Efficient Algorithm for Locating a New Transportation Facility in a Network

HUEL-SHEN TSAY AND LIANG-TAY LIN

The single-location problem is to locate a new transportation facility in a network that can serve all customers at the minimum distance or cost. There are four types of single-location problems. The absolute 1-center problem is considered in this paper. By definition, in that problem, the customers are on any vertex and the center may be a vertex or a point on an edge. There are two previous methods for finding the absolute 1-center: (a) the Hakimi method (1965) and (b) the Minieka method (1981). They considered all possible links of a network to determine the best candidate point. Later, Larson and Odoni proposed a shortcut to reduce the number of links needed for calculation. In this paper, a new shortcut with a stricter bound is first proposed to find the absolute 1-center directly. The Larson and Odoni shortcut is then introduced and integrated with the Minieka method to form a combined method. Finally, a new method is developed to find the absolute 1-center based on a spanning tree that is obtained from that of the vertex to all shortest distances. The number of iterations needed to perform the analysis is in proportion to the number of vertices instead of edges for any given network. To make a consistent comparison, four different methods have been programmed and tested with several networks. The results show that the new method or the new shortcut is fast and powerful in finding the absolute 1-center location. They provide the same solutions and belong to polynomial time-bounded algorithms. Therefore, we recommend use of the new method or shortcut for locating a new facility if the absolute 1-center problem is considered in a network.

In selecting the optimal facility, location plays a vital role in the fields of transportation, communications, and distribution management. Applications may include transit stops, fire stations, warehouses or plant locations, post offices, schools, and public buildings. A major concern in location models is to find the optimal placement of facilities on a network so the cost of locating, operating, and providing service is minimized. Here, the cost of serving customers can be defined as the cost incurred between customers and the assigned depot; it refers to the transportation cost that is primarily due to the distance traveled to and from the depot location. Therefore, the back-and-forth distance between two nodes is an important component in determining the location of new facilities.

Generally speaking, network location research can be categorized into two types: single-depot location and multiple-depot locations. The single-depot location problem considers locating only one depot in the network, either to minimize loss or to maximize benefit or to provide good service to customers. This facility and its customers may be located at the vertex (node) or anywhere along two vertices. The multiple-depot location problem, on the other hand, finds locations for more than one depot to serve all customers with an objective of minimizing total related cost, minimizing the maximum travel distance, or providing the best service.

Graduate School of Transportation and Communication Management Science, National Cheng Kung University, Taiwan, Republic of China.

Because locating a new transportation facility in a given network is our main consideration, it is necessary to know the differences between various types of single-location problems. There are four major types of single-location problems shown in Table 1. From Table 1, the vertex also represents nodes, and each link or edge has an infinite number of possible point locations. The vertex 1-center and general 1-center location problems have been solved and programmed through efficient methods (*1,2*). These are all polynomial, time-bounded algorithms.

The absolute 1-center problem is defined as a point located such that the maximum distance from this facility to any node is minimized. This new location can be anywhere on a link (edge) or at a node (vertex). Basically, it is a problem of one point serving multiple nodes. One application of the absolute 1-center problem, for example, locates a fire station in a rural community in a manner that minimizes the maximum response time from the station to any farmhouse. It was first presented by Hakimi (*3,4*). The literature on network location problems has grown rapidly since the appearance of Hakimi's paper (*5*). The Hakimi method, for each link, constructs upper envelopes continuously to compute the intersecting points from all nodes in the network. From all feasible intersecting points, we choose the best local minimum for the corresponding link. Once all links have been examined, the best among all such local minima is selected as the absolute 1-center of a given network. Its solution is more difficult and complex than that of either the vertex 1-center or general 1-center problem. In this paper, four methods for solving the absolute 1-center problem are extensively discussed and compared.

The general absolute 1-center location problem is, among four types of single-location problems, the most difficult to solve. This is a problem of one point serving an infinite number of customer points on each link. Recently, some algorithms have been developed and proved to be effective (*1,2,6*). Because the absolute 1-center is our focus, the general absolute 1-center location problem is not discussed here.

LARSON AND ODONI SHORTCUT

Because the Hakimi method requires the examination of each link before the best absolute center in a network is chosen, the number of calculations grows rapidly and sometimes becomes unacceptably large as the number of links increases.

TABLE 1 FOUR TYPES OF SINGLE LOCATION PROBLEMS

Type	Facility Location	Customer Locations
Vertex 1-center	Vertex	Vertex
General 1-center	Vertex	Link*
Absolute 1-center	Link*	Vertex
General Absolute 1-center	Link*	Link*

* **Represents infinite possible points located on each link or edge.**

Some links, in fact, cannot further improve the optimal solutions. Larson and Odoni proposed a shortcut to reduce the computational effort required to obtain the absolute 1-center (7). That shortcut takes advantage of the fact that it is simple to find the optimal solution of a vertex 1-center problem in a given network. This solution is then treated as the upper bound value to identify those links that actually cannot improve the final result. This shortcut is represented by the following equation:

$$\frac{m(r,a) + m(s,b) - l(r,s)}{2} < m(i^*) \quad (1)$$

where

$m(r,a)$ = the distance required for node r to serve the farthest node a in the network;
$m(s,b)$ = the distance required for node s to serve the farthest node b in the network;
$l(r,s)$ = link distance between nodes r and s; and
$m(i^*)$ = the optimal solution of vertex 1-center.

