum h_i^2]^{1/2} \text{ for } i, j \in I(\mathbf{X}).$$

It is easy to verify that when  $M$  is negative then (11) is equivalent to

$$\text{Minimize } \{\sum h_i^2\},$$

subject to (12)

$$w_{ij} \delta_{ij} (h_i - h_j) \leq -1 \text{ for } i, j \in I(\mathbf{X}).$$

Also, when there is no feasible solution to (11) when  $M$  is negative, then there is no feasible solution to (12). Note also that  $h_i = 0$  when point  $i$  is fixed.

Problem (12) is a quadratic programming problem. Solution procedures for it are well known: for example see Beale [2] or Wolfe [15]. As our objective function is merely a sum of squares, we found the following specialized approach useful. The problem is

$$\text{Minimize } \{\mathbf{X}^T \mathbf{X}\},$$

subject to (13)

$$\mathbf{A}\mathbf{X} \geq \mathbf{B}.$$

Applying the Kuhn–Tucker conditions to (13) and setting  $\mathbf{Y} = \mathbf{A}\mathbf{X} - \mathbf{B}$  yields the solution

$$\mathbf{X} = \mathbf{A}^T \mathbf{V}, \quad (14)$$

where  $V$  is the solution to

$$AA^T V - Y = B, \quad (15a)$$

$$V^T Y = 0, \quad (15b)$$

$$V, Y \geq 0. \quad (15c)$$

Finding a feasible solution to equations (15) is a complementary programming problem as presented by Dantzig and Cottle [3]; a solution method is given by Lemke [9]. A computer program for a complementary programming solution is presented in [14].

Let  $\mathbf{H}^*$  be the solution to (12). Recall that  $\mathbf{H}^*$  is a vector in the direction of steepest descent of  $F(\mathbf{X})$ . Essentially we will move  $\mathbf{X}$  in the direction  $\mathbf{H}^*$  until there is a new addition to  $I(\mathbf{X})$ . In other words, until a new pair  $(i, j)$ , where  $(i, j) \notin I(\mathbf{X})$ , fulfils  $F_{ij}(\mathbf{X}) = F(\mathbf{X})$ . Let the new position of  $\mathbf{X}$  along  $\mathbf{H}^*$  be  $\mathbf{X} + a\mathbf{H}^*$  where  $a \geq 0$ .

First, we note that for at least one pair  $(i, j) \in I(\mathbf{X})$ , the inequality  $w_{ij}\delta_{ij}(h_i^* - h_j^*) \leq -1$  will be an equality. For this pair (or pairs):

$$F_{ij}(\mathbf{X} + a\mathbf{H}^*) = F_{ij}(\mathbf{X}) - a = F(\mathbf{X}) - a.$$

For the entering pair(s), i.e. for those  $(i, j) \notin I(\mathbf{X})$  that join  $I(\mathbf{X})$ ,

$$F_{ij}(\mathbf{X} + a\mathbf{H}^*) = F(\mathbf{X}) - a \quad (16)$$

at the point of entry.

We solve (16) for the smallest possible  $a$ . The smallest  $a$  for all  $(i, j) \notin I(\mathbf{X})$  will give us the entry point and the pair(s) that enters  $I(\mathbf{X})$ . Note the similarity of this approach to the concept of the simplex method of linear programming. To find this  $a$ , define an  $a_{ij}$  as the smallest positive number such that

$$F_{ij}(\mathbf{X} + a_{ij}\mathbf{H}^*) = F(\mathbf{X}) - a_{ij} \text{ for } (i, j) \notin I(\mathbf{X}). \quad (17)$$

Recall that  $F_{ij}(\mathbf{X}) = w_{ij} |x_i - x_j|$ . Substituting it in (17) and solving for  $a_{ij}$  yields

$$a_{ij} = \min \{a_1, a_2\}, \quad (18)$$

where

$$a_1 = \left\{ \begin{array}{ll} [F(\mathbf{X}) - w_{ij}(x_i - x_j)]/[1 + w_{ij}(h_i^* - h_j^*)] & \text{if } 1 + w_{ij}(h_i^* - h_j^*) > 0 \\ \infty & \text{otherwise} \end{array} \right\},$$

$$a_2 = \left\{ \begin{array}{ll} [F(\mathbf{X}) + w_{ij}(x_i - x_j)]/[1 - w_{ij}(h_i^* - h_j^*)] & \text{if } 1 - w_{ij}(h_i^* - h_j^*) > 0 \\ \infty & \text{otherwise} \end{array} \right\}.$$

