Journal of the Operations Research
Society of Japan
Vol. 29, No. 1, March 1986

APPROXIMATION OF A TESSELLATION OF THE PLANE BY A VORONOI DIAGRAM

Atsuo Suzuki
University of Tokyo

Masao Iri
University of Tokyo

(Received September 18, 1985: Revised January 17, 1986)

Abstract In this paper the problem of obtaining the Voronoi diagram which approximates a given tessellation of the plane is formulated as the optimization problem, where the objective function is the discrepancy of the Voronoi diagram and the given tessellation. The objective function is generally non-convex and nondifferentiable, so we adopt the primitive descent algorithm and its variants as a solution algorithm. Of course, we have to be content with the locally minimum solutions. However the results of the computational examples suggest that satisfactory good solutions can be obtained by our algorithm. This problem includes the problem to restore the generators from a given Voronoi diagram (i.e., the inverse problem of constructing a Voronoi diagram from the given points) when the given diagram is itself a Voronoi diagram. We can get the approximate position of the generators from a given Voronoi diagram in practical time; it takes about 10s to restore the generators from a Voronoi diagram generated from thirty-two points on a computer of speed about 17 MIPS. Two other practical examples are presented where our algorithm is efficient, one being a problem in ecology and the other being one in urban planning. We can get the Voronoi diagrams which approximate the given tessellations (which have 32 regions and are defined by 172 points in the former example, 11 regions and 192 points in the latter example) within 10s in these two examples on the same computer.

1. Introduction

The Voronoi diagram has been recognized as a concept of fundamental importance in many kinds of problems in geometry, urban planning, environmental control, physics, biology, ecology, numerical analysis, etc. [8]. The computational problem of constructing the Voronoi diagram in the plane has been one of the main subjects of computational geometry, and many algorithms have been proposed. Recently, our research group developed a practical fast algorithm to construct a Voronoi diagram for n points in linear time, i.e., $O(n)$ on the average [9], [10], although its worst-case time complexity $O(n^2)$ is inferior to the theoretically optimal complexity $O(n \log n)$ of divide-and-conquer type algorithms.

This fast algorithm has made it possible to solve a class of location problems numerically within a practicable time, which had been thought to be far from being practically solvable because it needs many subroutine calls for the Voronoi diagram construction [7]. We call such a class of location problems geographical optimization problems.

In [7], the problem was formulated and solved as a most common geographical optimization problem, which is to obtain the locations of facilities in such a way that the total cost of people who enjoy the service from the facilities is minimized under the assumption that people should always access the nearest facility, i.e., the problem of minimizing

$$F(x_1,...x_n)= \int f(\min_i \|x-x_i\|)\phi(x)d^N x$$

where x_i $(i=1,...,n)$ is the locations of facilities, and $\|\cdot\|$ represents the Euclidean distance, f is the function repersenting the relation between distance and cost, and $\phi(x)$ is the function representing the population density.

In this paper, we formulate another type of geographical optimization problem, i.e., the problem of obtaining the Voronoi diagram which best approximates the given tessellation of the bounded subset of R^N as the minimization problem with the discrepancy between the given tessellation and the Voronoi diagram as the objective function. We propose a method to get a solution —— a method which belongs to a class of techniques we call the geographical optimization method. Computational results are shown and discussions are given.

The first case in which our method should be efficient is that the given diagram is itself known a priori to be a Voronoi diagram. The problem is to restore the generators from the given Voronoi diagram, that is, the inverse problem of constructing the Voronoi diagram from the given points. For this problem itself, geometrical approaches have been proposed as will be shown in section 2. If the exact Voronoi diagram were given, we could determine the position of the generators by such an elementary geometrical method. However, such a situation is unrealistic. Even if theoretical consideration tells us that the diagram which appears in a phenomenon should be a Voronoi diagram, the error in observation process must perturb the original diagram. Therefore, the geometrical method would always tell us that the diagram is not the Voronoi, i.e., it

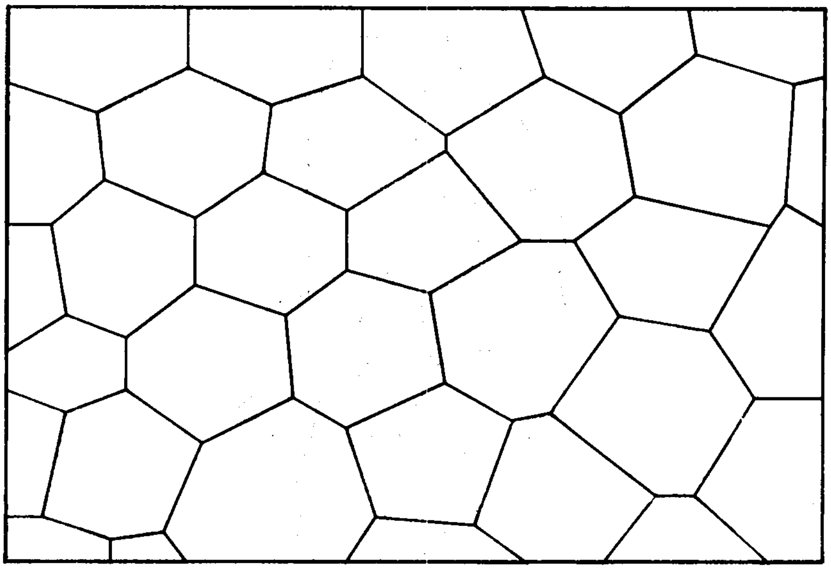


Fig. 1. Territories formed by *Tilapia mossambica* [13]

would give us no information in almost every case. However the method proposed in this paper always tells us at least approximate positions of the generators. Figure 1 is an example of this case. It is taken from [13, Fig. 1], which is a schematic diagram of the photograph in [3, Fig. 1]. The latter is the photograph of the sand pattern formed by male mouth breeder fish, *Tilapia mossambica*, kept in a large outdoor pool with an initially uniform sand floor. *Tilapia mossambica* excavates breeding pits by spitting sand away from the pit center toward his neighbors, then reciprocal spitting results in sand parapets, which are conspicuous territorial boundaries. These facts suggest that this diagram might be a Voronoi diagram.

The second case is the problem of voting precincts and school districts. In these problems, all the people living in an area have to access the facility (polling place or school) determined by administrative condition for enjoying the service. Therefore, if each voting precinct (school district) is the Voronoi region belonging to the polling place (school), these voting precincts (school districts) are equitable because people enjoy the service from the nearest facility. The discrepancy between the present voting precincts (school districts) and the Voronoi diagram may be an index of the equitableness in that sense [8]. Figure 2 is the junior high school districts of Tsukuba in Japan.

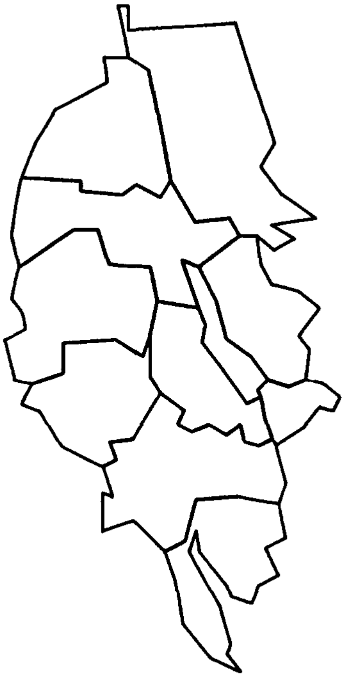


Fig. 2. Junior high school districts in Tsukuba

Judging from these examples, it is worth while in practice to consider the Voronoi diagram approximating a given tessellation. We apply our geographical optimization method to these examples in section 5.

2. A Geometrical Method to Restore the Generators from the Given Voronoi Diagram

First we show the formal definition of the Voronoi diagram. $P(x)$ denotes a point in the N-dimensional Euclidean space R^N, where x is an N-dimensional vector $(x^1, x^2, \ldots, x^N)$. For n distinct points $P_1(x_1)$, $P_2(x_2)$, ..., $P_n(x_n)$ given in R^N,

(2.1) $$V_i = \bigcap_{j:j\neq i} \{x \in R^N \mid \|x-x_i\| < \|x-x_j\|\}$$

is the set of points in R^N which are closer to $P_i(x_i)$ than to any other $P_j(x_j)$ $(j\neq i)$, where $\|\cdot\|$ denotes the Euclidean distance. V_i is a convex set because it is the intersection of half spaces. V_1, V_2, ..., V_n partition R^N into n convex regions in the sense that we have

(2.2) $$\bigcup_{i=1}^{n} V_i = R^N \text{ and } V_i \cap V_j = \phi \ (i\neq j),$$

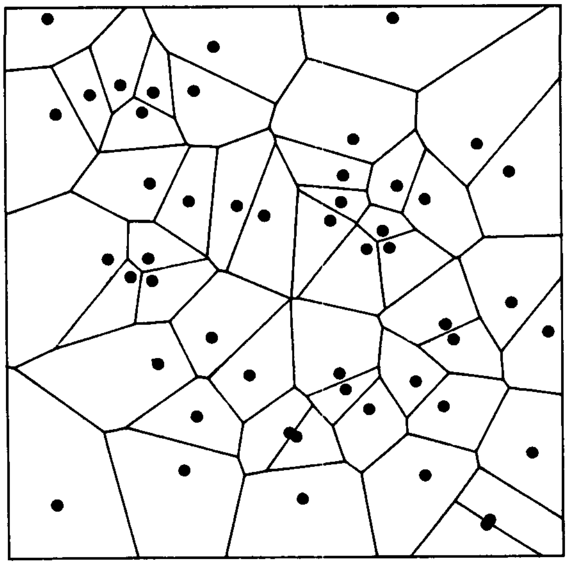


Fig. 3. An example of the Voronoi diagram (with 50 generators)

where $\bar{A}$ denotes the topological closure of set A. The partition determines in an obvious manner a polyhedral complex, which is called the Voronoi diagram for the given n points P_i (x_i)'s. This partition is also called the Dirichlet tessellation or the Thiessen tessellation [5], [12], [14], [15]. We sometimes call $P_i(x_i)$ $(i=1,\ldots,n)$ generators. Each V_i $(i=1,\ldots,n)$ is a kind of "territory" of point $P_i(x_i)$ $(i=1,\ldots,n)$ and is called the Voronoi region of P_i (x_i) $(i=1,\ldots,n)$. In the two-dimensional case, $N=2$, the vertices of the polygonal Voronoi region are called the Voronoi points and the edges the Voronoi edges. Figure 3 is an example of the Voronoi diagram with $n=50$ generators.