It implies that the Hakimi method can be applied to those links that do not violate Equation 1. So, for a link (r,s) that satisfies Equation 2:

$$\frac{m(r,a) + m(s,b) - l(r,s)}{2} \geq m(i^*) \quad (2)$$

The local 1-center of this link (r,s) cannot further improve the vertex 1-center solution $m(i^*)$. The fact is that the maximum distance associated with the vertex 1-center must be greater than or equal to the corresponding distance for the absolute 1-center (7). In other words, if Equation 2 is satisfied, the link (r,s) need not be examined further. Through such a test, considerable computational effort will be reduced. But the number of computations that can actually be saved depends on the network configuration. It is difficult to predict a specific number of reductions if the shortcut is applied. However, this shortcut shows its ability to eliminate several unnecessary calculations.

A NEW SHORTCUT

A new shortcut is proposed in this section. Nodes a and b are assumed to be the farthest nodes that can be reached by nodes r and s shown on Figure 1a. Then, we have

$$\overline{ra} = m(r,a)$$

$$\overline{sb} = m(s,b)$$

There exists one point p on the path r-a that makes $\overline{pa} = m(i^*)$. Similarly, there is another point q on $\overline{sb}$ with the property of $\overline{qb} = m(i^*)$. Equation 1 can be rearranged in the following form:

$$\frac{m(r,a) - m(i^*) + m(s,b) - m(i^*) - l(r,s)}{2} \leq 0 \quad (3)$$

Then, based upon the preceding definitions, we have

$$\overline{rp} + \overline{sq} - l(r,s) \leq 0 \quad (4)$$

It means that any link in the network satisfying Equation 4 may be able to improve the final solution of the absolute 1-center. Only such a link will be considered in making further calculations. From Figure 1b, x and y are defined as:

$$\{x \mid \overline{xr} + \overline{ra} = \overline{xs} + \overline{sa} \text{ , } x \in \text{link } (r,s)\} \quad (5)$$

$$\{y \mid \overline{ys} + \overline{sb} = \overline{yr} + \overline{rb} \text{ , } y \in \text{link } (r,s)\} \quad (6)$$

Let x' and y' have this relationship:

$$\overline{x'r} = 2\overline{xr} + \overline{rp}$$

$$\overline{y's} = 2\overline{ys} + \overline{sq}$$

As far as Δ rsa (Figure 1b) is concerned, the distance from x passing through node r to node a should be equal to the distance traveling from x through nodes s to a. Suppose x is the absolute 1-center of a given network; path x-r-a will have the longest distance. This value can be further decreased if the absolute 1-center is not located on x. There are two possibilities. The first way considers the center located on the left-hand side of x. In such circumstances, the best location obviously belongs to node r. The distance from node r to the farthest node a is $m(i^*)$ plus $\overline{rp}$, $\overline{ra} = m(i^*) + \overline{rp}$. It is greater than or at most equal to $m(i^*)$ and may not be the best choice. Another possibility is to move the center to the right-hand side of x. The distance from x' to a becomes $m(i^*)$ if only $\overline{rp}$ distance units are shifted from x to x'. Furthermore, once the length of link (r,s) is larger than $\overline{x'r}$ (i.e., $l(r,s) \geq 2\overline{xr} + \overline{rp}$,

