*Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*
*Edition by ATSUYUKI OKABE & BARRY BOOTS and SUNG NOK CHIU*


CHAPTER 9

# Locational Optimization Through Voronoi Diagrams

Voronoi diagrams are useful not only for the spatial analyses shown in the preceding chapters, but also for spatial optimization. In this chapter we discuss four classes of locational optimization problems which can be solved with the aid of Voronoi diagrams. The chapter consists of five sections. Section 9.1 is a preliminary section in which we briefly show a few non-linear non-convex optimization methods to be used in this chapter. The reader who is familiar with these methods can skip this preliminary section.

In Section 9.2 we deal with the locational optimization of points. In general, the problem is to determine the location of points so that the average distance (more generally, the average cost) to the nearest points is minimized. The problem varies slightly according to the nature of the points. We consider six kinds of points: point-like facilities used by independent individuals (e.g. public mail boxes); point-like facilities used by groups (e.g. tennis courts); a hierarchical facility (e.g. the main library and branch libraries); observation points for estimating some spatial quantities (e.g. pluviometers); service points of a mobile facility (e.g. a bookmobile) and nodes through which users go to a central point (e.g. bus stops for commuters). We show that these locational optimization problems can be solved with the ordinary and weighted Voronoi diagrams.

In Section 9.3 we discuss the locational optimization of lines. We deal with three types of lines. The first type of lines is a service route of a mobile facility (e.g. an ice-cream vendor). The optimization problem is to determine a service route to minimize the average distance (or cost) to the nearest point on a service route provided that the total length of the service route is given. The second type of lines is a network (e.g. a railway network). The optimization problem is to determine the location of nodes (stations) and links (railways) to minimize the total flow cost (or transportation cost) between any two points on a plane provided that the total length of a network and the flow density between any two points in a region are given. The third type of lines is the Steiner tree. The problem is to approximately obtain a big Steiner tree which cannot be obtained from combinatorial methods. We show that the locational optimization problems of these types of lines can be solved with the help of the ordinary Voronoi diagram, the line Voronoi diagram and the Delaunay triangulation.

In Section 9.4 we consider two types of space–time optimization problem. The first type of problem is to optimize the location of point-like facilities that are constructed over multiple time-stages. The optimization is to minimize the distance (or cost) to the nearest facility averaged over users in a region as well as over time. The second type of problem is to determine the location of facilities which open periodically (e.g. marketplaces open every Sunday) so that the average distance (or cost) to the nearest facilities and the waiting time is minimized. We show that these two types of problems can be solved with the aid of the space–time Voronoi diagram and the Voronoi diagram on a cylinder.

In Section 9.5 we show a method for fitting a Voronoi diagram to a given polygonal tessellation. With this method we can see to what extent a Voronoi diagram fits a given tessellation. Moreover, we can apply this method to the locational optimization problem of point-like facilities whose use is spatially restricted.

## 9.1 PRELIMINARIES

In solving the locational optimization problems mentioned above, we meet a certain type of mathematical optimization problem, called the non-linear, non-convex programming problem. In this section we briefly present a few computationally feasible methods to solve this problem. The reader who wants to know the full details of non-linear programming methods should consult, for example, Fiacco and McCormick (1968), Gill *et al.* (1981), and Konno and Yamashita (1978).

### 9.1.1 The non-linear, non-convex programming problem

The *programming problem* referred to in the mathematical optimization literature is the problem in which we seek to determine $n$ variables $x_1, \ldots, x_n$ which satisfy $m_1$ inequalities and/or $m_2$ equations, and, in addition, minimize a function of the $n$ variables. Mathematically, the programming problem is written as

**Problem OPT1 (the general programming problem)**

$$\min_{x_1,\ldots,x_n} F(x_1,\ldots,x_n), \tag{9.1.1}$$

subject to

$$\begin{aligned} &g_i(x_1,\ldots,x_n) \leq 0, \quad i = 1,\ldots,m_1, \\ &g_i(x_1,\ldots,x_n) = 0, \quad i = m_1+1,\ldots,m_1+m_2. \end{aligned} \tag{9.1.2}$$

We call the function in expression (9.1.1) an *objective function*. In this chapter we assume that the objective function has at least the first derivative and it is bounded below, i.e. $F(\boldsymbol{x}) > K > -\infty$. We call relations (9.1.2) *constraints* and the functions in the constraints *constraint functions*. The programming problems with constraints ($m_1 + m_2 \geq 1$) and without constraints ($m_1 + m_2 = 0$) are called the *constrained programming problem* and the *unconstrained programming problem*, respectively. For simplicity, $F(x_1, \ldots, x_n)$ and $g_i(x_1, \ldots, x_n)$ are written as $F(\boldsymbol{x})$ and $g_i(\boldsymbol{x})$, where $\boldsymbol{x}^{\mathrm{T}} = (x_1, \ldots, x_n)$ is a vector in the $n$-dimensional Euclidean space $\mathbb{R}^n$. Note that since the maximization of $F(\boldsymbol{x})$ is equivalent to the minimization of $-F(\boldsymbol{x})$, we consider only the minimization problem; since $g_i(\boldsymbol{x}) \geq 0$ is equivalent to $-g_i(\boldsymbol{x}) \leq 0$, we consider only the two relations $\leq$ and $=$ in the constraints of relation (9.1.2). In Figure 9.1.1 a simple example is depicted, where the objective function is a univariate function, $F(x) = x(x-1)(x-2)(x-4)(x-8)(x-10) + 3000$, and the constraints are $g_1(x) = -x \leq 0$ and $g_2(x) = x \leq 10$ ($m_1 = 2, m_2 = 0$).

The constraints in relation (9.1.2) determine a region in $\mathbb{R}^n$, which is given by

$$S = \{\boldsymbol{x} \mid g_i(\boldsymbol{x}) \leq 0, i = 1, \ldots, m_1; g_i(\boldsymbol{x}) = 0, i = m_1+1, \ldots, m_1+m_2\}. \quad (9.1.3)$$

We call the set $S$ a *feasible region*, and $\boldsymbol{x} \in S$ a *feasible solution*. In the above example the feasible region is given by $S = \{x \mid 0 \leq x \leq 10\}$, which is indicated by the heavy solid line in Figure 9.1.1. In this feasible region we notice three basins around the points indicated by $x^*, x'^*$ and $x^{**}$. We call the bottom points in these basins *local minima*. To be precise, let $N_\varepsilon(\boldsymbol{x}^*) = \{\boldsymbol{x} \mid \|\boldsymbol{x} - \boldsymbol{x}^*\| < \varepsilon\}$ be an open ball centred at $\boldsymbol{x}^*$ with radius $\varepsilon$. If there exist a positive $\varepsilon$ and $\boldsymbol{x}^* \in S$ for which the relation $F(\boldsymbol{x}^*) \leq F(\boldsymbol{x})$ holds for $\boldsymbol{x} \in S \cap N_\varepsilon(\boldsymbol{x}^*)$, we call $\boldsymbol{x}^*$ a *local minimum (optimum)*. In the example of Figure 9.1.1, local minima are $x^*, x'^*$ and $x^{**}$. Among them, the bottom

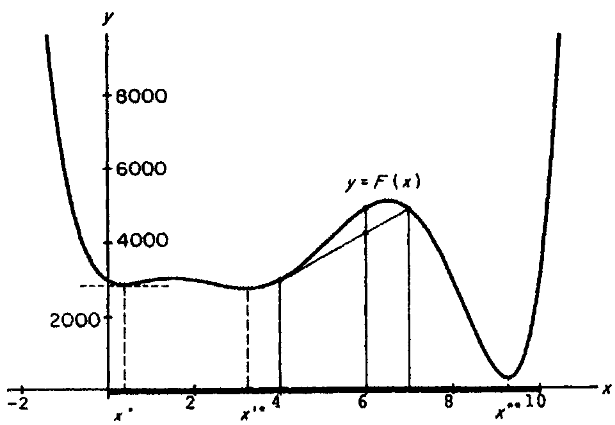



**Figure 9.1.1** A feasible region, local minima, and the global minimum $F(x) = x(x-1)(x-2)(x-4)(x-8)(x-10) + 3000, g_1(x) = -x \leq 0, g_2(x) = x \leq 10$.

at $x^{**}$ is the lowest. We call such a bottom point the *global minimum*. Mathematically, if there exists $\boldsymbol{x}^{**} \in S$ that satisfies $F(\boldsymbol{x}^{**}) \leq F(\boldsymbol{x})$ for $\boldsymbol{x} \in S$, we call $\boldsymbol{x}^{**}$ the *global minimum (optimum)*.

Since the functions $F(\boldsymbol{x})$ and $g_i(\boldsymbol{x})$ are general functions, programming problems subsumed under Problem OPT1 are various, but we can classify them into two classes: the linear programming problem and the non-linear programming problem. The *linear programming problem* is the programming problem in which each of the functions $F(\boldsymbol{x})$ and $g_i(\boldsymbol{x})$, $i \in I_{m_1+m_2}$, is a linear function, i.e. $F(x_1, \ldots, x_n) = c_1x_1 + \ldots + c_nx_n$, and $g_i(x_1, \ldots, x_n) = a_{i1}x_1 + \ldots + a_{in}x_n + b_i$. The *non-linear programming problem* is the programming problem in which either $F(\boldsymbol{x})$, or at least one of the functions $g_i(\boldsymbol{x})$, $i \in I_{m_1+m_2}$, is a non-linear function, such as $F(x_1, \ldots, x_n) = c_1x_1^2 + \ldots + c_nx_n^2$.

We can further classify the non-linear programming problem into two subclasses: the non-linear convex programming problem, and the non-linear, non-convex programming problem. If for any two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in $\mathbb{R}^n$ and for all $0 \leq \lambda \leq 1$, the following relation holds

$$F[\lambda\boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2] \leq \lambda F(\boldsymbol{x}_1) + (1-\lambda)F(\boldsymbol{x}_2), \tag{9.1.4}$$

we call the function $F(\boldsymbol{x})$ a *convex function*; otherwise, we call it a *non-convex function* (an alternative definition is given in Section 1.3.1). The function shown in Figure 9.1.1 is a non-convex function, because the relation (9.1.4) does not hold, for example, $F(0.5 \times 4 + 0.5 \times 7) > 0.5\,F(4) + 0.5\,F(7)$ (observe the two filled circles in Figure 9.1.1). If every non-linear function in the non-linear programming problem is convex, we call such a programming problem the *non-linear convex programming problem*; if at least one non-linear function in the non-linear programming problem is non-convex, we call it the *non-linear, non-convex programming problem*. As will be seen below, the programming problems encountered in the proceeding sections are the non-linear, non-convex programming problems.

### 9.1.2 The descent method

Let us first consider the unconstrained case of Problem OPT1, i.e.

**Problem OPT 2 (the unconstrained programming problem)**

$$\min_{x_1, \ldots, x_n} F(x_1, \ldots, x_n). \tag{9.1.5}$$

As is indicated by the horizontal broken line in Figure 9.1.1, the necessary condition for a local minimum of a univariate function is that the line tangential to the curve given by $y = F(x)$ at $x^*$ is horizontal. Algebraically, this necessary condition is written as $\mathrm{d}F(x)/\mathrm{d}x = 0$. Similarly, the necessary condition of a multivariate function $F(x_1, \ldots, x_n)$ for a local minimum is that the hyperplane tangential to the surface given by $y = F(\boldsymbol{x})$ at $\boldsymbol{x}^*$ is horizontal. Algebraically, this condition is written as

$$\begin{aligned} &\frac{\partial F(\boldsymbol{x}^*)}{\partial x_1} = 0, \\ &\quad\vdots \\ &\frac{\partial F(\boldsymbol{x}^*)}{\partial x_n} = 0. \end{aligned} \tag{9.1.6}$$

When the function $F(\boldsymbol{x})$ is a non-linear, non-convex function, we usually encounter analytical difficulty in solving the simultaneous equations of equation (9.1.6). It is almost impossible to obtain not only the global minimum but also a local minimum with the analytical method, or the so-called classical method. To overcome this difficulty, we may alternatively use a numerical method which gives a solution close to a local minimum with a certain precision level.

The difficulty in solving the non-linear programming problem may be intuitively understood by imagining the difficult situation in which we are lost in a rolling mountainous region (Figure 9.1.2(a)) covered with mist, and we are looking for the bottom of a basin in the region without a contour map (Figure 9.1.2(b)). Since we are surrounded by mist, we can see only the small area around us ($\boldsymbol{x}^{(k)}$ in Figure 9.1.2(b)). What we can do therefore is to walk down observing the slope in this small area. First, we decide in which direction ($\boldsymbol{d}^{(k)}$ in Figure 9.1.2(b)) we should walk down (Step 1). Once the direction is determined (the arrow in Figure 9.1.2(b)), we traverse straight in this direction (the straight line in Figure 9.1.2). Second, we decide at which point we should stop on this line (Step 2); third, when we reach that point ($\boldsymbol{x}^{(k+1)}$ in Figure 9.1.2(b)), we reconsider the direction and repeat the same procedure from that point (Step 3); finally, if we reach a flat place, we stop

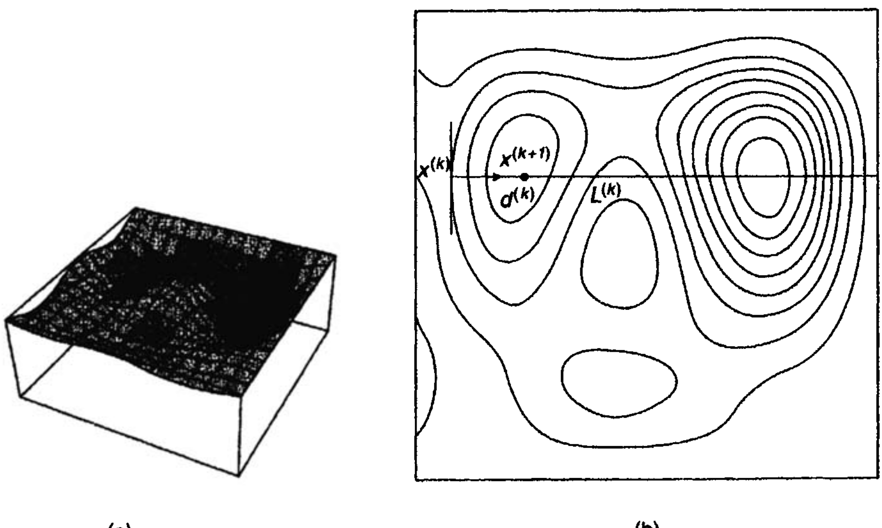



**Figure 9.1.2** The descent method: (a) the surface of a function $F(x_1, x_2)$ and (b) its contour lines.

at that point (Step 4). We consider that the reached point is a bottom point (a local minimum). Since we do not know how many bottoms exist in the region, we only know that the reached bottom is one of the many bottoms (local minima). If we are lucky enough, the bottom may be the lowest bottom (the global minimum) in the region. The above search method is called the *descent method*, and its computational procedure is written algorithmically as follows.

**Algorithm OPT1 (a prototype of the descent method)**

Step 0. Choose an arbitrary point $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, and substitute 0 for $k$ (denoted by $k \leftarrow 0$).

Step 1. Determine a direction vector $\boldsymbol{d}^{(k)}$ at the point $\boldsymbol{x}^{(k)}$.

Step 2. Determine a step size $\alpha^{(k)}$ at the point $\boldsymbol{x}^{(k)}$.

Step 3. $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^{(k)} + \alpha^{(k)} \boldsymbol{d}^{(k)}$.

Step 4. If $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\| < \varepsilon \approx 0$, report $\boldsymbol{x}^{(k+1)}$; otherwise, $k \leftarrow k+1$ and go to Step 1.

The descent method has several variations in the choice of direction vector $\boldsymbol{d}^{(k)}$ (Step 1). When the direction vector is given by the steepest direction, i.e.

$$\boldsymbol{d}^{(k)^{\mathrm{T}}} = -\nabla F(\boldsymbol{x}^{(k)}) = -\left(\frac{\partial F(\boldsymbol{x}^{(k)})}{\partial x_1^{(k)}}, \ldots, \frac{\partial F(\boldsymbol{x}^{(k)})}{\partial x_n^{(k)}}\right), \tag{9.1.7}$$

the descent method is called the *steepest descent method*. In the example of Figure 9.1.3, the steepest direction is shown by the solid arrow (note that the steepest direction is orthogonal to the contour line of $F(\boldsymbol{x})$). The steepest descent method has an advantage in that it requires only the first derivatives. On the other hand, it has a disadvantage in that the convergence to a local optimum solution is not always fast (compare the zig-zag solid line with the broken line in Figure 9.1.3). To overcome this disadvantage, the *Newton method* is proposed, in which the direction vector is given by

$$\boldsymbol{d}^{(k)^{\mathrm{T}}} = -[\nabla^2 F(\boldsymbol{x}^{(k)})]^{-1} \nabla F(\boldsymbol{x}^{(k)}), \tag{9.1.8}$$

where $\nabla^2 F(\boldsymbol{x}^{(k)})$ is the Hessian matrix of $F(\boldsymbol{x}^{(k)})$, i.e.

$$\nabla^2 F(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial^2 F}{\partial x_1^2} & \cdots & \dfrac{\partial^2 F}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 F}{\partial x_n \partial x_1} & \cdots & \dfrac{\partial^2 F}{\partial x_n^2} \end{bmatrix}. \tag{9.1.9}$$

In the example of Figure 9.1.3, this direction is indicated by the broken arrow. The calculation of the Hessian matrix is, however, sometimes difficult. To diminish this difficulty, we may use a direction vector

$$\boldsymbol{d}^{(k)^{\mathrm{T}}} = -[H^{(k)}]^{-1} \nabla F(\boldsymbol{x}^{(k)}), \tag{9.1.10}$$

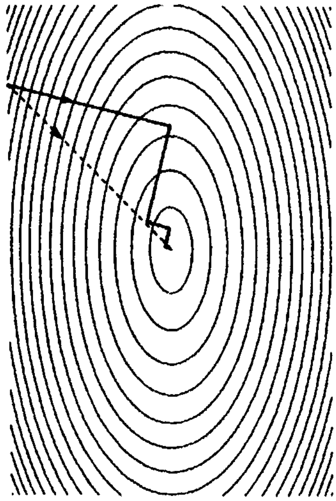


**Figure 9.1.3** Direction vectors: the steepest direction (the solid arrow) and the direction used in the Newton method (the broken line).

where $H^{(k)}$ is a 'simple' approximation of the Hessian matrix in the sense that the matrix operation with $H^{(k)}$ is simpler than that with $\nabla^2 F(\boldsymbol{x})$. For example, $H^{(k)}$ is given by the matrix in which the diagonal elements are given by those of $\nabla^2 F(\boldsymbol{x})$ and the off-diagonal elements are zeros. We call this alternative method the *quasi-Newton method*. Formally, if $H^{(k)} = I$, the quasi-Newton method becomes the steepest descent method; if $H^{(k)} = \nabla^2 F(\boldsymbol{x}^{(k)})$, the method becomes the Newton method.

Having determined the direction vector $\boldsymbol{d}^{(k)}$ at the point $\boldsymbol{x}^{(k)}$ (Step 1), we next consider the straight line $L^{(k)}$ radiating from $\boldsymbol{x}^{(k)}$ in the direction $\boldsymbol{d}^{(k)}$, and search a point $\boldsymbol{x}^{(k+1)}$ on $L^{(k)}$ at which we reconsider the direction (Step 2). We call this search the *line search*. In the example of Figure 9.1.2(b), the direction vector $\boldsymbol{d}^{(k)}$ and the line $L^{(k)}$ are indicated by the arrow and the hair line, respectively. In the literature, several rules are proposed for the line search, for example, Curry's rule, Altman's rule, and Goldstein's rule. Among them, Goldstein's rule has an advantage in that we can always terminate the line search in a finite number of steps.

To show Goldstein's rule explicitly, let $\alpha$ be the distance from $\boldsymbol{x}^{(k)}$ along the line $L^{(k)}$ toward the direction $\boldsymbol{d}^{(k)}$, and

$$h^{(k)}(\alpha) = F(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}). \tag{9.1.11}$$

The function $h^{(k)}(\alpha)$ shows the value of $F(\boldsymbol{x})$ along the line $L^{(k)}$ (see $\beta = h^{(k)}(\alpha)$ on the $\alpha$–$\beta$ plane in Figure 9.1.4 obtained from $L^{(k)}$ in Figure 9.1.2(b)). The line tangential to $\beta = h^{(k)}(\alpha)$ at $\alpha = 0$ is given by

$$\tau^{(k)}(\alpha) = h^{(k)}(0) + \frac{\partial h^{(k)}(0)}{\partial \alpha}\alpha \tag{9.1.12}$$

(see $\beta = \tau^{(k)}(\alpha)$ in Figure 9.1.4). We now consider two lines passing through the point $(0, h^{(k)}(0))$ which are slightly flatter than the line of equation (9.1.12) (see Figure 9.1.4). Mathematically, these lines are written as

$$\eta_i^{(k)}(\alpha) = h^{(k)}(0) + \mu_i \frac{\partial h^{(k)}(0)}{\partial \alpha}\alpha, \quad i = 1,2, \tag{9.1.13}$$

where $0 < \mu_1 \leq \mu_2 < 1$ (in practice, $\mu_1 = 0.4$ and $\mu_2 = 0.6$ are often used). With this function, we define the set

$$M^{(k)} = \{\alpha \mid \eta_2^{(k)}(\alpha) \leq h^{(k)}(\alpha) \leq \eta_1^{(k)}(\alpha)\} \tag{9.1.14}$$

(see the heavy solid line segment on the $\alpha$-axis in Figure 9.1.4). The set $M^{(k)}$ is always finite because $F(x)$ is assumed to be bounded below. *Goldstein's rule* is the rule in which we choose $\alpha^{(k)}$ in $M^{(k)}$. We can find such an $\alpha^{(k)}$ in a finite number of steps using the following algorithm.

