CSIS Discussion Paper #19

A computational method for optimizing the location of a store on a continuum of a network when users' choice behavior follows the Huff model

Kei-ichi Okunuki and Atsuyuki Okabe

May 31, 1999

The Department of Urban Engineering, School of Engineering,
University of Tokyo7-3-1 Hongo, Bunkyo-ku, Tokyo113-8656
E-mail: nuki@okabe.t.u-tokyo.ac.jp

Abstract

This paper shows a computational method for optimizing the location of one store on a network, assuming that users probabilistically choose stores following the Huff model (Huff, 1963) and that the store can be located on a continuum of a network. This method gives the exact globally optimal solution with the computational order of $n_L^2 \log n_L$ where n_L is the number of links of the network.

Keywords: locational optimization, network, Huff's model, shortest path distance

1 Introduction

The objective of this paper is to develop a computational method for optimizing the location of one store on a network, assuming that users probabilistically choose stores following the Huff model (Huff, 1963) and that the store can be located on a continuum of a network.

In the literature of Operations Research, we can find many papers related to locational optimization on a network, such as the median problem (Hakimi, 1964). In the median problem, it is proved that an optimal location exists at a node of a network (Hakimi, 1963). Thus we can obtain the optimal location by searching all nodes of the network. In the locational optimization with the Huff model, however, the optimal location may be at a point on links of the network (not always at a node). Since the number of locatable points is infinite, we cannot find an optimal location by searching all these points. We should develop an efficient computational method for this optimization. In this paper, we propose this method. First, using the Huff model, we explicitly formulate the demand for a store as a mathematical function, assuming that consumers are distributed over links; the distance between two points on a network is measured by the shortest path distance. Second, using this demand function, we show a method for finding the globally optimal location, at which the store can attain the maximum demand.

2 The Locational Optimization Problem of a Store on a Network

We wish to find the point on a network at which a store maximizes its profit. Under the assumption of a constant marginal cost, maximizing profit is equivalent to capturing maximum demand. In this paper, we develop a method for finding the location at which the store attains the maximum demand.

We consider a network $\mathcal{N}$ in which $n-1$ stores, labelled store 1, ..., store $n-1$, are located. Given these stores, we optimize the location of a new store, store n, on $\mathcal{N}$. Let $P_n(p)$ be the probability that a consumer at an arbitrary point, p, chooses store n at p_n; $w(p)$ be the demand density at p. Then, the amount, $D(p_n)$, of demand for store n is given by

$$D(p_n) = \sum_{p \in \mathcal{N}} P_n(p) w(p). \tag{1}$$

We assume that the probability $P_n(p)$ is given by the Huff model (1963). To be explicit, let a_i be the attractiveness of store i (which may be measured by, for example, the area of the store and the number of items); $d(p, p_i)$ be the shortest path distance between a consumer at p and store i at p_i; $F(d(p, p_i))$ be the distance deterrence function, i.e., F is a monotonously decreasing function with respect to the shortest path distance, $d(p, p_i)$, between p and p_i. Then, the Huff model is written as

$$P_n(p) = \frac{a_n F(d(p, p_n))}{\sum_{i=1}^{n} a_i F(d(p, p_i))}, \tag{2}$$

where the distance deterrence function is usually given by

$$F(d(p, p_i)) = \exp(-\lambda d(p, p_i)), \;\; \lambda > 0, \tag{3}$$

(Wilson, 1970). We call this model *network Huff model* (Miller, 1994, and Okabe & Okunuki, 1997).

Using the network Huff model, we formulate the optimization problem mentioned above. Let $\boldsymbol{x}$ and $\boldsymbol{x}_i$ be the location vectors of p and p_i respectively; L be the set of n_L links of $\mathcal{N}$; X_l be the set of points forming link l on $\mathcal{N}$; X be the set of points forming $\mathcal{N}$ ($X = X_1 \cup X_2 \cup \ldots \cup X_{n_L}$). Using equation (1) with equations (2) and (3) into, we formulate the problem of maximizing the demand of store n as follows.

Locational Optimization Problem of a Store on a Network:

$$\max_{\boldsymbol{x}_n \in X} D(\boldsymbol{x}_n) = \sum_{l \in \mathcal{N}} \int_{\boldsymbol{x} \in X_l} \frac{a_n \exp(-\lambda d(\boldsymbol{x}, \boldsymbol{x}_n))}{\sum_{i=1}^{n} a_i \exp(-\lambda d(\boldsymbol{x}, \boldsymbol{x}_i))} w(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \tag{4}$$

When all of the location vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are fixed, we can obtain the integration in equation (4), as is shown in Okabe & Okunuki (1997). In equation (4), however, $\boldsymbol{x}_n$ is a variable, and hence we have to obtain the function $D(\boldsymbol{x}_n)$ of $\boldsymbol{x}_n$ (not a value of the integration). In the next section, we show how to obtain the function $D(\boldsymbol{x}_n)$.

3 The Integration of the Demand Equation

The integration of equation (4) requires a fairly long derivation, which is divided into the following four steps.

Step 1 (See Figure 1)

Step 1 is to obtain the shortest paths from an arbitrary point p to store 1, …, store $n-1$ on $\mathcal{N}$.

Let N be the set of n_N nodes of network $\mathcal{N}$; N_s be the set of nodes at which $n-1$ stores are located ($N_s \subseteq N$); L be the set of n_L links of $\mathcal{N}$.

