

Discretization and Validation of the Continuum Approximation Scheme for Terminal System Design

Yanfeng Ouyang, Carlos F. Daganzo

Institute of Transportation Studies and Department of Civil and Environmental Engineering,
 University of California, Berkeley, California 94720
 {yfouyang@berkeley.edu, daganzo@ce.berkeley.edu}

This paper proposes an algorithm that automatically translates the “continuum approximation” (CA) recipes for location problems into discrete designs. It applies to terminal systems, but can also be used for other logistics problems. The study also systematically compares the logistics costs predicted by the CA approach with the actual costs for discrete designs obtained with the automated procedure. The predictions are quite accurate. The paper also gives conditions under which the discrete solution has a small optimality gap.

Key words: terminal design; continuum approximation

History: Received: October 2003; revision received: July 2004; accepted: November 2004.

1. Background

Designing a physical distribution system for minimal logistics cost is a complex task. The objective function usually includes complicated cost expressions for the various distribution stages, i.e., *inbound costs* for deliveries into the terminals, *outbound costs* for deliveries from terminals to customers, and *terminal costs* for setup and operation of the terminals. Furthermore, the decision variables are usually discrete and very numerous, including the number of terminals, their locations, delivery routes, schedules, and the allocation of customers to terminals.

The paper focuses on the strategic design of a terminal system in a continuous service area S , where customer demand is distributed with a spatial density $\lambda(x)$, $x \in S$. The goal is to find a set of terminal locations, $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$, and a partition of S into a set of *influence areas* served by these terminals, $\mathbf{I} = \{I_1, I_2, \dots, I_N\}$, that minimize the total logistics cost, $Z_D(\mathbf{x}, \mathbf{I})$. The number of terminals N is itself a decision variable. The minimization problem is

$$\begin{aligned} \text{Min } Z_D(\mathbf{x}, \mathbf{I}) &= \sum_{i=1}^N \left(\int_{I_i} z_D(x, x_i, I_i) \cdot \lambda(x) dx \right), \quad (1) \\ \text{s.t. } x_i &\in S, \quad i = 1, 2, \dots, N, \\ I_i \cap I_j &= \phi \quad \text{for } \forall i \neq j, \\ \bigcup_i I_i &= S, \end{aligned}$$

where $z_D(x, x_i, I_i)$ is the cost of serving a unit of demand at $x \in I_i$ via a terminal at x_i . This should include all fixed and variable costs.

In the applied mathematics literature a simpler version of (1) is called the “optimal resource allocation problem” (Okabe, Boots, and Sugihara 1992; Du, Faber, and Gunzburger 1999). These problems also allow pointlike service facilities to be located among a continuum of customers. However, for the problems to be tractable, z_D must be a simple function of a norm, $\|x - x_i\|$; e.g., $\|x - x_i\|^2$. Unfortunately, these simple forms are not realistic for typical logistics problems (e.g., including inbound costs).

Facility location problems can also be formulated by considering a finite number of possible locations for customers and terminals. Optimal locations are then selected with a mixed-integer program. An extensive literature also exists on this subject; see, e.g., Daskin (1995) and Drezner and Hamacher (2002). This approach is effective if the number of candidates is small, but for a problem like (1), the number of possible choices is so large that a discrete optimization process is not practical even if done heuristically.

To circumvent some of those drawbacks, and building on the work in Newell (1971, 1973), Daganzo and Newell (1986) proposed a continuum approximation (CA) approach for terminal system design. It was argued in this reference that a near-optimum solution should have influence areas as “round” as possible, with terminals located near their centers. It was also argued that if in addition $\lambda(x)$ varies slowly with x , and the areas $|I_i|$ can be approximated by a slow-varying function of x , $A(x)$, such that $|I_i| \cong A(x)$ if $x \in I_i$, then the set function $z_D(x, x_i, I_i)$ in (1) can be approximated by a simpler function of two real arguments, $z_C(x, A(x))$.

The function A is a decision variable representing the desired influence area size for locations near x . With this approximation, (1) can be replaced by

$$\text{Min } Z_C(A) = \int_S z_C(x, A(x)) \cdot \lambda(x) dx, \quad (2)$$

where $Z_C(A)$ is a functional of A . More details about the procedure for obtaining z_C from z_D are given in §3, and also in Daganzo (1999).

The advantage of (2) is that it can be optimized point by point by finding the value of $A(x)$ that minimizes $z_C(x, A(x))$ at every x . This result is denoted $A^*(x)$, and the corresponding cost $Z_C^*(A^*)$. One then looks for a partition of S with “round” influence areas such that

$$|I_i| \cong A^*(x), \quad \forall x \in I_i, \quad (3)$$

and for a set of centrally located terminals. The hope is that the discrete solution so identified, $\{x_C, I_C\}$, will satisfy $Z_D(x_C, I_C) \cong Z_C^*(A^*)$. The extent to which this happens is explored in this paper. The paper also proposes a discretization algorithm to obtain the solution $\{x_C, I_C\}$, because to the authors’ best knowledge, no systematic procedure has yet been proposed for the discretization step.