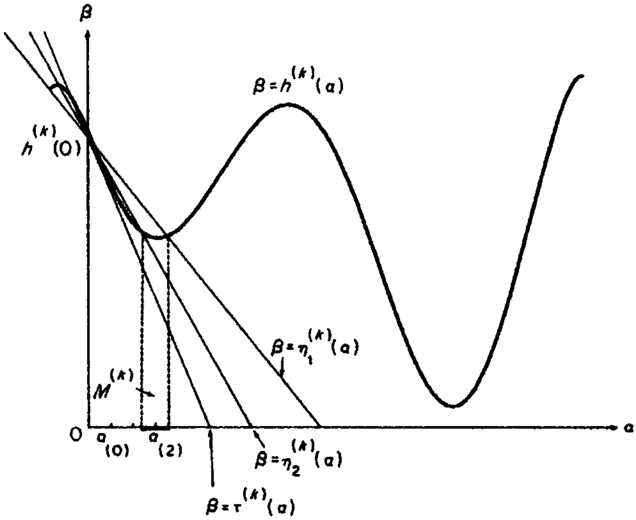



**Figure 9.1.4** Goldstein's rule.

**Algorithm OPT2 (Goldstein's rule)**

Step 0. Choose an arbitrary initial value $\alpha_{(0)} > 0$, and $i \leftarrow 0$.

Step 1. $\alpha_{(i)} \leftarrow (i+1)\alpha_{(0)}$.

Step 2. If $\eta_2^{(k)}(\alpha_{(i)}) \leq h^{(k)}(\alpha_{(i)})$, then go to Step 3; otherwise, $i \leftarrow i+1$ and go to Step 1.

Step 3. If $h^{(k)}(\alpha_{(i)}) \leq \eta_1^{(k)}(\alpha_{(i)})$, then report $\alpha_{(i)}$; otherwise, $\alpha_{(i)} \leftarrow \frac{1}{2}\{\alpha_{(i)} + \alpha_{(i-1)}\}$, and go to Step 2 (where $\alpha_{(-1)} = 0$).

### 9.1.3 The penalty function method

The descent method is developed for the unconstrained non-linear programming problem (Problem OPT2); it cannot be directly applied to the constrained non-linear programming problem (Problem OPT1, $m_1 + m_2 \geq 1$). To solve the latter problem, we can use the 'transformation method'. The essential idea is that we first transform a constrained non-linear programming problem into an unconstrained non-linear programming problem (Problem OPT2); next we solve this unconstrained programming problem with a method that solves the unconstrained programming problem, say the descent method. The transformation method has several variations, such as the penalty function method and the multiplier method. In this subsection we show the penalty function method, because this method is one of the most basic methods and many other methods are developed from this method.

In the constrained programming problem of Problem OPT1, we have to find a local minimum in the feasible region $S$ determined by the relation (9.1.2). In the process of searching a local minimum, however, we may choose an infeasible solution. In such a case we are supposed to pay a penalty. If this penalty is extremely high, we give up that solution and try to find another solution. In the end, we may reach the local minimum in the feasible region $S$.

To state this method a little more precisely, suppose that a penalty $R_1(\boldsymbol{x})$ for an infeasible solution is given by

$$R_1(\boldsymbol{x}) = \begin{cases} +\infty & \text{if } \boldsymbol{x} \notin S, \\ 0 & \text{if } \boldsymbol{x} \in S, \end{cases} \tag{9.1.15}$$

and let $L(\boldsymbol{x}) = F(\boldsymbol{x}) + R_1(\boldsymbol{x})$. Noticing that $L(\boldsymbol{x})$ coincides with $F(\boldsymbol{x})$ for any feasible solution and it is $+\infty$ for any infeasible solution, we can restate the constrained programming problem (Problem OPT1) as the unconstrained programming problem:

$$\min_{\boldsymbol{x}} L(\boldsymbol{x}) = \min_{\boldsymbol{x}} [F(\boldsymbol{x}) + R_1(\boldsymbol{x})]. \tag{9.1.16}$$

In principle, this idea appears to work well, but we have a difficulty in implementing this method in practice. In the descent method, we assume that the objective function has at least the first derivative. The objective function $L(\boldsymbol{x})$ is, however, discontinuous at the boundary of the feasible region $S$, even if $+\infty$ is replaced by a very large number.

To overcome this difficulty, we approximate the penalty function $R_1(\boldsymbol{x})$ by a smooth function whose value (i.e. the penalty) smoothly increases as $\boldsymbol{x}$ moves farther from the feasible region $S$. For instance,

$$R_2(\boldsymbol{x}) = \sum_{i=1}^{m_1} [\max\{g_i(\boldsymbol{x}, 0)\}]^2 + \sum_{i=m_1+1}^{m_1+m_2} g_i(\boldsymbol{x})^2. \tag{9.1.17}$$

To show how this penalty function works well, let us consider an illustrative example: $F(x) = (x+1)^2$, $g_1(x) = -x$ and $g_2(x) = x - 1$, which are depicted in Figure 9.1.5. In this case, the penalty function of equation (9.1.17) is written as

$$R_2(x) = [\max\{-x, 0\}]^2 + [\max\{x-1, 0\}]^2, \tag{9.1.18}$$

which is shown in Figure 9.1.5(b). Then the transformed function is given by $L(x) = F(x) + R_2(x)$, which is depicted in Figure 9.1.5(c).

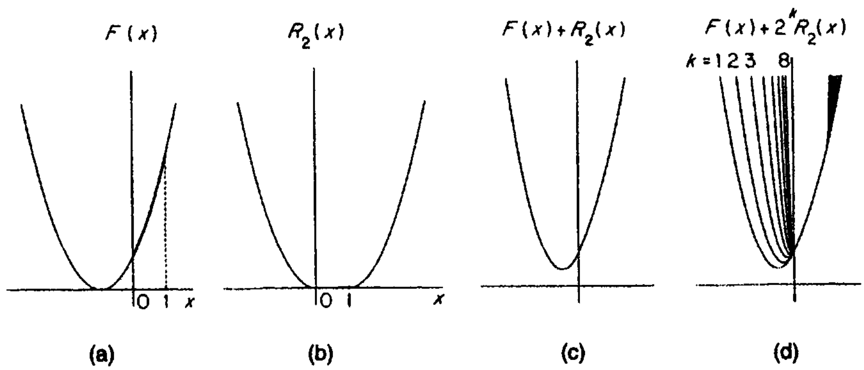



**Figure 9.1.5** The penalty function method: (a) $F(x) = (x+1)^2$, $g_1(x) = -x \leq 0$, $g_2(x) = x-1 \leq 0$; (b) $R_2(x) = [\max\{-x, 0\}]^2 + [\max\{x-1, 0\}]^2$; (c) $L(x) = F(x) + R_2(x)$; (d) $L(x)^{(k)} = F(x) + 2^k R_2(x)$.

In this example we can analytically obtain the global minimum of $L(x)$, and the solution is $x^* = -0.5$. Obviously, this solution violates the constraint $g_1(x) = -x \leq 0$. This violation suggests that the penalty should be higher. We do not know in advance, however, how high the penalty should be. We hence increase the penalty from low to high according to $2^k R_2(x)$, $k = 0, 1, \ldots$. As a result, we obtain a series of solutions, $-0.50$, $-0.33$, $-0.20$, $-0.11$, $-0.05$, $-0.03$, $-0.02$, $-0.01$ for $k = 0, \ldots, 8$ (Figure 9.1.5(d)). As is easily noticed in Figure 9.1.5(a), the minimum is 0. The above series appears to approach this minimum.

In general, with the penalty function of equation (9.1.17), we can transform the constrained programming problem of Problem OPT1 into the unconstrained programming problem

$$\min_{x} L^{(k)}(\boldsymbol{x}) = \min_{x} [F(\boldsymbol{x}) + r_0^k R_2(\boldsymbol{x})], \tag{9.1.19}$$

where $r_0 > 1$. It is shown that if there exists a local optimum $\boldsymbol{x}^{*(k)}$, for this unconstrained programming problem, then there exists a series $\boldsymbol{x}^{*(0)}, \boldsymbol{x}^{*(1)}, \boldsymbol{x}^{*(2)}, \ldots, \boldsymbol{x}^{*(k)}, \ldots$, that converges to a local minimum of the constrained programming problem, Problem OPT1. Hence, as we expected, the series in the above example actually converges to 0. This technique is called the *sequential unrestrained minimization technique* (Fiacco and McCormick, 1968).

We call the above method the *penalty function method* or, more specifically, the *exterior penalty function method.* With this method we can solve the constrained programming problems in the following sections. Of course we can use alternative methods which are modified from the exterior penalty function method, such as the *interior penalty function method* and the *mixed penalty function method.* Furthermore, to gain fast convergence, we may use the *multiplier method.* The reader who wishes to use these alternative methods should see, for example, Gill *et al.* (1981, Chapters 4–6).

## 9.2 LOCATIONAL OPTIMIZATION OF POINTS

Using the computational methods in Section 9.1, we now wish to solve the locational optimization problems mentioned in the introduction. We first discuss the locational optimization of points in a plane or in a space, where the points represent point-like facilities, service points, observation points, nests of birds, crystallites, points from which some influential power is generated, and so forth.

The study of locational optimization of point-like facilities may date back to Launhardt (1882) or Weber (1909) (the so-called Weberian problem; a historical review is provided by Wesolowsky, 1993), but the study has been rapidly progressed since Hakimi (1964) in Operations Research. The recent progress can be seen, for example, in Handler and Mirchandani (1979), the *European Journal of Operations Research* (1985, Vol. 20, No. 3), Love *et al.* (1988), Mirchandani and Francis (1990) and Drezner (1995). It should be noted that the problem dealt with there is mostly locational optimization on

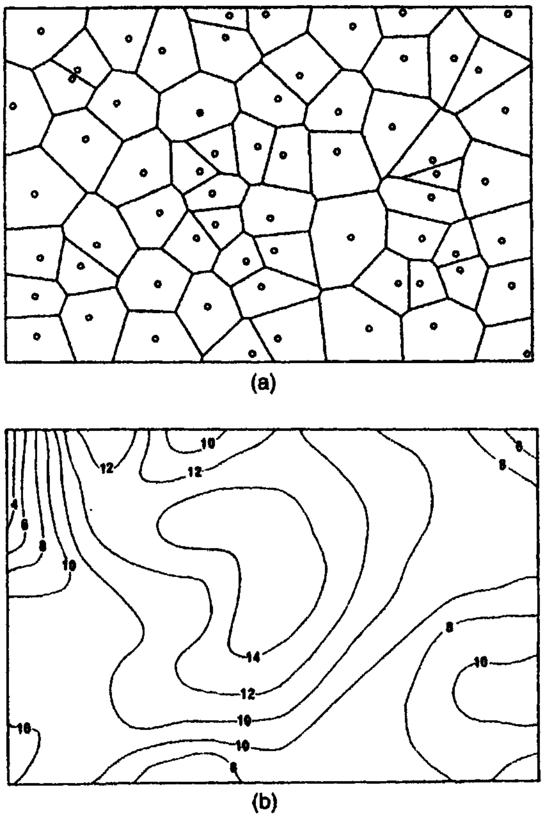


**Figure 9.2.1** (a) Public mail boxes in Koganei district in Tokyo and (b) the distribution of population (×10/ha).

a network where the demand for services supplied by facilities is supposed to arise only on nodes of the network.

Compared with this rapid progress, progress was slow in the study of locational optimization in a continuous plane where demand arises at any point in a plane and feasible locations of facilities are any point in the plane. This slow progress was due to the lack of efficient computational methods for managing complicated geometrical calculation on a plane. In fact, the computational method in the 1960s could deal with less than twenty facilities, e.g. Leamer (1968). In the 1980s, however, computational geometry (Preparata and Shamos, 1985), in particular efficient computational methods for constructing a Voronoi diagram (Chapter 4), have been rapidly developed to overcome this difficulty (Iri *et al.*, 1984; Sugihara and Iri, 1992). We can now practically solve locational optimization problems of a large number of points in a continuous plane or space with Voronoi diagrams (see the review by Okabe and Suzuki, 1997).

### 9.2.1 Locational optimization of point-like facilities used by independent users

We first consider the locational optimization of point-like facilities which are used by independent users. A good example of such a facility is a public mail box. For illustrative purposes, we show an actual example in Figure 9.2.1 where the empty circles and the filled circle indicate the locations of public mail boxes in Koganei, Tokyo. Our concern is with whether or not the configuration of those mail boxes is 'optimal'; if not, to what extent we can improve the configuration by relocating the mail boxes?

To answer these questions, we should explicitly define what is 'optimal'. To define it, let us discuss two factors: the cost borne by the post office and the cost borne by users. Regarding the former cost, the major cost is the cost of collecting the mail. In Koganei, the mail boxes are managed by the main post office indicated by the filled circle in Figure 9.2.1. A collector wagon starts from this office, visits all mail boxes successively, and returns to the office. We have to take this cost into account. However, since a public organization is supposed to minimize the user's cost or maximize the user's convenience instead of maximizing profit, the primary factor is the user's convenience rather than the cost borne by the post office. For the moment, let us consider the primary factor. The secondary factor will be taken into account later (see the locational optimization of service points of a mobile facility in Section 9.3.1).

The user's cost or user's convenience may be measured in terms of the distance from a user to a mail box. Since mail boxes are indifferent, most users go to their nearest mail box. As a result, every mail box, say a mail box $i \in I_n = \{1, \ldots, n\}$, has its own catchment area or the area in which the nearest mail box from any point in this area is the mail box $i$. Recalling Property V6 in Chapter 2, we notice that this region is indeed the Voronoi polygon (strictly speaking, since users use streets, this region is given by the

Voronoi region of the network area-Voronoi diagram defined in Section 3.8.3) associated with the mail box $i$.

Obviously, a street distance is different from a crow-flight distance (the Euclidean distance). This difference is empirically examined by Kobayashi (1983), Koshizuka and Kobayashi (1983), and Love *et al.* (1988, Chapter 10). At first glance, this difference appears very big, but according to Koshizuka and Kobayashi (1983), this difference is not so large as we might expect. If this difference is allowable in practice, the catchment areas of mail boxes can be approximated by the ordinary Voronoi diagram. The solid lines in Figure 9.2.1 indicate the ordinary Voronoi diagram generated by the mail boxes in Koganei.

As mentioned above, we measure the cost of a user by the Euclidean distance from the user to his/her nearest mail box. The cost of users as a whole is hence measured by the average nearest neighbour distance, i.e. the distance to the nearest mail box averaged over all users in a region. In these terms, we can say that the optimal configuration of mail boxes implies the configuration which provides the minimum average nearest neighbour distance.

To formalize the above model mathematically, let $S$ be a closed subset of $\mathbb{R}^2$ representing a region; $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be $n$ points (location vectors) in $S$ indicating the locations of $n$ facilities; $\phi(\boldsymbol{x})$ be the density of users over region $S$, where the integral of $\phi(\boldsymbol{x})$ over $S$ is assumed to be unity without loss of generality; and $\mathcal{V}_{\cap S} = \{V_1 \cap S, \ldots, V_n \cap S\}$ be the Voronoi diagram generated by $P = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ bounded by $S$. For simplicity, we write $\mathcal{V} = \{V_1, \ldots, V_n\}$ for $\mathcal{V}_{\cap S} = \{V_1 \cap S, \ldots, V_n \cap S\}$.

The Euclidean distance from a user at $\boldsymbol{x}$ to a mail box at $\boldsymbol{x}_i$ is written as $\|\boldsymbol{x} - \boldsymbol{x}_i\|$. More generally, we can consider the travel cost from a user at $\boldsymbol{x}$ to a facility at $\boldsymbol{x}_i$ as a function of the Euclidean distance or the squared Euclidean distance, $f(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2)$ (note that we use the squared Euclidean distance simply for analytical convenience). One of the simplest cost functions is $f(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2) = c\sqrt{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2} = c\|\boldsymbol{x} - \boldsymbol{x}_i\|$. With this cost function, the objective function discussed above is written as

$$F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \int_{V_i} f(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2)\, \phi(\boldsymbol{x})\, d\boldsymbol{x}. \tag{9.2.1}$$

Note that since the total number of users is assumed to be unity, the total travel cost is the same as the average cost.

When $f(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2) = \|\boldsymbol{x} - \boldsymbol{x}_i\|$, and $\phi(\boldsymbol{x}) = 1/S$ (the density of users is uniform over $S$), we can calculate the integral in equation (9.2.1). The explicit form is shown in Section 8.3.

In terms of the objective function given by equation (9.2.1), we can formally state the locational optimization problem of public mail boxes as the following programming problem.

**Problem OPT3 (minimization of the average travel cost to the nearest points without constraints)**

$$\min_{x_1,\ldots,x_n} \sum_{i=1}^{n} \int_{V_i} f(\|x - x_i\|^2)\,\phi(x)\,dx. \tag{9.2.2}$$

The network problem corresponding to Problem OPT3 is called the *multi-median problem.* In this problem, permissible locations are only on nodes and a distance is given by the shortest path distance on a network. When a network density is dense, we may expect that the solution of Problem OPT3 is close to the solution of the multi-median problem. This approximation is quite useful when a network is huge, because the multi-median problem on a huge network is hard to solve.

The function $F$ of equation (9.2.1) is apparently non-linear, and moreover it is non-convex. We can see the latter property from the following fact. If $(x_1^*, \ldots, x_i^*, \ldots, x_j^*, \ldots, x_n^*)$ is a local minimum of $F$, then $(x_1^*, \ldots, x_j^*, \ldots, x_i^*, \ldots, x_n^*)$ is also a local minimum of $F$, because the equation $F(x_1^*, \ldots, x_i^*, \ldots, x_j^*, \ldots, x_n^*) = F(x_1^*, \ldots, x_j^*, \ldots, x_i^*, \ldots, x_n^*)$ holds (confirm this equation in equation (9.2.1)). This implies that we have multiple local minima unless $x_1^* = \ldots = x_n^*$. The solution, $x_1^* = \ldots = x_n^*$, means that all facilities are located at the same place. Obviously, this solution does not give a minimum. Therefore the function $F$ has multiple local minima, implying that the function $F$ is non-convex.

Because of this non-convex property, as we mentioned in Section 9.1, we have to use a numerical method. If we employ the steepest descent method, we need the first derivatives of $F$. To derive them, suppose that the $i$th generator point slightly changes its location from $x_i$ to $x_i + \delta x_i$ ($\|\delta x_i\| \approx 0$) provided that all other generator points remain at the same location. Let $\mathcal{V}$ be the Voronoi diagram generated by $\{x_1, \ldots, x_i, \ldots, x_n\}$ and $\mathcal{V}'$ be that generated by $\{x_1, \ldots, x_i + \delta x_i, \ldots, x_n\}$; $V_i$ be the Voronoi polygon of the $i$th generator in $\mathcal{V}$ and $V_i'$ be that in $\mathcal{V}'$; and $J_i$ be the set of indices of the Voronoi polygons adjacent to $V_i$. Owing to the slight move $\delta x_i$, the Voronoi diagram slightly changes, but, as is indicated by the broken lines in Figure 9.2.2, this change occurs only in $V_i$ and its adjacent Voronoi polygons, i.e. $V_j, j \in J_i$. Moreover, the Voronoi edges between adjacent Voronoi polygons $V_j$ and $V_k$, $j, k \in J_i$, and the Voronoi edges between adjacent Voronoi polygons $V_j'$ and $V_k'$, $j, k \in J_i$, are on the same line. From this property and equation (9.2.1), we obtain

$$\begin{aligned}\Delta F &= F(x_1, \ldots, x_i + \delta x_i, \ldots, x_n) - F(x_1, \ldots, x_i, \ldots, x_n)\\ &= \int_{V_i'} f(\|x_i + \delta x_i - x\|^2)\,\phi(x)\,dx - \int_{V_i} f(\|x_i - x\|^2)\,\phi(x)\,dx \\ &\quad + \sum_{j \in J_i} \left[\int_{V_j'} f(\|x_j - x\|^2)\,\phi(x)\,dx - \int_{V_j} f(\|x_j - x\|^2)\,\phi(x)\,dx\right].\end{aligned} \tag{9.2.3}$$

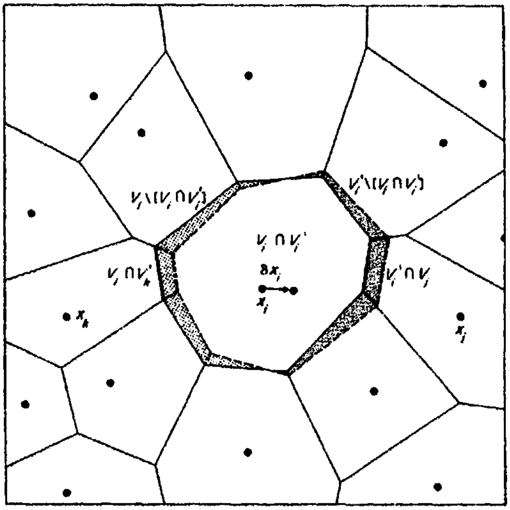



**Figure 9.2.2** Variation in a Voronoi polygon.

Since the equations $V_i = (V_i \cap V_i') \cup (V_i \backslash V_i \cap V_i')$, $V_i' = (V_i \cap V_i') \cup (V_i' \backslash V_i \cap V_i')$, $V_j = (V_j \cap V_j') \cup (V_i' \cap V_j)$ and $V_j' = (V_j \cap V_j') \cup (V_i \cap V_j')$ hold, (see Figure 9.2.2), equation (9.2.3) is written as

$$\begin{aligned}\Delta F = &\int_{V_i' \cap V_i} \left[ f(\| x_i + \delta x_i - x \|^2) - f(\| x_i - x \|^2) \right] \phi(x)\,dx \\ &+ \int_{V_i' \backslash V_i \cap V_i'} f(\| x_i + \delta x_i - x \|^2)\,\phi(x)\,dx \int_{V_i \backslash V_i \cap V_i'} f(\| x_i - x \|^2)\,\phi(x)\,dx \\ &+ \sum_{j \in J_i} \left[ \int_{V_i \cap V_j'} f(\| x_j - x \|^2)\,\phi(x)\,dx - \int_{V_i' \cap V_j} f(\| x_j - x \|^2)\,\phi(x)\,dx \right]. \end{aligned} \tag{9.2.4}$$