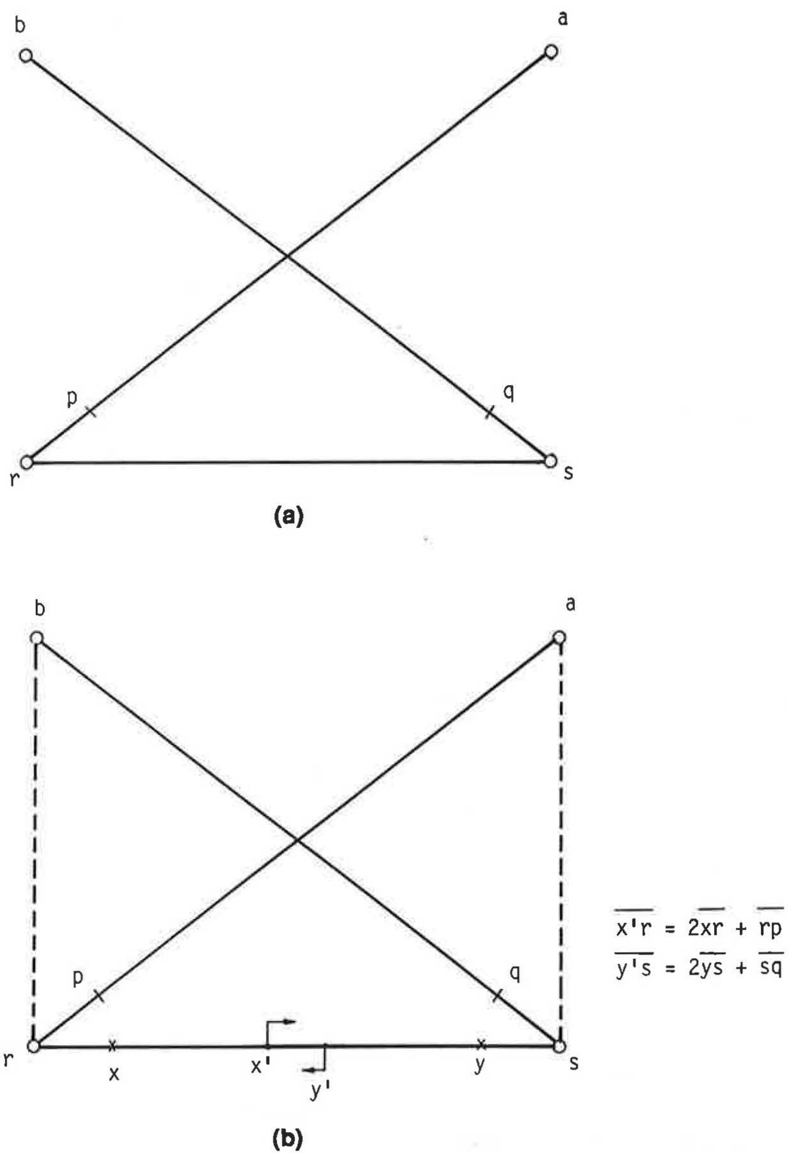


FIGURE 1 (a) Graphic representation of service from link (r,s). (b) Graphic representation of different service range for link (r,s).

the service distance from x' to node a will be less than or equal to $m(i^*)$. It is also necessary to make $l(r,s) \geq 2\overline{ys} + \overline{sq}$ to have the same property. Besides, x' must lie in the left side of y' to guarantee that the shortest distance from x' to node a or from y' to node b is smaller than $m(i^*)$. Thus,

$$l(r,s) \geq 2\overline{xr} + \overline{rp} + 2\overline{ys} + \overline{sq} \tag{7}$$

$$l(r,s) \geq 2\left[\frac{m(r,a) + l(r,s) + m(s,a)}{2} - m(r,a)\right] + m(r,a) - m(i^*)$$
$$+2\left[\frac{m(s,b) + l(r,s) + m(r,b)}{2} - m(s,b)\right] + m(s,b) - m(i^*)$$

$$l(r,s) \geq l(r,s) + m(s,a) - m(i^*) + l(r,s) + m(r,b) - m(i^*)$$

$$\frac{m(s,a) + m(r,b) + l(r,s)}{2} \leq m(i^*) \tag{8}$$

$m(r,b)$ = the distance from node r to the farthest node b (for s) and

$m(s,a)$ = the distance from node s to the farthest node a (for r).

Any link that violates Equation 8 cannot improve the final solution of absolute 1-center and will not be further considered.

Because

$$m(r,a) \leq m(s,a) + l(r,s)$$

$$m(s,b) \leq m(r,b) + l(r,s)$$

$$\frac{m(r,a) + m(s,b) - l(r,s)}{2}$$
$$\leq \frac{m(s,a) + m(r,b) + l(r,s)}{2} \leq m(i^*) \tag{9}$$

It is noted that the proposed shortcut has a more strict bound than the Larson and Odoni shortcut does based on Equation 9. If the proposed shortcut given in Equation 8 is considered, then the local 1-center of link (r,s) is located on the middle point of $\overline{x'y'}$. Its location has DI distance units from node r.

$$\begin{aligned} DI &= \overline{x'r} + 1/2[l(r,s) - \overline{x'r} - \overline{y's}] \\ &= l(r,s) + m(s,a) - m(i^*) + [l(r,s) \\ &\quad - 2l(r,s) - m(s,a) \\ &\quad - m(r,b) + 2m(i^*)] \\ &= 1/2[l(r,s) + m(s,a) - m(r,b)] \end{aligned} \quad (10)$$

The service distance of this local 1-center to the farthest node is

$$\begin{aligned} SS &= m(i^*) - 1/2[l(r,4s) - \overline{x'r} - \overline{y's}] \\ &= m(i^*) - 1/2[l(r,s) - 2l(r,s) - m(s,a) \\ &\quad - m(r,b) + 2m(i^*)] \\ &= 1/2[m(s,a) + m(s,b) + l(r,s)] \end{aligned} \quad (11)$$

Based on the preceding discussion, the proposed shortcut can be performed as follows. First, for any link in the network, we check whether it satisfies Equation 8. If the answer is yes, then Equations 10 and 11 will be applied to find the local 1-center of that link. Otherwise, the link need not be further considered. After all links have been examined, the location and service distance of absolute 1-center for the given network can easily be determined. The foregoing procedure is rather simple and makes it easy to obtain the final solution without using elaborate computations. Comparisons of this new shortcut with other methods are given later.

MINIEKA METHOD

A polynomial time algorithm for finding the absolute 1-center of a network was proposed by Minieka (*8*). This algorithm is combinatorial in nature and requires only knowledge of the shortest path distances between all pairs of nodes. Conceptually, it is different from the Hakimi method. Consider p on a link (r,s) as one point on the link (r,s) that is p units from r and $l(r,s) - p$ units from s, where $0 \le p \le l(r,s)$. Those nodes that are best reached from p by traveling through node r are set in node set R. Similarly, others best reached through node s belong to set S. On the basis of this definition, the Minieka method for finding p^*, the local 1-center on a link (r,s), follows these steps:

Step 1: Obtain the shortest matrix between all nodes through any efficient algorithm.

Step 2: Place all nodes in R, and arrange the sequence of nodes according to the order of their distance from node r, with the most distant node first. Compare the maximum distance from node r to all other nodes of the network with the link length $l(r,s)$ and then store the higher value as the first point-to-node distance.

Step 3: Remove from R and place into S the node that is currently most distant from node r.