First, we construct the shortest path tree, T_i, rooted at each existing store at p_i ($i = 1, \ldots, n-1$) on $\mathcal{N}$. This construction can be efficiently done by the Dijkstra's shortest path tree algorithm (Dijkstra, 1959, Freedman & Tarjan, 1984). Let L_i be the set of links forming T_i. On each link $l_j \in (L \setminus L_i)$ with end nodes p_{j1} and p_{j2}, there always exists one point, q_{ji}, satisfying that the shortest path distance from store i to the point q_{ji} through p_{j1} is equal to that through p_{j2} of l_j. We call such points *collision points* produced by p_i (Okabe & Kitamura, 1996). We obtain collision points for all links in $(L \setminus L_i)$ for all existing stores ($i = 1, \ldots, n-1$).

Having obtained a collision point on link l, we next insert the point as a node and break l into two links at the collision point. This insertion gives a modified network, denoted by $\mathcal{N}^+$. We denote the set of nodes of $\mathcal{N}^+$ by N^+ and the set of links $\mathcal{N}^+$ by L^+, and index the elements in N^+ and L^+ as $N^+ = \{p_1, \ldots, p_{n-1}, p_{n+1}, \ldots, p_{n_{N+}}\}$ and $L^+ = \{l_1, \ldots, l_{n_{L+}}\}$ (note that p_n is not included in N^+ yet).

Step 2

Step 2 is to introduce a parametric representation of the shortest path distance between an arbitrary point p on $\mathcal{N}^+$ and an existing store at p_i (i=1,...,$n-1$) . This representation is useful for integrating equation (4) along links.

The parametric representation is based upon the following nice property (Okabe, Yomono & Kitamura, 1992). The shortest path distance between the point p and p_i monotonically increases or decreases as the point p moves along l_j. It never occurs that the distance increases (decreases) to a certain point and then decreases (increases). Using this property, we can parametrically represent the moving point p on a link l_j with end nodes p_{j1}, p_{j2} by

$$\begin{aligned} d(p, p_i) &= d(p, p_{j1}) + d(p_{j1}, p_i) \\ &= (-1)^{\delta_{ji}} t + c_{ji} \quad \text{for } 0 \le t \le l_j, \end{aligned} \tag{5}$$

where l_j indicates the length of link l_j, $c_{ji}(= d(p_{j1}, p_i))$ is a constant with respect to t, and δ_{ji} is given by

$$\delta_{ji} = \begin{cases} 0 & \text{for } d(p_{j1}, p_i) < d(p_{j2}, p_i), \\ 1 & \text{for } d(p_{j1}, p_i) > d(p_{j2}, p_i). \end{cases} \tag{6}$$

Step 3

Step 3 is to introduce the same parametric representation for the new store n at p_n.

Suppose that the new store n is located at a point, p_n, on link l_a whose end nodes are p_{a1} and p_{a2}, and that the point p_n is located at distance u from p_{a1} along l_a, i.e.,

$$d(p_n, p_{a1}) = u. \tag{7}$$

First we construct the shortest path tree rooted at p_n, and obtain the collision points. Next, we classify all links in L^+ into four groups in relation to these collision points and p_n. (I) links on which p_n exists but any collision point produced by p_n does not exist; (II) links on which a collision point produced by p_n exists but p_n does not exist; (III) links on which p_n and a collision point produced by p_n exist; (IV) links on which neither p_n nor any collision point produced by p_n exists. We denote these sets of links by L_{I}^{+}, L_{II}^{+}, L_{III}^{+} and L_{IV}^{+}, respectively.

For each set, we can parametrically represent a moving point p on a link l_j.

Case I: p on link $l_a \in L_{\mathrm{I}}^{+}$ (Figure 2).

On l_a there exist p_n but no collision point produced by p_n. The moving point p along link l_a is written as

$$\begin{aligned} d(p, p_n) &= |d(p, p_{j1}) - d(p_n, p_{j1})| \\ &= \begin{cases} -d(p, p_{a1}) + d(p_n, p_{a1}) & \text{for } d(p, p_{a1}) \leq d(p_n, p_{a1}), \\ d(p, p_{a1}) - d(p_n, p_{a1}) & \text{for } d(p, p_{a1}) \geq d(p_n, p_{a1}) \end{cases} \\ &= \begin{cases} -t + u & \text{for } 0 \leq t \leq u, \\ t - u & \text{for } u \leq t \leq l_j. \end{cases} \end{aligned} \tag{8}$$

Case II: p on link $l_j \in L_{\mathrm{II}}^{+}$ (Figure 3).