In Figure 9.2.2 the integral domains $V_i' \backslash (V_i \cap V_i')$ and $V_i \backslash (V_i \cap V_i')$ are indicated by the sparsely and densely shaded regions, respectively; the integral domains $V_i' \cap V_j$ and $V_i \cap V_k'$ are indicated by the regions surrounded by the heavy lines (note that $V_i \cap V_j'$ or $V_i' \cap V_j$ may be empty or may not, but at least one of the two is non-empty). Observing in Figure 9.2.2 that the lightly shaded region $V_i \backslash (V_i \cap V_i')$ consists of subregions $V_i \cap V_k'$, $k \in J_i$ (one of which is indicated by the heavy solid lines on the left-hand side), and the densely shaded region $V_i' \backslash (V_i \cap V_i')$ consists of subregions $V_i' \cap V_j$, $j \in J_i$ (one of which is indicated by the heavy solid lines on the right-hand side), we notice that the following equations hold:

$$V_i' \backslash (V_i \cap V_i') = \sum_{j \in J_i} V_i' \cap V_j, \quad V_i \backslash (V_i \cap V_i') = \sum_{j \in J_i} (V_i \cap V_j'). \tag{9.2.5}$$

Using these equations, equation (9.2.4) is written as

$$\Delta F=\int_{V_i'\cap V_i}\left\{f(\|\boldsymbol{x}_i+\delta\boldsymbol{x}_i-\boldsymbol{x}\|^2)-f(\|\boldsymbol{x}_i-\boldsymbol{x}\|^2)\right\}\phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$
$$+\sum_{j\in J_i}\int_{V_i'\cap V_j}\left\{f(\|\boldsymbol{x}_i+\delta\boldsymbol{x}_i-\boldsymbol{x}\|^2)-f(\|\boldsymbol{x}_j-\boldsymbol{x}\|^2)\right\}\phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \tag{9.2.6}$$
$$+\sum_{j\in J_i}\int_{V_i\cap V_j'}\left\{f(\|\boldsymbol{x}_j-\boldsymbol{x}\|^2)-f(\|\boldsymbol{x}_i-\boldsymbol{x}\|^2)\right\}\phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$$

As $\delta\boldsymbol{x}_i$ approaches zero, the integral domains $V_i\cap V_j'$ and $V_i'\cap V_j$ in equation (9.2.6) reduce to the Voronoi edge shared by $V_i$ and $V_j$, and

$$\lim_{\delta\boldsymbol{x}_i\to 0}|V_i'\cap V_j|=0,\quad \lim_{\delta\boldsymbol{x}_i\to 0}|V_i\cap V_j'|=0. \tag{9.2.7}$$

From this property and the property that the equation $\|\boldsymbol{x}_i-\boldsymbol{x}\|=\|\boldsymbol{x}_j-\boldsymbol{x}\|$ (or $\|\boldsymbol{x}_i+\delta\boldsymbol{x}_i-\boldsymbol{x}\|=\|\boldsymbol{x}_j-\boldsymbol{x}\|$) holds if the point $\boldsymbol{x}$ is on the Voronoi edge shared by $V_i$ and $V_j$ (or $V_i'$ and $V_j'$), we have the following relations:

$$\begin{aligned}&\|\boldsymbol{x}_i-\boldsymbol{x}\|\approx\|\boldsymbol{x}_j-\boldsymbol{x}\|,\ \boldsymbol{x}\in V_i\cap V_j',\quad j\in J_i,\\&\|\boldsymbol{x}_i+\delta\boldsymbol{x}_i-\boldsymbol{x}\|\approx\|\boldsymbol{x}_j-\boldsymbol{x}\|,\ \boldsymbol{x}\in V_i'\cap V_j,\ j\in J_i,\end{aligned} \tag{9.2.8}$$

for sufficiently small $\delta\boldsymbol{x}_i$. From this equation and the assumption that $f(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|)$ and $\phi(\boldsymbol{x})$ are continuous, the value of $|\{f(\|\boldsymbol{x}_i+\delta\boldsymbol{x}_i-\boldsymbol{x}\|^2)-f(\|\boldsymbol{x}_j-\boldsymbol{x}\|^2)\}\,\phi(\boldsymbol{x})|$ is of order $\|\delta\boldsymbol{x}_i\|$, denoted by $O(\|\delta\boldsymbol{x}_i\|)$, for $\boldsymbol{x}\in V_i'\cap V_j$. From equation (9.2.7), the area of the integral domain $V_i'\cap V_j$ is also $O(\|\delta\boldsymbol{x}_i\|)$ for sufficiently small $\delta\boldsymbol{x}_i$. Hence, $|V_i'\cap V_j|\{f(\|\boldsymbol{x}_i+\delta\boldsymbol{x}_i-\boldsymbol{x}\|^2)-f(\|\boldsymbol{x}_j-\boldsymbol{x}\|^2)\}\,\phi(\boldsymbol{x})$ is $O(\|\delta\boldsymbol{x}_i\|^2)$. Similarly, $|V_i\cap V_j'|\{f(\|\boldsymbol{x}_i-\boldsymbol{x}\|^2)-f(\|\boldsymbol{x}_j-\boldsymbol{x}\|^2)\}$ $\phi(\boldsymbol{x})$ is $O(\|\delta\boldsymbol{x}_i\|^2)$. Therefore, equation (9.2.6) is written as

$$\Delta F=\int_{V_i'\cap V_i}\left\{f(\|\boldsymbol{x}_i+\delta\boldsymbol{x}_i-\boldsymbol{x}\|^2)-f(\|\boldsymbol{x}_i-\boldsymbol{x}\|^2)\right\}\phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}+O(\|\delta\boldsymbol{x}_i\|^2), \tag{9.2.9}$$

from which we obtain

$$\frac{\partial F}{\partial x_{i\kappa}}=\lim_{\delta x_{i\kappa}\to 0}\frac{\Delta F}{\delta x_{i\kappa}}$$
$$=\lim_{\delta x_{i\kappa}\to 0}\left[\int_{V_i'\cap V_i}\frac{f(\|\boldsymbol{x}_i+\delta\boldsymbol{x}_i-\boldsymbol{x}\|^2)-f(\|\boldsymbol{x}_i-\boldsymbol{x}\|^2)}{\delta x_{i\kappa}}\phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \right. \tag{9.2.10}$$
$$\left. -+\frac{O(\|\delta\boldsymbol{x}_i\|^2)}{\delta x_{i\kappa}}\right],\quad \kappa=1,2.$$

Therefore, we obtain the first derivatives as

$$\frac{\partial F}{\partial x_{i\kappa}}=\int_{V_i}2(x_{i\kappa}-x_\kappa)\,f'(\|\boldsymbol{x}_i-\boldsymbol{x}\|^2)\,\phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},\quad \kappa=1,2. \tag{9.2.11}$$

Note that this result holds not only for $\boldsymbol{x}_i^{\mathrm{T}}=(x_{i1},x_{i2})$, $\boldsymbol{x}^{\mathrm{T}}=(x_1,x_2)\in\mathbb{R}^2$, but also for $\boldsymbol{x}_i^{\mathrm{T}}=(x_{i1},\ldots,x_{i\kappa},\ldots,x_{im})$ and $\boldsymbol{x}^{\mathrm{T}}=(x_1,\ldots,x_\kappa,\ldots,x_m)\in\mathbb{R}^m$. In the latter case, $\kappa=1,2$ is simply replaced by $\kappa=1,\ldots,m$.

Using the steepest descent method with this derivative, we can obtain a local optimal solution. Alternatively, as mentioned in Section 9.1, we may use the Newton method to gain faster convergence. For this use, we need the second partial derivatives of $F$. The derivation of the second partial derivatives is much more lengthy than the above derivation. Since space is limited, we show only the result here. The detailed derivation is given by Suzuki (1987) and Asami (1991). Using the notation $x_i^T = (x_{i1}, \ldots, x_{i\kappa}, \ldots, x_{i\lambda}, \ldots, x_{im})$ and $x^T = (x_1, \ldots, x_\kappa, \ldots, x_\lambda, \ldots, x_m) \in \mathbb{R}^m$, the second partial derivatives are written as

$$\frac{\partial^2 F}{\partial x_{i\kappa}\partial x_{i\lambda}} = \int_{V_i}\left[2\delta(\kappa,\lambda)\, f'(\|x_i - x\|^2) + 4(x_{i\kappa} - x_\kappa)(x_{i\lambda} - x_\lambda)\, f''(\|x_i - x\|^2)\, \phi(x)\right] dx$$

$$- \sum_{j\in J_i}\int_{V_i\cap V_j} \frac{2}{\|x_j - x_i\|^2}(x_{i\kappa} - x_\kappa)(x_{i\lambda} - x_\lambda) f'(\|x_i - x\|^2)\, \phi(x)\, dx, \tag{9.2.12}$$

where

$$\delta(\kappa,\lambda) = \begin{cases} 1 & \text{if } \kappa = \lambda, \\ 0 & \text{if } \kappa \neq \lambda,\ \kappa,\lambda \in I_m, \end{cases} \tag{9.2.13}$$

$$\frac{\partial^2 F}{\partial x_{i\kappa}\partial x_{j\lambda}} = \int_{V_i\cap V_j} \frac{2}{\|x_j - x_i\|}(x_{i\kappa} - x_\kappa)(x_{j\lambda} - x_\lambda) \times f'(\|x_i - x\|^2)\, \phi(x)\, dx, \quad j \neq i,\ j\in J_i\ \kappa,\lambda \in I_m, \tag{9.2.14}$$

$$\frac{\partial^2 F}{\partial x_{i\kappa}\partial x_{j\lambda}} = 0, \quad j \neq i,\ j \notin J_i, \tag{9.2.15}$$

(note that $V_i \cap V_j$ is the Voronoi edge shared by $V_i$ and $V_j$). Although we may use the Newton method with these second derivatives, the calculation of the Hessian becomes time consuming for a large number of facilities when the density $\phi(x)$ is not uniform. To reduce computing time, Iri *et al.* (1984) recommend, based upon their numerical experiment, using the quasi-Newton method with the diagonal matrix, $H$, whose diagonal elements are given by the first term in equation (9.2.12). Using this quasi-Newton method, we can solve Problem OPT3 in practice for a large number of facilities. (Note that when $\phi(x)$ is not uniform, Iri *et al.*, 1984, carry out the integration in equations (9.2.11)–(9.2.15) using the numerical integration formula shown in Abramowitz and Stegun, 1964.)

Figure 9.2.3 shows the optimal configuration of $n = 128$ points in a square $S$ for the three different density functions $\phi(x)$ (Iri *et al.*, 1984).

Using the same method, we can obtain the local optimal configuration of public mail boxes in Koganei. The result is shown in Figure 9.2.4. It is of interest to compare this local optimal configuration with the present configuration shown in Figure 9.2.1(a).

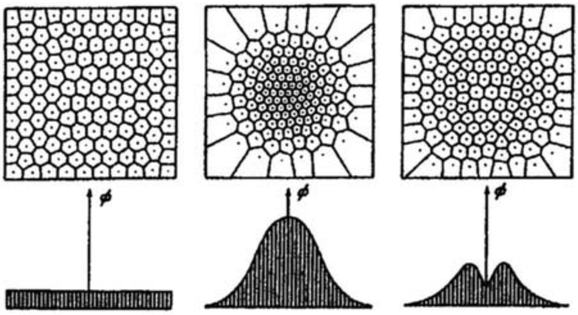



**Figure 9.2.3** Optimal location of 128 points in a square for: (a) the density function $\phi(\boldsymbol{x})$ given by $\phi(\boldsymbol{x}) = c, \boldsymbol{x} \in S$; (b) the density function $\phi(\boldsymbol{x})$ given by $\phi(\boldsymbol{x}) = c \exp(-25\|\boldsymbol{x}\|^2), \boldsymbol{x} \in S$; and (c) the density function $\phi(\boldsymbol{x})$ given by $\phi(\boldsymbol{x}) = c \exp(-\|\boldsymbol{x}\|(25\|\boldsymbol{x}\| - 10)), \boldsymbol{x} \in S$. (*Source*: Iri *et al.*, 1984, Figures 1 and 2.)

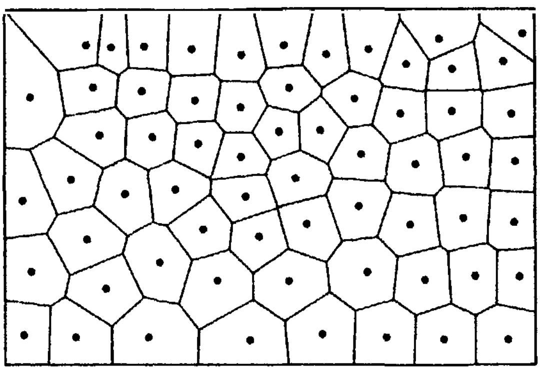


**Figure 9.2.4** Locational optimization of public mail boxes in Koganei, Tokyo.

### 9.2.2 Locational optimization of points in a three-dimensional space

In the above example we formulated Problem OPT3 in the plane $\mathbb{R}^2$. We can also formulate the same problem in the space $\mathbb{R}^3$ and can solve the problem almost in the same manner using the three-dimensional Voronoi diagram. Figure 9.2.5 shows the optimal configuration of 16 points in a cube where the density function $\phi(\boldsymbol{x})$ is given by the uniform distribution (Takeda, 1985). This result is quite suggestive for crystallography. It is shown in Rogers (1964) that Voronoi polyhedra become truncated octahedra when generators are given by cubic lattice points in an infinite space (see also Chapter 7).

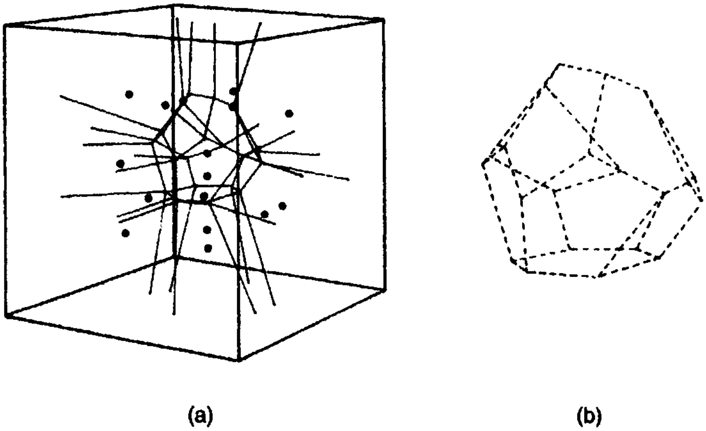



**Figure 9.2.5** Optimal location of 16 points in a cube: (a) the Voronoi diagram of the optimized points; (b) a Voronoi polyhedron in (a). (*Source*: Takeda, 1985, Figure 3.7.)

### 9.2.3 Locational optimization of point-like facilities used by groups

Facilities are not always used by independent users; some kinds of facilities are used by groups of users. For example, a tennis court is used by at least a pair of players; a convention hall is used by groups of certain members. This type of locational optimization problem is examined by Ohsawa and Suzuki (1987), upon which this subsection is based.

Suppose that there are $n$ facilities located at $x_1, \ldots, x_n$ in region $S$, and that every group consists of $m$ members. Groups are distributed over $S$ according to a density function $\phi(y_1, \ldots, y_m)$, where $y_1, \ldots, y_m$ indicate the locations of $m$ members. The value of $\phi(y_1, \ldots, y_m)\, dy_1, \ldots, dy_m$ indicates the number of groups whose members $1, \ldots, m$ are in a small region $dy_1$ around $y_1, \ldots,$ and in a small region $dy_m$ around $y_m$, respectively. Since $m$ members in a group use the same facility, say facility $i$, the total travel cost of the group is given by the sum of the travel costs of the $m$ members from their locations to facility $i$, i.e. $\sum_{j=1}^{m} f(\|y_j - x_i\|^2)$.

We make two assumptions. First, every group chooses the facility that gives the minimum total travel cost, i.e. $\min_i \sum_{j=1}^{m} f(\|y_j - x_i\|^2)$. Second, every group uses the facility $\alpha$ times per unit time regardless of the travel cost to their nearest facility. For convenience, $\alpha = 1$ without loss of generality. Under these assumptions, the total travel cost of groups over region $S$ is given by

$$F(x_1, \ldots, x_n) = \int_S \min_i \sum_{j=1}^{m} f(\| y_j - x_i \|^2)\, \phi(y_1, \ldots, y_m)\, dy_1 \ldots dy_m. \quad (9.2.16)$$

The locational optimization problem is hence to minimize this objective function with respect to $x_1, \ldots, x_n$. This problem is difficult to solve for a general travel cost function. The problem, however, becomes tractable in a specific case where the travel cost function is given by $f(\|\boldsymbol{y}_j - \boldsymbol{x}_i\|^2) = \|\boldsymbol{y}_j - \boldsymbol{x}_i\|^2$. For this travel cost function, the following equation holds:

$$\sum_{j=1}^{m} \| \boldsymbol{y}_j - \boldsymbol{x}_i \|^2 = m \| \bar{\boldsymbol{y}} - \boldsymbol{x}_i \|^2 + \sum_{j=1}^{m} \| \boldsymbol{y}_j - \bar{\boldsymbol{y}} \|^2, \tag{9.2.17}$$

where $\bar{\boldsymbol{y}}$ is the centroid of $m$ members, i.e. $\bar{\boldsymbol{y}} = \sum_{i=1}^{m} \boldsymbol{y}_i/m$. Substituting equation (9.2.17) into equation (9.2.16), we obtain

$$\begin{aligned} F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) &= \int_S \min_i \sum_{j=1}^{m} f(\| \boldsymbol{y}_j - \boldsymbol{x}_i \|^2)\, \phi(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)\, \mathrm{d}\boldsymbol{y}_1 \ldots \mathrm{d}\boldsymbol{y}_m \\ &= \int_S \min_i \left[ m \| \bar{\boldsymbol{y}} - \boldsymbol{x}_i \|^2 + \sum_{j=1}^{m} f(\| \boldsymbol{y}_j - \bar{\boldsymbol{y}} \|^2) \right] \phi(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)\, \mathrm{d}\boldsymbol{y}_1 \ldots \mathrm{d}\boldsymbol{y}_m. \end{aligned} \tag{9.2.18}$$

The first term, $\min_i m \|\bar{\boldsymbol{y}} - \boldsymbol{x}_i\|^2$, implies the distance to the nearest facility from the centroid of $m$ members (multiplied by $m$). Obviously, if the centroid $\bar{\boldsymbol{y}}$ is in the Voronoi polygon $V_i$, the nearest facility is the facility at $\boldsymbol{x}_i$. The second term, $\min_i \sum_{j=1}^{m} \|\boldsymbol{y}_j - \bar{\boldsymbol{y}}\|^2$, is the total cost from $m$ members to their centroid, which is independent of $i$, and so $\min_i \sum_{j=1}^{m} \|\boldsymbol{y}_j - \bar{\boldsymbol{y}}\|^2 = \sum_{j=1}^{m} \|\boldsymbol{y}_j - \bar{\boldsymbol{y}}\|^2$. Thus, equation (9.2.18) is written as

$$F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = m \sum_{i=1}^{n} \int_{V_i} \| \bar{\boldsymbol{y}} - \boldsymbol{x}_i \|^2\, \phi_g(\bar{\boldsymbol{y}})\, \mathrm{d}\bar{\boldsymbol{y}} + c, \tag{9.2.19}$$

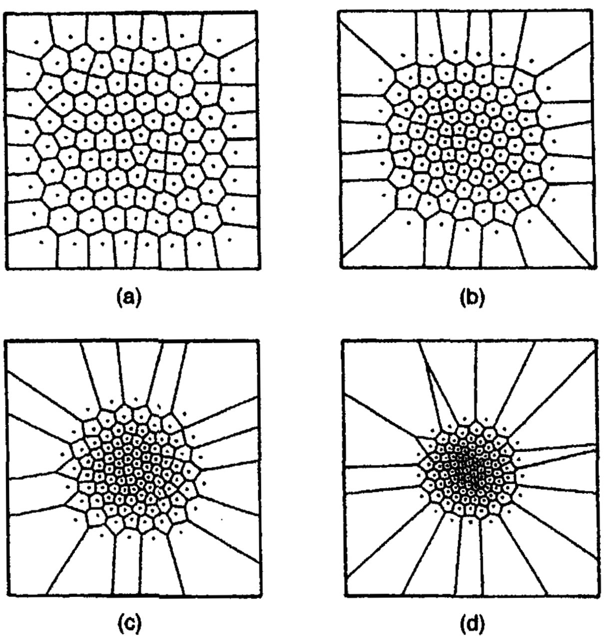


**Figure 9.2.6** Optimal locations of 100 facilities used by a group of (a) $m = 2$ members, (b) $m = 4$ members, (c) $m = 8$ members, (d) $m = 16$ members. (*Source*: Ohsawa and Suzuki, 1987, Figure 3.)

where $\phi_g(\bar{y})$ is the density function of $\bar{y}$ obtained from $\phi(y_1, \ldots, y_m)$, and $c$ is a constant. Note that when $\phi(y_1, \ldots, y_m)$ is uniform over a unit square region, $\phi_g(\bar{y}) = \phi_g(\bar{y}_1, \bar{y}_2)$ is written as

$$\begin{aligned}\phi_g(\bar{y}_1, \bar{y}_2) = \frac{m^2}{\{(m-1)!\}^2} \sum_{i=1}^{[m\bar{y}_1]} \frac{(-1)^i i!\,(m\bar{y}_1 - i)^{m-1}}{m!\,(m-i)!} \\ \times \sum_{j=1}^{[m\bar{y}_2]} \frac{(-1)^j j!\,(m\bar{y}_2 - j)^{m-1}}{m!\,(m-j)!},\end{aligned} \tag{9.2.20}$$

where $[m\bar{y}_i]$ is the maximum integer less than or equal to $m\bar{y}_i$ (Mood *et al.*, 1974).