Step 4: Compare the distance from node s to the node that has the maximum distance in R with the largest value of the current set S. If this new distance is smaller than the existing maximum distance, go to Step 3; otherwise go to Step 5.

Step 5: Calculate the maximum distance needed from both sets R and S, using the equation

$$MD = [d(r,z_i) + d(s,z_k) + l(r,s)]/2 \quad (12)$$

where

MD = the current maximum distance needed to serve customers in the both sets R and S;
$d(r,z_i)$ = maximum distance from node r to node z_i in the current set R;
$d(s,z_k)$ = maximum distance from node s to node z_k in the current set S; and
$l(r,s)$ = actual link distance between nodes r and s.

Step 6: Compute the p^* by subtracting $d(r,z_i)$ from MD.

Step 7: Go to Step 3 until all other nodes have been examined and moved to set S. Compare the length of link (r,s) with the maximum distance from S, and then store the higher value as a MD with p^* equal to the length of link (r,s).

Step 8: Choose the smallest MD and its related p^* value among all candidates. This is the local 1-center of link (r,s).

The foregoing procedure can be used for finding the local 1-center of link (r,s). Obviously, it is also applicable to all other links. Thus, the local centers of other links are found through the same steps. After all links have been examined, the best absolute 1-center of the network is determined simply by choosing the minimum among all local 1-center candidates. This method performs the preceding steps easily and can be used to solve large network problems. Its computational effort mainly lies in obtaining the all-to-all shortest-distance matrix. Therefore, this is a polynomial time-bounded algorithm and is easy to program.

A COMBINED METHOD

Although the Minieka method is efficient in computing the local 1-center on a link, it still requires much effort to examine all links of a given network if no bounding technique is applied. For the Larson and Odoni shortcut, considerable reduction in computational effort can be achieved by omitting many unnecessary links before searching for the absolute 1-center. After the shortcut is applied, however, the inefficient Hakimi method is used to find local centers for those critical links that do not violate Equation 1. Therefore, it becomes feasible to combine the Minieka method with the Larson and Odoni shortcut to reduce further the number of calculations and computer time. The basic idea of this combination is simply to consider the Larson and Odoni shortcut first in deleting links that cannot improve the solution. Then only those links satisfying Equation 1 are examined and calculated to determine their local centers using the Minieka method. It is expected that the computational effort will be reduced through this combined method. The steps of this combined method are summarized next.

Step 1: Obtain the shortest distance matrix between all nodes.

Step 2: Apply the Larson and Odoni shortcut to delete those links that satisfy Equation 2.

Step 3: Use the Minieka method to find the local 1-center for each critical link and store it as a candidate.

Step 4: Repeat Step 3 until all critical links have been examined.

This combined method takes advantages of the most efficient parts of the Minieka method and the Larson and Odoni shortcut. Tests of a newly developed computer program show that the program works well and reduces some computer time. These tests are discussed more extensively later.

A NEW METHOD

In this section, a new method for finding the absolute 1-center is proposed. The solution is obtained from a spanning tree based on the vertex's one-to-all shortest distances. It first considers the longest and second longest distances of the spanning tree from each node in a network (*9,10*). For each such tree, the local 1-center is found. Then the minimum of local centers is selected as the absolute 1-center for the entire network. Because the new method finds the local 1-center from the spanning tree of each node, the maximum number of iterations needed to perform the computation is in proportion to the number of nodes, instead of links, for the given network. In other words, conceptually, the new method can reduce computer time more than the previous method if larger networks are considered. The steps included in this new method are as follows:

Step 1: Obtain the shortest path for each vertex to all other nodes.

Step 2: Find the farthest node i and second longest distance node j to form the nonoverlap distance $x(i,j)$ according to the minimum spanning tree of each node from Step 1.

Step 3: Determine the service distance of local 1-center for each node and store it as a candidate:

$$1/2[\max x(i,j)] \tag{13}$$

Step 4: Repeat Steps 2 to 3 until all nodes have been examined.

Step 5: Choose the minimum value among all candidates. This is the absolute 1-center for the given network.

It is easy to perform the preceding steps by using the graphic method manually. Steps 2 and 3 need to be modified, however, if the new method is to be programmed. After several tests, it is found that the local 1-center of the designated node may not always be located on the path that includes the longest and second longest distances rooted at each node. More verifying steps must be added to obtain a better solution of the local 1-center. The best way to perform this analysis is to check all connecting links from that node. This can be observed in Figure 2a. Suppose node I, with the longest path I-A and the second longest path I-B, is under consideration. The pivotal local 1-center of node I is located on M with MI distance from node I. Link (I, K) represents one connecting links originating from node I. The service distance SS of the initial local 1-center based on the previous steps equals $\frac{1}{2}(IA + IB)$. If the shortest distances from node K to nodes A and B satisfy Equation 14,