It is easy to obtain the generators from the given Voronoi diagram in the two-dimensional case by purely geometrical method [8, p.100]. This method is based on the geometrical property of the Voronoi diagram given below: In Fig. 4, P_1, P_2 and P_3 are generators; Q_1 is a Voronoi point which is the circumcenter of $\Delta P_1P_2P_3$, Q_2, Q_3 and Q_4 are the neighboring Voronoi points. Let

$$\angle Q_2Q_1Q_3 = \alpha$$

then

$$\angle P_1P_3P_2 = \pi-\alpha$$

and from the theorem of the angle at circumference

$$\angle P_1Q_1Q_4 = \angle P_2Q_1Q_4 = \pi-\alpha.$$

Therefore, if we are given an exact Voronoi diagram, a generator can be determined as the intersection of rays such as r_1 and r_2 in Fig. 5

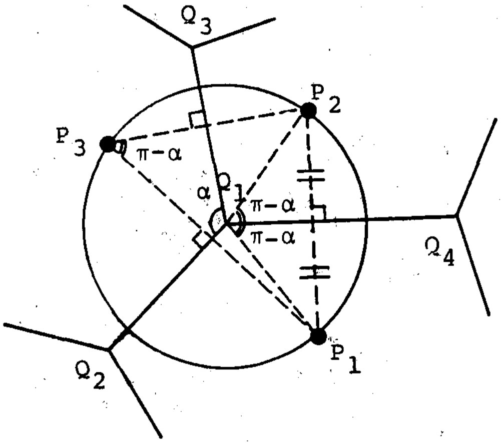


Fig. 4. Properties of the Voronoi diagram
(P_1, P_2, P_3 : generators; Q_1, Q_2, Q_3, Q_4 : Voronoi points)

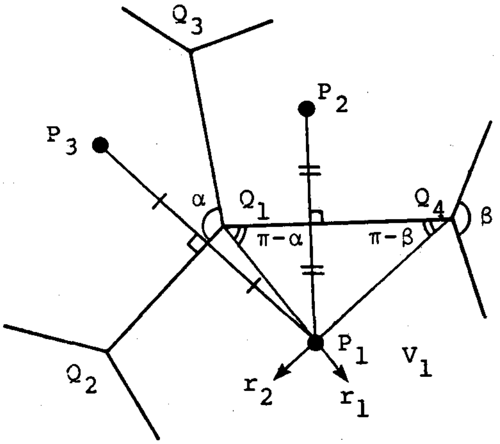


Fig. 5. The geometrical method to obtain the generators from the Voronoi diagram

emanating from the endpoints of a Voronoi edge, for example, Q_1 and Q_4. Once the generator P_1 of a Voronoi region V_1 is obtained, we can get the generators of the Voronoi regions which share a Voronoi edge in common with V_1 as the mirror images of P_1 with respect to the Voronoi edges bounding V_1. Then, by repeating this procedure, all the generators can be determined --- at least in principle. Furthermore, it is proved in [2] that a proper convex plane tessellation, all of whose vertices have degree 3, is the Voronoi diagram if and only if all such rays as r_1, r_2

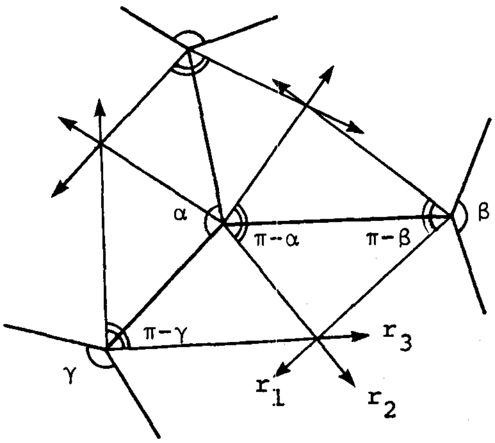


Fig. 6. A necessary and sufficient condition for a tessellation on the plane to be a Voronoi diagram

and r_3 shown in Fig. 6 have a point in common for each region. Using this condition, we can determine whether a given tessellation is the Voronoi diagram or not. However, in practical situations, we would hardly have a chance to be given an exact Voronoi diagram. Even if the diagram is known to be the Voronoi diagram from the theoretical point of view, the errors in observation process would perturb the shape of the original diagram. Thus in practical situations, nothing can be obtained from the geometrical method explained above.

3. Problem Formulation and a Solution Algorithm

Let $P_i(x_i)$ $(i=1,\ldots,n)$ be the generators of a Voronoi diagram and $\phi(x)$ be a positive, finite-valued and smooth function defined on a bounded subset $\overline{\bigcup A_j}$ $(A_i \cap A_j=\emptyset\ (i\neq j))$ of the N-dimensional Euclidean space R^N. Intuitively, $\phi(x)$ is thought to represent a population density in practical situations. Our objective function is the discrepancy

$$(3.1) \qquad F(x_1,\ldots,x_n)=\sum_{i\neq j}\int_{V_i\cap A_j}\phi(x)d^Nx$$

between the given tessellation $\{A_j\}_{j=1}^n$ of the bounded subset of R^N, and the Voronoi diagram $\{V_i\}_{i=1}^n$ generated by $P_i(x_i)$ $(i=1,\ldots,n)$, the discrepancy being measured with $\phi(x)$ as the weighting function. Before entering into the discussion of the solution algorithm, we should make some observation on the properties of the objective function F of (3.1). F qua function of the Nn vector X:

$$X=(x_1', x_2', \ldots, x_n')'$$

is generally non-convex, and has nondifferentiable points. In fact, it has a local minimum which is not the global minimum, such as shown in Fig. 7. In Fig. 7, $\{A_j\}_{j=1}^{n}$ is a Voronoi diagram so that the minimum value of F should be 0. However, if $\tilde{X}=(\tilde{x}_1', \ldots, \tilde{x}_n')'$ be the exact solution constituted from the coordinates of $P_i(x_i)$'s generating the Voronoi diagram, and if we put $X=(x_1', \ldots x_n')'$, where $x_l=\tilde{x}_m$, $x_m=\tilde{x}_l$, and $x_i=\tilde{x}_i$ $(i\neq l, m)$ (A_l is not adjacent to A_m), then F is one of the local minima which is not global minimum because F is not equal to 0 and any small change of X increases the discrepancy. Figure 8 shows one of the typical cases of nondifferentiable points of F (see the legenda of the Figure).

Next we note that the minimization of F is equivalent to the maximization problem of

$$F_1=\sum_{i=1}^{n}\int_{V_i\cap A_i}\phi(x)d^N x.$$

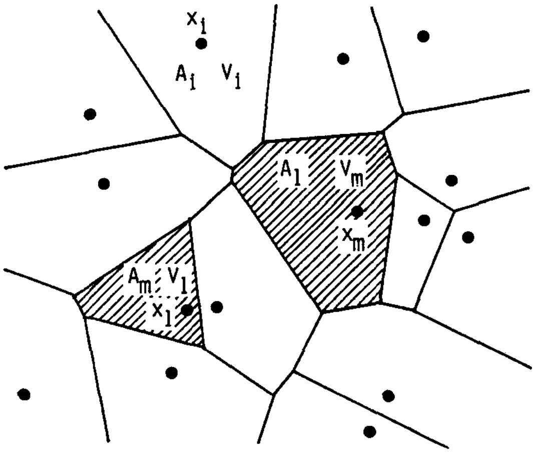


Fig. 7. An example of local minima of F
(Shaded areas represent the discrepancy.)

It is easy to show that

$$\bigcup_{\substack{i,j\\ i\neq j}}(V_i\cap A_j)=\bigcup_{j=1}^{n}\{(\bigcup_{i:i\neq j}V_i)\cap A_j\}=\bigcup_{j=1}^{n}\{(\bigcup_{i=1}^{n}A_i-V_j)\cap A_j\}$$

$$=\bigcup_{j=1}^{n}A_j-\bigcup_{j=1}^{n}(V_j\cap A_j).$$

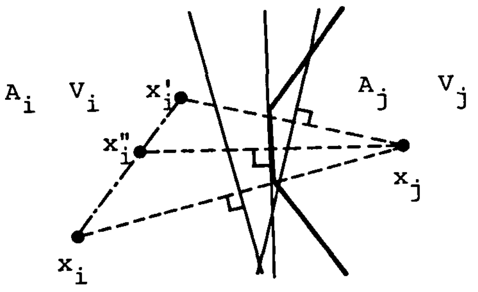


Fig. 8. An example of nondifferentiable points of F
If x_i moves toward x'_i, the partial derivative of F with respect to x_i changes discontinuously at x''_i.