Comparing equation (9.2.19) with equation (9.2.2), we notice that the objective function of equation (9.2.19) is the same as that of equation (9.2.2) except that $f(\|x - x_i\|^2)$ is replaced by $\|x - x_i\|^2$. Therefore, we can apply the same computational method used for Problem OPT3 to this locational optimization problem. Actually, using that method, Ohsawa and Suzuki (1987) solved the problem for $n = 100$ facilities used by groups of $m = 2, 4, 8, 16$ members. The results are shown in Figure 9.2.6. We notice from Figure 9.2.6 that facilities tend to gather near the centre as the number of members constituting a group increases.

### 9.2.4 Locational optimization of a hierarchical facility

We implicitly assumed above that all facilities supply the same quality service, or that facilities are indifferent, like public mail boxes. This assumption is not always appropriate for some kinds of facilities. For example, in Japan,

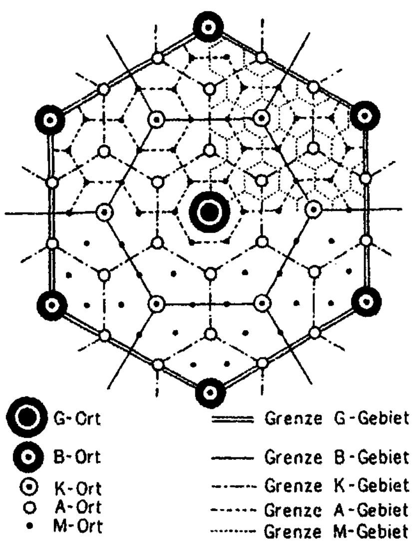



**Figure 9.2.7** Hierarchical system of cities. (*Source*: Christaller, 1933, Figure 2.)

post offices are ranked from the first to the third according to the level of postal service; a university library system usually consists of the main library and a few branch libraries. We call such a set of facilities a *hierarchical facility*.

Besides a hierarchical facility, we also observe a hierarchy in a set of cities in a country. The study of this hierarchy, which has been investigated in spatial economics since Christaller (1933) and Lösch (1940), is worth noting in conjunction with Voronoi diagrams. Christaller (1933) formulates hierarchical systems of cities with Voronoi diagrams generated by triangular lattice points. A few examples are shown in Figures 9.2.7 and 9.2.10 (which will be discussed with the optimal configurations obtained in Figure 9.2.9).

The locational optimization of a hierarchical facility has been studied mainly in Operations Research. Dokmeci (1973) solves this problem for discrete demand with the integer programming. Moore and ReVelle (1982) viewed a hierarchical facility from a covering problem. Tien *et al.* (1983) dealt with the problem of a hierarchical health facility for discrete demand. O'Kelley and Storbeck (1984) took probabilistic allocation into account. Narula (1984) examined the characteristics of several hierarchical facilities and classified the types of hierarchy. Hodgson (1984) solved a multi-level locational problem with the gravity model. Mirchandani (1987) generalized the Tien *et al.* (1983) model.

As is noticed in those articles, most of the models are formulated on a network; few models are formulated on a continuous plane. Recalling the computational method used in Problem OPT3, we notice that the same method can be applied to the above problem. In fact, T. Suzuki (1989) and Okabe *et al.* (1997) have applied that method to the locational optimization problem of a hierarchical facility on a continuous plane. In this subsection we follow T. Suzuki's (1989) model.

Suppose that there are $n$ facilities in a region $S$, and those facilities are ranked from 1 to $m$. Let $n_i$ be the number of facilities of rank $i$, $(\sum_{i=1}^{m} n_i = n)$,

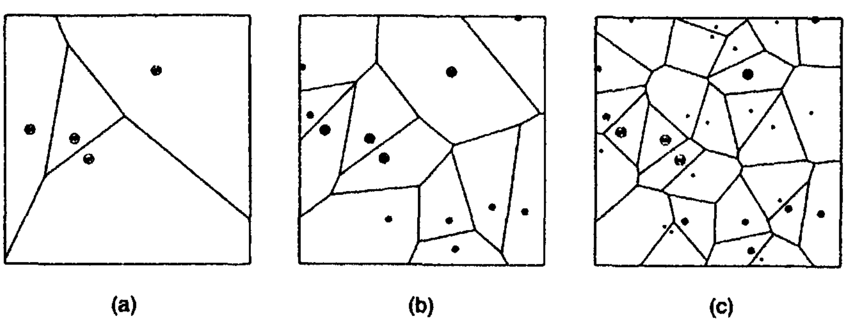



**Figure 9.2.8** A nested hierarchical facility: (a) the Voronoi diagram $\mathcal{V}^{(1)}$ generated by the locations of facilities supplying service $s_1$ (facilities of rank 1); (b) the Voronoi diagram $\mathcal{V}^{(2)}$ generated by the locations of facilities supplying service $s_2$ (facilities of rank 1 and rank 2); (c) the Voronoi diagram $\mathcal{V}^{(3)}$ generated by the locations of facilities supplying service $s_3$ (facilities of ranks 1, 2 and 3).

and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be the locations of $n$ facilities. For convenience, the first $n_1$ locations of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are those of rank 1 facilities; the next $n_2$ locations are those of rank 2 facilities, and so forth.

Suppose next that the $n$ facilities supply $k$ kinds of services, $s_1, \ldots, s_k$. Service $s_j$ may be supplied by all facilities or may be supplied by facilities of certain ranks. First, we consider the case in which facilities of rank $i$ supply services $s_i, \ldots, s_m$, implying that facilities of rank $i$ supply their proper service as well as the services supplied by facilities of lower ranks $i+1, \ldots, m$. This type of hierarchy is called a *successively inclusive hierarchy* (Tien *et al.*, 1983) or a *nested hierarchy* (Banerji and Fisher, 1974).

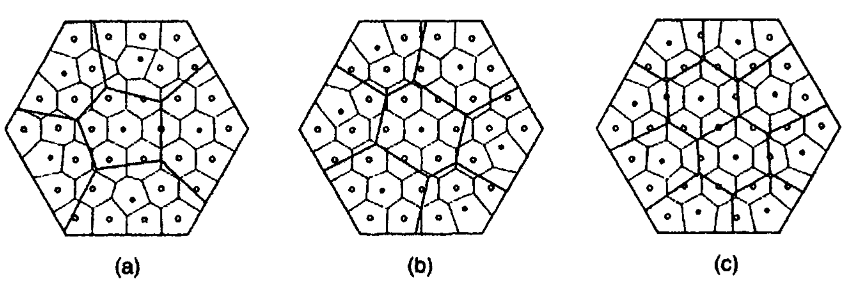
We make two assumptions. First, a user who needs service $s_j$ goes to the nearest facility among the facilities supplying service $s_j$ (the facilities of rank $1, \ldots, j$). As a result, service areas of the facilities supplying service $s_j$ are given by the Voronoi diagram $\mathcal{V}^{(j)} = \{V_1^{(j)}, \ldots, V_{n_1+\ldots+n_j}^{(j)}\}$, generated by $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_1+\ldots+n_j}\}$. An example of $m = 3$ is depicted in Figure 9.2.8. Second, every user consumes $\alpha_j$ units of service $s_j$, $j \in I_k$, per unit time regardless of the Euclidean distance to the facilities. For convenience, the $\alpha_i$'s are standardized as $\sum_{j=1}^{k} \alpha_j = 1$.

Under these assumptions the average travel cost of users over the region $S$ is written as

$$F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{k} \alpha_i \sum_{j=n_1+\ldots+n_{i-1}+1}^{n} \int_{V_j^{(i)}} \| \boldsymbol{x}_j - \boldsymbol{x} \| \phi(\boldsymbol{x})\, d\boldsymbol{x}. \tag{9.2.21}$$

The computational method for minimizing this objective function is the same as that of Problem OPT3. Using that method, T. Suzuki (1989) obtained the optimal configurations for $n = 37, m = 2$ and $\alpha_1 = \alpha_2 = 0.5$. The results are shown in Figure 9.2.8 where $n_1 = 6, 7, 10$.

As is noticed from Figure 9.2.9, the optimal configuration varies according to the ratio of $n_2$ to $n_1$. This ratio, $K$, is an important parameter in Christaller's (1933) model. In Figure 9.2.9, the $K$ values are 5.2 in (a), 4.3 in (b), and 2.7 in (c). Christaller (1933) refers to $K = 3, 4, 7$ (Figure 9.2.10) as the hierarchical systems resulting from the market principle, the transportation principle, and the administrative principle, respectively. Since Christaller



(a) (b) (c)

**Figure 9.2.9** Optimal configurations of a hierarchical facility, $n = 37, m = 2$, $\alpha_1 = \alpha_2 = 0.5$: (a) $n_1 = 6$, (b) $n_1 = 7$, (c) $n_1 = 10$. (*Source*: T. Suzuki, 1989, Figure 4.10.)

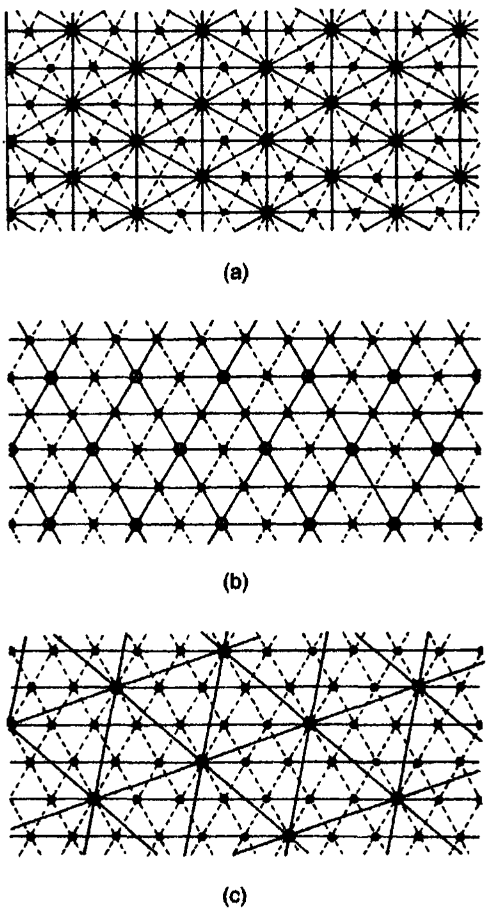



**Figure 9.2.10** Christaller's (1933) hierarchical systems of cities: (a) $K = 3$; (b) $K = 4$; (c) $K = 7$.

(1933) formulated the model on an unbounded plane, whereas T. Suzuki (1989) formulated it on a bounded plane, we cannot directly compare their results. Figure 9.2.9, however, provides a new look at hierarchical systems of cities from an optimization viewpoint (see also Okabe and Sadahiro, 1996, who studied the problem from a statistical viewpoint).

In the above model, optimization is achieved with respect to the locations of facilities provided that the number of rank $i$ facilities are given. We may relax this condition and optimize not only the locations but also the number of rank $i$ facilities, provided that the total number of all rank facilities is fixed. This relaxation is made by Okabe *et al.* (1997).

A hierarchy is not always nested; it may be partially nested or successively exclusive. We can formulate the optimization problems of such a hierarchical facility almost in the same manner as in the above, and can solve them with the descent method. The reader who is interested in this problem should consult T. Suzuki (1989).

### 9.2.5 Locational optimization of observation points for estimating the total quantity of a spatial variable continuously distributed over a plane

As we mentioned in Chapter 1, the Voronoi diagram is alternatively called the Thiessen diagram. The original use of the Thissen diagram was to estimate the total precipitation in a region $S$. To be explicit, suppose that there are $n$ pluviometers placed at $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ in the region $S$. Let $z(\boldsymbol{x}_i)$ be the precipitation at $\boldsymbol{x}_i$, and $V_i$ be the Voronoi polygon associated with a pluviometer at $\boldsymbol{x}_i$. Thiessen (1911) estimated the true total precipitation in the region $S$, i.e.

$$Z = \int_S z(\boldsymbol{x})\, d\boldsymbol{x} = \sum_{i=1}^{n} \int_{V_i} z(\boldsymbol{x})\, d\boldsymbol{x}, \tag{9.2.22}$$

by

$$\hat{Z} = \sum_{i=1}^{n} |V_i|\, z(\boldsymbol{x}_i). \tag{9.2.23}$$

In Thiessen (1911), the locations of pluviometers are fixed, and no locational optimization problem is discussed (see also Lebel *et al.*, 1987). However, if we can freely choose those locations, we have a locational optimization problem to minimize the estimation error. A similar problem is considered by Hori and Nagata (1985), who obtain the optimal configuration of monitor points to estimate $NO_x$ in a region.

To formulate the above optimization problem, let $z(\boldsymbol{x})$ be a random variable at $\boldsymbol{x}$ in a region $S$ given by

$$z(\boldsymbol{x}) = m(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}), \tag{9.2.24}$$

where $m(\boldsymbol{x})$ is a continuous function of $\boldsymbol{x}$, which is constant for a given $\boldsymbol{x}$, and $\varepsilon(\boldsymbol{x})$ is a random variable whose expected value is given by $\mathrm{E}[\varepsilon(\boldsymbol{x})] = 0$. From equation (9.2.24) and $\mathrm{E}[\varepsilon(\boldsymbol{x})] = 0$, we obtain $\mathrm{E}[z(\boldsymbol{x})] = m(\boldsymbol{x})$, which indicates the average trend of the variable $z$ over $S$. To measure the estimation error, we consider the expected squared error over the region $S$, i.e.

$$F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \mathrm{E}\left[\sum_{i=1}^{n} \int_{V_i} \{z(\boldsymbol{x}) - z(\boldsymbol{x}_i)\}^2\, d\boldsymbol{x}\right]. \tag{9.2.25}$$

Upon substituting equation (9.2.24) into equation (9.2.25), we obtain

$$\begin{aligned} F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \int_{V_i} & \mathrm{E}\left[\{m(\boldsymbol{x}) - m(\boldsymbol{x}_i)\}^2\right] \\ & + \mathrm{E}\left[\varepsilon(\boldsymbol{x})^2\right] + \mathrm{E}\left[\varepsilon(\boldsymbol{x}_i)^2\right] - 2\mathrm{E}\left[\varepsilon(\boldsymbol{x})\,\varepsilon(\boldsymbol{x}_i)\right] \\ & + 2\mathrm{E}\left[\varepsilon(\boldsymbol{x})\{m(\boldsymbol{x}) - m(\boldsymbol{x}_i)\}\right] + 2\mathrm{E}\left[\varepsilon(\boldsymbol{x}_i)\{m(\boldsymbol{x}) - m(\boldsymbol{x}_i)\}\right] d\boldsymbol{x}. \end{aligned} \tag{9.2.26}$$

The first term, $\mathrm{E}[\{m(\boldsymbol{x}) - \mathrm{m}(\boldsymbol{x}_i)\}^2]$, is written as $\{m(\boldsymbol{x}) - m(\boldsymbol{x}_i)\}^2$ because $m(\boldsymbol{x})$ is constant for a given $\boldsymbol{x}$. The second and third terms are variances, which are denoted by $\sigma(\boldsymbol{x})$ and $\sigma(\boldsymbol{x}_i)$, respectively. The fourth term is written as $\mathrm{E}[\varepsilon(\boldsymbol{x})\,\varepsilon(\boldsymbol{x}_i)] = \sigma(\boldsymbol{x})\sigma(\boldsymbol{x}_i)\ \mathrm{Cov}\,(\boldsymbol{x},\boldsymbol{x}_i)$ where $\mathrm{Cov}\,(\boldsymbol{x},\boldsymbol{x}_i)$ is the covariance. The fifth and sixth terms are zero, because $\mathrm{E}[\varepsilon(\boldsymbol{x})\ \{m(\boldsymbol{x}) - \mathrm{m}(\boldsymbol{x}_i)\}] = \{m(\boldsymbol{x}) - \mathrm{m}(\boldsymbol{x}_i)\}\ \mathrm{E}[\varepsilon(\boldsymbol{x})] = 0$. Thus, equation (9.2.26) is written as

$$F(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) = \sum_{i=1}^{n}\int_{V_i}\Big[\{m(\boldsymbol{x}) - m(\boldsymbol{x}_i)\}^2 + \{\sigma(\boldsymbol{x}) - \sigma(\boldsymbol{x}_i)\}^2 \\ + 2\sigma(\boldsymbol{x})\,\sigma(\boldsymbol{x}_i)\,\{1 - \mathrm{Cov}\,(\boldsymbol{x},\boldsymbol{x}_i)\}\Big]\,d\boldsymbol{x}. \tag{9.2.27}$$

The locational optimization problem is hence to minimize $F$ of equation (9.2.27) with respect to $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$.

First, let us consider the case in which the variance $\sigma(\boldsymbol{x})$ is independent of $\boldsymbol{x}$, and constant for all points in the region $S$, i.e. $\sigma(\boldsymbol{x}) = \sigma$; the change in the average trend, $m(\boldsymbol{x})$, is small compared with the variance $\sigma$. In this case, equation (9.2.27) is approximated by

$$F(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) = \sum_{i=1}^{n}\int_{V_i} 2\sigma^2\{1 - \mathrm{Cov}\,(\boldsymbol{x},\boldsymbol{x}_i)\}\,d\boldsymbol{x}. \tag{9.2.28}$$

Empirically, the covariance $\mathrm{Cov}\,(\boldsymbol{x},\boldsymbol{x}_i)$ tends to decrease monotonically as $\|\boldsymbol{x} - \boldsymbol{x}_i\|$ increases, and so $1 - \mathrm{Cov}\,(\boldsymbol{x},\boldsymbol{x}_i)$ increases monotonically as $\|\boldsymbol{x} - \boldsymbol{x}_i\|$ increases. Let $f(\|\boldsymbol{x} - \boldsymbol{x}_i\|)$ be this increasing function. Then, the optimization problem reduces to

$$\min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n}\sum_{i=1}^{n}\int_{V_i} 2\sigma^2\{1 - \mathrm{Cov}\,(\boldsymbol{x},\boldsymbol{x}_i)\}\,d\boldsymbol{x} = \min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n}\sum_{i=1}^{n}\int_{V_i} f(\|\boldsymbol{x} - \boldsymbol{x}_i\|)\,d\boldsymbol{x}. \tag{9.2.29}$$

We readily notice from equation (9.2.2) that the problem given by expression (9.2.29) is a special case of Problem OPT3.

Let us next consider the case in which the variance $\sigma(\boldsymbol{x})$ is small compared with the change in the average trend, $m(\boldsymbol{x})$. In this case, equation (9.2.27) is approximated by

$$F(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) = \sum_{i=1}^{n}\int_{\boldsymbol{x}\in V_i}\{m(\boldsymbol{x}) - m(\boldsymbol{x}_i)\}^2\,d\boldsymbol{x}. \tag{9.2.30}$$

The locational optimization problem is to minimize this function with respect to $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$. The exact solution of this minimization problem is difficult to obtain for a general function, $m(\boldsymbol{x})$. We may, however, obtain a solution which is possibly close to the optimal solution from the following relation. For a given $V_i$, the following relation holds:

$$\int_{V_i}\{m(\boldsymbol{x}) - m(\boldsymbol{x}_i)\}^2\,d\boldsymbol{x} \geq \int_{V_i}\{m(\boldsymbol{x}) - m_i\}^2\,d\boldsymbol{x}, \tag{9.2.31}$$

where

$$m_i = \frac{1}{|V_i|}\int_{V_i} m(\boldsymbol{x})\,d\boldsymbol{x}. \tag{9.2.32}$$

We notice from this relation that if $m(\boldsymbol{x}_i)$ happens to be equal to $m_i$, that is, if the value of $m(\boldsymbol{x}_i)$ at the generator $\boldsymbol{x}_i$ of the Voronoi polygon $V_i$ happens to be equal to the average value of $m(\boldsymbol{x})$ over $V_i$, the value on the left-hand

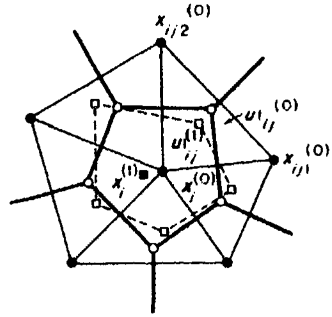



**Figure 9.2.11** A heuristic method for searching for $m(\boldsymbol{x}_i) = m_i$.

side of equation (9.2.31) achieves the minimum value. This property suggests that the solution may be given by a set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ that satisfies $m(\boldsymbol{x}_i) = m_i$ (or $m(\boldsymbol{x}_i) \approx m_i$ in practice). To find such a set, Hori and Nagata (1985) propose the following heuristic or crude method.