On l_j, there exists a collision point q_{jn} produced by p_n on l_a. In this case as shown in Figure 3, we have two shortest paths between p_n and q_{jn}, and these paths form a loop, denoted by C. When the configuration of p_{a1}, p_{a2}, p_{j1} and p_{j2} is shown in Figure 3(a), the length, l_c, of the loop C is given by

$$l_c = l_j + l_a + d(p_{j1}, p_{a1}) + d(p_{j2}, p_{a2}), \tag{9}$$

where l_j and l_a indicate the length of links, l_j and l_a, respectively. Obviously, the shortest path distance $d(q_{jn}, p_n)$ is always equal to $\frac{l_c}{2}$, which is given by

$$\begin{aligned} d(q_{jn}, p_n) &= \frac{l_c}{2} \\ &= \frac{1}{2}(l_j + l_a + d(p_{j1}, p_{a1}) + d(p_{j2}, p_{a2})). \end{aligned} \tag{10}$$

The shortest path distance $d(q_{jn}, p_{j1})$ is given by

$$\begin{aligned} d(q_{jn}, p_{j1}) &= d(q_{jn}, p_n) - d(p_{j1}, p_{a1}) - d(p_n, p_{a1}) \\ &= \frac{1}{2}(l_j + l_a - d(p_{j1}, p_{a1}) + d(p_{j2}, p_{a2})) - u. \end{aligned} \tag{11}$$

Thus the moving point p is written as

$$\begin{aligned} d(p, p_n) &= \begin{cases} d(p, p_{j1}) + d(p_{j1}, p_{a1}) + d(p_n, p_{a1}) \\ \quad \text{for } 0 \le d(p, p_{j1}) \le d(q_{jn}, p_{j1}); \\ l_j + l_a + d(p_{j2}, p_{a2}) - d(p, p_{j1}) - d(p_n, p_{a1}) \\ \quad \text{for } d(q_{jn}, p_{j1}) \le d(p, p_{j1}) \le l_j \end{cases} \\ &= \begin{cases} t + u + d(p_{j1}, p_{a1}) \\ \quad \text{for } 0 \le t \le -u + \frac{1}{2}(l_j + l_a + d(p_{j2}, p_{a2}) - d(p_{j1}, p_{a1})); \\ -t - u + l_a + l_k + d(p_{j2}, p_{a2}) \\ \quad \text{for } -u + \frac{1}{2}(l_j + l_a + d(p_{j2}, p_{a2}) - d(p_{j1}, p_{a1})) \le t \le l_j. \end{cases} \end{aligned} \tag{12}$$

In the same way, when the configuration of p_{a1}, p_{a2}, p_{j1} and p_{j2} is given by Figure 3(b), the moving point p is written as

$$d(p, p_n) = \begin{cases} t - u + l_a + d(p_{j1}, p_{a2}) \\ \quad \text{for } 0 \le t \le u + \frac{1}{2}(l_j - l_a + d(p_{j2}, p_{a1}) - d(p_{j1}, p_{a2})); \\ -t + u + l_j + d(p_{j2}, p_{a1}) \\ \quad \text{for } u + \frac{1}{2}(l_j - l_a + d(p_{j2}, p_{a1}) - d(p_{j1}, p_{a2})) \le t \le l_j. \end{cases} \tag{13}$$

Case III: p on link $l_a \in L_{\mathrm{III}}^+$ (Figure 4).

In this case, as in Figure 4, we have two shortest paths between p_n and q_{an}, and the distance of the shortest path is shorter than the length of the link l_a, i.e.,

$$d(p_{a1}, p_{a2}) < l_a. \tag{14}$$

The two shortest paths form a loop C' and its length l_c is given by

$$l_c = d(p_{a1}, p_{a2}) + l_a. \tag{15}$$

Since the shortest path distance $d(q_{an}, p_n)$ is equal to $\frac{l_c}{2}$, $d(q_{an}, p_n)$ is given by

$$\begin{aligned} d(q_{an}, p_n) &= \frac{l_c}{2} \\ &= \frac{1}{2}(d(p_{a1}, p_{a2}) + l_a). \end{aligned} \tag{16}$$

When the configuration of p_{a1}, p_n, q_{na} and p_{a2} is given by Figure 4(a), the shortest path distance $d(q_{an}, p_{a1})$ is given by

$$\begin{aligned} d(q_{an}, p_{a1}) &= d(p_n, p_{a1}) + d(q_{an}, p_n) \\ &= u + \frac{1}{2}(d(p_{a1}, p_{a2}) + l_a). \end{aligned} \tag{17}$$

Thus the moving point p is written as

$$\begin{aligned} d(p, p_n) &= \begin{cases} -d(p, p_{a1}) + d(p_n, p_{a1}) \\ \quad \text{for } 0 \le d(p, p_n) \le d(p_n, p_{a1}); \\ d(p, p_{a1}) - d(p_n, p_{a1}) \\ \quad \text{for } d(p_n, p_{a1}) \le d(p, p_n) \le d(q_{an}, p_{a1}); \\ -d(p, p_{a1}) + d(p_n, p_{a1}) + l_a + d(p_{a1}, p_{a2}) \\ \quad \text{for } d(q_{an}, p_{a1}) \le d(p, p_n) \le l_a \end{cases} \\ &= \begin{cases} -t + u & \text{for } 0 \le t \le u; \\ t - u & \text{for } u \le t \le u + \frac{1}{2}(l_a + d(p_{a1}, p_{a2}); \\ -t + u + l_a + d(p_{a1}, p_{a2}) & \text{for } u + \frac{1}{2}(l_a + d(p_{a1}, p_{a2}) \le t \le l_a. \end{cases} \end{aligned} \tag{18}$$

In the same way, when the configuration of p_{a1}, q_{na}, p_n and p_{a2} is given by Figure 4(b), the shortest path distance $d(q_{an}, p_{a1})$ is given by

$$d(q_{an}, p_{a1}) = u - \frac{1}{2}(d(p_{a1}, p_{a2}) + l_a). \tag{19}$$

Thus the moving point p is written as

$$d(p, p_n) = \begin{cases} t - u + l_a + d(p_{a1}, p_{a2}) & \text{for } 0 \le t \le u - \frac{1}{2}(l_a + d(p_{a1}, p_{a2}); \\ -t + u & \text{for } u - \frac{1}{2}(l_a + d(p_{a1}, p_{a2}) \le t \le u; \\ t - u & \text{for } u \le t \le l_a. \end{cases} \tag{20}$$

Case IV: p on link $l_j \in L^{+}_{\mathrm{IV}}$ (Figure 5).