$$\text{Max}\,[D(K,A),D(K,B)] = GL < SS \tag{14}$$

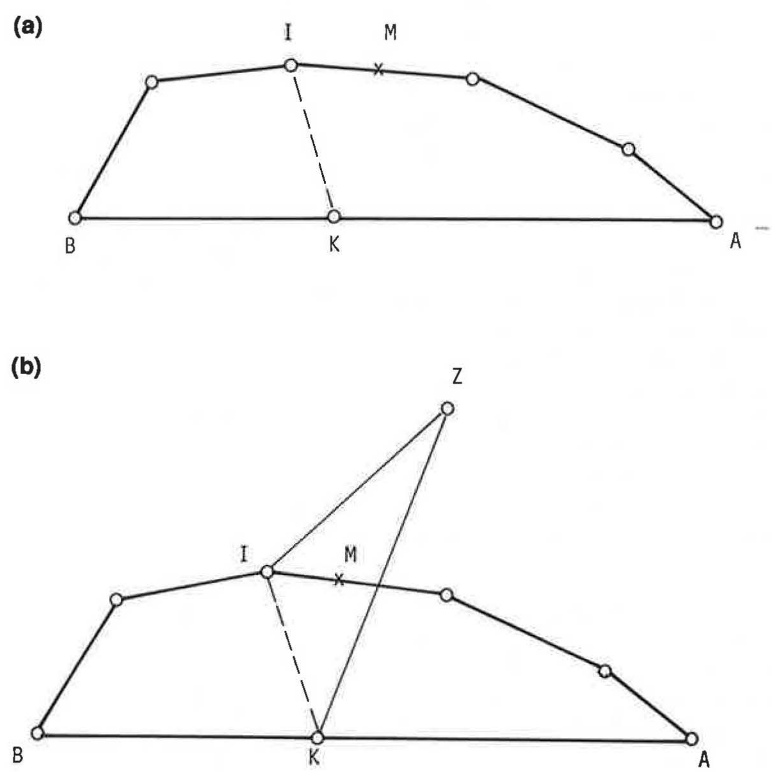


FIGURE 2 (a) The farthest node *A* and second longest distance node *B* for node *I* and one of connecting links (*I*,*K*). (b) Checking steps for connecting links rooted from node *I*.

where

$D(K,A)$ = the shortest distance from nodes K to A;
$D(K,B)$ = the shortest distance from nodes K to B; and
GL = higher value of $D(K,A)$ and $D(K,B)$,

then it is possible for the local 1-center to be located on the connecting link (I,K) instead of the original shortest path. What would be the most desired location of local 1-center for node I? Before performing more analyses, we denote W as the nodes that are different from nodes I, K, A, and B but satisfy the following two conditions.

$$D(K,W) \geq GL$$

$$D(I,W) + D(I,K) \geq GL$$

where

$D(K,W)$ = the shortest distance from nodes K to W;
$D(I,W)$ = the shortest distance from nodes I to W; and
$D(I,K)$ = the shortest distance from nodes I to K.

Let Z be the node that has the largest value among all $D(I,W)$. The new local 1-center stays on link (I,K) if Equation 15 is met.

$$D(I,Z) + D(I,K) \geq GL \tag{15}$$

Otherwise, the local 1-center remains at the node K. The service distance and location of local 1-center for node I become SS and p^*, respectively.

$$SS_1 = 1/2[D(I,Z) + D(I,K) + GL] \tag{16}$$

$$p^* = SS_1 - D(I,Z) \tag{17}$$

After obtaining the new SS_1, if SS_1 is smaller than SS, then SS_1 will substitute SS as the new service distance of local 1-center. The location of this local 1-center is located on the connecting link (I,K) with p^* distance units from node I. Otherwise, the SS value still represents the service distance of local 1-center. After all connecting links have been examined, the smallest value among all SS is chosen. The smallest value and its corresponding location p^* are considered the local 1-center of node I.

To put the preceding discussion into sequential steps, we substitute the following Steps 1a through 5a for Steps 2 and 3 and add Steps 6a to 8a for checking connecting links. Before describing these steps, let $c(i,j)$ be the shortest distance from nodes i to j and $b(i,j) = A$ be the nearest node number on the shortest path from node i to all other nodes j; $g(i,j)$ represents the largest value among all $c(i,j)$, and $h(i,j')$ denotes the second largest value in the all remaining $c(i,j)$. B gives the node letter j that has the second longest distance from node i. Besides, $N(i)$ shows the node letter i currently under consideration.

Step 1a: List all $c(i,j)$ and $b(i,j) = A$ for node i;
Step 2a: Find the largest value $g(i,j)$ among $c(i,j)$ and its nearest node letter A from node i to all other nodes;
Step 3a: Determine the second largest value among remaining $c(i,j)$ that the nearest node number is not A and denote it as $h(i,j')$ and $b(i,j') = B$;
Step 4a: Calculate $Q(i)$ through the following equation:

$$Q(i) = \frac{g(i,j) + h(i,j')}{2} \tag{18}$$

Step 5a: If $[Q(i) - h(i,j')]$ is less than or equal to $c(i,A)$, then $Q(i)$ is the local 1-center distance for node i. Determine the suitable location p^* for $Q(i)$ and go to Step 6a. If $[Q(i) - h(i,j'] > c(i,A)$, go back to Step 4;
Step 6a: Check one connecting link from node i to node k:

$$GL = \text{Max}[D(k,A),D(k,B)]$$

If $GL \leq Q(i)$, go to Step 7a; otherwise, go to Step 8a;

Step 7a: Find z that satisfies the following two conditions and has the largest value:

$$D(i,z) + D(i,k) \geq GL \text{ and } D(k,z) \geq GL$$

If there is no z available, go to Step 8a.

$$Q'(i) = 1/2[D(i,z) + D(i,k) + GL]$$

If $Q'(i) < Q(i)$, then $Q(i) = Q'(i)$, $p^* = Q'(i) - D(i,z)$. Go to Step 8a.