Thus, we have

$$(3.2)\qquad F(x_1,\ldots,x_n)=\int_{\bigcup_{j=1}^{n} A_j}\phi(x)\mathrm{d}^N x-\sum_{i=1}^{n}\int_{V_i\cap A_i}\phi(x)\mathrm{d}^N x.$$

The first term of (3.2) is the total measure of $\bigcup A_j$ which is constant and the second is F_1, the coincidence of $\{A_i\}_{i=1}^{n}$ and $\{V_i\}_{i=1}^{n}$.

Since no general optimization algorithm is available at present which works for such functions better than a most primitive class of descent methods, we have to resort to a variant of primitive descent algorithms. We also have to be content with one of local minima. Thus we shall investigate the algorithms of the following type:

[Algorithm] Starting with a given initial guess $X^{(0)}$, repeat (1)-(3) for $\nu=0,1,2,\ldots$ until some stopping criterion is satisfied.

(1) Search direction:--- Compute the gradient $\nabla F(X^{(\nu)})$ of F at the ν-th approximate solution $X^{(\nu)}$. Then determine the search direction $d^{(\nu)}$ using $\nabla F(X^{(\nu)})$ and some other auxiliary quantities if we want.

(2) Line search:--- Determine $\hat{\alpha}^{(\nu)}$ (up to a certain degree of approximation) such that

$$F(X^{(\nu)}+\hat{\alpha}^{(\nu)}d^{(\nu)})=\min_{\alpha} F(X^{(\nu)}+\alpha d^{(\nu)}).$$

(3) New approximation:--- Set

$$X^{(\nu+1)}=X^{(\nu)}+\omega\hat{\alpha}^{(\nu)}d^{(\nu)}.$$

Here, ω is an acceleration factor to avoid undesirable stagnation at nondifferentiable points [6], [11].

There are a number of variants of the algorithm of the above type with different choices of the search direction in (1), of the acceleration factor in (3) and of the stopping rule. We have tested several variants as will be described in section 5.

4. Calculation of the Partial Derivatives

Since the objective function in (3.1) is not familiar in form, it may not be useless to explicitly write down its partial derivatives (the gradient and the Hessian). In so doing, we adopt the tensor notation in R^N in order to keep the geometrical meanings of the relevant expressions as clear as possible. Thus, we denote by $g_{\lambda\kappa}$ the metric tensor in R^N and adopt Einstein's summation convention. For example, the inner product of two vectors x, y in R^N is expressed as

$$(4.1) \qquad (x,y)= g_{\lambda\kappa}x^{\kappa}y^{\lambda} \quad (= \sum_{\kappa=1}^{N} \sum_{\lambda=1}^{N} g_{\lambda\kappa}x^{\kappa}y^{\lambda}).$$

We need notation for the Voronoi diagram $\{V_i\}_{i=1}^{n}$ and the given tessellation $\{A_j\}_{j=1}^{n}$. The $(N-1)$-dimensional face bounding two adjacent Voronoi regions V_i and V_j is denoted by

$$(4.2) \qquad W_{ij}=\partial V_i \cap \partial V_j$$

and the intersection of W_{ij} or W_{ji} and A_i by

$$(4.3) \qquad L_{i,j}=W_{ij} \cap A_i$$

(see Fig. 9). $L_{i,j}$ is of essential importance when we calculate the partial derivatives because F varies as $L_{i,j}$'s move. It is easily shown that $L_{i,j}=\emptyset$ when $W_{ij}=\emptyset$ or $V_i \cap A_i=\emptyset$. The distance between two generators x_i and x_j will be denoted by

$$(4.4) \qquad \alpha_{ij}=\|x_i-x_j\| .$$

The partial derivative of the F of (3.1) with respect to x_i is due to the variation of the regions $V_i \cap A_j$ $(i \neq j,\ W_{ij} \neq \emptyset)$.

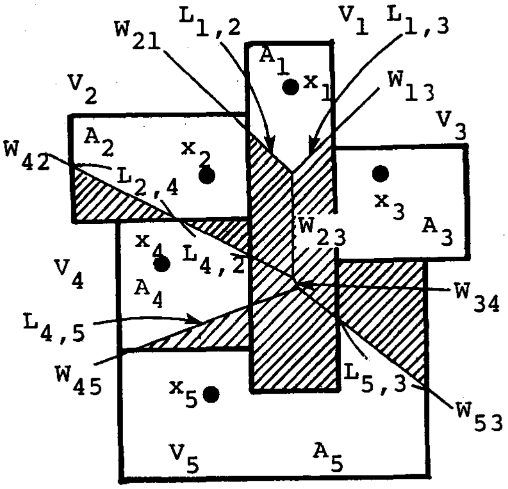


Fig. 9. Components of $\{V_i\}$ and $\{A_j\}$

——— : boundary of V_i and V_j (W_{ij})

▬▬▬ : boundary of A_i and A_j

(Shaded areas represent the discrepancy.)

$$(4.5)\qquad \frac{\partial F}{\partial x_i^\lambda} = \sum_{j:\mathrm{W}_{ij}\neq\emptyset} \Big\{\int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}(x_i^\kappa - x^\kappa)\phi(x)\,\mathrm{d}^{N-1}x - \int_{\mathrm{L}_{j,i}} \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}(x_i^\kappa - x^\kappa)\phi(x)\,\mathrm{d}^{N-1}x\Big\}.$$

The equation (4.5) can be rewritten into the form,

$$(4.6)\qquad \frac{\partial F}{\partial x_i^\lambda} = \sum_{j:\mathrm{W}_{ij}\neq\emptyset} \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}\{\mu(\mathrm{L}_{i,j})(x_i^\kappa - \bar{x}_{i,j}^\kappa) - \mu(\mathrm{L}_{j,i})(x_i^\kappa - \bar{x}_{j,i}^\kappa)\},$$

where $\mu(\mathrm{L}_{i,j}) = \int_{\mathrm{L}_{i,j}} \phi(x)\,\mathrm{d}^{N-1}x$ is the "length" of $\mathrm{L}_{i,j}$ and $\bar{x}_{i,j}^\kappa = \int_{\mathrm{L}_{i,j}} x^\kappa \phi(x)\,\mathrm{d}^{N-1}x/\mu(\mathrm{L}_{i,j})$ is the "centroid" of $\mathrm{L}_{i,j}$, each defined with respect to the weight $\phi(x)$.

The second derivatives of F may be calculated in a similar vein but with more complicated manipulation of formulas as follows. (See Appendix for detailed derivation.)

$$(4.7)\qquad \frac{\partial^2 F}{\partial x_j^\kappa \partial x_i^\lambda} = \begin{cases} \displaystyle\sum_{k:\mathrm{W}_{ik}\neq\emptyset} \{(H_{\lambda\kappa}^{ik} - H_{\lambda\kappa}^{ki}) + G_{\lambda\kappa}^{iki}\} & (j=i), \\ \displaystyle G_{\lambda\kappa}^{ijj} + \sum_{k:\mathrm{W}_{ij}\cap\partial\mathrm{V}_k\neq\emptyset} K_{\lambda\kappa}^{ijk} & (j\in\{l\,|\,l\neq i,\ \mathrm{W}_{il}\neq\emptyset\}), \\ 0 & (\text{otherwise}), \end{cases}$$

where

$$(4.8)\qquad H_{\lambda\kappa}^{ij} = \int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}\, \phi(x)\,\mathrm{d}^{N-1}x,$$

$$\begin{aligned}(4.9)\qquad G_{\lambda\kappa}^{ijk} = &-\int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}^3}\, g_{\lambda\nu}(x_i^\nu - x^\nu)\, g_{\kappa\mu}(x_i^\mu - x_j^\mu)\,\phi(x)\,\mathrm{d}^{N-1}x \\ &+\int_{\mathrm{L}_{j,i}} \frac{1}{\alpha_{ij}^3}\, g_{\lambda\nu}(x_i^\nu - x^\nu)\, g_{\kappa\mu}(x_i^\mu - x_j^\mu)\,\phi(x)\,\mathrm{d}^{N-1}x \\ &+\int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}^3}\, g_{\lambda\nu}(x_i^\nu - x^\nu)\, g_{\kappa\mu}(x_k^\mu - x^\mu)\frac{\partial\phi}{\partial x^\xi}(x_j^\xi - x_i^\xi)\,\mathrm{d}^{N-1}x \\ &-\int_{\mathrm{L}_{j,i}} \frac{1}{\alpha_{ij}^3}\, g_{\lambda\nu}(x_i^\nu - x^\nu)\, g_{\kappa\mu}(x_k^\mu - x^\mu)\frac{\partial\phi}{\partial x^\xi}(x_j^\xi - x_i^\xi)\,\mathrm{d}^{N-1}x\end{aligned}$$

$$+\int_{\partial L_{i,j}} \frac{1}{\alpha_{ij}^2}\frac{1}{\tan\theta(x)} g_{\lambda\nu}(x_i^\nu - x^\nu) g_{\kappa\mu}(x_k^\mu - x^\mu)\phi(x)\,d^{N-2}x$$

$$-\int_{\partial L_{j,i}} \frac{1}{\alpha_{ij}^2}\frac{1}{\tan\theta(x)} g_{\lambda\nu}(x_i^\nu - x^\nu) g_{\kappa\mu}(x_k^\mu - x^\mu)\phi(x)\,d^{N-2}x,$$

$$(4.10)\quad K^{ijk}_{\lambda\kappa} = \int_{L_{i,j}\cap\partial V_k} \frac{1}{\alpha_{ij}} g_{\lambda\nu}(x_i^\nu - x^\nu)\frac{\beta(x)}{\alpha_{ik}\alpha_{jk}} g_{\kappa\mu}(x_k^\mu - x^\mu)\phi(x)\,d^{N-2}x$$

$$-\int_{L_{j,i}\cap\partial V_k} \frac{1}{\alpha_{ij}} g_{\lambda\nu}(x_i^\nu - x^\nu)\frac{\beta(x)}{\alpha_{ik}\alpha_{jk}} g_{\kappa\mu}(x_k^\mu - x^\mu)\phi(x)\,d^{N-2}x.$$