When the shortest path from p_n to p_{j2} goes through p_{a1} and p_{j1} (Figure 5(a)), we can write the moving point p on l_j as

$$\begin{aligned} d(p, p_n) &= d(p, p_{j1}) + d(p_{j1}, p_{a1}) + d(p_n, p_{a1}) \\ &= t + u + d(p_{j1}, p_{a1}) \ \text{ for } \ 0 \leq t \leq l_a. \end{aligned} \tag{21}$$

When the shortest path from p_n to p_{j2} goes through p_{a2} and p_{j1} (Figure 5(b)), we can write the moving point p on l_j as

$$\begin{aligned} d(p, p_n) &= d(p, p_{j1}) + d(p_{j1}, p_{a2}) + d(p_n, p_{a2}) \\ &= t - u + l_a + d(p_{j1}, p_{a2}) \ \text{ for } \ 0 \leq t \leq l_a. \end{aligned} \tag{22}$$

When the shortest path from p_n to p_{j1} goes through p_{a1} and p_{j2} (Figure 5(c)), we can write the moving point p on l_j by

$$\begin{aligned} d(p, p_n) &= d(p, p_{j2}) + d(p_{j2}, p_{a1}) + d(p_n, p_{a1}) \\ &= -t + u + l_j + d(p_{j2}, p_{a1}) \ \text{ for } \ 0 \leq t \leq l_a. \end{aligned} \tag{23}$$

When the shortest path from p_n to p_{j1} goes through p_{a2} and p_{j2} (Figure 5(d)), we can write the moving point p on l_j by

$$\begin{aligned} d(p, p_n) &= d(p, p_{j2}) + d(p_{j2}, p_{a2}) + d(p_n, p_{a2}) \\ &= -t - u + l_a + l_j + d(p_{j2}, p_{a2}) \ \text{ for } \ 0 \leq t \leq l_a. \end{aligned} \tag{24}$$

In the above we have explicitly obtained the function $d(p, p_n)$ of t for Cases I, II, III and IV. These functions vary from case to case, but formally we can write them in the same form as

$$d(p, p_n) = (-1)^{\delta_{jnk}} t + c_{jnk}(u) \ \text{ for } \ l_{jk1}(u) \leq t \leq l_{jk2}(u). \tag{25}$$

Note that $c(u)$ and $l(u)$ are all linear funcitions of u.

Step 4:

Step 4 is to integrate equation (4) on each link l_j on L^+.

We assume that $l_j \in L^+$ is divided into n_j links (l_{jk} for $k = 1, \ldots, n_j$) at a store n at p_n and at the collision point produced by p_n. Substituting equation (5) and (25) into equation (4), we obtain the demand, $D_n(u)$, for store n as

$$D_n(u) = \sum_{l_j \in L^+} \sum_{k=1}^{n_j} D_{njk}(u), \tag{26}$$

where

$$D_{njk}(u) = \int_{l_{jk1}(u)}^{l_{jk2}(u)} \frac{A(u,t)}{A(u,t) + B(u)} w(t) \mathrm{d}t, \tag{27}$$

$$A(u,t) = a_n \exp(-\lambda(c_{jnk}(u) + (-1)^{\delta_{jnk}} t)), \tag{28}$$

$$B(t) = \sum_{i=1}^{n-1} a_i \exp(-\lambda(c_{ji} + (-1)^{\delta_{ji}} t)). \tag{29}$$

Let $I_{j1} = \{i \mid \delta_{ji} = \delta_{jnk}\}$ and $I_{j2} = \{i \mid \delta_{ji} \neq \delta_{jnk}\}$. Then, equation (27) is written as

$$D_{njk}(u) = \int_{l_{jk1}(u)}^{l_{jk2}(u)} \frac{a_n \exp(-\lambda c_{jn}(u))}{K(t)} w(t) \mathrm{d}t, \tag{30}$$

where

$$K(t) = \sum_{i \in I_{j1}} a_i \exp(-\lambda c_{ji}) + \exp(2\lambda(-1)^{\delta_{jnk}} t) \sum_{i \in I_{j2}} a_i \exp(-\lambda c_{ji}). \tag{31}$$