First, we construct the Voronoi diagram $\{V_1^{(0)}, \ldots, V_n^{(0)}\}$ (the heavy solid lines in Figure 9.2.11) generated by an initial generator set $\{\boldsymbol{x}_1^{(0)}, \ldots, \boldsymbol{x}_n^{(0)}\}$ (the filled circles in Figure 9.2.11). Hori and Nagata (1985) recommend using the initial generator set given by the optimal solution of Problem OPT3, where $f(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2) = \|\boldsymbol{x} - \boldsymbol{x}_i\|^2$, $\phi(\boldsymbol{x}) = m(\boldsymbol{x})$. Second, we construct the Delaunay triangulation for this Voronoi diagram (the light solid lines in

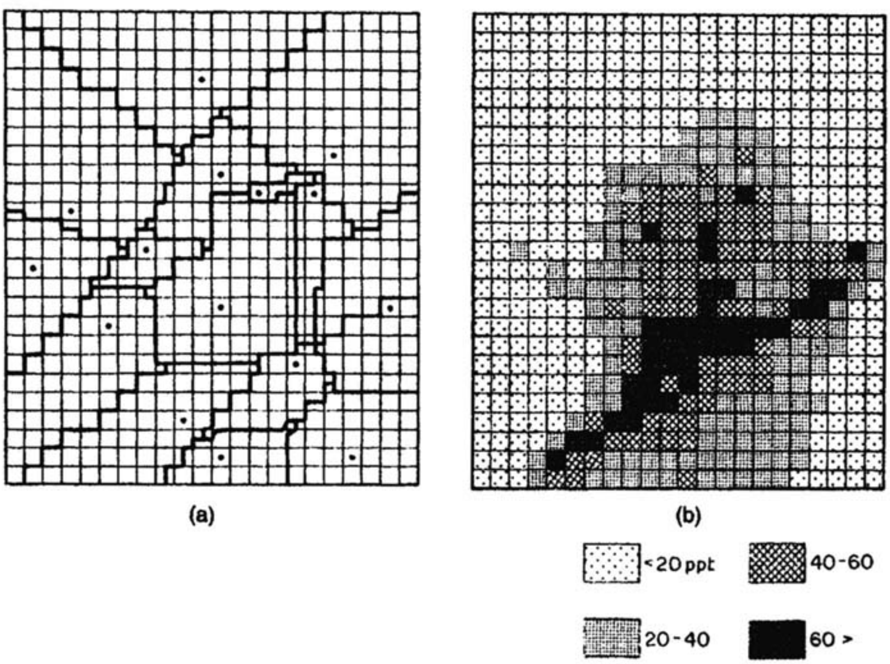



**Figure 9.2.12** (a) The optimal configuration of 16 observation points, and (b) density of $NO_x$ in Kyoto. (*Source*: Hori and Nagata, 1985, redrawn from Figures 6 and 7.)

Figure 9.2.11). Let $u_{ij}^{(0)}, j \in I_{n_i}$, be the Voronoi points of the Voronoi polygon $V_i^{(0)}$, where $n_i$ is the number of Voronoi vertices of $V_i^{(0)}$ (the unfilled circles in Figure 9.2.11); $T_{ij}^{(0)}$ be a Delaunay triangle which corresponds to the vertex $u_{ij}^{(0)}$; and $x_{ij1}^{(0)}$ and $x_{ij2}^{(0)}$ be two vertices of $T_{ij}^{(0)}$ other than $x_i^{(0)}$. Third, we find a point $u_{ij}^{(1)}$ in $T_{ij}^{(0)}$ or in the circumcircle of $T_{ij}^{(0)}$ that satisfies $m(u_{ij}^{(1)}) = \{m(x_i^{(0)}) + m(x_{ij1}^{(0)}) + m(x_{ij2}^{(0)})\}/3$ (if such a point does not exist, we find a point $u_{ij}^{(0)}$ in such a way that $m(u_{ij}^{(1)})$ is close to $\{m(x_i^{(0)}) + m(x_{ij1}^{(0)}) + m(x_{ij2}^{(0)})\}/3$) (the unfilled squares in Figure 9.2.11). Fourth, we construct the polygon $V_i'$ whose vertices are given by $u_{ij}^{(1)}, j \in I_{n_i}$ (the broken lines in Figure 9.2.11). Fifth, in this polygon we find a point $x_i^{(1)}$ that satisfies equation (9.2.33) where $m_i = m(x_i^{(1)})$, $V_i = V_i'$ (if we cannot find such a point, we find a point $x_i^{(1)}$ in such a way that $m(x_i^{(1)})$ is close to the left-hand-side of equation (9.2.33)) (the filled square in Figure 9.2.11). Last, we generate a Voronoi diagram with $\{x_1^{(1)}, \ldots, x_n^{(1)}\}$. We continue this procedure until $x_i^{(k)}, i \in I_n$ converges. It should be noted, however, that this procedure does not guarantee convergence.

According to Hori and Nagata (1985), the variance in the density of $NO_x$ is fairly small. Hori and Nagata (1985) hence optimize the objective function given by equation (9.2.31) with the above heuristic method. Figure 9.2.12(a) shows the near-optimal configuration of 16 monitor points placed in Kyoto provided that $m(x)$ is given by Figure 9.2.12(b).

### 9.2.6 Locational optimization of service points of a mobile facility

Facilities are not always fixed at the same locations. Some kinds of facilities are mobile and stop at several points in a region to provide service. An example is a bookmobile (Takeda, 1985). A bookmobile offers a library service at some fixed points in a region for a certain length of time. Library users go to their nearest service points when the bookmobile stops at those points. If the bookmobile could stop at every house, the user's convenience would be maximum. Because of a time constraint, however, the sum of the library service time and travel time should be not greater than a certain limit. The locational optimization problem is hence to minimize the average distance (or cost) of users to their nearest service points provided that the number of service points and the total travel distance of the bookmobile are given.

The mathematical formulation of this locational optimization problem is almost the same as that of Problem OPT3 except for constraints. The objective function is given by equation (9.2.1). If we assume that the total travel distance of the bookmobile is given by the sum of the Euclidean distance between the successive service points, the bookmobile problem is formulated as follows.

**Problem OPT4 (minimization of the average travel cost to the nearest points provided that the length of a line successively connecting those service points is not greater than a certain limit)**

$$\min_{x_1,\ldots,x_n} \sum_{i=1}^{n} \int_{V_i} f(\|x - x_i\|^2)\, \phi(x)\, dx, \qquad (9.2.33)$$

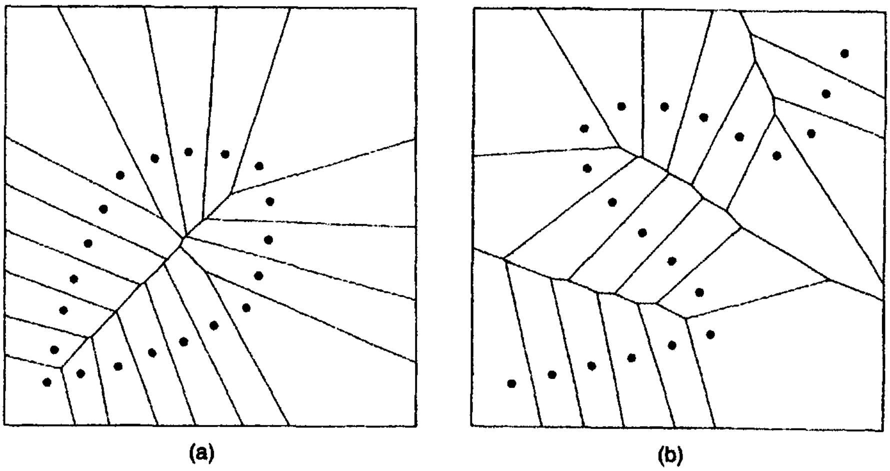



**Figure 9.2.13** Locational optimization of $n = 20$ service points: (a) $d^{**} = 0.1$, $x_1^T = (0.1, 0.1)$, $x_{20}^T = (0.9, 0.9)$; (b) $d^{**} = 0.1$, $x_1^T = x_{20}^T = (0.15, 0.15)$. (*Source*: Takeda, 1985, Figure 5.3.)

subject to

$$\sum_{i=1}^{n-1} d(x_i, x_{i+1}) \leq d^*. \tag{9.2.34}$$

An alternative constraint is that every distance between service points is not greater than $d^{**} = d^*/n$, i.e.

$$d(x_i, x_{i+1}) \leq d^{**}, \quad i = 1, \ldots, n-1. \tag{9.2.35}$$

Obviously, this constraint is stronger than the constraint of equation (9.2.34).

We may also add the constraint that the starting and finishing points are fixed, or that those points are the same and fixed, i.e.

$$x_1 = c, \quad x_n = c'(c \neq c') \quad \text{or} \quad x_1 = x_n = c. \tag{9.2.36}$$

In this case, the variables in expression (9.2.33) are $x_2, \ldots, x_{n-1}$.

Problem OPT4 is a constrained non-linear, non-convex programming problem. As we showed in Section 9.1, we can solve this problem using the penalty function method. Note that when $n$ is large, the penalty function method is not efficient. Alternatively, we may use the multiplier method.

Figure 9.2.13 shows the optimal configuration of $n = 20$ service points obtained by Takeda (1985).

### 9.2.7 Locational optimization of terminal points through which users go to the central point

On a network we have terminal facilities, such as stations on a railway network, bus stops on a bus route, access points of a communication network,

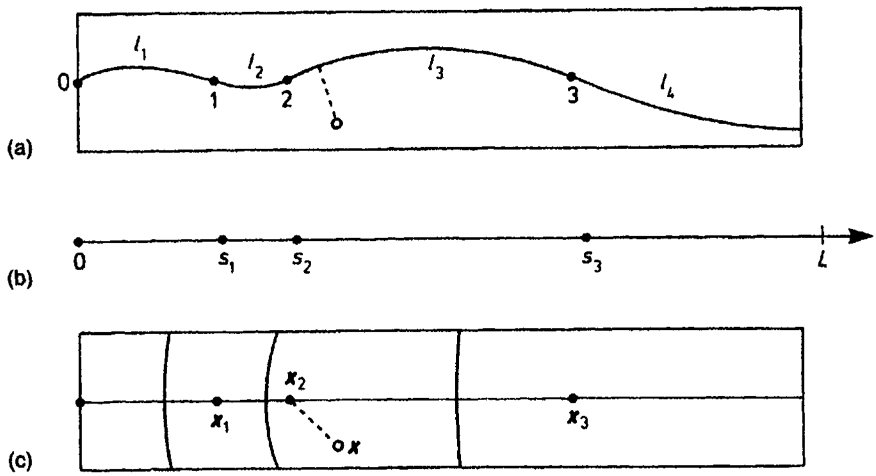



**Figure 9.2.14** Bus stops on a bus route.

and so forth. Such networks sometimes have central points. For example, consider a bus route in a residential area. Commuters walk from their homes to bus stops, and take a bus to a railway station, from which they go to their places of work. In this case the central point is the railway station at which a bus terminates. Obviously, commuters want to reach the station as fast as possible. The problem is thus to seek a set of the locations of bus stops that minimizes the average travel time of commuters. This type of optimization problem is examined by Lesley (1976), Vaughan and Cousins (1977), Wirasinghe and Ghoneim (1981), and T. Suzuki (1987), among others.

To formulate the above problem mathematically, let us consider a bus route, $L$, in a region $S$ (Figure 9.2.14), on which $n$ bus stops are placed. Those bus stops are indexed from 1 to $n$ from the station. Let $l_i$ be the route distance from bus stop $i$ to bus stop $i+1$, $i = 0, 1, \ldots, n$. Note that bus stop 0 is fixed at the station, and $l_{n+1}$ is the distance from bus stop $n$ to the end of the bus route $L$ from which a bus departs (a bus garage). The location of a point on the route $L$ can be represented by a route distance, $s$, from the station. The location of bus stop $i$ is hence indicated by $s = s_i = \sum_{j=1}^{i} l_j$. Commuters are distributed over $S$ according to a density function $\phi(x)$.

First, we assume that every commuter walks from his/her home (the unfilled circle in Figure 9.2.14(a)) to the nearest point (called an access point) on the bus route $L$, and then walks from this access point to a bus stop along the bus route $L$. Under this assumption, the choice of a bus stop depends upon only the distances from the access point to alternative bus stops; the choice does not depend upon the distance from his/her home to the access point. Hence, the two-dimensional problem reduces to a one-dimensional problem (Figure 9.2.14(b)). To be explicit, let $\phi_L(s)$ be the density of commuters whose nearest point on the bus route $L$ from their homes is a point $s$. Once $\phi_L(s)$ is obtained from $\phi(x)$ (the density of commuters on a plane), the problem becomes the problem of optimizing $n$ bus stops on the line segment $L$ over which commuters are distributed with the density function $\phi_L(s)$.

Let $v_b$ be the speed of a bus, $v_w$ be walking speed, and $b$ be the time spent at every bus stop. We assume that every commuter has a time schedule and arrives at a bus stop when a bus just reaches it. Then, the total travel time from a point $s$ to bus stop $i$ on foot and from this bus stop to the station by bus is given by $|s - s_i|/v_w + s_i/v_b + ib$. Obviously, the relation

$$b + \frac{l_i}{v_b} < \frac{l_i}{v_w} \tag{9.2.37}$$

should be satisfied; otherwise, walking is faster than taking a bus. Since every commuter minimizes the travel time, commuters at the point $s$ choose the bus stop that gives the minimum travel time, i.e. $\min_i\{|s - s_i|/v_w + s_i/v_b + ib\}$. As a result, every bus stop has its catchment area, which is given by

$$V_i = \left\{ s \mid \frac{|s - s_i|}{v_w} + \frac{s_i}{v_b} + ib \le \frac{|s - s_j|}{v_w} + \frac{s_j}{v_b} + jb, \text{ for } j \in I_n \backslash \{i\} \right\}. \tag{9.2.38}$$

As we defined in Chapter 3, this region is the weighted Voronoi region generated on the line $L$.

From equation (9.2.38), the total travel time of all commuters in the region $L$ (or $S$) is given by

$$F(l_1, \ldots, l_n) = \sum_{i=0}^{n} \int_{V_i} \left\{ \frac{|s - s_i|}{v_w} + \frac{s_i}{v_b} + ib \right\} \phi_L(s)\, \mathrm{d}s. \tag{9.2.39}$$

The optimization problem is hence written as

$$\min_{l_1, \ldots, l_n} \sum_{i=0}^{n} \int_{V_i} \left\{ \frac{|s - s_i|}{v_w} + \frac{s_i}{v_b} + ib \right\} \phi_L(s)\, \mathrm{d}s, \tag{9.2.40}$$

subject to

$$b + \frac{l_i}{v_b} < \frac{l_i}{v_w} \quad \text{for } i \in I_n, \tag{9.2.41}$$

$$\sum_{i=1}^{n} l_i \le L, \tag{9.2.42}$$

$$l_i > 0 \quad \text{for } i \in I_{n+1}. \tag{9.2.43}$$

This problem is a constrained non-linear programming problem, which can be solved, as shown in Section 9.1, by use of the penalty function method. When the density function $\phi_L(s)$ is the uniform distribution, the objective function is explicitly written as

$$F(l_1,\ldots,l_n)=\frac{1}{4v_b^2v_w}\left(v_b^2+2v_bv_w-v_w^2\right)\sum_{i=1}^{n}l_i+\frac{1}{v_b}\sum\sum_{i\neq j}l_il_j$$
$$+b\sum_{i=1}^{n}(i-1)l_i+\frac{v_b-v_w}{2v_b}b\sum_{i=1}^{n}l_i-\frac{v_wb^2}{4}n \tag{9.2.44}$$
$$+\frac{l_{n+1}}{v_b}\sum_{i=1}^{n}l_i+nbl_{n+1}+\frac{l_{n+1}^2}{2v_w}.$$

In this case, the problem becomes a constrained quadratic programming problem (a special case of the constrained non-linear programming problem).

In the above, we assumed that commuters first go to the nearest point on the route $L$. Instead, if we assume that commuters take the Euclidean path to a bus stop (the broken line in Figure 9.2.14(c)), the problem cannot be reduced to the one-dimensional problem; the problem should be formulated on a plane. Essentially, the formulation is the same as in the one-dimensional case. In place of equation (9.2.38), the catchment area of the bus stop at $s_i$ is given by

$$V_i=\left\{x \mid \frac{\|x-x_i\|}{v_w}+\frac{s_i}{v_b}+ib\leq\frac{\|x-x_j\|}{v_w}+\frac{s_j}{v_b}+jb,\ j\neq i,\ j\in I_n\right\}. \tag{9.2.45}$$

An example is shown in Figure 9.2.14(c). The objective function is thus given by

$$F(l_1,\ldots,l_n)=\sum_{i=0}^{n}\int_{V_i}\left\{\frac{\|x-x_i\|}{v_w}+\frac{s_i}{v_b}+ib\right\}\phi(x)\,dx. \tag{9.2.46}$$

The constraints are the same as those given by relations (9.2.37), (9.2.41), (9.2.42) and (9.2.43).

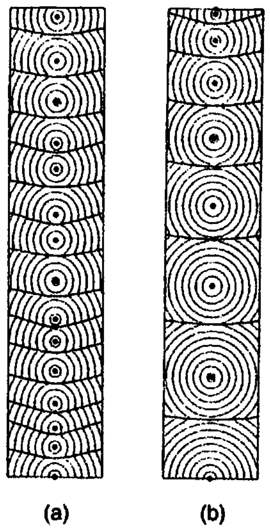



**Figure 9.2.15** Locational optimization of bus stops on the Motoyahata–Takatsuka line: (a) the present location of bus stops; (b) the optimal location of bus stops. (*Source*: T. Suzuki, 1987, Figure 8.)

The calculation of the integral in equation (9.2.46) is fairly complex, but we can carry it out in practice. In particular, when the density $\phi(x)$ is uniform, this integral is explicitly obtained. The result is, however, extremely lengthy, and so it is omitted here (see T. Suzuki, 1987, who obtained the results with the help of REDUCE-III).

T. Suzuki (1987) applies the above optimization problem to the Motoyahata–Takatsuka bus line in Ichikawa. This line is chosen because it is mainly used for commuting to Motoyahata station; the bus route is almost straight; and the population density $\phi(x)$ is fairly uniform. From the empirical survey, the parameter values are observed as $v_b = 8.00$ m/s, $v_w = 1.20$ m/s, $b = 30$ s, $L = 4865.7$ m. T. Suzuki (1987) obtained the optimal solutions for $n = 2, \ldots, 34$. These solutions indicate that the total travel time is minimized when $n = 8$. The optimal locations of the eight bus stops are depicted in Figure 9.2.15(b). It is of interest to compare this figure with the actual locations of bus stops shown in Figure 9.2.15(a). The travel time in the optimal solution is shorter than the actual travel time by 2 min on average. Obviously, the optimal solution depends upon an objective function. Instead of minimizing the total travel time, we may maximize the profit of a bus company. This alternative optimization (including other criteria) is discussed in T. Suzuki (1987).

### 9.2.8 Locational optimization of points on a continuous network

In the preceding subsections we dealt with locational optimization problems on a continuous plane. In this subsection we deal with those on a continuous network (i.e. point-like facilities are locatable on any point on the network (including nodes)).

In Section 3.8.2 we introduced the network Voronoi link-diagram, $\mathcal{V}_{\text{link}} = \{V_{\text{link}}(p_1), \ldots, V_{\text{link}}(p_n)\}$ generated by $\{p_1, \ldots, p_n\}$ on a network $\mathcal{N}(N, L)$, where $N$ denotes the set of nodes and $L$ denotes the set of links forming the network. To measure the Voronoi link set $V_{\text{link}}(p_i)$, Hakimi *et al.* (1992) define the 'size' of $V_{\text{link}}(p_i)$. To be explicit, let $w(q_i)$ be the weight of the $i$th node in $N$. Then the *size*, $s(V_{\text{link}}(p_i))$, of $V_{\text{link}}(p_i)$ is defined by $s(V_{\text{link}}(p_i)) = \sum_{q_i \in V_{\text{link}}(p_i)} w(q_i)$. In terms of $s(V_{\text{link}}(p_i))$, we consider the following problem.

**Problem OPT5 (Voronoi $p$-centre problem)** Find a set $P$ of points such that

$$\min_{p_1, \ldots, p_n} \max\left\{s\left(V_{\text{link}}(p_1)\right), \ldots, s\left(V_{\text{link}}(p_n)\right)\right\}. \tag{9.2.47}$$

Hakimi *et al.* (1992) show that the Voronoi $p$-centre problem is NP-hard (Section 1.3.4). They also develop an algorithm to solve this problem.

In the study of market area delineation, one of the most popular models is the Huff model (1963). Originally this model assumed the Euclidean distance. This assumption, however, is not always acceptable when we deal with market area delineation in a small district, such as the market areas of fast food stores in a down town district. To deal with such market area delineation, we use the Huff model on a continuous network of streets (Miller, 1994; Okabe and Kitamura, 1996). Okabe and Okunuki (1999) attempt to optimize the location of one store to maximize its profit provided that the other stores are fixed and the choice behaviour of consumers is given by the Huff model.

## 9.3 LOCATIONAL OPTIMIZATION OF LINES

In the previous section we discussed the locational optimization of points. In this section, extending the method developed there, we consider the locational optimization of lines. We deal with three types of lines: a service route (Section 9.3.1), a network (Section 9.3.2) and an Euclidean Steiner minimum tree (Section 9.3.3).

### 9.3.1 Locational optimization of a service route

We first consider the locational optimization of a service route. Generally, the problem is to seek an optimal service route of a mobile facility that provides some service at any point on the service route. Such mobile services are numerous in Japan. The well-known example is a wagon collecting old newspapers. The wagon travels a route calling with a loudspeaker, 'we are exchanging old newspapers for tissue paper'. Hearing that call, a resident who wants to sell old newspapers goes to the nearest service route to catch the wagon. The wagon then comes to the resident's home and the resident gets tissue paper in exchange for old newspapers. More traditional examples in Japan are an ice-cream vendor riding on a bicycle with a bell, a bean card (tofu) vendor pulling a cart blowing a bugle, a fermented soybeans (natto) vendor calling 'natto' with a resonant voice, and so forth. The problem of these vendors is to find the optimal service route in a region that yields the maximum profit.