Step 8a: Check other connecting links originating from node i. If all links have been examined, go to Step 4; otherwise, go to Step 6a.

EXAMPLE

Find the absolute 1-center of the network shown in Figure 3 using the new method. This example requires that the shortest distance from each node to all vertices be calculated in the first step. Then, the spanning tree of the designated node based on the shortest distance from each node to all vertices is calculated in the second step. The spanning tree of the designated node based on the shortest distance is then obtained. Figure 4 shows the spanning tree of node 7. After several checking steps, the initial local 1-center becomes the center of node 7. From this figure, it can be seen that the distance between nodes 5 and 10 is 155. Thus, the initial local 1-center for node 7 equals 77.5 units, according to Equation 13. This center will be located on link (7,8) at a distance of 0.5 units from node 7. Similarly, the initial local 1-center of node 6 can be easily obtained from Figure 5a. This initial local 1-center has 62.5 distance units and stays on link (6,10) with a distance

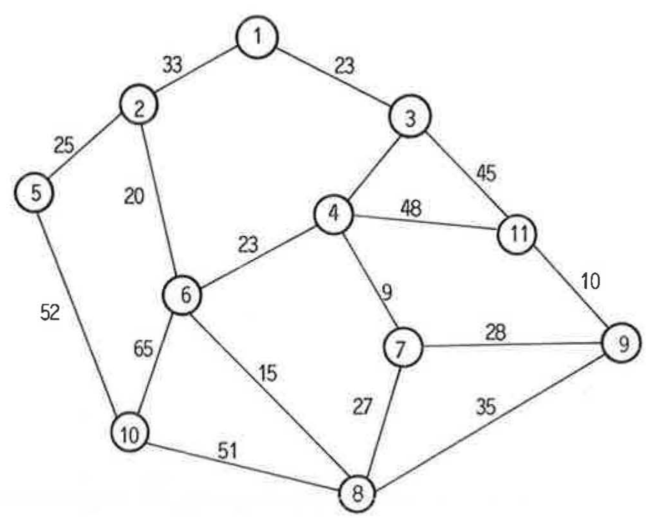


FIGURE 3 Distance and configuration of given network.

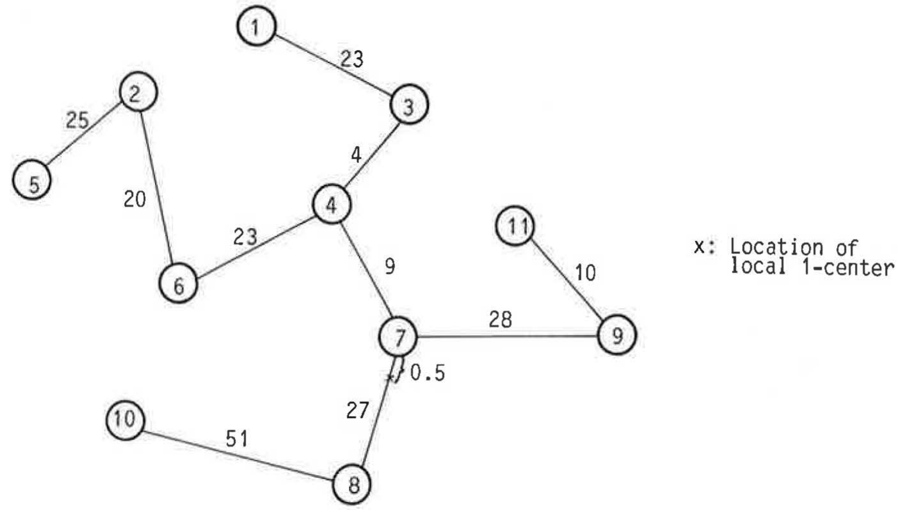
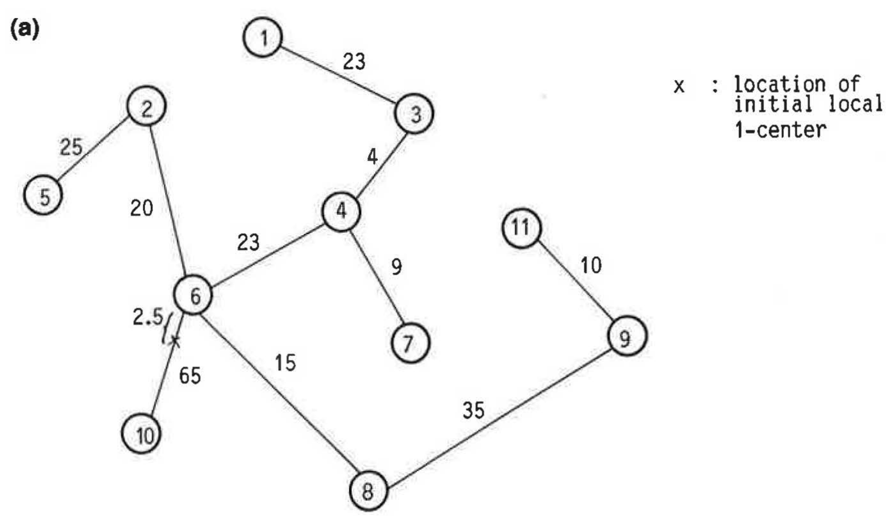


FIGURE 4 Spanning tree of node 7 and its location of local 1-center.