For  $(i, j) \notin I(\mathbf{X})$  the numerators in  $a_1$  and  $a_2$  are positive and at least one of the denominators is positive (because their sum is 2). Therefore, there exists a positive solution to  $a_{ij}$ . Let  $a^*$  be the maximal  $a$  such that  $F(\mathbf{X} + a\mathbf{H}^*) = F(\mathbf{X}) - a$ ; then

$$a^* = \min_{(i,j) \in I(\mathbf{X})} \{a_{ij}\}. \quad (19)$$

At  $\mathbf{X} + a^*\mathbf{H}^*$  we have  $F(\mathbf{X} + a^*\mathbf{H}^*) = F(\mathbf{X}) - a^*$ . Let

$$I_1 = \{(i, j) \mid a_{ij} = a^*\};$$

$$I_2 = \{(i, j) \mid (i, j) \in I(\mathbf{X}) \text{ and } F_{ij}(\mathbf{X} + a^*\mathbf{H}^*) < F(\mathbf{X}) - a^*\}.$$

Table 1. Randomly generated problems

No. of free points	No. of fixed points	Algorithm 1		Algorithm 2	
		Iter.	Time	Iter.	Time
5	100	69	0.17	5	0.07
10	200	2	0.02	2	0.05
20	400	66	0.32	4	0.17
40	800	7	0.13	5	0.53
100	2000	68	1.12	*	*
200	4000	7	0.50	*	*

\*Algorithm 2 was not run due to excessively large computer storage needed.

Then

$$I(\mathbf{X} + a^*\mathbf{H}^*) = [I(\mathbf{X}) \cup I_1] - I_2. \quad (20)$$

We sum up the algorithm as follows:

*Algorithm 2:* (steepest descent)

- (1) Set  $k = 0$  ( $k$  is the iteration counter); choose a starting point  $\mathbf{X}^{(0)}$ .
- (2) Compute  $I(\mathbf{X}^{(0)})$ ,  $F(\mathbf{X}^{(0)})$ .
- (3) Compute  $\mathbf{H}^*$  by (12). If there is no feasible solution, then  $\mathbf{X}^{(k)}$  is optimal; stop.
- (4) Compute  $a^*$  by (19).
- (5) Set  $\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + a^*\mathbf{H}^*$ .
- (6) Update  $I(\mathbf{X}^{(k)})$  to get  $I(\mathbf{X}^{(k+1)})$  by (20).
- (7) Update  $F(\mathbf{X}^{(k+1)}) = F(\mathbf{X}^{(k)}) - a^*$ .
- (8) Return to step (3) with  $k = k + 1$ .

#### COMPUTATIONAL RESULTS

In Table 1, results from randomly generated problems are presented. All parameters were generated by a uniform distribution. For each free point we generated 20 weights to a group of 20 fixed points and nonzero weights to all other free points. As a starting point we employ the solution of the one-center problem obtained by forcing all free points to one site. We used (for algorithm 1)  $\epsilon = 10^{-5} F(\mathbf{X})$ , giving a relative accuracy of  $10^{-5}$ . Time is in seconds on a CDC/6400 system; the algorithms were coded in FORTRAN IV. We also generated a problem with 750 free points and 15 000 fixed points. We circumvented the need for huge computer storage by setting all weights between free points to 10 (this did not decrease the "difficulty" of the problem). This problem was solved by algorithm 1 in 49 iterations and 5.50 sec! Note that Algorithm 2 requires more computer memory than Algorithm 1.

#### REFERENCES

1. J. W. Bandler, T. V. Srinivasan and C. Charalambus, Minimax optimization of networks by grazor search. *IEEE Trans. Microwave Theory Tech.* **20**, 596–604 (1972).
2. E. M. L. Beale, *Mathematical Programming in Practice*. John Wiley, New York (1968).
3. G. B. Dantzig and W. Cottle, Complementary pivot theory of mathematical programming. *J. Linear Algebra Appl.* **1**, 103–125 (1968).
4. V. F. Demjanov, Algorithms for some minimax problems. *J. Comput. Systems Sci.* **2**, 342–380 (1968).
5. Z. Drezner, DISCON—A new method for the layout problem. *Oper. Res.* **28**, 1375–1384 (1980).
6. Z. Drezner and G. O. Wesolowsky, A new method for the multifacility minimax location problem. *J. Oper. Res. Soc.* **29**, 1095–1101 (1978).
7. Z. Drezner and G. O. Wesolowsky, Single facility  $\ell_p$  distance minimax location. *SIAM J. Alg. Discrete Methods* **1**, 315–321 (1980).
8. Y. Ishizaki, and H. Watanabe, An iterative Chebyshev approximation method for network design. *IEEE Trans. Circ. Theory* **CT-15**, 326–336 (1968).
9. C. Lemke and J. T. Howson, Jr., Equilibrium points of bimatrix games. *J. SIAM*, **12**, 413–423 (1964).
10. N. Megiddo, The weighted Euclidean one-center problem. *Math. Oper. Res.* **8**, 498–504 (1983).
11. N. Megiddo, Linear programming in linear time when the dimension is fixed. *J. ACM* **31**, 114–127 (1984).