The $\theta(x)$ in (4.9) is the angle between the hyperplane containing W_{ij} and that containing $\bar{A}_i \cap \bar{A}_j$ at each point x on $\partial L_{i,j}$, and the $\beta(x)$ in (4.10) is

$$\beta(x) = \{g_{\lambda\kappa}(x_k^\kappa - x^\kappa)(x_k^\lambda - x^\lambda)\}^{1/2}.$$

5. Numerical Examples

In the following numerical examples we deal with the two-dimensional case (N=2) with metric tensor

$$g_{\lambda\kappa} = \delta_{\lambda\kappa} \quad (\lambda,\kappa=1,2)$$

where the density function $\phi(x)$ is equal to 1 in $\bigcup A_j$ and vanishes outside it.

We performed a number of experimental computations with two different kinds of search directions using various acceleration factors in §5.1 and §5.2. As the simpler search direction $d^{(\nu)}$, we took the direction of steepest descent (abbreviated as S):

$$(5.1)\quad \text{S:}\ d^{(\nu)} = -\nabla F(X^{(\nu)}).$$

In the terminology of tensor analysis, $d^{(\nu)}$ is a contravariant vector whereas $\nabla F(X^{(\nu)})$ is a covariant vector. Hence, the steepest descent direction does not have an invariant meaning under the $2n$-dimensional affine transformation and that even under the rescaling of coordinate axes in R^2. We also investigated a more sophisticated direction (abbreviated as M):

$$(5.2)\quad \text{M:}\ d^{(\nu)} = -H^{-1}\nabla F(X^{(\nu)}),$$

which is obtained by modifying the steepest descent direction with the following approximation H of the Hessian ($2n\times 2n$ matrix) of F. Since the exact Hessian (see (4.7)) is too complicated to incorporate in the iteration process, we adopted as the approximate Hessian

$$(5.3) \qquad H^{i}_{\lambda\kappa} = \sum_{j:W_{ij}\neq\emptyset} (H^{ij}_{\lambda\kappa}+H^{ji}_{\lambda\kappa}).$$

It is numerically not a good approximation, but is a symmetric and positive-definite covariant tensor of valence 2, having the same tensorial character as the exact Hessian. Therefore, descent direction (5.2) is invariant under the 2-dimensional rescaling.

It is difficult to compare theoretically these two kinds of descent directions from the viewpoint of computational efficiencies, but it will be good for a numerical method to have such a property of invariance. In fact, it is reported in [7] that the descent direction M is superior to S with respect to computational time for another kind of geographical optimization problem.

For the line search, we adopted the so-called "Goldstein method" [4], which chooses $\alpha^{(\nu)}$ so as to satisfy the inequalities

$$(5.4) \qquad \mu_2 d^{(\nu)}\nabla F(X^{(\nu)}) < F(X^{(\nu)}+\alpha^{(\nu)}d^{(\nu)})-F(X^{(\nu)}) < \mu_1 d^{(\nu)}\nabla F(X^{(\nu)})$$

with appropriately prescribed parameters μ_1 and μ_2 ($0<\mu_1<\mu_2<1$). (We chose $\mu_1=0.2$ and $\mu_2=0.8$ throughout our experiments.)

For the selection of the acceleration factor ω, we investigated the speed of convergence of the objective function and the properties of the resulting solutions numerically for various values of ω ranging from 1.0 to 2.6. The integrals in the expressions of F, ∇F and H were computed by means of numerical quadrature formulas: The integration on $V_i \cap A_j$ ($i\neq j$) was done with the seven-point formula of the Gaussian type for a triangle given in [1, p.893] and that on $L_{i,j}$ with the three point Gaussian formula. HITAC M-280H (about 17 MIPS with array processor) at the Computer Centre of the University of Tokyo was used in FORTRAN throughout the experiments.

5.1. The inverse problem of the Voronoi diagram construction

We applied our algorithm to the problem of restoring the generators from a given tessellation which is known to be a Voronoi diagram. We

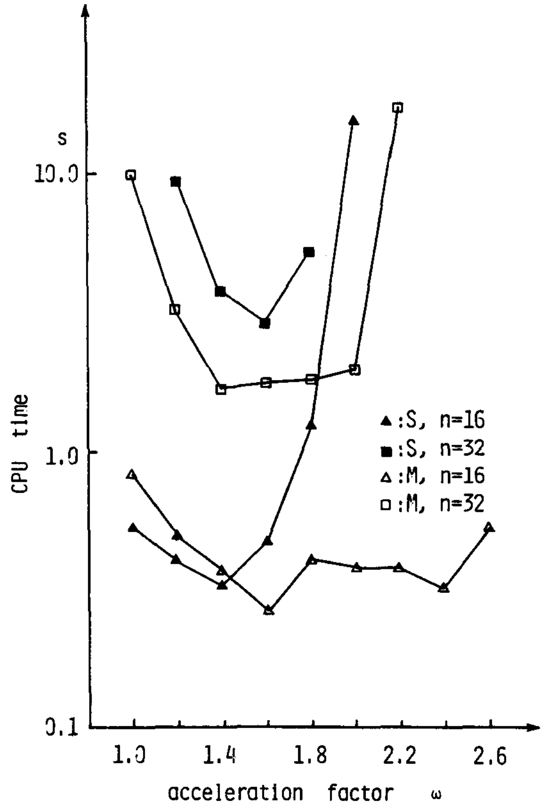
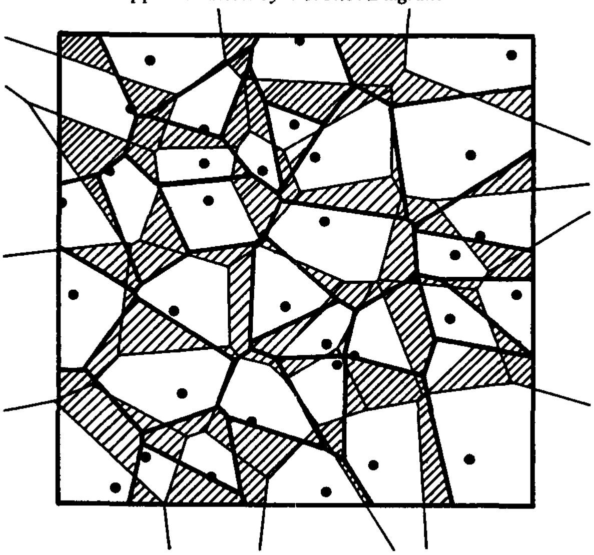


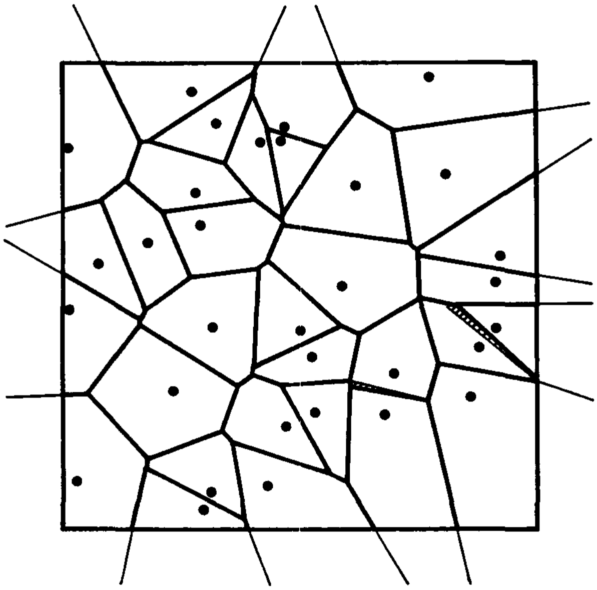
Fig. 10. CPU time for obtaining the solution (average of five different initial guesses)
$\{A_j\}$: a Voronoi diagram
Stopping criterion: $F \leq 0.01\times($ area of $\bigcup A_j)$
S : steepest descent direction (5.1)
M : modified direction using H (5.2)

adopted for sample tessellations two Voronoi diagrams generated from sixteen and thirty-two points, respectively, distributed in the unit square (-0.5,0.5)×(-0.5,0.5), and investigated the effect of the descent direction (S or M) and the acceleration factor ω (varying from 1.0 to 2.6 by 0.2) on the speed of the convergence. We measured the CPU times needed, starting with an initial guess $x_i^{(0)}$ randomly located in each A_i, to have the discrepancy between the given Voronoi diagram and the solution reduced to 1.0 % of the total area of $\bigcup A_j$. The plots in Fig. 10 are the average CPU times on five different initial guesses for different search directions and acceleration factors. From Fig. 10, it is seen that as to the search direction, M is superior to S in the large and is less sensitive than S to the choice of acceleration factor.