We assume that the demand density $w(t)$ is constant on each link l_j (the density varies from link to link), i.e.,

$$w(t) = w_j \text{ for } 0 \leq t \leq l_j,\ j = 1, \ldots, n_{L+} \tag{32}$$

Under this assumption, we can carry out the integration in equation (30) and obtain

$$\begin{aligned} D_{njk}(u) \quad = \quad & \frac{\left(2\lambda(-1)^{\delta_{jnk}} l_{jk2}(u) + \log\left(\frac{a_n \exp(-\lambda c_{jnk}(u)) + \sum_{i=1}^{n-1} a_i \exp(-\lambda c_{ji})}{K(l_{jk2}(u))}\right)\right) w_j}{2\lambda(-1)^{\delta_{jnk}} \left(1 + \sum_{i \in I_{j1}} \frac{a_i}{a_n} \exp(-\lambda(c_{ji} - c_{jnk}(u))\right)} \\ & - \frac{\left(2\lambda(-1)^{\delta_{jnk}} l_{jk1}(u) + \log\left(\frac{a_n \exp(-\lambda c_{jnk}(u)) + \sum_{i=1}^{n-1} a_i \exp(-\lambda c_{ji})}{K(l_{jk1}(u))}\right)\right) w_j}{2\lambda(-1)^{\delta_{jnk}} \left(1 + \sum_{i \in I_{j1}} \frac{a_i}{a_n} \exp(-\lambda(c_{ji} - c_{jnk}(u))\right)} \end{aligned} \tag{33}$$

We can exactly obtain the demand function of equation (4) as a function of u shown in equation (33).

4 Computational Method for the Optimization

When p_n and all of the collision points produced by p_n are inserted on $\mathcal{N}^+$, we obtain a new network, denoted by $\mathcal{N}^{++}$ (Figure 6). The topological relations between nodes of this network $\mathcal{N}^{++}$ depends on the location of p_n. The topology of this network $\mathcal{N}^{++}$ holds during p_n moves within a certain length along links in L^+, but changes when either p_n or a collision point produced by p_n passes over a node in N^+ (Figure 7).

To obtain the length, within which p_n moves along links in L^+ so that the topological relations holds, we construct the shortest path trees rooted at all nodes in N^+ on the network $\mathcal{N}^+$, and insert all of the collision points produced by all nodes on links in L^+ as nodes. As a result, we obtain a new network $\mathcal{N}^+_{div}$ (Figure 8). Let L^+_{div} be the set of n^{L+}_{div} links of $\mathcal{N}^+_{div}$. When p_n moves within each link $l_h \in L^+_{div}$, neither p_n nor any collision point produced by p_n passes over a node in N^+. Then, the topological relations holds and we can obtain the amount, $D_n(u)$, of store n by equation (26) and (33) in the previous section. Let X_{lh} be the set of location parameter u on the link l_h. The locational optimization problem of equation (4) is written as

$$\max_{l_h \in L^+_{div}} \max_{u \in X_{lh}} D_n(u) = \sum_{l_j \in L^+} \sum_{k=1}^{n_j} D_{njk}(u), \tag{34}$$

where $D_{njk}(u)$ is given by equation (33).

$D_{njk}(u)$ is a continuous and differentiable function with respect to $u \in X_{lh}$. We can obtain the local optimal solution u^*_h by a nonlinear programming method, such as the descent method (Gill *et al.*, 1981). In all of u_h^* for $1, \ldots, n_{div}^{L+}$, we can find the globally optimal solution, that is, the optimal location for store n.

Now, we consider the order of the computational procedure for obtaining the globally optimal solution (for details, see Okabe & Okunuki, 1997).

First, to obtain the network $\mathcal{N}^+$ from $\mathcal{N}$, we construct $n-1$ shortest path trees (in step 1 in the preceding section). The order of this construction is $O(n_N \log n_N)$ under the assumption that the order of the number of stores is less than that of nodes.

Second, to obtain the network $\mathcal{N}_{div}^+$ from $\mathcal{N}^+$, we construct the shortest path trees rooted at all nodes of $\mathcal{N}^+$. The order of the number of nodes of $\mathcal{N}^+$ is $O(n_L)$, and the order of the construction is $O(n_L^2 \log n_L)$. The order of links of $\mathcal{N}_{div}^+$ is $O(n_L^2)$.

Third, to obtain the globally optimal solution, we optimize for every links of $\mathcal{N}_{div}^+$. The order is $O(n_L^2)$.

Summing up, we notice that the order of the total computational time is $O(n_L^2 \log n_L)$.

Last we show two examples. The results are shown in Figure 9 and 10. Figure 9(b) is a very simple case in which one store (store 1) is located at an end of a link and the location of one new store (store 2) is optimized. The second example (Figure 10) deals with a little more complicated network on which one store (store 1) is located at a node and the location of one new store (store 2) is optimized.

5 Conclusion

This paper shows a computational method for finding the globally optimal location at which the store attains the maximum demand when users' choice behavior follow the Huff model and the store is locatable at any point on a continuum of a network. The solution is exact and the computational tool is tractable. Thus our method is useful for locational decision in practice.

References

Dijksrtra, E. (1959) A note on two problems in connexion with graphs, *Numerishe Mathematik*, **1**, 269-271.

Freedman & Tarjan (1984) Fibonacci heaps and their uses in improved network optimization algorithms, *Proceedings of the 25th Annual IEEE Symposium on Foundations of Computer Science*, Florida: Singer Island, 338-346

Gill, P.E., Murry, W. & Wright, M.H. (1981) *Practical Optimization*, London: Academic Press.

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

Hakimi, S.L. (1986) p-median theorems for competitive locations, *Annals of Operations Research*, **6**, 77-98.