12. M. R. Osborn and G. A. Watson, An algorithm for minimax approximation in the nonlinear case. *Comput. J.* **12**, 63–68 (1969).
13. J. C. Picard and H. D. Ratliff, A cut approach to the rectilinear distance facility location. *Oper. Res.* **26**, 422–433 (1978).
14. A. Ravindran, Algorithm 431: A computer routine for quadratic and linear programming problems. *Commun. ACM* **15**, 818–820 (1972).
15. P. Wolfe, The simplex method for quadratic programming. *Econometrica* **27**, 382–398 (1959).

## APPENDIX

In this Appendix we give an example of using Euclidean distances, in which the univariate search may converge to a nonoptimal point.

The problem is described in Fig. A1.

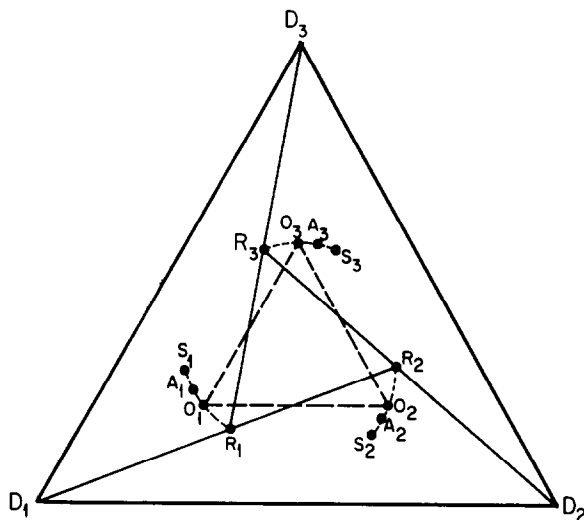


Fig. A1. An Euclidean Distance Problem.

Three fixed points are located at  $D_1$ ,  $D_2$ ,  $D_3$ . The three free points are connected among themselves by weights of “1”, and each free point is connected to one fixed point by a weight of 1. The coordinates of the points marked with  $D$ ,  $O$ ,  $R$ , and  $S$  (three points of each type) are given in Table A1.

Suppose the heuristic achieved a solution  $R_1$ ,  $R_2$ ,  $R_3$ , where all nonzero weighted distances are equal to  $1/\sqrt{7} = 0.377\ 96$ . Every free point is equidistant with three other equally weighted points and is on the line connecting two of them. Hence, Algorithm 1 will not change their locations [7].

However, if we rotate the inner triangle and “contract” it a little so that its edges are parallel to the edges of the bigger triangle, we can achieve another “equilibrium” solution with a lower maximum distance. The solution consists of points  $O_1$ ,  $O_2$ ,  $O_3$  which give the optimal objective function of  $(\sqrt{3} - 1)/2 = 0.366\ 03$ .

Actually, the solution “triangle” can be rotated from the  $R$ 's to the  $S$ 's and every such solution would be terminal for Algorithm 1. With our usual starting point, Algorithm 1 terminated with  $A_1$ ,  $A_2$ ,  $A_3$ ; this is depicted in Fig. A1; the objective function was 0.367 39.

Table A1. Coordinates

	$x_1$	$y_1$	$x_2$	$y_2$	$x_3$	$y_3$
$D$	$-1/2$	$0$	$1/2$	$0$	$0$	$\sqrt{3}/2$
$O$	$-(\sqrt{3} - 1)/4$	$(\sqrt{3} - 1)/4$	$(\sqrt{3} - 1)/4$	$0$	$0$	$1/2$
$R$	$-1/7$	$\sqrt{3}/14$	$3/14$	$\sqrt{3}/7$	$-1/14$	$2\sqrt{3}/7$
$S$	$-3/14$	$\sqrt{3}/7$	$1/7$	$\sqrt{3}/14$	$1/14$	$2\sqrt{3}/7$