(a) $\{A_j\}_{j=1}^{n}$: given tessellation (thick lines)

$\{V_i\}_{i=1}^{n}$: initial Voronoi diagram for generators (●) each chosen in A_j (thin lines)



(b) $\{A_j\}_{j=1}^{n}$: given tessellation (thick lines)

$\{V_i\}_{i=1}^{n}$: approximate solution Voronoi diagram when the discrepancy (shaded areas) has been reduced to 1.0 % of the area of $\cup A_j$ (thin lines)

Fig. 11. Initial guess (a) and obtained solution (b) ——— $\{A_j\}$ is the Voronoi diagram generated by thirty-two points randomly distributed in the unit square.

Figure 11 is the example when the given Voronoi diagram is generated from the randomly distributed thirty-two points in the unit square: (a) is the initial guess $x_i^{(0)}$ randomly located in each A_i, and (b) is the nearly optimum solution when the discrepancy is reduced to 1.0 % of the area of $\cup A_j$.

5.2. A small but practical example

As another sample tessellation $\{A_j\}_{j=1}^{n}$, we took part of Fig. 1 of section 1 (the territories of *Tilapia mossambica*) which has ten regions in the rectangular area $(0,4.3)\times(0,3.5)$, and investigated how well $\{A_j\}_{j=1}^{n}$ can be approximated by a Voronoi diagram. In section 4 we have shown that our algorithm yields in general a local minimum but not always the global. Thus the solution we ultimately have is expected to be highly dependent on the descent direction, the acceleration factor and the initial guess. In order to examine this dependence numerically, we took five different initial guesses by choosing each $x_i^{(0)}$ randomly in A_i, and for each initial guess, we applied the variation of our algorithm with search directions S and M and with different acceleration factors ω ranging from 1.0 to 2.0 (step 0.2). Iteration was continued until we have either

$$\max_{\kappa,i}|x_i^{\kappa(\nu+1)}-x_i^{\kappa(\nu)}| < 1.0\times10^{-2}$$

or

$$\nu=50.$$

Hence, we had 60 "solutions" in all, among which the solution with the 4th initial guess, search direction M and acceleration factor $\omega=1.4$ gave the smallest value, $F_{\min}=0.4472$ of the objective function. (Note that the total area $\mu(\cup A_j)=15.05$.) The plots in Fig. 12 show the value $(F-F_{\min})/F_{\min}$ for each initial guess, each search direction and each value of ω, where it is seen that the solution depends on the initial guess considerably. Thus it seems important to start with a physically meaningful initial guess. How to do it depends on the problem (see also §5.3 and §5.4). We furthermore tested another fifteen initial guesses, but no solution gave a value of the objective function less than $F_{\min}$. This would mean that it is not of much use to repeat solutions starting with randomly chosen initial guesses but we had better start with a few physically plausible initial guesses, which are chosen, for example, by inspection.

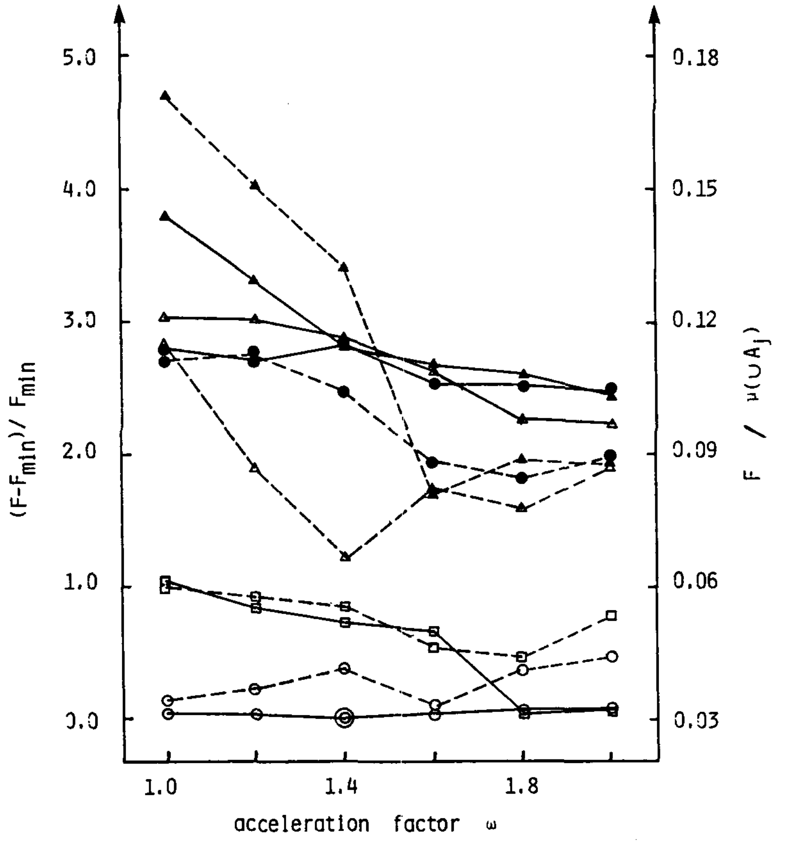


Fig. 12. The value of $(F-F_{min})/F_{min}$ (left axis) and $F/\mu(\cup A_j)$ (right axis) for a part of Fig. 1

(F_{min} is the minimum value of F among the obtained solutions.)

▲ ● □ ○ △ : five different initial guess

- - - - - - - : steepest descent direction S (5.1)

———— : modified direction M (5.2)

5.3. Territories of *Tilapia mossambica*

We adopted Fig. 1 in section 1, the territories of *Tilapia mossambica*, as $\{A_j\}_{j=1}^{n}$. The density $\phi(x)$ is 1 if $x \in \cup A_j$ and otherwise 0. The number n of territories is equal to thirty-two and the number of points defining the tessellation is equal to 172. In [13], the distribution of the angle of Fig. 1 was compared statistically with the distribution of the angles of the Voronoi diagram obtained from the computer simulation under some mathematical model in order to back up the validity of the model proposed. Our method should offer a way to compare

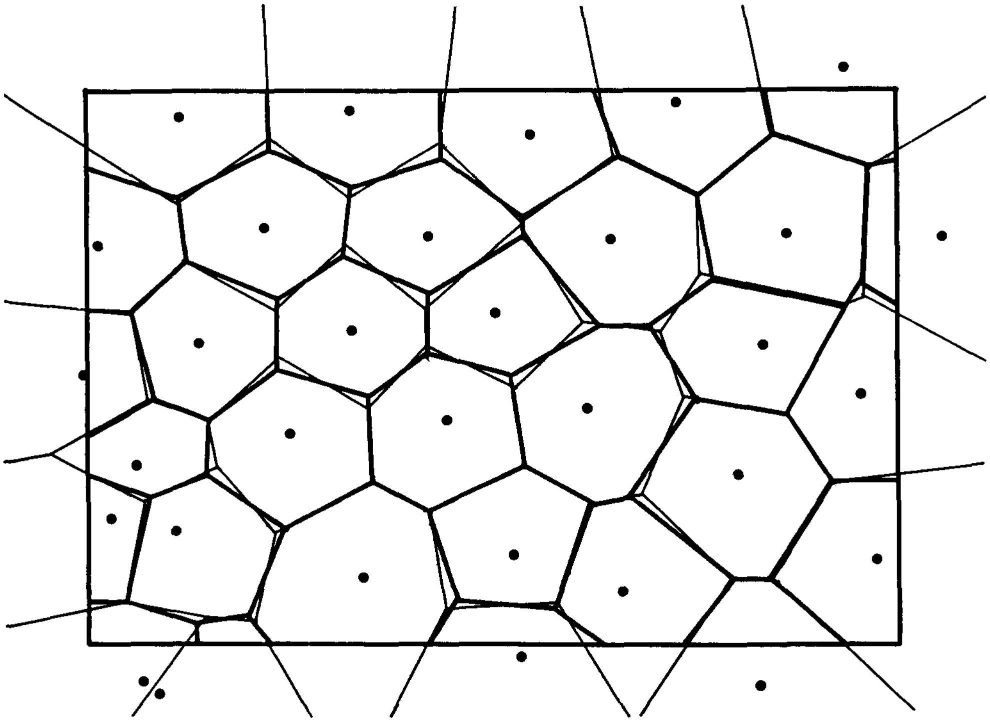


Fig. 13. Territories of *Tilapia mossambica* (thick lines) and solution Voronoi diagram after 50 iterations (thin lines) (● ; generators)

Fig. 1 directly with the Voronoi diagram. Starting with an appropriate initial guess obtained by inspection, we got the Voronoi diagram of Fig. 13 using descent direction M and $\omega = 1.2$ after 50 iterations. The discrepancy between the tessellation of Fig. 1 and the Voronoi diagram was reduced from 24.8 % to 3.8 % of the total area $\mu(\cup A_j)$. Computation time for one iteration was about 0.15s. This result tells us that Fig. 1 in [3] may be regarded approximately as a Voronoi diagram.

5.4. School districts in Tsukuba

As the last example we took the school districts of junior high schools in Tsukuba as $\{A_j\}_{j=1}^{n}$ (Fig. 2). The density $\phi(x)$ is 1 if $x \in \cup A_j$ and otherwise 0. The number n of the districts is eleven and the number of points defining $\{A_j\}$ is 192. We adopted the descent direction M and $\omega = 1.4$. Starting with the present locations of those junior high schools as the initial guess, we could reduce the discrepancy between $\{V_i\}_{i=1}^{n}$ and $\{A_j\}_{j=1}^{n}$ from 20 % to 10 % of the total area $\mu(\cup A_j)$ after 20 iterations. Computation time was 65-70 ms for one iteration (Fig. 14).