Huff, D.L. (1963) A probabilistic analysis of shopping center trade area, *Land Economics*, **39**, 81-90

Miller, H. (1994) Market area delineation within networks using geographic information systems, *Geographical Systems*, **1**, 157-175

Okabe, A., Yomono, H. & Kitamura, M. (1995) Statistical analysis of the distribution of points on a network, *Geographical Analysis*, **27**, 152-175

Okabe, A. & Kitamura, M. (1996) A computational method for market area analysis on a network, *Geographical Analysis*, **28**, 330-349

Okabe, A. & Okunuki,K. (1997) A Computational Method for Estimating the Demand of Retail Stores on a Street Network Using GIS, *Discussion Paper* , **71**, Department of Urban Engineering, University of Tokyo

Wilson, A.G. (1970) *Entropy in Urban and Regional Modelling*, London: Pion

Acknowledgments

We thank to Hidehiko Yomono for his valuable comments.

Figure captions

Figure 1: An illustrative example of a network $\mathcal{N}$ with the shortest path tree and collision points, and the modified network $\mathcal{N}^+$

(a) a network in which nodes are indicated by the black circles and the location of an existing store, store 1, is indicated by the arrow

(b) the shortest path tree rooted at store 1 (the continuous lines) and collision points (the white circles) (the lengths of two broken lines with arrows starting from store 1 are the same)

(c) the modified network $\mathcal{N}^+$ obtained by inserting collision points in the links of the network $\mathcal{N}$.

Figure 2: The link l_a with end nodes p_{a1} and p_{a2} (the black circles) on which p_n (the white circle) exists, and an arbitrary point p (the small black circle) moving along the link l_a

Figure 3: The link l_j with end nodes p_{j1} and p_{j2} (the black circles) on which a collision point, q_{jn} (the white circle), produced by p_n on l_a exist (the dotted lines indicating the shortest path tree rooted at p_n), and an arbitrary point p (the small black circle) moving along link l_j

Figure 4: The link l_a on which p_n and a collision point, q_{an}, produced by p_n (the white circles) exist (the dotted lines indicating the shortest path tree rooted at p_n), and an arbitrary point p (the small black circle) moving along link l_j

Figure 5: The link l_j on which neither p_n nor any collision point produced by p_n exists, and an arbitrary point p (the small black circle) moving along link l_j

Figure 6: An illustrative example of a network $\mathcal{N}^{++}$

(a) a network $\mathcal{N}^+$ (Figure 1(c)), on which a new store, store 2 (the white circle), is located, and the collision points (the white squares) produced by store 2

(b) the modified network $\mathcal{N}^{++}$ obtained by inserting store 2 and collision points

Figure 7: An illustrative example of changes in the topological relations between nodes of a network $\mathcal{N}^{++}$ ((b), (c) and (d)) as store 2 moves along the dotted line in this Figure (a)

Figure 8: The modified network $\mathcal{N}^{+}_{div}$ obtained inserting all collision points (the white circles) produced by all nodes of $\mathcal{N}^{+}$

Figure 9: An example of the locational optimization on a link, where the demand density is a constant

(a) a network with two nodes (the black circles) on which store 1 exists

(b) the optimal location of a new store, store 2 (the white circle)

Figure 10: An example of the locational optimization on a network, where the demand density is a constant

(a) a network with three nodes (the black circles) on which store 1 exists

(b) the optimal location of a new store, store 2 (the white circle)