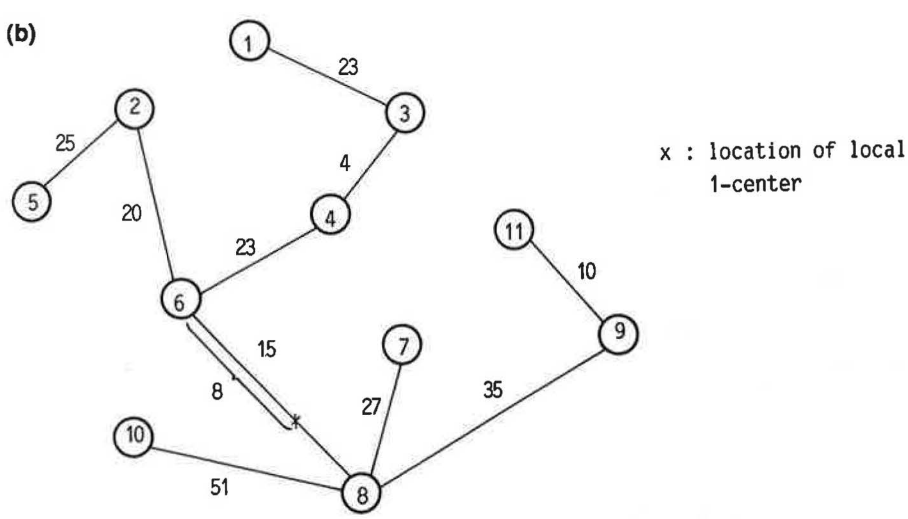


FIGURE 5 (a) Initial local 1-center of node 6 with 62.5 units service distance. (b) Location of local 1-center for node 6 with 58 units.

of 2.5 units from node 6. It is not the best solution for node 6. In fact, the local 1-center of node 6 is located on link (6,8) with 58 distance units from node 6 to serve all other nodes by applying Steps 6a to 8a to examine each connecting link rooted from node 6. Its final location can be seen from Figure 5b. This location and service distance also represents the absolute 1-center for the given network. Therefore, from this example, it is important to examine all connecting links of any designated node before its local 1-center is finally determined.

Another way of searching the local 1-center of each node is simply to apply the given Steps 1a to 8a. A complete computational procedure for node 7 is shown on Table 2 based on these steps. Definitions of all variables in the table are referred to this section. Table 3 gives the computational result of local 1-center for node 6. This case provides the user a better solution after checking each connecting link originated from node 6. It is noted that the results obtained from Table 3 are the same as those shown in Figure 5b.

COMPARISON OF FOUR METHODS

Thus far, four different methods for finding the absolute 1-center have been discussed: the new shortcut, the Minieka method, the combined method, and the new method. As far as computer time and computational complexity are concerned, it is necessary to understand the capabilities and limitations of these four methods. One analytic strategy is to apply four methods to the same given network. To make such a comparison, nine networks with different numbers of nodes and links are selected. In these networks, an absolute 1-center will be sought such that the maximum service distance from this center to all nodes is minimized. The absolute 1-center can be located anywhere on a link or at a node.

Four computer programs have been developed separately for the four methods. Each program reads the same network input file and prints the output in an identical format. Each program was run on a PC/AT with a math coprocessor 80287-10. For each network, the final distance and location

TABLE 2 COMPUTATION OF LOCAL 1-CENTER FOR NODE 7 BY APPLYING STEPS 1a TO 8a

Node i	Node j										
N(7)=7	1	2	3	4	5	6	7	8	9	10	11
c(7,j)=	36	52	13	9	77	32	0	27	28	78	38
b(7,j)=	4	4	4	4	4	4	7	8	9	8	9

A. Search the initial local 1-center:

$g(7,10) = 78, \quad A = 10, \quad h(7,5) = 77, \quad B = 5$

$b(7,10) = 8 \neq b(7,5) = 4$

$$Q(7) = \frac{g(7,10) + h(7,5)}{2} = 77.5$$

$Q(7) - h(7,5) = 77.5 - 77 = 0.5 < c(7,8) = 27$, O.K.

The initial local 1-center of node 7 is located on link (7,8) and 0.5 distance units from node 7.

B. Check each connecting link

(1) link (7,4), k = 4

$GL = \max[D(4,10) = 87, D(4,5) = 68] = 87 \nless Q(7)$

(2) link (7,9), k = 9

$GL = \max[D(9,10) = 86, D(9,5) = 95] = 95 \nless Q(7)$

No connecting links can provide a better solution, so the local 1-center of node 7 is still located on link (7,8) and 0.5 distance units from node 7.

TABLE 3 COMPUTATION OF LOCAL 1-CENTER FOR NODE 6 BY APPLYING STEPS 1a TO 8a

Node i	Node j										
N(6)=6	1	2	3	4	5	6	7	8	9	10	11
c(6,j)=	50	20	27	33	45	0	42	15	50	65	60
b(6,j)=	4	2	4	4	2	6	4	8	4	10	8

A. Search the initial local 1-center:

g(6,10) = 65, A = 10, h(6,11) = 60, B = 11

b(6,10) ≠ b(7,11)

$$Q(6) = \frac{g(6,10) + h(6,11)}{2} = 62.5$$

Q(6) - h(6,11) = 62.5 - 60 = 2.5 < c(6,10) = 60, O.K.

The initial local 1-center of node 6 is located on link (6,10) and 2.5 distance units from node 6.