To discuss the above problem more precisely, let us consider, for example, an optimal service route of an ice-cream vendor. We suppose, for simplicity, that only one ice-cream vendor exclusively sells ice-cream in a region, and that a service route can be freely placed in a region (a road network is so dense that the route can be almost freely placed in a region). The starting point and the finishing point of the service route may be the same or may not. We assume that every customer buys ice-cream at the nearest point on the service route; the demand for ice-cream is inversely proportional to the average travel cost from all customers to their nearest points on the service route; the ice-cream vendor can satisfy all demand; and the marginal cost of a unit amount of ice-cream is constant. Under these assumptions, the profit maximization problem reduces to the problem of minimizing the average travel cost. Obviously, as the length of the service route becomes long, the average travel cost becomes shorter, and the profit increases. Because of a time constraint, however, the vendor cannot walk infinitely; the travel distance in a day is limited. The problem is hence to determine the service route to minimize the average travel cost provided that the length of the service route is not greater than a certain limit.

To formulate the above problem mathematically, we represent a service route in a region $S$ by the chain of straight line segments (a curved line is approximated by the chain of small straight line segments). Let $x_1, \dots, x_n$ be the end points of the straight line segments; $\overline{x_i x_{i+1}}$ be the open straight line segment connecting points $x_i$ and $x_{i+1}$; and $\{V(x_1), \dots, V(x_n), V(\overline{x_1 x_2}), \dots, V(\overline{x_{n-1} x_n})\}$ be the line Voronoi diagram generated by $\{x_1, \dots, x_n, \overline{x_1 x_2}, \dots, \overline{x_{n-1} x_n}\}$ (Section 3.5). From the definition of the line Voronoi diagram, if a point $x$ is in the Voronoi region $V(x_i)$, the Euclidean distance from the point $x$ to the nearest point on the service route is given by $\|x - x_i\|$; if a point $x$ is in the Voronoi region $V(\overline{x_i x_{i+1}})$, the Euclidean distance $d_l(x, \overline{x_i x_{i+1}})$ from the point $x$ to the nearest point on the service route is given by the Euclidean distance from the point $x$ to the line segment $\overline{x_i x_{i+1}}$, which is explicitly given by

$$d_l(x, \overline{x_i x_{i+1}}) = \left[ \| x - x_i \|^2 - \frac{\{(x - x_i)^{\mathrm{T}} (x_{i+1} - x_i)\}^2}{\| x_{i+1} - x_i \|^2} \right]^{1/2}. \tag{9.3.1}$$

Let $\phi(x)$ be the demand density at $x$; and $f(\|x - x_i\|^2)$ be a strictly increasing travel cost function of the squared Euclidean distance, $\|x - x_i\|^2$. Note that the total demand is assumed to be unity without loss of generality.

In these terms, the average travel cost to the service route is written as

$$\begin{aligned} F(x_1, \dots, x_n) = &\sum_{i=1}^{n} \int_{V(x_i)} f(\| x - x_i \|^2)\, \phi(x)\, dx \\ &+ \sum_{i=1}^{n-1} \int_{V(\overline{x_i x_{i+1}})} f\left(d_l(x, \overline{x_i x_{i+1}})^2\right) \phi(x)\, dx. \end{aligned} \tag{9.3.2}$$

When $f(\|x - x_i\|^2) = \|x - x_i\|$, and the demand density is uniform, i.e. $\phi(x) = 1/S$ over $S$, we can explicitly obtain the value of equation (9.3.2) for given $x_1, \dots, x_n$ (see Section 8.3).

With the above objective function, we can formulate the optimal service route problem as follows.

**Problem OPT6 (minimization of the average travel cost to the nearest point on a service line provided that the length of the service line is not greater than a certain limit)**

$$\begin{aligned} \min_{x_1, \dots, x_n} \Bigg[ &\sum_{i=1}^{n} \int_{V(x_i)} f(\| x - x_i \|^2)\, \phi(x)\, dx \\ &+ \sum_{i=1}^{n-1} \int_{V(\overline{x_i x_{i+1}})} f\left(d_l(x, \overline{x_i x_{i+1}})^2\right) \phi(x)\, dx \Bigg] \end{aligned} \tag{9.3.3}$$

subject to

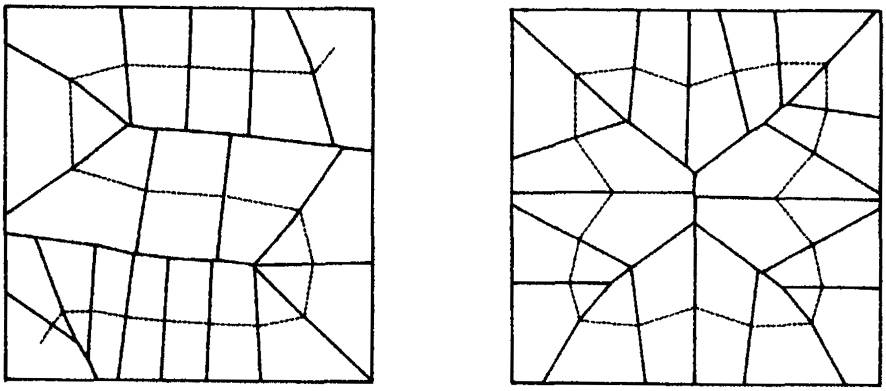
$$\sum_{i=1}^{n-1} \| x_i - x_{i+1} \| \leq d^*. \tag{9.3.4}$$

We may add the constraints $x_1 = c, x_n = c'$, or $x_1 = x_n = c$, where $c$ and $c'$ ($c \neq c'$) are constants. In this case the variables in expression (9.3.3) are $x_2, \ldots, x_{n-1}$.

Like Problem OPT3, Problem OPT6 is a non-linear, non-convex programming problem. To solve this problem, as mentioned in Section 9.1, we should use a numerical method, such as the penalty function method. If we adopt the steepest descent method in the unconstrained programming problem transformed from the constrained programming problem of Problem OPT3 (recall Section 9.1.3), we need the first derivative. The derivation of the first derivative is almost the same as was shown in Problem OPT3 (equation (9.2.2)). The slight move of $x_i$ produces the changes in the Voronoi regions $V(x_i)$, $V(\overline{x_{i-1} x_i})$ and $V(\overline{x_i x_{i+1}})$, and their adjacent Voronoi regions. Since the changes in the integral domains are cancelled out (recall the derivation from equation (9.2.3) to equation (9.2.11)), the first derivative is obtained from the first-order derivative of the integrand,

$$\begin{aligned} \frac{\partial F}{\partial x_{i\kappa}} &= \int_{V(x_i)} \frac{\partial}{\partial x_{i\kappa}} f(\| x - x_i \|^2)\, \phi(x)\, dx \\ &+ \int_{V(\overline{x_{i-1} x_i})} \frac{\partial}{\partial x_{i\kappa}} f(d_l(x, \overline{x_{i-1} x_i})^2)\, \phi(x)\, dx \\ &+ \int_{V(\overline{x_i x_{i+1}})} \frac{\partial}{\partial x_{i\kappa}} f(d_l(x, \overline{x_i x_{i+1}})^2)\, \phi(x)\, dx, \quad \kappa = 1,2. \end{aligned} \tag{9.3.5}$$

Since $d_l(x, \overline{x_i x_{i+1}})$ is explicitly given by equation (9.3.1), and the function $f(x)$ is assumed to be differentiable, we can calculate the integrals in equation



(a) (b)

**Figure 9.3.1** Locally optimal service routes in a unit square. Parameter values are: $d^* = \sqrt{2}/10$, $\phi(x) = 1$, $f(\|x - x_i\|^2) = \|x - x_i\|^2$; $n = 20$ in (a), $n = 21$ in (b). (*Source*: Takeda, 1985, Figure 5.10.)

(9.3.5) once the function $f(x)$ is explicitly given (the first integral is the same as equation (9.2.11)).

Using the penalty function method and the steepest descent method with this derivative, we can solve Problem OPT6. Actually, Takeda (1985) solved this problem with a slight modification (the quasi-Newton method) for $f(\|x - x_i\|^2) = \|x - x\|^2$, $n = 20$, $\phi(x)$ being uniform over a unit square, $d^* = \sqrt{2}/10$ and $x_1^{\mathrm{T}} = (0.1, 0.1), x_{20}^{\mathrm{T}} = (0.9, 0.9)$; $n = 21$, $x_1^{\mathrm{T}} = x_{21}^{\mathrm{T}} = (0.15, 0.15)$. The results are depicted in Figure 9.3.1.

### 9.3.2 Locational optimization of a network

In most of network optimization problems the location of nodes and links, or at least the location of links, is fixed (for example, recall the bus stop problem in Section 9.2.1). The locational optimization of both nodes and links was rarely studied because of complicated geometrical computation. The recent progress in the Voronoi diagrams, however, provides a clue to this difficult problem. In fact, using the Voronoi diagram, Suzuki and Iri (1986c) attempted to solve the optimization problem in which the locations of both nodes and links were to be optimized so that the total travel time was minimized provided that the connection of links of the network and the traffic volumes between origins and destinations were given.

Let us consider a railway network in a region $S$ in which nodes are stations, and links are railways connecting the stations. We assume that the speed of a train on this railway network is controlled as follows: first, the speed is accelerated from a station with $\alpha$ until it reaches $v_{\max}$; next, the speed is kept constant at $v_{\max}$; last, the speed is decelerated with $-\alpha$ until it becomes zero at the next station. To avoid unnecessary complicated equations, we assume that the distance between two adjacent stations is greater than $v_{\max}{}^2/\alpha$, implying that a train achieves the maximum speed between stations.

When a traveller makes a trip from an origin $x_0$ to a destination $x_{\mathrm{d}}$, we assume that the traveller first walks to the nearest station, $x^*(x_0)$, from the origin $x_0$ at walking speed $v_{\mathrm{w}}$ (since the traveller knows the time schedule of trains, he/she reaches the station just when a train arrives there); next, the traveller takes a train from station $x^*(x_0)$ to station $x^*(x_{\mathrm{d}})$, which is the nearest station from the destination $x_{\mathrm{d}}$; last, the traveller walks from the station $x^*(x_{\mathrm{d}})$ to the destination $x_{\mathrm{d}}$ at walking speed $v_{\mathrm{w}}$. To obtain the total travel time of this trip, let $N[x^*(x_0), x^*(x_{\mathrm{d}})]$ be the number of stations between station $x^*(x_0)$ and station $x^*(x_{\mathrm{d}})$; $d_{\mathrm{net}}(x^*(x_0), x^*(x_{\mathrm{d}}))$ be the railway distance between station $x^*(x_0)$ and the station $x^*(x_{\mathrm{d}})$; and $b$ be the time spent at each station. A train runs a distance $v_{\max}{}^2/\alpha$ while it is accelerating and decelerating, and the train runs a distance $d_{\mathrm{net}}(x^*(x_0), x^*(x_{\mathrm{d}})) - N[x^*(x_0), x^*(x_{\mathrm{d}})]\, v_{\max}{}^2/\alpha$ while it maintains a constant speed $v_{\max}$. Since it takes $2v_{\max}/\alpha$ (minutes) for acceleration and deceleration, the total travel time from the origin $x_0$ to the destination $x_{\mathrm{d}}$ is given by

$$\begin{aligned} T(\boldsymbol{x}_{\text{o}},\boldsymbol{x}_{\text{d}}) = & \frac{1}{v_{\text{w}}}\{\| \boldsymbol{x}_{\text{o}} - \boldsymbol{x}^*(\boldsymbol{x}_{\text{o}}) \| + \| \boldsymbol{x}_{\text{d}} - \boldsymbol{x}^*(\boldsymbol{x}_{\text{d}}) \|\} \\ & + (\frac{v_{\max}}{\alpha} + b)\{N[\boldsymbol{x}^*(\boldsymbol{x}_{\text{o}}), \boldsymbol{x}^*(\boldsymbol{x}_{\text{d}})] + 1\} \\ & + \frac{1}{v_{\max}} d_{\text{net}}(\boldsymbol{x}^*(\boldsymbol{x}_{\text{o}}), \boldsymbol{x}^*(\boldsymbol{x}_{\text{d}})). \end{aligned} \tag{9.3.6}$$

To consider the traffic volume between two points in $S$, suppose that a traveller in a small region around $\boldsymbol{x}_0$ makes trips to see friends who are distributed over the region $S$. The number of friends in a small region $(\mathrm{d}\boldsymbol{x})$ around $\boldsymbol{x}_{\text{d}}$ is proportional to the number of inhabitants there. In terms of the density function of inhabitants, $\phi(\boldsymbol{x}_{\text{d}})$, this number is given by $c_1\phi(\boldsymbol{x}_{\text{d}})\,\mathrm{d}\boldsymbol{x}$, where $c_1$ is a constant. We assume that the traveller makes $c_1\phi(\boldsymbol{x}_{\text{d}})\,\mathrm{d}\boldsymbol{x}$ trips to a small region around $\boldsymbol{x}_{\text{d}}$ during a unit period of time. We also assume that the number of trip-makers in a small region around $\boldsymbol{x}_{\text{o}}$ is proportional to the number of inhabitants there, i.e. $c_2\phi(\boldsymbol{x}_{\text{o}})\,\mathrm{d}\boldsymbol{x}$, where $c_2$ is a constant. Thus the number of trips from a small region around $\boldsymbol{x}_{\text{o}}$ to a small region around $\boldsymbol{x}_{\text{d}}$ is given by $c_1 c_2\,\phi(\boldsymbol{x}_{\text{o}})\,\phi(\boldsymbol{x}_{\text{d}})\,\mathrm{d}\boldsymbol{x}_{\text{o}}\mathrm{d}\boldsymbol{x}_{\text{d}} = \phi(\boldsymbol{x}_{\text{o}})\,\phi(\boldsymbol{x}_{\text{d}})\,\mathrm{d}\boldsymbol{x}_{\text{o}}\mathrm{d}\boldsymbol{x}_{\text{d}}$, where we fix $c_1c_2 = 1$ without loss of generality. In the region $S$, the total round travel time of all travellers in a unit period of time is given by

$$F(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) = 2\int_S T(\boldsymbol{x}_{\text{o}},\boldsymbol{x}_{\text{d}})\,\phi(\boldsymbol{x}_{\text{o}})\,\phi(\boldsymbol{x}_{\text{d}})\,\mathrm{d}\boldsymbol{x}_{\text{o}}\,\mathrm{d}\boldsymbol{x}_{\text{d}}. \tag{9.3.7}$$

The nearest stations, $\boldsymbol{x}^*(\boldsymbol{x}_{\text{o}})$ and $\boldsymbol{x}^*(\boldsymbol{x}_{\text{d}})$, are readily obtained from the Voronoi diagram, $\{V_1,\ldots,V_n\}$, generated by a set of the $n$ stations at $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$. Obviously, if $\boldsymbol{x}_{\text{o}} \in V_i$, then $\boldsymbol{x}^*(\boldsymbol{x}_{\text{o}}) = \boldsymbol{x}_i$; if $\boldsymbol{x}_{\text{d}} \in V_j$, then $\boldsymbol{x}^*(\boldsymbol{x}_{\text{d}}) = \boldsymbol{x}_j$. We hence write $N[\boldsymbol{x}^*(\boldsymbol{x}_{\text{o}}), \boldsymbol{x}^*(\boldsymbol{x}_{\text{d}})]$ as $N[\boldsymbol{x}_i, \boldsymbol{x}_j]$ for $\boldsymbol{x}_{\text{o}} \in V_i$, $\boldsymbol{x}_{\text{d}} \in V_j$, and equation (9.3.7) is written as

$$\begin{aligned} F(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) = & 2\sum_{i=1}^{n}\int_{V_i}\frac{1}{v_{\text{w}}}\|\boldsymbol{x}_{\text{o}} - \boldsymbol{x}_i\|\,\phi(\boldsymbol{x}_{\text{o}})\,\mathrm{d}\boldsymbol{x}_{\text{o}} \\ & + 2\sum_{j=1}^{n}\int_{V_i}\frac{1}{v_{\text{w}}}\|\boldsymbol{x}_{\text{o}} - \boldsymbol{x}_j\|\,\phi(\boldsymbol{x}_{\text{d}})\,\mathrm{d}\boldsymbol{x}_{d} \\ & + 2\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\left\{\left(\frac{v_{\max}}{\alpha} + b\right)\left(N[\boldsymbol{x}_i,\boldsymbol{x}_j] + 1\right) + \frac{1}{v_{\max}}d_{\text{net}}(\boldsymbol{x}_i,\boldsymbol{x}_j)\right\} \\ & \times \int_{V_i}\phi(\boldsymbol{x}_{\text{o}})\,\mathrm{d}\boldsymbol{x}_{\text{o}}\int_{V_j}\phi(\boldsymbol{x}_{\text{d}})\,\mathrm{d}\boldsymbol{x}_{\text{d}}. \end{aligned} \tag{9.3.8}$$

The first term is the round travel time from (to) origins in the region $V_i$ to (from) the station $\boldsymbol{x}_i$. The second term is the round travel time from (to) the station $\boldsymbol{x}_j$ to (from) destinations in the region $V_j$. The last term is the round travel time from (to) the station $\boldsymbol{x}_i$ to (from) the station $\boldsymbol{x}_j$. Notice that the first term and the second term are the same.

To sum up, the network locational optimization problem is formulated as follows.

**Problem OPT7 (locational optimization of a network by minimizing the total travel time)**

$$\min_{x_1,\ldots,x_n}\left[\sum_{i=1}^{n}\int_{V_i}\frac{1}{v_w}\|x-x_i\|\phi(x)\,dx\right.$$
$$+\sum_{i=1}^{n}\sum_{j\geq i+1}^{n}\left\{\left(\frac{v_{\max}}{\alpha}+b\right)\left(N[x_i,x_j]+1\right)+\frac{1}{v_{\max}}d_{\text{net}}(x_i,x_j)\right\} \tag{9.3.9}$$
$$\left.\times\int_{V_i}\phi(x)\,dx\int_{V_j}\phi(x)\,dx\right]$$

(note that the factor 4 is omitted in expression (9.3.9)). We may add the constraints that $x_1 = c$ and $x_n = c'$, or $x_1 = x_n = c$ $(c \neq c')$.

Again, Problem OPT7 is a non-linear, non-convex programming problem. We may solve it with the steepest descent method. The first-order derivative is obtained from

$$\frac{\partial}{\partial x_{i\kappa}}F(x_1,\ldots,x_n)=\int_{V_i}\frac{1}{v_w}\frac{\partial}{\partial x_{i\kappa}}\|x-x_i\|\phi(x)\,dx$$
$$+\sum_{i=1}^{n}\sum_{j\geq i+1}^{n}\frac{1}{v_{\max}}\left\{\frac{\partial}{\partial x_{i\kappa}}d_{\text{net}}(x_i,x_j)\right\}\int_{V_i}\phi(x)\,dx\int_{V_j}\phi(x)\,dx$$
$$+\sum_{i=1}^{n}\sum_{j=i+1}^{n}\left\{\left(\frac{v_{\max}}{\alpha}+b\right)\left(N[x_i,x_j]+1\right)+\frac{1}{v_{\max}}d_{\text{net}}(x_i,x_j)\right\} \tag{9.3.10}$$
$$\times\left\{\int_{V_j}\phi(x)\,dx\frac{\partial}{\partial x_{i\kappa}}\int_{V_i}\phi(x)\,dx+\int_{V_i}\phi(x)\,dx\frac{\partial}{\partial x_{i\kappa}}\int_{V_j}\phi(x)\,dx\right\},\quad \kappa=1,2.$$

The first term is the same as equation (9.3.5). The second term indicates the change in the railway distance due to the slight move of the station $x_i$. If the railway is assumed to be a chain of lines connecting stations with straight line segments, the value of this term is obtained from

$$\frac{\partial}{\partial x_{i\kappa}}d_{\text{net}}(x_i,x_j)=\frac{\partial}{\partial x_{i\kappa}}\left\{\|x_{i-1}-x_i\|+\|x_i-x_{i+1}\|\right\},\quad \kappa=1,2. \tag{9.3.11}$$

The last term in equation (9.3.10) treats the change in the area of Voronoi polygons. The derivation is almost the same as that in Section 9.3.1 (we will also refer to this derivation in Section 9.5).

Using the steepest descent method with this derivative, we can solve Problem OPT7. An actual computation is carried out by Suzuki and Iri (1986c) with a modified method (the quasi-Newton method). Figure 9.3.2 shows one of their results where $n = 16$, $S = \{(x_1, x_2) \mid -0.5 \leq x_1, x_2 \leq 0.5\}$, $x_1^{\mathrm{T}} = (-0.1, -0.1)$, $x_{16}^{\mathrm{T}} = (0.9, 0.9)$, $v_{\max} = 0.01$, $\alpha = 0.002$, $v_w = 0.001$, and

$$\phi(x)=\begin{cases}\exp(-25\|x\|) & \text{if } x\in S,\\ 0 & \text{if } x\notin S.\end{cases} \tag{9.3.12}$$

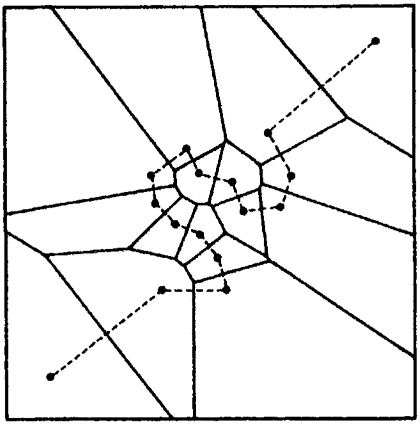


**Figure 9.3.2** A locally optimal railway stations $n = 16$, $S = \{(x_1, x_2) \mid -0.5 \le x_1, x_2 \le 0.5\}$, $x_1^T = (-0.1, -0.1)$, $x_{16}^T = (0.9, 0.9)$, $v_{max} = 0.01, \alpha = 0.002$, $v_w = 0.001$, and $\phi(x)$ is given by equation (9.3.12). (*Source*: Suzuki and Iri, 1986c, Figure 1.)