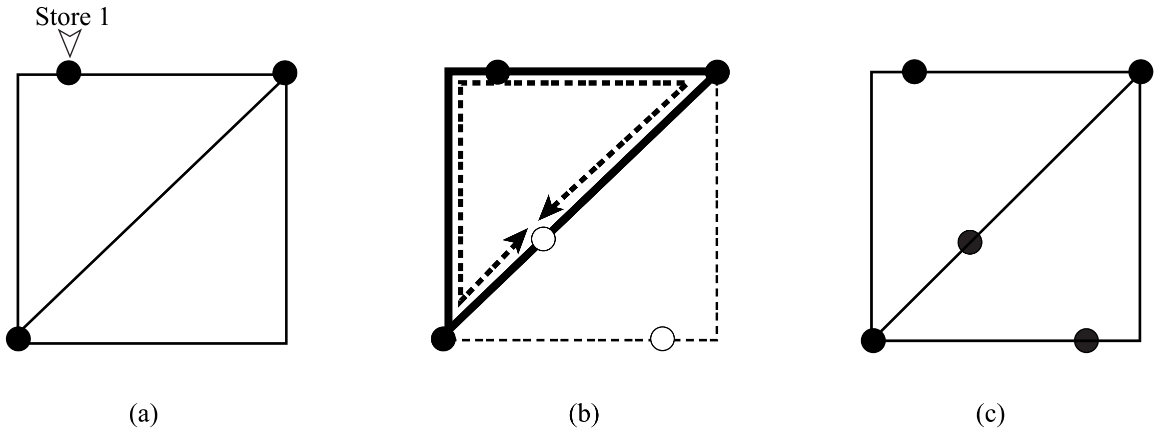

Figure 1

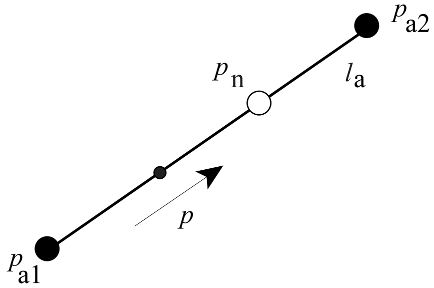

Figure 2

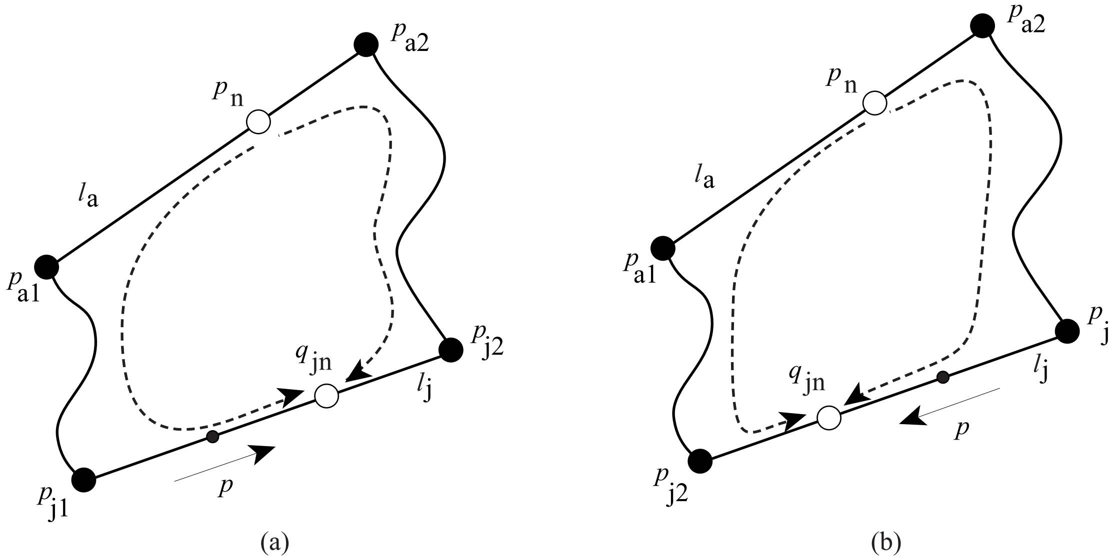

Figure 3

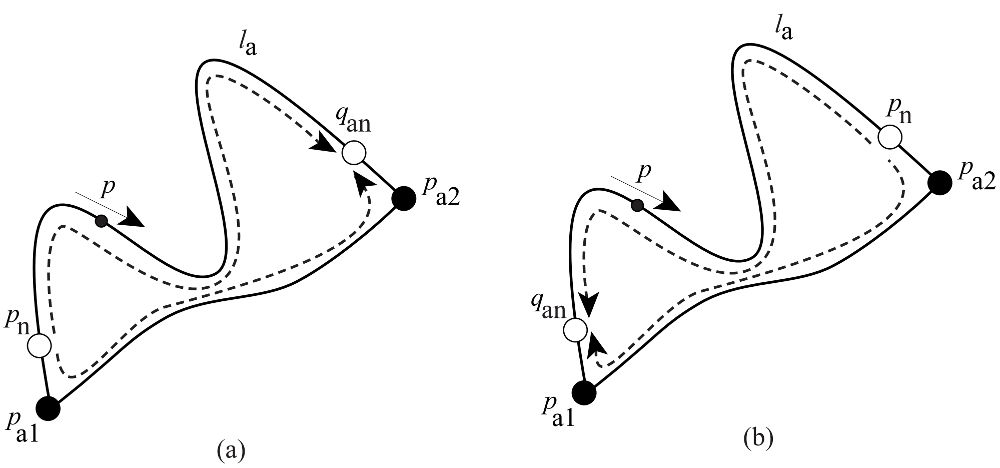

Figure 4

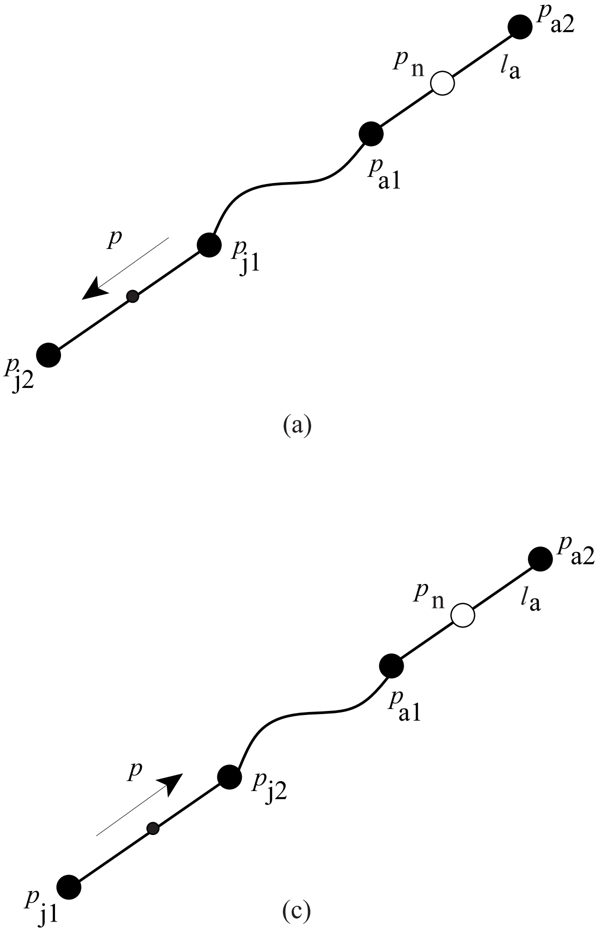

Figure 5