B. Check each connecting link:

(1) link (6,2), k = 2

GL = max[D(2,10) = 85, D(2,11) = 80] = 85 ≮ Q(6)

(2) link (6,4), k = 4

GL = max[D(4,10) = 87, D(2,11) = 47] = 87 ≮ Q(6)

(3) link (6,8), k = 8

GL = max[D(8,10) = 51, D(2,11) = 47] = 51 < Q(6)

One connecting link (6,8) may provide a better solution. More checking steps need to be undertaken.

D(6,1) + D(6,8) = 65 ≥ GL = 51

D(8,1) = 63 ≥ GL = 51

Q'(6) = [D(6,1)+D(6,8)+GL] = [50+15+51] = 58 ≤ Q(6)

P* = Q'(b) - D(b,1) = 58 - 50 = 8 ≤ c(6,8) = 15 O.K.

Thus, the local 1-center of node 6 is located on link (6,8) and 8 distance units from node 6.

of the absolute 1-center are the same according to the output of the four computer programs. They all provide the best solution. Comparisons of computer time used and the numbers of links and nodes considered by each method are summarized in Table 4. It can be seen that the combined method is more efficient than the Minieka method, because the former skips many unnecessary links before searching the local 1-center. The new method and the new shortcut are obviously better than the combined method. Both the new method and the shortcut use almost the same computer running time. For a network with 80 nodes and 141 links, the new method and shortcut need only 45 percent of the Minieka's computer time and 61 percent of the time required by the combined method. The results show that the new method and the shortcut are computationally fast and powerful if larger networks are considered. Also, both new methods can be categorized as polynomial time-bounded algorithms.

CONCLUSIONS

The combined method finds the absolute 1-center with fewer link computations than the Minieka method does, if the latter is assumed to examine all links. The computational complexity of this technique relies on the efforts of finding the all-to-all shortest distance paths and requires $0(N^3)$ calculations. Hence, the combined method is a polynomially bounded algorithm and requires less computational effort. The proposed new

TABLE 4 COMPARISONS OF FOUR METHODS BY RUNNING ON PC/AT WITH A MATH COPROCESSOR 80287-10

Network Number	No. of Nodes	No. of Links	Minieka Method (sec)	Combined Method (sec)	New Shortcut (sec)	New Method (sec)
1	15	27	3.6	4.5	4.1	3.7
2	17	36	4.7	5.3	4.5	4.0
3	25	50	9.9	9.5	7.5	7.2
4	30	40	13.1	12.4	10.0	9.5
5	40	56	27.7	23.3	17.5	17.0
6	45	95	44.0	32.8	22.6	22.2
7	50	74	51.7	41.0	28.4	28.0
8	65	100	109.5	82.2	53.4	53.0
9	80	141	207.4	148.0	90.4	90.0

shortcut gives a stricter bound than does the Larson and Odoni shortcut. After this new shortcut is applied, the location and its service distance to the local 1-center for the desired link can be obtained directly. The new method finds the absolute 1-center by considering the number of nodes instead of the number of links in a network. Although the combined method has reduced the number of links needed in calculating the local 1-center, the number of remaining links, in most cases, is still greater than the number of nodes considered for the given network. Therefore, after several tests, it can be concluded that the new method and the new shortcut are faster and more powerful than the combined method or the Minieka method, especially for a large network. On this basis, the new method or the new shortcut is recommended for use in locating a new transportation facility if the absolute 1-center location problem is being considered.

ACKNOWLEDGMENT

The authors would like to express their deepest thanks for the assistance of Gwo-Feng Yang during the phases of developing computer programs and preparing this final manuscript.

REFERENCES

1. H. S. Tsay. *The Combined Facility Location and Vehicle Routing Problem: Formulation and Solution.* Ph.D. dissertation. Purdue University, West Lafayette, Ind., Aug. 1985.
2. H. S. Tsay. New Efficient Algorithms for Solving 1-Center Location Problems. *Transportation Planning Journal*, Vol. 15, No. 2, June 1986, pp. 207–220. (Printed in Taiwan)
3. S. L. Hakimi. Optimum Distribution of Switching Centers and the Absolute Centers and Medians of a Graph. *Operations Research*, Vol. 12, 1964, pp. 450–459.
4. S. L. Hakimi. Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems. *Operations Research*, Vol. 13, 1965, pp. 462–475.
5. B. C. Tansel, R. L. Francis, and T. J. Lowe. Location on Networks: Parts I and II. *Management Science*, Vol. 29, 1983, pp. 482–511.
6. J. D. Fricker and H. S. Tsay. A Polynomial Time Algorithm to Solve the General Absolute 1-Center Problem. Presentation at ORSA/TIMS Joint National Meeting, Atlanta, Ga., Oct. 1985.
7. R. C. Larson and A. R. Odoni. *Urban Operations Research.* Prentice-Hall, Englewood Cliffs, N.J., 1981, pp. 439–442.
8. E. Minieka. A Polynomial Time Algorithm for Finding the Absolute Center of a Network. *Networks*, Vol. 11, 1981, pp. 351–355.
9. G. Y. Handler and P. B. Mirchandani. *Location on Networks: Theory and Algorithms.* MIT Press, Cambridge, Mass., 1979.
10. H. S. Tsay and L. T. Lin. A New Method for Finding Absolute 1-Center in a Network. Presented at 1987 TIMS/ORSA Joint National Meeting, New Orleans, La., May 1987.