The closest discretization literature describes *centroidal Voronoi tessellation* (CVT) methods for partitioning spatial regions according to various criteria; see Okabe, Boots, and Sugihara (1992) for a review. One of the relevant problems in this field is finding optimum locations for measuring stations for rainfall estimation. These methodologies are described under the rubric “location optimization of observation points for estimating the total quantity of a continuous spatial variable” in Okabe, Boots, and Sugihara (1992). Unfortunately, estimation methods (e.g., Hori and Nagata 1985) do not control for influence area size or shape, and cannot be used for our application. More relevant are methods for finding *weighted CVTs* (e.g., Du, Faber, and Gunzburger 1999) because by adjusting the weights one can control for area size. Unfortunately, shape cannot be controlled with these methods. This is a serious drawback because the subareas of a weighted CVT may not be round. Because weighted CVT methods are also computationally slow, our algorithm uses an entirely different approach.

Feasible designs obtained with this algorithm are shown to exhibit costs $Z_D(x_C, I_C)$ very close to the CA prediction $Z_C^*(A^*)$. The paper also gives sufficient conditions under which $Z_C^*(A^*)$ is a tight lower bound for the exact optimal system costs. Because $Z_D(x_C, I_C)$ and $Z_C^*(A^*)$ are close to each other, the optimality gap between $Z_D(x_C, I_C)$ and the true optimum should be small under these conditions.

This paper is organized as follows. Section 2 develops a simple discretization algorithm that explicitly

controls for influence area shape; §3 shows the numerical examples; and §4 shows conditions under which $Z_C^*(A^*)$ bounds from below the true minimum. A final section discusses generalizations.

2. The Model and Algorithm

As discussed before, a near-optimum design $\{x_C, I_C\}$ should: (i) satisfy the size requirement (3), (ii) have influence areas as round as possible, and (iii) have terminals located near the centers of the influence areas. (We assume from now on that distances are given by the Euclidean metric, unless stated otherwise.)

2.1. A Disk Model

To capture (ii) and (iii), we will imagine that each influence area contains a round disk centered at the terminal, and instead of $\{x_C, I_C\}$ we will look for a set of N nonoverlapping disks, where $N \cong \int_S [A^*(x)]^{-1} dx$. By sliding the disks within S , different designs can be obtained. Two examples are displayed in Figure 1. We use $r(x) = \{r(x_i)\}$ for the set of disk radii; see dotted arrows. For a good design, disks should jointly cover most of S without protruding outside it; see Figure 2. Because each influence area must contain one disk, this ensures that the influence areas are “round.” In addition, for a good design, the area of each disk should be as close as possible to $A^*(x)$; i.e.,

$$r(x_i) \approx \sqrt{\frac{A^*(x_i)}{\pi}}, \quad i = 1, 2, \dots, N. \quad (4a)$$

It should be possible to satisfy these two conditions simultaneously because there always are many ways to cover most of S with disks of different sizes, as illustrated by Figure 2.

Of course, because disks cannot tessellate convex Euclidean regions, we cannot expect the equality in (4a) to be satisfied exactly. Therefore, we look instead for radii that satisfy

$$r(x_i) = \sqrt{\frac{A^*(x_i)}{k}}, \quad i = 1, 2, \dots, N, \quad (4b)$$

for k as small as possible. (Given our definition for N , $k \geq \pi$.)

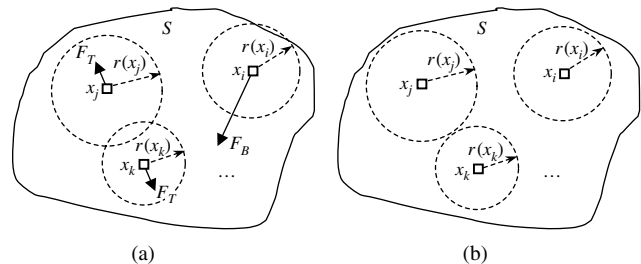


Figure 1 Disks and Terminals: (a) An Infeasible Overlapping Pattern; (b) A Feasible Nonoverlapping Pattern

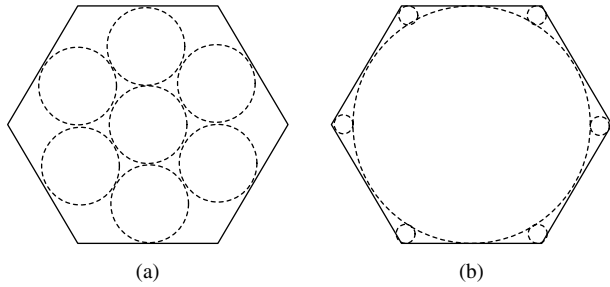


Figure 2 Two Possible Layouts of Seven Disks in a Hexagon: (a) Homogeneous Pattern; (b) Inhomogeneous Pattern

To automate the sliding procedure, we now introduce two types of repulsive “forces” that act on the centers of the disks. The first type, terminal force F_T , acts along the line connecting the centers of overlapping disks. The other type, boundary force F_B , acts on disks touching the boundary, pointing toward the interior of S in a direction normal to the boundary. Solid arrows in Figure 1(a) depict these forces.