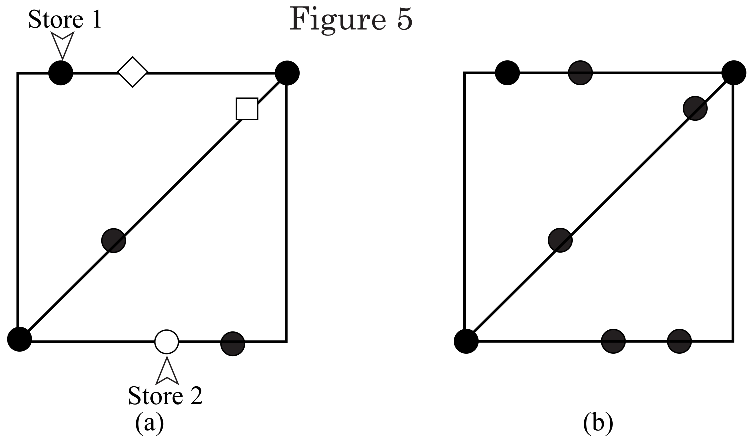

Figure 6

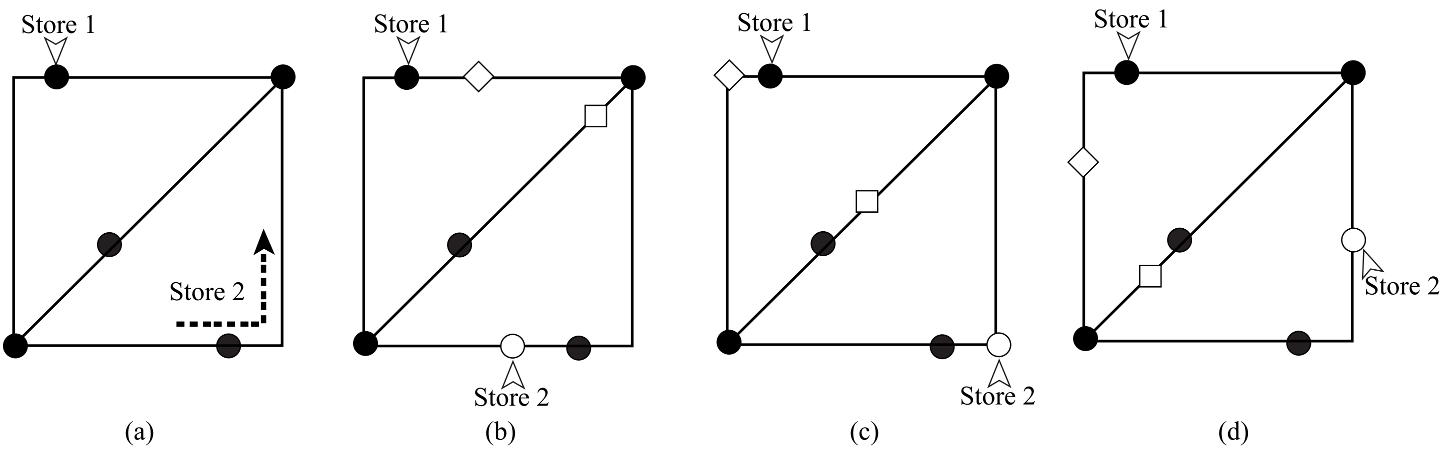

Figure 7

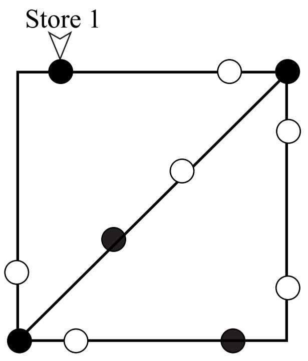

Figure 8

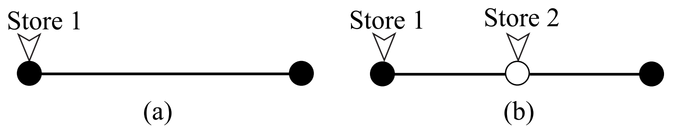

Figure 9

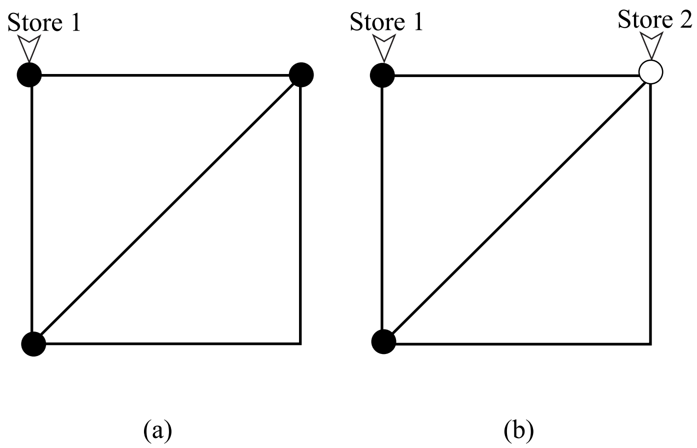

Figure 10