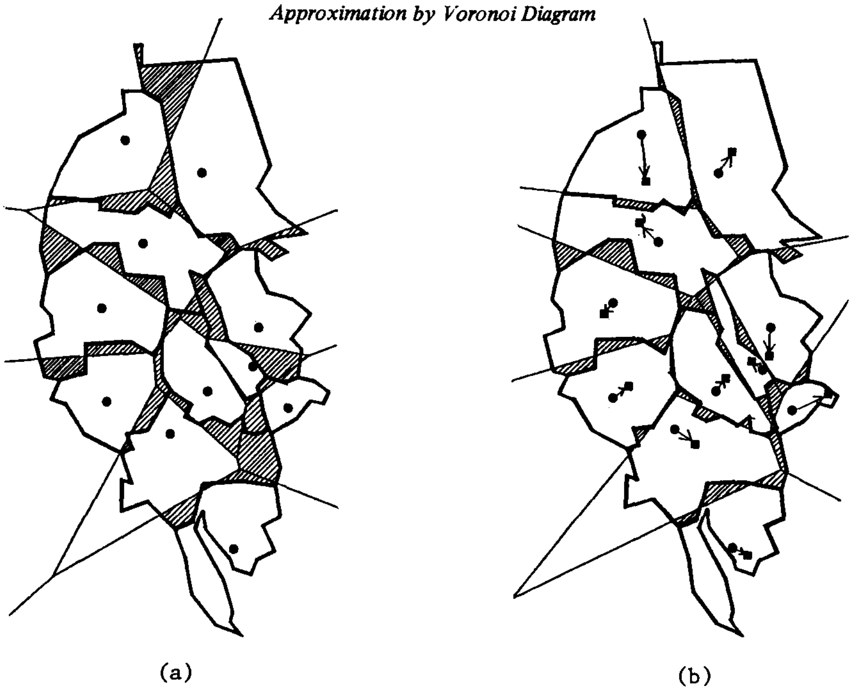


Fig. 14. Junior high schools of Tsukuba and their school districts

(a) $\{A_j\}_{j=1}^{n}$: school districts of Tsukuba (thick lines)
$\{V_i\}_{i=1}^{n}$: initial Voronoi diagram (thin lines) generated by the present junior high schools (●)
Discrepancy (shaded area) = 20 % of $\mu(\cup A_j)$.

(b) $\{A_j\}_{j=1}^{n}$: school districts of Tsukuba (thick lines)
$\{V_i\}_{i=1}^{n}$: solution Voronoi diagram after 20 iterations (thin lines)
● : present locations of junior high schools
■ : generators of the solution Voronoi diagram
Discrepancy (shaded area) = 10 % of $\mu(\cup A_j)$.

6. Conclusions and Discussions

The problem which minimizes the discrepancy between a given tessellation of a bounded subset of R^2 and a Voronoi diagram has been formulated, and a practical algorithm for approximately solving it has been proposed. This problem includes as a special case the inverse problem of constructing the Voronoi diagram when the given tessellation is itself a Voronoi diagram. We have shown that our algorithm is practical also in this case.

From the methodological point of view, the solution obtained by our algorithm is only one of the local minima. However, if we can get the physically meaningful initial guess, it is possible to obtain even the global minimum with appropriately chosen descent direction and acceleration factor. Furthermore, any solution obtained by our method is certainly an improvement on the initial solution. We have shown through examples in §5.1 and §5.2 that the invariant descent direction with respect to the *Nn*-dimensional affine transformation is better than the steepest descent direction in CPU time, sensitivity for the acceleration factor, and quality of the solutions. Also we have shown that it is efficient to use the acceleration factor even in a primitive way such as constant acceleration factor.

How satisfactory the obtained solution is may be evaluated from the standpoint of Operations Research, but not from mathematical consideration alone. For example, in §5.3 we get the solution that the discrepancy between the territories of *Tilapia mossambica* and the Voronoi diagram is 3.8 % of the total area concerned. Although we do not know whether this solution is the global minimum or only one of the local minima, we can see this solution satisfactory by considering the magnitude of errors associated with the fluctuation inherent to the phenomenon and with the physical measurement to make the schematic diagram from the photograph, which is supposed to be of the order at least 3-5 %.

The example in §5.4 might seem unrealistic because it is impossible to relocate the junior high schools. However, it can be said that the solution of this example gives us a quantitative index about the equitableness of the present distribution of the schools and the present definition of the school districts. Furthermore, each region A_i was approximated by one Voronoi region V_i in this example. We can easily extend our method to the case where A_i is approximated by more than one Voronoi region $V_{i1},\ldots,V_{il}(l\geq 2)$. If A_i is approximated well by the union of several Voronoi regions, it is helpful to geographical information processing because the Voronoi diagram has many a nice property for computational analysis [8].

Acknowledgements

The authors thank Dr. Kazuo Murota and Dr. Masaaki Sugihara of the University of Tsukuba for their helpful advice and suggestions, and Mr.

Takao Ohya of Central Research Institute of Electric Power Industry who developed the fast Voronoi diagram algorithm with its program, and members of their research group, especially Mr. Hiroshi Imai for their counsel and assistance. Also they are grateful to Professor Takeshi Koshizuka of the University of Tsukuba for the data of the school districts in Tsukuba and their colleague Mr.Kouji Kurata for his valuable comments.

Appendix. Calculation of the Gradient and the Hessian of F

In this appendix, the detailed derivation of (4.5)-(4.10), the gradient and the Hessian of F, is shown. Before entering into the derivation, we note some fundamental relations for a given tessellation $\{A_j\}$ and the Voronoi diagram $\{V_i\}$ (see Fig. A1):

$$\partial V_i=\bar{V}_i\backslash V_i,\ W_{ij}=\partial V_i\cap\partial V_j,\ L_{i,j}=W_{ij}\cap A_i,\ \partial L_{i,j}=\bar{L}_{i,j}\backslash L_{i,j}.$$

Also we assume that the angle θ between the hyperplane containing W_{ij} and that containing $\bar{A}_i\cap\bar{A}_j$ is known at each point on $\partial L_{i,j}$. Our objective function is

$$\text{(A.1)}\qquad F(x_1,\dots,x_n)=\sum_{i\neq j}\int_{V_i\cap A_j}\phi(x)d^N x.$$

First the gradient of F is considered (see Fig. A2). The hyperplane containing W_{ij} is represented as $(w,\ c)$:

$$\text{(A.2)}\qquad w_\kappa x^\kappa=c \qquad\qquad (g^{\lambda\kappa}w_\kappa w_\lambda=1).$$

The hyperplane $(w,\ c)$ moves to $(w+\delta w\ ,\ c+\delta c)$ corresponding to the increment δx_i of the variable x_i. Let h be the distance between two hyperplanes $(w,\ c)$ and $(w+\delta w,\ c+\delta c)$, then the increment of F caused by the change of $L_{i,j}$ is given by $\int_{L_{i,j}} h\phi(x)d^{N-1}x$.

The hyperplane $(w,\ c)$ is the perpendicular bisector hyperplane of P_iP_j, i.e.,

$$\text{(A.3)}\qquad w_\kappa(x_j^\kappa+x_i^\kappa)/2=c,$$

$$\text{(A.4)}\qquad w_\lambda=\frac{1}{\alpha_{ij}}g_{\lambda\kappa}(x_j^\kappa-x_i^\kappa),$$

where $\alpha_{ij}=\|x_i-x_j\|$. Thus, we have

$$(\mathrm{A}.5)\qquad h=w_\kappa \delta x^\kappa,$$

where δx is the vector, of whose endpoints one is on (w, c) and the other on $(w+\delta w, c+\delta c)$. Eliminating c from (A.2) and (A.3), and differentiating by x_i, we obtain the equation

$$(\mathrm{A}.6)\qquad \delta w_\kappa (x_j^\kappa+x_i^\kappa-2x^\kappa)+w_\kappa(\delta x_i^\kappa-2\delta x^\kappa)=0.$$

Substituting (A.4) and its derivative

$$(\mathrm{A}.7)\qquad \delta w_\lambda= \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}\delta x_i^\kappa$$

into (A.6), we get

$$(\mathrm{A}.8)\qquad w_\kappa \delta x^\kappa= \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}(x_i^\kappa-x^\kappa)\delta x_i^\lambda.$$

Thus, we have

$$(\mathrm{A}.9)\qquad h= \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}(x_i^\kappa-x^\kappa)\delta x_i^\lambda.$$

Therefore the increment of F for $\mathrm{L}_{i,j}$ is

$$(\mathrm{A}.10)\qquad \int_{\mathrm{L}_{i,j}} h\phi_{,}(x)\mathrm{d}^{N-1}x = \int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}} g_{\lambda\kappa}(x_i^\kappa-x^\kappa)\delta x_i^\lambda\phi(x)\mathrm{d}^{N-1}x.$$

By similar calculation for $\mathrm{L}_{j,i}$, we obtain (4.5), i.e.,

$$(\mathrm{A}.11)\qquad \frac{\partial F}{\partial x_i^\lambda} = \sum_{j:\mathrm{W}_{ij}\neq\emptyset}\Big\{\int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}(x_i^\kappa-x^\kappa)\phi(x)\mathrm{d}^{N-1}x \\ - \int_{\mathrm{L}_{j,i}} \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}(x_i^\kappa-x^\kappa)\phi(x)\mathrm{d}^{N-1}x\Big\}.$$