### 9.3.3 Euclidean Steiner minimal tree

In Section 2.5 we referred to a Euclidean minimum spanning tree (EMST). As is noticed from the term 'minimum', the EMST problem may be regarded as a locational optimization problem of lines, and this problem can be solved with the Delaunay triangulation (Property D23). In this subsection we show another locational optimization problem of lines, called the Steiner problem, which can also be solved with the Delaunay triangulation.

To give an example of the Steiner problem, suppose that there are three factories on a plane where we can freely lay out roads (Figure 9.3.3(a)). The problem is to construct a network of roads connecting these factories with the minimum length of roads, provided that roads are directly connected between these factories. Recalling the definition of Euclidean minimum spanning trees in Section 2.5, we realize that the solution is given by a Euclidean minimum spanning tree for the three points at which the factories are located (Figure 9.3.3(b)). This solution, however, is not optimal if we are allowed to add a new point (an empty circle in Figure 9.3.3(c)). The problem is to find the fourth point that minimizes the length of roads connecting the three

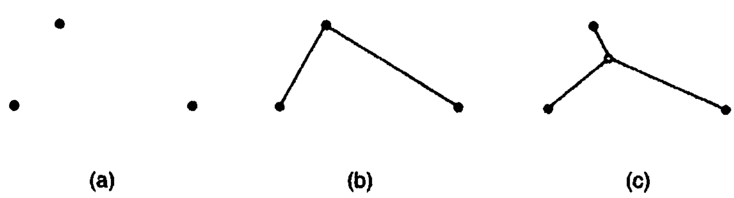



**Figure 9.3.3** An example of a Euclidean minimum spanning tree (b) and an Euclidean Steiner minimal tree (c).

factories and the fourth point. The solution is shown in Figure 9.3.3(c). The length of the roads of Figure 9.3.3(c) is shorter than that of Figure 9.3.3(b).

According to Kuhn (1974), the above problem dates back to Fermat in the seventeenth century, and was introduced by Courant and Robbins (1941, VII, §6) as the Steiner (Steiner's) problem. Since then, the Steiner problem has been generalized in various directions. In this subsection we consider the following generalized problem.

**Problem OPT8 (Euclidean Steiner minimal tree problem)** Suppose that a certain number of points are fixed in the Euclidean plane (space). We wish to minimize the length of line segments connecting the given points. We can add new points (through which the given points are connected) if the additional points reduce the length. The problem is to find the number of additional points and their locations that minimize the length of the line segments connecting the given points and the additional points.

We call the collection of line segments of this solution a *Euclidean Steiner minimal tree*, and the additional points *Steiner points*.

To discuss the above problem mathematically, let $P$ be a set $\{x_1, \ldots, x_m\}$ of $m$ $(2 \le m < \infty)$ distinct points which are fixed in $\mathbb{R}^2$; $Q_n$ be a set of $n$ additional points where $n$ and their locations $\{x_{m+1}, \ldots, x_{m+n}\}$ are variables; $C_{m+n} = [c_{ij}]$ be the $(m+n) \times (m+n)$ adjacency matrix of points in $P \cup Q_n$ where $c_{ij} = 1$ if nodes $x_i$ and $x_j$ are connected by a line segment and $c_{ij} = 0$ if they are not connected; $c_i = \sum_{j=1,\, j\neq i}^{m+n} c_{ij}$ be the degree of node $i$ (the number of links connected to node $i$); $l_{ij}$ be $\overline{x_i x_j}$ (the line segment joining nodes $x_i$ and $x_j$) if $c_{ij} = 1$ and $l_{ij} = \varnothing$ if $c_{ij} = 0$; $L_{m+n} = \{l_{ij} \mid i \neq j, i, j \in I_{m+n}\}$; and $\mathscr{G}$ be all possible connected geometric graphs of $G(P \cup Q_n, L_{m+n})$ for $n = 0, 1, \ldots$ and $x_{m+1}, x_{m+2}, \ldots \in \mathbb{R}^2$. In these terms, Problem OPT8 is written mathematically as

$$\min_{n; x_{m+1}, \ldots, x_{m+n}} \sum_{i=1}^{m+n} \sum_{j=i+1}^{m+n} c_{ij} \| x_i - x_j \|, \quad \text{subject to } G(P \cup Q_n, L_{m+n}) \in \mathscr{G}. \quad (9.3.13)$$

To solve this problem, we first solve its subproblem. We fix $n$ and the adjacency matrix $C_{m+n}$, and consider a set, $\mathscr{G}(C_{m+n})$, of all possible geometric graphs whose adjacency matrix is given by $C_{m+n}$. The subproblem is written as

$$\min_{x_{m+1}, \ldots, x_{m+n}} \sum_{i=1}^{m+n} \sum_{j=i+1}^{m+n} c_{ij} \| x_i - x_j \|, \quad \text{subject to } G(P \cup Q_n, L_{m+n}) \in \mathscr{G}(C_{m+n}). \quad (9.3.14)$$

We call the tree of this solution the *relatively minimal tree*. Gilbert and Pollak (1968) show that the relatively minimal tree is uniquely determined for a given $C_{m+n}$. Figure 9.3.4 depicts relatively minimal trees for different adjacency matrices. Obviously, a Euclidean Steiner minimal tree is one of the relatively minimal trees including points $P$.

Euclidean Steiner minimal trees have many interesting properties, which are discussed by Courant and Robbins (1941), Melzak (1961), Chang (1972),

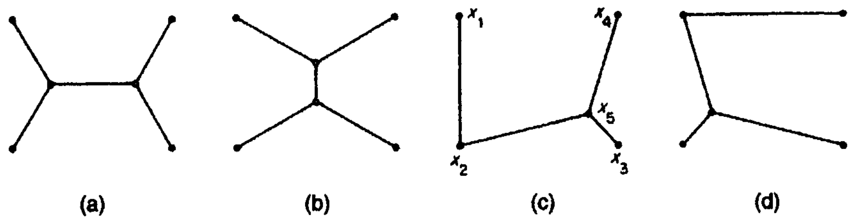



**Figure 9.3.4** (a) Euclidean Steiner minimal tree; (a)(b) Euclidean Steiner trees; (a)–(d) relatively minimal trees.

Gilbert and Pollak (1968), Megiddo (1978), Smith *et al.* (1981), Winter (1985), among others. We refer to only three of them which are to be used here.

**Property ST1** Exactly three links meet at every Steiner point, and the three angles ($\alpha_{i1}, \alpha_{i2}, \alpha_{i3}$) made by those links at any Steiner point are all 120° (i.e. $c_i = 3$ and $\alpha_{i1} = \alpha_{i2} = \alpha_{i3} = 120°$).

**Property ST2** At most three links meet at every point of $P$ ($c_i \leq 3$). If exactly three links meet at a point in $P$ ($c_i = 3$), then $\alpha_{i1} = \alpha_{i2} = \alpha_{i3} = 120°$. If exactly two links meet at a point in $P$ ($c_i = 2$), then $\alpha_{i1} \geq 120°$, where $\alpha_{i1}$ and $\alpha_{i2}$ are angles at a point of degree 2, and $\alpha_{i1} \leq \alpha_{i2}$.

**Property ST3** The number $n^*$ of Steiner points is less than or equal to the number of given points minus 2, i.e. $n^* \leq m - 2$.

We call the relatively minimal tree satisfying these three properties a *Euclidean Steiner tree*. Figures 9.3.4(a) and (b) are Euclidean Steiner trees. A *Euclidean Steiner minimal tree* is the Euclidean Steiner tree whose length is minimum among all Euclidean Steiner trees (Figure 9.3.4(a)).

To obtain Euclidean Steiner minimal trees, several combinatorial methods are proposed in the literature, for example Chang (1972), Gilbert and Pollak (1968), Meggido (1978), and Winter (1985). In practice, however, their methods can hardly solve the problem with more than twenty points ($m > 20$). In fact, Garey *et al.* (1977) show that the Steiner problem is NP-hard (Section 1.3.4). For a large number of points, it is almost impossible to obtain a Euclidean Steiner minimal tree. We may, however, obtain an approximate Euclidean minimal Steiner tree with a computational method. A few heuristic methods are proposed in the literature, for example Smith *et al.* (1981) and Suzuki and Iri (1986b). We show the latter method here, because it is more efficient than the former.

A clue to the problem is shown in Figure 9.3.5. Figure 9.3.4(c) is a relatively minimal tree, but not a Euclidean Steiner tree, because the links at $x_2$ do not satisfy Property ST2. To avoid such a point, we *split* (Gilbert and Pollak, 1968) the point as follows. First, we add a new point, $x_6$, in the close neighbourhood of $x_2$ (see Figure 9.3.5(a)); second we connect $x_2$ with $x_6$,

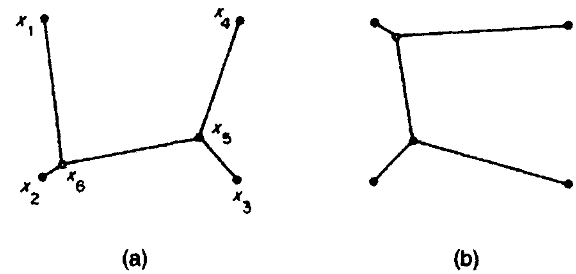



**Figure 9.3.5** Splitting a point in relatively minimal trees.

$x_6$ with $x_1$, and $x_6$ with $x_5$. As a result (Figure 9.3.5a), the adjacency matrix of the resulting network becomes the same as that of the Euclidean Steiner tree in Figure 9.3.4(a). Solving the optimization problem given by expression (9.2.14), we obtain Figure 9.3.4(a), which is one of the Euclidean Steiner trees. Similarly, we obtain another Euclidean Steiner tree (Figure 9.3.4(b)) from the relatively minimal tree in Figure 9.3.4(d) through splitting shown in Figure 9.3.5(b).

We now formulate the above procedure as the following algorithm.

**Algorithm OPT1 (Euclidean Steiner tree algorithm)**

Step 1. Choose an arbitrary number $n$ larger than $m-2$, and place the $n$ points $Q_n$ randomly in the convex hull of the given points $P$.

Step 2. Construct the Delaunay triangulation for points $P \cup Q_n$.

Step 3. Obtain the minimum spanning tree from the Delaunay triangulation using, for example, Cheriton–Tarjan's method (Cheriton and Tarjan, 1976).

Step 4. Given the adjacency matrix of this minimum spanning tree, solve the problem of expression (9.3.14) with the descent method.

Step 5. Delete the points of degree $c_i \leq 2$ in $Q_n$ of the resulting relatively minimal tree, and place a new point in the close neighbourhood of each point of degree $c_i \geq 4$ in $Q_n$ (as a result, $n$ changes).

Step 6. For each point of degree $c_i = 2$ in $P$ (denoted by $x_i$*), place a new point in the close neighbourhood of each $x_i$* inside the angle $\alpha_{i1} < 120°$ in the direction of the bisector of angle $\alpha_{i1}$.

For each point of degree $c_i > 3$ in $P$ and each point of degree $c_i = 3$ in $P$ that does not satisfy $\alpha_{i1} = \alpha_{i1} = \alpha_{i1} = 120°$ (denoted by $x_i$**), place a new point in the close neighbourhood of each $x_i$** inside the smallest angle $\alpha_i^* = \min_j \{\alpha_{ij}, j \in I_{ci}\}$ in the direction of the bisector of the angle $\alpha_i^*$.

Step 7. If there is no point that can be deleted or added in Steps 5 and 6, report $n$ and $x_{m+1}, \ldots, x_{m+n}$. Else, go to Step 2.

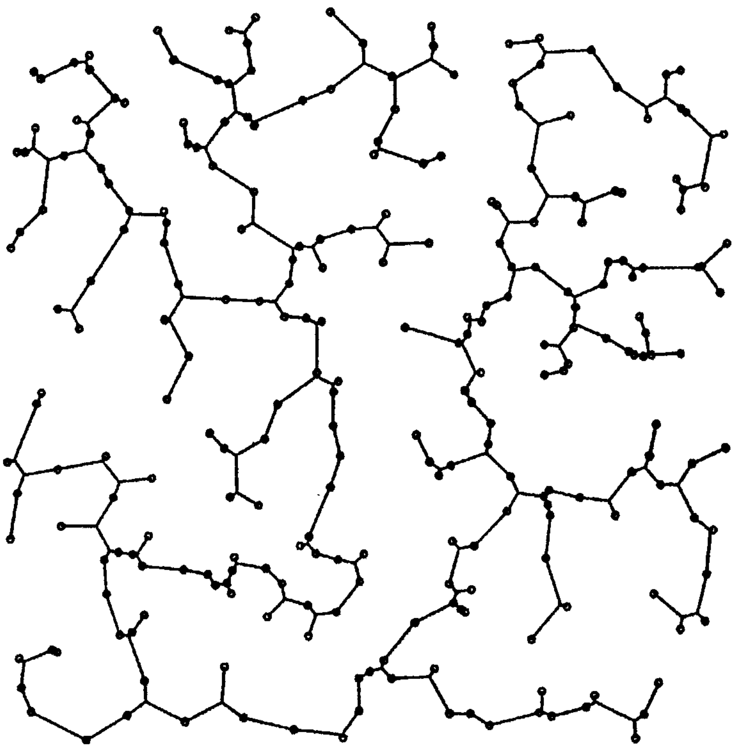


**Figure 9.3.6** An approximate Euclidean Steiner tree. (*Source*: Suzuki and Iri, 1986b, Figure 9.)

Step 5 is the procedure to satisfy Property ST1, and Step 6 is the procedure to satisfy Property ST2. Step 4 is to solve the non-linear, non-convex optimization problem of expression (9.3.14). The objective function is given by $F(\boldsymbol{x}_{m+1}, \ldots, \boldsymbol{x}_{m+n}) = \sum_{i=1}^{m+n} \sum_{j=i+1}^{m+n} c_{ij} \| \boldsymbol{x}_i - \boldsymbol{x}_j \|$. The first-order derivative is given by

$$\frac{\partial F}{\partial x_{i\kappa}} = \sum_{j \in \{j | c_{ij} \neq 0\}} \frac{\partial}{\partial x_{i\kappa}} \| \boldsymbol{x}_i - \boldsymbol{x}_j \|, \quad \kappa = 1, 2. \tag{9.3.15}$$

The second-order derivatives are also obtained from this equation. Using the steepest descent method or the (quasi-)Newton method, we can solve the problem in Step 4.

With Algorithm OPT1, we can obtain a Euclidean Steiner tree. In the case of Figure 9.3.4, we have only two Euclidean Steiner trees. Hence, comparing those trees, we can obtain a Euclidean Steiner minimal tree (Figure 9.3.4(a)). In the case of a large number of given points, we can hardly find all Euclidean Steiner trees. We hence obtain a computationally feasible number of Euclidean Steiner trees and find the minimum tree among them. Obviously, this tree is not guaranteed to be a Euclidean Steiner minimal tree, but it may be close to it.

Figure 9.3.6 shows an approximate Euclidean Steiner minimal tree obtained by Suzuki and Iri (1986b) for $m = 2^8$ points.

Recalling that Section 9.2 dealt with the locational optimization of points, and Section 9.3 dealt with that of lines, the reader might expect that the next section will deal with the locational optimization of areas. At present, however, few papers deal with this problem except for an initial attempt. Shiode (1995) considers the locational optimization of a park in a region to minimize the average distance to the boundary of the park provided that the area of the park and the length of the boundary of the park are fixed.

## 9.4 LOCATIONAL OPTIMIZATION OVER TIME

In the preceding sections we formulated the locational optimization problems without considering the time dimension. In this section we take this dimension into account.

### 9.4.1 Multi-stage locational optimization

In Section 9.2 we implicitly assumed that all facilities can be constructed at the same time. This assumption, however, is not always acceptable in reality. One reason is the budget constraint. The number of facilities that can be constructed in a year is sometimes limited because of the budget constraint; we have to construct the required number of facilities over years. Another reason is inefficiency. When the density of users is expected to increase from low density to high density, it is inefficient to construct all facilities in the early stage; the facilities should be constructed in accordance with the increase of users. Because of these reasons, we sometimes meet the problem in which the location of facilities is to be optimized not only over a region but also over time. The problems related to this locational optimization are studied by several researchers in Operations Research, for example Love (1976), Erlenkotter (1981), Van Roy and Erlenkotter (1982), Gunawardane (1982) and Scott (1975). Among them, the problems considered by Scott (1971) and Suzuki *et al.* (1991) are the closest to the above problem. The former model is formulated on a network, whereas the latter model is formulated on a continuous plane. We show the latter model here, because it is formulated with the Voronoi diagram.

Suppose that we construct $n$ facilities in a region $S$, and that we can construct only one facility in one stage. Consequently it takes $n$ stages to construct all the facilities. The length of the $i$th stage (referred to as stage $i$) is $\tau_i$, which may be the same constant like a year ($\tau_i = 1$) or may not. The $n$ facilities are indexed according to the order of construction; facility $i$ is constructed at the beginning of stage $i$. Thus, facilities $j = 1, \dots, i$ are usable in stage $i$. We assume that once a facility is located, the facility remains there; it cannot be replaced (Figure 9.4.1).

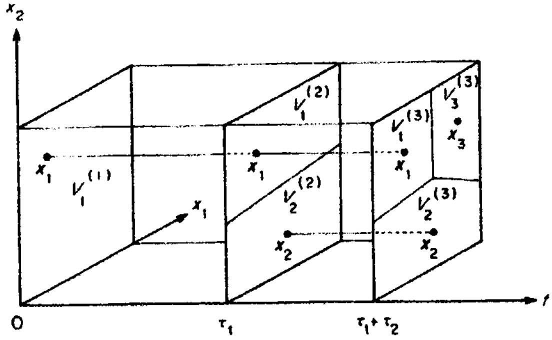



**Figure 9.4.1** Locating facilities over many stages.

We further assume, as in Section 9.2, that every user uses the nearest facility with respect to the Euclidean distance. Let $\mathcal{V}^{(i)} = \{V_1^{(i)}, \ldots, V_i^{(i)}\}$ be the Voronoi diagram generated by a set $\{x_1, \ldots, x_i\}$ of points. Obviously, $\mathcal{V}^{(1)} = S$. In stage $i$, users in $V_j^{(i)}$ use facility $j$; in stage $i+1$, users in $V_j^{(i+1)}$ use facility $j$ (Figure 9.4.1). Let $f(\|x - x_j\|^2)$ be the travel cost function of the squared Euclidean distance, which is assumed to be strictly increasing and differentiable, and $\phi_i(x)$ be the density of users at $x$ in stage $i$. We assume that $\phi_i(x)$ is constant in stage $i$ (note that $\phi_i(x) \neq \phi_{i+1}(x)$ in general).

In the above terms, the travel cost averaged over region $S$ and over $n$ stages is written as

$$F(x_1, \ldots, x_n) = \sum_{i=1}^{n} \tau_i \sum_{j=1}^{i} \int_{V_j^{(i)}} f(\|x - x_j\|^2)\, \phi_i(x)\, dx. \qquad (9.4.1)$$

With this objective function, we formulate the above problem as the following mathematical programming problem.

**Problem OPT9 (minimization of the average travel cost to the nearest points over many stages)**

$$\min_{x_1, \ldots, x_n} \left[ \sum_{i=1}^{n} \tau_i \sum_{j=1}^{i} \int_{V_j^{(i)}} f(\|x - x_j\|^2)\, \phi_i(x)\, dx \right]. \qquad (9.4.2)$$

In conjunction with this far-sighted locational optimization problem, it may be of interest to compare a near-sighted locational optimization problem. In the latter problem, we optimize the location of facility 1 in stage 1; in stage 2, provided that the location of facility 1 is fixed at $x_1$ obtained in stage 1, we optimize the location of facility 2; in stage 3, provided that locations of facilities 1 and 2 are fixed at $x_1$ and $x_2$ obtained in stages 1 and 2, we optimize the location of facility 3; ... ; in stage $i$, provided that the locations of the facilities $j = 1, \ldots, i-1$ are fixed at $x_1, \ldots, x_{i-1}$ obtained in stages $1, \ldots, i-1$, we optimize the location of facility $i$ without considering the facilities $j = i+1, \ldots, n$ to be located in the future. Mathematically, this optimization problem is written as

**Problem OPT10 (myopic locational optimization problem)**

$$\min_{x_i}\left[\sum_{j=1}^{i}\int_{V_i^{(i)}} f(\|x-x_j\|^2)\,\phi_i(x)\,\mathrm{d}x\right],\tag{9.4.3}$$

where $x_1,\ldots,x_{i-1}$ are fixed and obtained from

$$\min_{x_{i-1}}\left[\sum_{j=1}^{i-1}\int_{V_i^{(i-1)}} f(\|x-x_j\|^2)\,\phi_i(x)\,\mathrm{d}x\right],\tag{9.4.4}$$

for $k = 2,3,\ldots,i$, where, $V_j^{(0)} = S$.