We investigate the increment Δ_λ^1 of the first term in the braces of (A.11) corresponding to the increment δx_j of the variable x_j ($j\in\{l \mid l\neq i,\ \mathrm{W}_{il}\neq\emptyset\}$, see Fig. A3). In Fig. A3, δx is perpendicular to the hyperplane $(w+\delta w, c+\delta c)$. Then Δ_λ^1 is given by

$$(\mathrm{A}.12)\qquad \Delta_\lambda^1= \int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}+\delta\alpha_{ij}}\, g_{\lambda\kappa}\{x_i^\kappa-(x^\kappa+\delta x^\kappa)\}\phi(x+\delta x)\mathrm{d}^{N-1}x \\ + \int_{\partial\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}}\, g_{\lambda\nu}(x_i^\nu-x^\nu)\frac{1}{\tan\theta(x)}\, \frac{1}{\alpha_{ij}} g_{\kappa\mu}(x_j^\mu-x^\mu)\delta x_j^\kappa\phi(x)\mathrm{d}^{N-2}x \\ - \int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}} g_{\lambda\kappa}(x_i^\kappa-x^\kappa)\phi(x)\mathrm{d}^{N-1}\mathrm{x}.$$

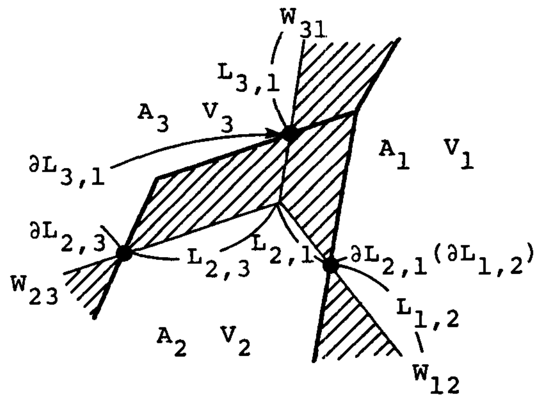


Fig. A1. Components of $\{A_j\}_{j=1}^n$ and $\{V_i\}_{i=1}^n$

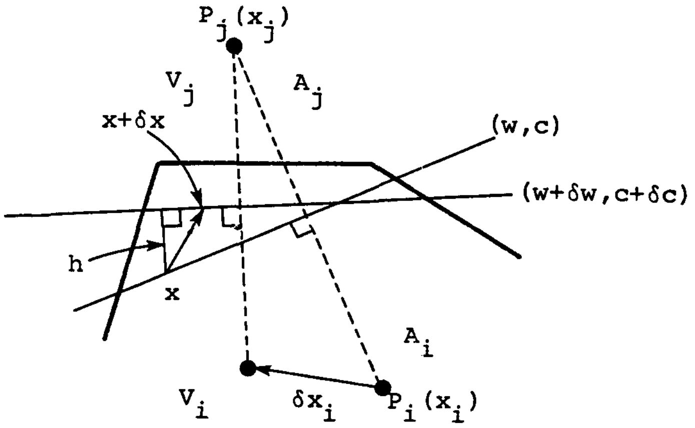


Fig. A2. Derivation of the gradient of *F*

Substituting the following relations (A.13)-(A.15) into (A.12), we obtain (A.16).

$$\text{(A.13)} \qquad \delta x^\nu = \frac{1}{\alpha_{ij}^2} g_{\kappa\lambda}(x_i^\lambda - x^\lambda)\delta x_j^\kappa (x_j^\nu - x_i^\nu),$$

$$\text{(A.14)} \qquad \delta\alpha_{ij} = \frac{1}{\alpha_{ij}} g_{\kappa\lambda}(x_j^\lambda - x_i^\lambda)\delta x_j^\kappa,$$

$$\text{(A.15)} \qquad \phi(x+\delta x)-\phi(x) = \frac{1}{\alpha_{ij}^2} g_{\kappa\lambda}(x_j^\lambda - x^\lambda)\delta x_j^\kappa (x_j^\xi - x_i^\xi)\frac{\partial\phi}{\partial x^\xi}.$$

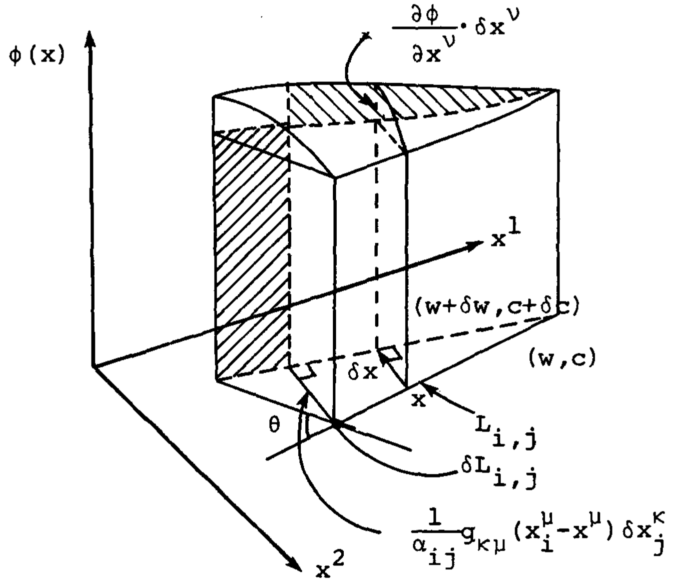

$\frac{\partial\phi}{\partial x^\nu}\cdot\delta x^\nu$
$\phi(x)$
x^1
$(w+\delta w, c+\delta c)$
(w,c)
δx
x
θ
$L_{i,j}$
$\delta L_{i,j}$
$\frac{1}{\alpha_{ij}} g_{\kappa\mu}(x_i^\mu - x^\mu)\delta x_j^\kappa$
x^2

Fig. A3. Weight of the increment Δ_λ^1 of $\partial F/\partial x_i^\lambda$

▧ : weight of the second term of (A.16)

▨ : weight of the third term of (A.16)

$$\begin{aligned}(\text{A.16})\qquad \Delta_\lambda^1 = &- \int_{L_{i,j}} \frac{1}{\alpha_{ij}^3} g_{\lambda\nu}(x_i^\nu - x^\nu) g_{\kappa\mu}(x_i^\mu - x_j^\mu)\delta x_j^\kappa \phi(x)\, d^{N-1}x \\ &+ \int_{L_{i,j}} \frac{1}{\alpha_{ij}^3} g_{\lambda\nu}(x_i^\nu - x^\nu) g_{\kappa\mu}(x_j^\mu - x^\mu)\delta x_j^\kappa (x_j^\xi - x_i^\xi)\frac{\partial\phi}{\partial x^\xi}\, d^{N-1}x \\ &+ \int_{\partial L_{i,j}} \frac{1}{\alpha_{ij}^2} \frac{1}{\tan\theta(x)} g_{\lambda\nu}(x_i^\nu - x^\nu) g_{\kappa\mu}(x_j^\mu - x^\mu)\delta x^\kappa \phi(x)\, d^{N-2}x.\end{aligned}$$

Next, we investigate the increment Δ_λ^2 of the first term in the braces of (A.11) corresponding to the increment δx_k of the variable x_k ($k \in \{l \mid W_{ij} \cap \partial V_l \neq \emptyset, i \neq j, W_{ij} \neq \emptyset\}$, see Fig. A4).

$$(\text{A.17})\qquad \Delta_\lambda^2 = \int_{L_{i,j} \cap \partial V_k} \frac{1}{\alpha_{ij}} g_{\lambda\nu}(x_i^\nu - x^\nu)\frac{\beta(x)}{\alpha_{ik}\alpha_{jk}} g_{\kappa\mu}(x_k^\mu - x^\mu)\delta x_k^\kappa \phi(x)\, d^{N-2}x,$$

where

$$(\text{A.18})\qquad \beta(x) = \{g_{\lambda\kappa}(x_k^\kappa - x^\kappa)(x_k^\lambda - x^\lambda)\}^{1/2}.$$

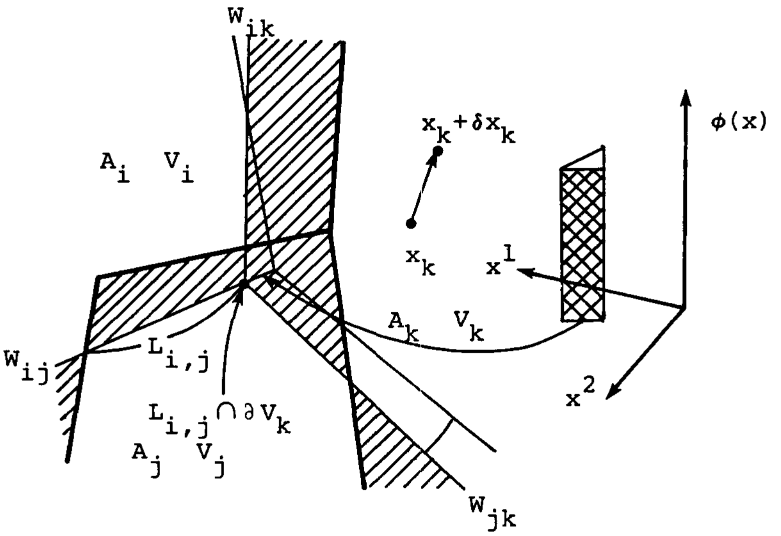


Fig. A4. Weight of the increment Δ_λ^2 of $\partial F/\partial x_i^\lambda$

▨ : discrepancy

▩ : weight of Δ_λ^2

When $j=i$, the first term of (A.12) is slightly changed, i.e.,

$$\text{(A.19)}\quad \int_{L_{i,j}} \frac{1}{\alpha_{ij}+\delta\alpha_{ij}} g_{\lambda\kappa}\{x_i^\kappa+\delta x_i^\kappa-(x^\kappa+\delta x^\kappa)\}\phi(x+\delta x)\mathrm{d}^{N-1}x.$$

Thus the increment of the first term in the braces of (A.11), Δ_λ^0, corresponding to the increment δx_i of the variable x_i is given by

$$\begin{aligned}\text{(A.20)}\quad \Delta_\lambda^0= &\int_{L_{i,j}} \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}\delta x_i^\kappa \phi(x)\mathrm{d}^{N-1}x\\ &-\int_{L_{i,j}} \frac{1}{\alpha_{ij}^3}\, g_{\lambda\nu}(x_i^\nu-x^\nu)g_{\kappa\mu}(x_i^\mu-x_j^\mu)\delta x_i^\kappa\phi(x)\mathrm{d}^{N-1}x\\ &+\int_{L_{i,j}} \frac{1}{\alpha_{ij}^3}\, g_{\lambda\nu}(x_i^\nu-x^\nu)g_{\kappa\mu}(x_i^\mu-x^\mu)\delta x_i^\kappa(x_j^\xi-x_i^\xi)\frac{\partial\phi}{\partial x^\xi}\,\mathrm{d}^{N-1}x.\\ &+\int_{\partial L_{i,j}} \frac{1}{\alpha_{ij}^2}\,\frac{1}{\tan\theta(x)}\, g_{\lambda\nu}(x_i^\nu-x^\nu)g_{\kappa\mu}(x_i^\mu-x^\mu)\delta x_i^\kappa\phi(x)\mathrm{d}^{N-2}x.\end{aligned}$$

From (A.16), (A.17) and (A.20), and by similar calculation for the second term of (A.11), we obtain

$$\text{(A.21)}\qquad \frac{\partial^2 F}{\partial x_j^{\kappa}\partial x_i^{\lambda}} = \begin{cases} \sum_{k:\mathrm{W}_{ik}\neq\emptyset} \{(H_{\lambda\kappa}^{ik}-H_{\lambda\kappa}^{ki})+G_{\lambda\kappa}^{iki}\} & (j=i), \\ G_{\lambda\kappa}^{ijj}+\sum_{k:\mathrm{W}_{ij}\cap\partial\mathrm{V}_k\neq\emptyset} K_{\lambda\kappa}^{ijk} & (j\in\{l\mid l\neq i,\ \mathrm{W}_{il}\neq\emptyset\}), \\ 0 & (\text{otherwise}), \end{cases}$$

where

$$\text{(A.22)}\qquad H_{\lambda\kappa}^{ij}= \int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}}\, g_{\lambda\kappa}\phi(x)\,d^{N-1}x,$$

$$\begin{aligned} \text{(A.23)}\qquad G_{\lambda\kappa}^{ijk}= &-\int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}^3}\, g_{\lambda\nu}(x_i^{\nu}-x^{\nu})g_{\kappa\mu}(x_i^{\mu}-x_j^{\mu})\phi(x)\,d^{N-1}x \\ &+\int_{\mathrm{L}_{j,i}} \frac{1}{\alpha_{ij}^3}\, g_{\lambda\nu}(x_i^{\nu}-x^{\nu})g_{\kappa\mu}(x_i^{\mu}-x_j^{\mu})\phi(x)\,d^{N-1}x \\ &+\int_{\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}^3}\, g_{\lambda\nu}(x_i^{\nu}-x^{\nu})g_{\kappa\mu}(x_k^{\mu}-x^{\mu})\frac{\partial\phi}{\partial x^{\xi}}(x_j^{\xi}-x_i^{\xi})\,d^{N-1}x \\ &-\int_{\mathrm{L}_{j,i}} \frac{1}{\alpha_{ij}^3}\, g_{\lambda\nu}(x_i^{\nu}-x^{\nu})g_{\kappa\mu}(x_k^{\mu}-x^{\mu})\frac{\partial\phi}{\partial x^{\xi}}(x_j^{\xi}-x_i^{\xi})\,d^{N-1}x \\ &+\int_{\partial\mathrm{L}_{i,j}} \frac{1}{\alpha_{ij}^2}\,\frac{1}{\tan\theta(x)}\, g_{\lambda\nu}(x_i^{\nu}-x^{\nu})g_{\kappa\mu}(x_k^{\mu}-x^{\mu})\phi(x)\,d^{N-2}x \\ &-\int_{\partial\mathrm{L}_{j,i}} \frac{1}{\alpha_{ij}^2}\,\frac{1}{\tan\theta(x)}\, g_{\lambda\nu}(x_i^{\nu}-x^{\nu})g_{\kappa\mu}(x_k^{\mu}-x^{\mu})\phi(x)\,d^{N-2}x, \end{aligned}$$

$$\begin{aligned} \text{(A.24)}\qquad \mathrm{K}_{\lambda\kappa}^{ijk}= &\int_{\mathrm{L}_{i,j}\cap\partial\mathrm{V}_k} \frac{1}{\alpha_{ij}}\, g_{\lambda\nu}(x_i^{\nu}-x^{\nu})\frac{\beta(x)}{\alpha_{ik}\alpha_{jk}}g_{\kappa\mu}(x_k^{\mu}-x^{\mu})\phi(x)\,d^{N-2}x \\ &-\int_{\mathrm{L}_{j,i}\cap\partial\mathrm{V}_k} \frac{1}{\alpha_{ij}}\, g_{\lambda\nu}(x_i^{\nu}-x^{\nu})\frac{\beta(x)}{\alpha_{ik}\alpha_{jk}}g_{\kappa\mu}(x_k^{\mu}-x^{\mu})\phi(x)\,d^{N-2}x. \end{aligned}$$

When N=2, $\partial\mathrm{L}_{i,j}$ and $\mathrm{L}_{i,j}\cap\partial\mathrm{V}_k$ are points in R^2, then the integration of the fifth and the sixth term of (A.23) and (A.24) become the summation of each intebgand which has the value corresponding to the $\partial\mathrm{L}_{i,j}$ or $\mathrm{L}_{i,j}\cap\partial\mathrm{V}_k$.

References

[1] Abramowitz, M., and Stegun, I. A., eds.: *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards Applied Mathematics Series 55, 1964 (10th printing, 1972).

[2] Ash, P., and Bolker, E. D.: Recognizing Dirichlet Tessellations. *Geometriae Dedicata*, Vol. 19, No. 2 (1985), 175-206.

[3] Barlow, G. W.: Hexagonal Territories. *Animal Behaviour*, Vol. 22 (1974), 876-878.

[4] Goldstein, A. A.: *Constructive Real Analysis*. Harper & Row, 1968.

[5] Dirichlet, G. L.: Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Journal für reine und angenwandte Mathematik*, Bd. 40 (1850), 209-227.

[6] Fukushima, M.: A Summary of Numerical Algorithms in Nonsmooth Optimization. *Proceedings of the 3rd Mathematical Programming Symposium, Japan*, Tokyo, 1982, 63-78.

[7] Iri, M., Murota, K., and Ohya, T.: A Fast Voronoi Diagram Algorithm with Applications to Geographical Optimization Problems. *Proceedings of the 11th IFIP Conference on System Modelling and Optimization*, 1983, Copenhagen, Lecture Notes in Control and Information Science 59, "System Modelling and Optimization" (P. Thoft-Christensen, ed.), Springer-Verlag, Berlin, 273-288.

[8] Iri, M., et al.: *Fundamental Algorithms for Geographical Data Processing* (in Japanese). Technical Report T-83-1, Operations Research Society of Japan, 1983.

[9] Ohya, T., Iri, M., and Murota, K.: A Fast Voronoi-Diagram Algorithm with Quaternary Tree Bucketing. *Information Processing Letters*, Vol. 18 (1984), 227-231.

[10] Ohya, T., Iri, M., and Murota, K.: Improvements of the Incremental Method for the Voronoi Diagram with Computational Comparison of Various Algorithms. *Journal of the Operations Research Society of Japan*, Vol. 27, No.4 (1984), 306-336.

[11] Polyak, B. T.: Minimization of Unsmooth Functionals. *USSR Computational Mathematics and Mathematical Physics*, Vol. 9, No. 3 (1969), 14-29.

[12] Rogers, C. A.: Packing and Covering. *Cambridge Tracts in Mathematics and Mathematical Physics*, No. 54, Cambridge University Press, London, 1964.

[13] Hasegawa, M., and Tanemura, M.: On the Pattern of Space Division by Territories. *Annals of the Institute of Statistical Mathematics*, Vol. 28 (1976), Part b, 509-519.

[14] Thiessen, A. H.: Precipitation Averages for Large Area. *Monthly Weather Review*, Vol. 39 (1911), 1082-1084.

[15] Voronoi, G.: Nouvelles Applications des Parametres Continus ou la Theorie des Formes Quadratiques. *Journal für reine und angewandte Mathematik*, Bd. 134 (1908), 198-287.

Atsuo SUZUKI, Masao IRI:
Department of Mathematical Engineering
and Instrumentation Physics
Faculty of Engineering
University of Tokyo
Bunkyo-ku, Tokyo 113, Japan