The computational methods for Problems OPT9 and OPT10 are almost the same as that for Problem OPT3. Regarding Problem OPT10, the first derivative of the objective function in expression (9.4.3) is just the same as that of equation (9.2.11). Regarding Problem OPT9, the first derivative of equation (9.4.1) is given by

$$\frac{\partial F}{\partial x_{j\kappa}}=\sum_{i=j}^{n}\tau_i\int_{V_i^{(i)}}\frac{\partial}{\partial x_{j\kappa}}f(\|x-x_j\|^2)\,\phi_i(x)\,\mathrm{d}x,\quad \kappa=1,2.\tag{9.4.5}$$

Each term in the summation is the same as equation (9.2.11). We can thus solve Problem OPT9 using the descent method with the derivatives obtained from equation (9.4.5). (Note that Voronoi diagrams $V^{(i)}$, $i = 1,\ldots,n$, are efficiently obtained from the incremental method shown in Chapter 4.) Figure 9.4.2 shows the locally optimal locations of $n = 3,\ldots,8$ facilities obtained by T. Suzuki *et al.* (1991) for the uniform distribution over a unit square and $f(\|x-x_i\|^2) = \|x-x_i\|$. The filled circles in panel (a) show the locally optimal locations of Problem OPT9 for equal time intervals ($\tau_i = 1, i = 1,$

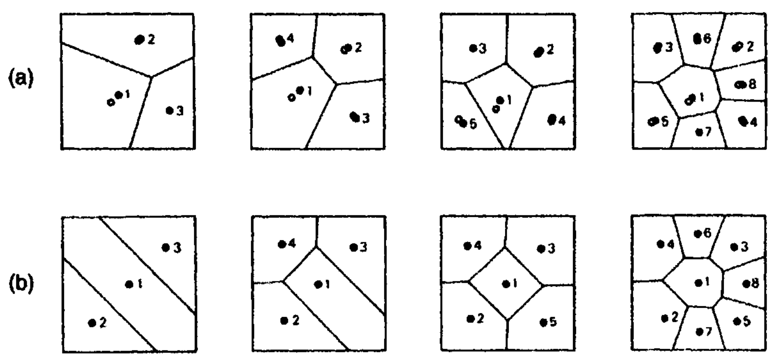



**Figure 9.4.2** Locally optimal locations of $n = 1,\ldots,8$ firms in a unit square region with a uniform distribution and $f(\|x-x_i\|^2) = \|x-x_i\|$: (a) locally optimal locations for Problem OPT9 (optimized over time) $\tau_i = 1$ for filled circles and $\tau_i = i$ for unfilled circles; (b) locally optimal locations for Problem OPT10 (optimized at each time stage). (*Source*: T. Suzuki *et al.*, 1991, Figure 4.)

$\dots, n$), and the unfilled circles in the same panel show those for linearly increasing time intervals ($\tau_i = i, i = 1, \dots, n$). The numbers associated with the filled and unfilled circles indicate the time of construction. As is noticed from these circles, the difference appears quite small. Panel (b) shows the locally optimal locations of facilities under the myopic location policy. Compared with panel (a), the configurational difference is large for a small number of facilities, but it becomes small as the number of facilities increases. In fact, T. Suzuki *et al.* (1991) show that the difference in the average nearest neighbour distance over time is within only 3% for $n = 8$, implying that the myopic location policy is not as bad as expected for this case.

### 9.4.2 Periodic locational optimization

A few hundreds years ago, market places were open periodically. In Japan, we can find such traces in the names of places, for instance 3rd-Day-Market, 7th-Day-Market, etc. In a 3rd-Day-Market, the market opened on the 3rd, 13th, and 23rd of each month. Nowadays, there are few such old-style markets, but we still have such periodic services, for example a monthly mobile clinic visiting several places in a region. In this subsection we consider the locational optimization of facilities that open periodically.

Suppose that there are $n$ market places, $i = 1, \dots, n$, in a region $S \subset \mathbb{R}^2$ and these market places are located at $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in S$, respectively. Market place $i$ is open at $t_i + jT, j = 0, 1, \dots$, where $t_1 < t_2 < \dots < t_n < T$ is assumed without loss of generality. A market opens for a certain length of time, but we assume that this time is very short compared with $t_i - t_{i-1}$. Under this assumption, market $i$ located at $\boldsymbol{x}_i$ and open at $t_i$ is represented by the point $(\boldsymbol{x}_i, t_i)$ in $\mathbb{R}^3$. Let $t_i^*(t)$ be the length of time from time $t$ to the nearest time when market $i$ opens, i.e.

$$t_i^*(t) = \min_j \{t_i + jT \mid t_i + jT \geq t\}. \tag{9.4.6}$$

Suppose that at time $t$ a consumer at a location $\boldsymbol{x}$ plans to go to a market. If the consumer chooses market place $i$, he/she has to wait for $t_i^*(t)$ days and walk $\|\boldsymbol{x} - \boldsymbol{x}_i\|$ km; if the consumer chooses market place $k$, he/she has to wait for $t_k^*(t)$ days and walk $\|\boldsymbol{x} - \boldsymbol{x}_k\|$ km. If the waiting time for market $i$ is shorter than that for market $k$ and the walking distance to market $i$ is shorter than market $k$, the consumer's decision is simple; he/she decides to go to market $i$. If the former inequality holds, but the latter inequality does not hold (or vice versa), the consumer's decision becomes slightly complex; he/she makes a decision by trading off between waiting time and walking distance. To be explicit, let $\alpha$ be this trade-off rate, that is, the patience of waiting for one day is equivalent to that of walking $\alpha$ km, and

$$d\left((\boldsymbol{x}, t), (\boldsymbol{x}_i, t_i^*(t))\right) = \sqrt{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2 + \alpha^2 (t - t_i^*(t))^2} \tag{9.4.7}$$

(recall that this distance is equivalent to the space–time distance defined in Chapter 3). If a consumer does not mind waiting, then $\alpha = 0$. If a consumer dislikes waiting, $\alpha$ is very large. In terms of this distance, the above choice assumption is written as: a consumer at $\boldsymbol{x}$ who considers to go to a market at time $t$ chooses the market which is the nearest with respect to the distance of equation (9.4.7), i.e. $\min_i [\sqrt{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2 + \alpha^2(t - t_i^*(t))^2}]^{1/2}$.

With the distance of equation (9.4.7), we can define the Voronoi diagram $\{V_1, \ldots, V_n\}$ generated by points $(\boldsymbol{x}_i, t_i)$, $i \in I_n$, in the space given by $\{(\boldsymbol{x}, t) \mid \boldsymbol{x} \in S, 0 \le t \le T\}$. We can see the consumer's choice from this Voronoi diagram. If $(\boldsymbol{x}, t)$ is in the Voronoi polygon $V_i$, the market place at $i$ is chosen. This Voronoi polygon can be depicted on a plane if $S$ is given by a line segment. In this case the Voronoi diagram becomes the space–time Voronoi diagram on a cylinder (Chapter 3), which is extendable on a plane (Figure 9.4.3(a)).

The optimization problem is to place $n$ market places so that the average space–time distance to the nearest market places is minimized. Mathematically, this problem is written as follows.

**Problem OPT11 (minimization of the average space–time distance to the nearest points)**

$$\min_{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n} \sum_{i=1}^{n} \int_{V_i} \sqrt{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2 + \alpha^2 (t - t_i^*(t))^2}\; \phi(\boldsymbol{x}, t)\, d\boldsymbol{x}\, dt. \tag{9.4.8}$$

The computational method for this problem is almost the same as that for Problem OPT3. Figure 9.4.3 shows a locally optimal solution in the simplest case in which two market places are placed in a unit line segment with $\phi(\boldsymbol{x}, t) = 1$, $t_1 = 0.4$, $t_2 = 0.8$ and $\alpha = 1$ (Seoung, 1990). Obviously, the solution changes according to the value of $\alpha$. When $\alpha = 0$ (implying that consumers do not mind waiting), the optimization problem in $\{(\boldsymbol{x}, t) \mid \boldsymbol{x} \in S, 0 \le t \le T\}$ reduces to the optimization problem in $S$ (Problem OPT3). The solution is given by $\boldsymbol{x}_1 = 0.25$ and $\boldsymbol{x}_2 = 0.75$. When $\alpha$ is very large (implying that consumers do not mind walking to a market place), the solution is given by $\boldsymbol{x}_1 = \boldsymbol{x}_2 = 0.5$. As is shown in Figure 9.4.3(b), the locations of market places approach the centre as the value of $\alpha$ increases. Note that

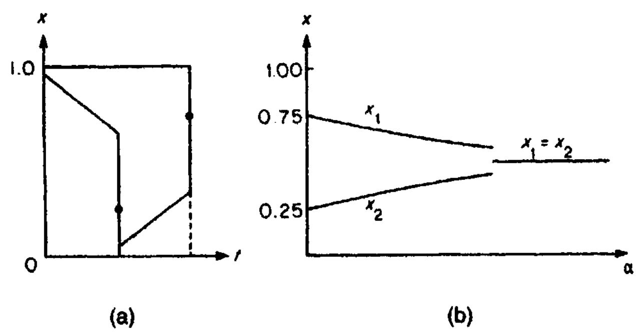



**Figure 9.4.3** Local optimal locations of two market places ($n = 2$) for $\phi(\boldsymbol{x}, t) = 1$, $t_1 = 0.5$, $t_2 = 1.0$: (a) $\alpha = 1$; (b) local optimal locations with respect to $\alpha$.

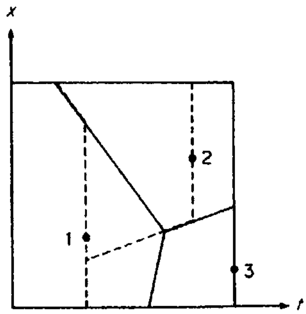



**Figure 9.4.4** The difference between the space–time Voronoi diagram (dashed lines) and the ordinary Voronoi diagram (solid lines).

this change is not continuous; after α exceeds a certain value, both market places locate at the centre.

The case of a two-dimensional region $S$ is studied by Takeda (1985) with a slightly different distance, i.e. equation (9.4.7) is replaced by

$$t_i^*(t) = \min_j \left\{ \| t_i + jT - t \| \;\middle|\; j = 0, 1, \ldots \right\}. \tag{9.4.9}$$

An example of the Voronoi diagram with this distance is depicted by the solid lines in Figure 9.4.4. In the same figure, the space–time Voronoi diagram is shown by the broken lines. The difference is, as is seen in Figure 9.4.4, distinct, and so their implications of time should be different. To consider a possible implication, suppose that a consumer wants to eat fresh fish on every Friday. If a fish market opens on every Friday, the consumer is happy. If a fish market opens on every Sunday, the consumer eats fish on Sunday, because he or she does not want to eat refrigerated fish. In this case, the consumer has to wait two days, and hence the degree of dissatisfaction may be measured as two days. If the market opens on every Thursday, the degree of dissatisfaction may be measured as one day. Under this understanding, the distance

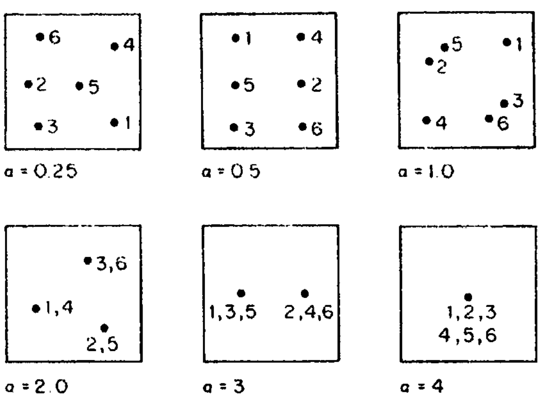



**Figure 9.4.5** Locally optimal locations of six market places in a unit square with a uniform distribution. (*Source*: Takeda, 1985, Figure 4.5.)

of equation (9.4.9) may be regarded as the measure of dissatisfaction with respect to time.

Figure 9.4.5 shows the local optimal locations of six market places in a unit square obtained by Takeda (1985). The numbers associated with the filled circles indicate $t_i$. For $\alpha \leq 1$, market places open at six different places, but for $\alpha = 2.0$, 3.0 and 4.0, they open at three, two and one place, respectively.

## 9.5 VORONOI FITTING AND ITS APPLICATION TO LOCATIONAL OPTIMIZATION PROBLEMS

In Chapter 7 we observed that the territories of mouthbreeder fish looked like a Voronoi diagram, and discussed how to measure the fitness of a polygonal tessellation to a Voronoi diagram. In this section we show another method in which we seek to find a Voronoi diagram that gives the ' best' fit to a given polygonal tessellation. We also show that this method is applicable to some locational optimization problems.

### 9.5.1 Method of fitting a Voronoi diagram to a polygonal tessellation

Let $S_1, \ldots, S_n$ be a finite number of closed polygons in $S \subset \mathbb{R}^2$ satisfying $[S_i \backslash \partial S_i] \cap [S_j \backslash \partial S_j] \neq \varnothing$ for $i \neq j$, $i, j \in I_n$ and $\cup_{i=1}^{n} S_i = S$. Then, $\mathcal{S} = \{S_1, \ldots, S_n\}$ forms a tessellation of $S$ (the dash-dot lines in Figure 9.5.1). Our objective is to fit a Voronoi diagram to the tessellation $\mathcal{S}$. To this end, we first choose an arbitrary point $\boldsymbol{x}_i$ in $S_i$ and generate the Voronoi diagram $\mathcal{V} = \{V_1, \ldots, V_n\}$ by the set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of points (the solid lines in Figure 9.5.1). Since we are concerned with the Voronoi diagram in $S$, the Voronoi diagram should be bounded by $S$, i.e. $\{V_1 \cap S, \ldots, V_n \cap S\}$, which is written, for notational simplicity, as $\{V_1, \ldots, V_n\}$. In general, we can treat a non-convex $S$ using the shortest-path Voronoi diagram (Section 3.4), but we assume, for simplicity, that $S$ is convex.

The similarity between the Voronoi diagram $\mathcal{V}$ and the tessellation $\mathcal{S}$ may be measured in terms of the intersection area $\sum_{i=1}^{n} |S_i \cap V_i|$ (the shaded regions in Figure 9.5.1). Obviously, if this value is large (i.e. close to $|S|$), we can say that the tessellation $\mathcal{S}$ is close to the Voronoi diagram $\mathcal{V}$. Mathematically, this intersection area is written as

$$F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \int_{V_i \cap S_i} \mathrm{d}\boldsymbol{x}. \tag{9.5.1}$$

This function may be slightly generalized as

$$F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \int_{V_i \cap S_i} \phi(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}, \tag{9.5.2}$$

where $\phi(\boldsymbol{x})$ indicates a weight (an explicit meaning will be given in Section 9.5.2). With this generalized objective function, the Voronoi fitting problem is formulated as follows.

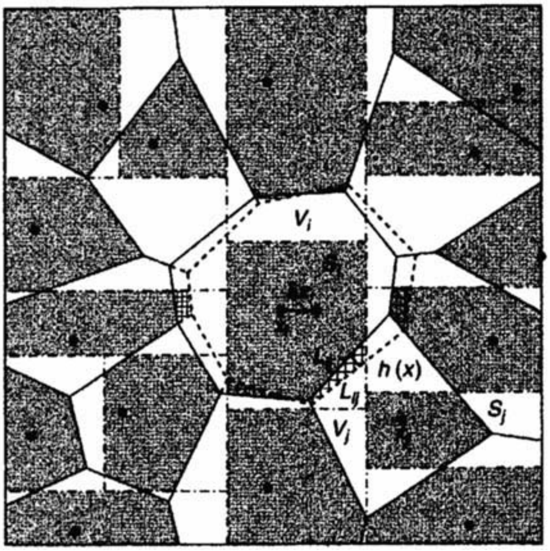



**Figure 9.5.1** Change in $V_i \cap S_j$ produced by a slight move of $x_i$.

**Problem OPT12 (maximization of the common area between a given tessellation and a Voronoi diagram)**

$$\max_{x_1,\ldots,x_n} \sum_{i=1}^{n} \int_{V_i \cap S_i} \phi(x)\,dx. \tag{9.5.3}$$

The computational method for this maximization problem is almost the same as that in expression (7.3.1) in Section 7.3. As is seen in Figure 9.5.1, a slight move, $\delta x_i$, of $x_i$ produces the changes in $V_i \cap S_j$, $j \in J_i = \{j \mid V_i \cap S_j \neq \emptyset, j \in I_n\}$ (the shaded areas in Figure 9.5.1). To measure these changes, let $V_i'$ be the Voronoi polygon of the point $x_i + \delta x_i$; $L_{ij} = (V_i \cap V_j) \cap S_i$ (the Voronoi edge shared with Voronoi polygons $V_i$ and $V_j$ included in $S_i$); $L_{ij}' = (V_i' \cap V_j) \cap S_i$; and $h(x)$ be the length of the perpendicular line segment from a point on $L_{ij}'$ to the foot point, $x^T = (x_1, x_2)$, on $L_{ij}$ (see Figure 9.5.1). After a few steps of calculation (Suzuki and Iri, 1986b), $h(x)$ is obtained as

$$h(x) = \frac{(x_{i1} - x_1)\,\delta x_{i1}}{\| x_i - x_j \|} + \frac{(x_{i2} - x_2)\,\delta x_{i2}}{\| x_i - x_j \|}. \tag{9.5.4}$$

The change in the area of $V_i \cap S_j$ (the grid area in Figure 9.5.1) is obtained from the integral of $h(x)$ along the line $L_{ij}$. Thus, the first derivative of equation (9.5.2) is given by

$$\begin{aligned}\frac{\partial F}{\partial x_{i\kappa}} &= \sum_{j \in J_i} \left\{ \int_{L_{ii}} h(x)\,\phi(x)\,dx - \int_{L_{ji}} h(x)\,\phi(x)\,dx \right\} \\ &= \sum_{j \in J_i} \left\{ \int_{L_{ij}} \frac{x_{i\kappa} - x_\kappa}{\| x_i - x_j \|}\,\phi(x)\,dx - \int_{L_{ji}} \frac{x_{i\kappa} - x_\kappa}{\| x_i - x_j \|}\,\phi(x)\,dx \right\}.\end{aligned} \tag{9.5.5}$$

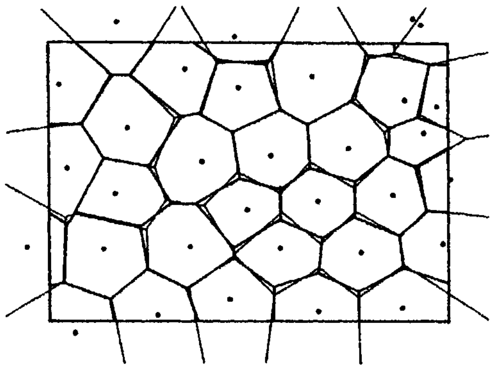


**Figure 9.5.2** Fitting a Voronoi diagram to the territories of mouthbreeder fish. (*Source*: Suzuki and Iri, 1986a, Figure 13.)

Using the steepest descent method with this derivative, we can solve Problem OPT12. Figure 9.5.2 illustrates an actual application carried by Suzuki and Iri (1986b) who fit a Voronoi diagram (the heavy solid lines) to the territories of mouthbreeder fish studied by Barlow (1974) (followed by Hasegawa and Tanemura, 1976, and Honda, 1978). Suzuki and Iri (1986a) also fitted a Voronoi diagram to basaltic columnar jointing studied by Koch (1974), Stoyan and Stoyan (1980) and Stoyan and Hermann (1986). Judging from

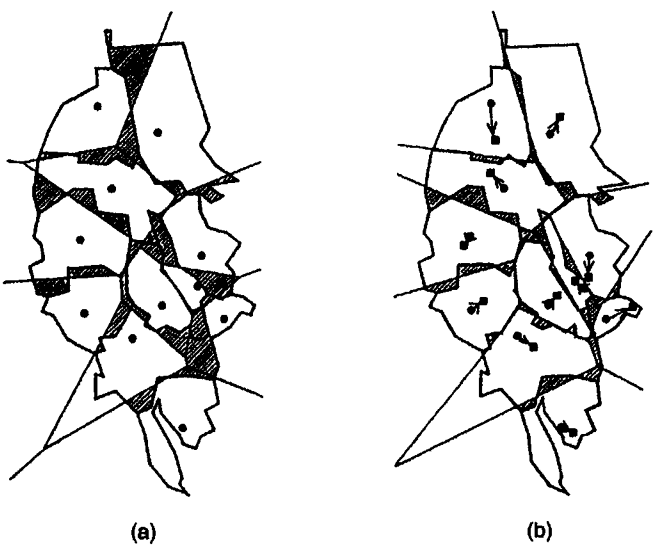


**Figure 9.5.3** School districts in Tsukuba: (a) school districts and the present sites of schools (filled circles), the students in the shaded region cannot go to their nearest schools; (b) school sites (filled squares) that minimize the area in which students cannot go to their nearest schools. (*Source*: Suzuki and Iri, 1986a, Figure 14.)

their examination, the Voronoi diagrams appear to give a good fit to these phenomena.

### 9.5.2 Locational optimization for minimizing restricted areas

The method of fitting a Voronoi diagram to a given tessellation is also useful to solve the locational optimization of facilities whose use is spatially restricted. An example is the locational optimization of schools constrained by school districts. Figure 9.5.3(a) shows the districts of junior high schools in Tsukuba (the filled circles indicate the locations of the schools). In Japan, students in a school district are supposed to go to the school assigned to that district. As a result, students in some areas have to go to the school which is farther than the nearest school. If we assume that students can take the Euclidean path in Figure 9.5.3, the area in which students cannot go to their nearest schools is given by the hatched region in Figure 9.5.3(a), where the polygons are the Voronoi diagram generated by the school sites. The locational optimization problem is to relocate the schools so that the number of students who cannot go to their nearest schools is minimized. This minimization problem is equivalent to the maximization problem of expression (9.5.3), where $\phi(x)$ is given by the density of students. Figure 9.4.5 shows the locally optimal locations obtained by Suzuki and Iri (1986a). The area in which students cannot go to their nearest schools reduces from 20% to 10% by this relocation.