Figure 3 defines our choices for the magnitudes of F_T and F_B . They depend on \mathbf{x} and $\mathbf{r}(\mathbf{x})$, vanishing when no disks overlap or protrude outside the boundary. We use $(N + 1)f$ for the magnitude of F_B , where f is the maximal value of F_T , to ensure that disks are never pushed out of S .

We call a pattern with zero forces an *equilibrium*. The disk centers of an equilibrium give \mathbf{x}_C . This is sufficient to obtain a solution because S can then be easily partitioned into influence areas, I_C , that contain the disks, as will be explained shortly. Although such an equilibrium solution $\{\mathbf{x}_C, I_C\}$ may not be unique, it should satisfy the near-optimality requirements (i)–(iii).

We assume that S is “simply connected,” in the sense that a disk of proper size can always be slid between any two points in S without touching the boundary. No generality is lost by this assumption, because complex areas (e.g., Japan) can usually be partitioned into simply connected components to which the model can be applied separately.

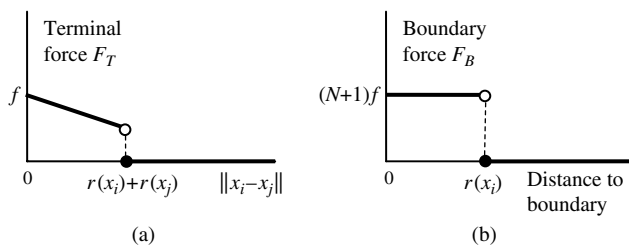


Figure 3 Possible Definitions of Forces: (a) Repulsive Force for Terminal Pair (i, j) ; (b) Boundary Force for Terminal i

2.2. The Algorithm

The forces defined above are used to slide the disks within S for small distances, while $\mathbf{r}(\mathbf{x})$ and the forces themselves are updated. The algorithm stops when all forces vanish. An equilibrium obviously exists and can be found for a sufficiently large k (because all disk radii vanish for $k \rightarrow \infty$). Conversely, an equilibrium will not exist if k is too small. Therefore, the algorithm increases k by a small increment, Δk , if the current value does not yield an equilibrium.

Step sizes for disk movements should not be too large for fast convergence. One could use constant step sizes, μ , comparable with the tolerance level ε (in distance units) or, even better, step sizes μ_m that gradually decrease with the iteration count m , such that $\mu_m \rightarrow 0$ and $\sum \mu_m \rightarrow \infty$.

Even for reasonably large k , this algorithm may not converge to an equilibrium if we encounter sets of *degenerate* terminal locations (also called singular points in Okabe, Boots, and Sugihara 1992). This happens, for example, if points are on a straight line that intersects the boundaries of S orthogonally. In this case, points would remain trapped on this line, because all ensuing terminal movements would have to be along the line. Fortunately, such degeneracy is usually unstable and can be eliminated by small location perturbations. Therefore, we add perturbations of random direction with a displacement size $\delta < \varepsilon$ at each step of the procedure.

Once an equilibrium has been obtained, S is partitioned into I_C with a *weighted-Voronoi tessellation* (WVT) that ensures each I_i contains one entire disk. The recipe is simple: First, partition S into very small squares, and then allocate each square to one I_i with the rule $i = \arg \min_i \{\|x - x_j\|/r(x_j)\}$, where x is the center of the square. This rule ensures that every disk is a subset of its influence area.

A possible implementation of the algorithm is:

- (1) Choose N arbitrary locations in area S and initialize all parameters: tolerance ε , perturbation size δ , and increment for k , Δk ; set initial $k \cong \pi$ and $m = 1$.
- (2) Calculate the disk radii with (4b) and then the forces on every terminal as per Figure 3; if all the forces equal zero (equilibrium reached), go to Step 5; otherwise, move each terminal along the direction of its resultant force by a step size μ_m , and add a random-direction perturbation of size δ .
- (3) If $\mu_m < \varepsilon$, reset $m = 0$, and increase k by Δk .
- (4) $m = m + 1$; go to Step 2.
- (5) Tessellate S with the WVT recipe.

3. Illustrations

3.1. Convergence Test

The algorithm’s convergence is illustrated with a problem that has a known optimal solution (with $k \approx$

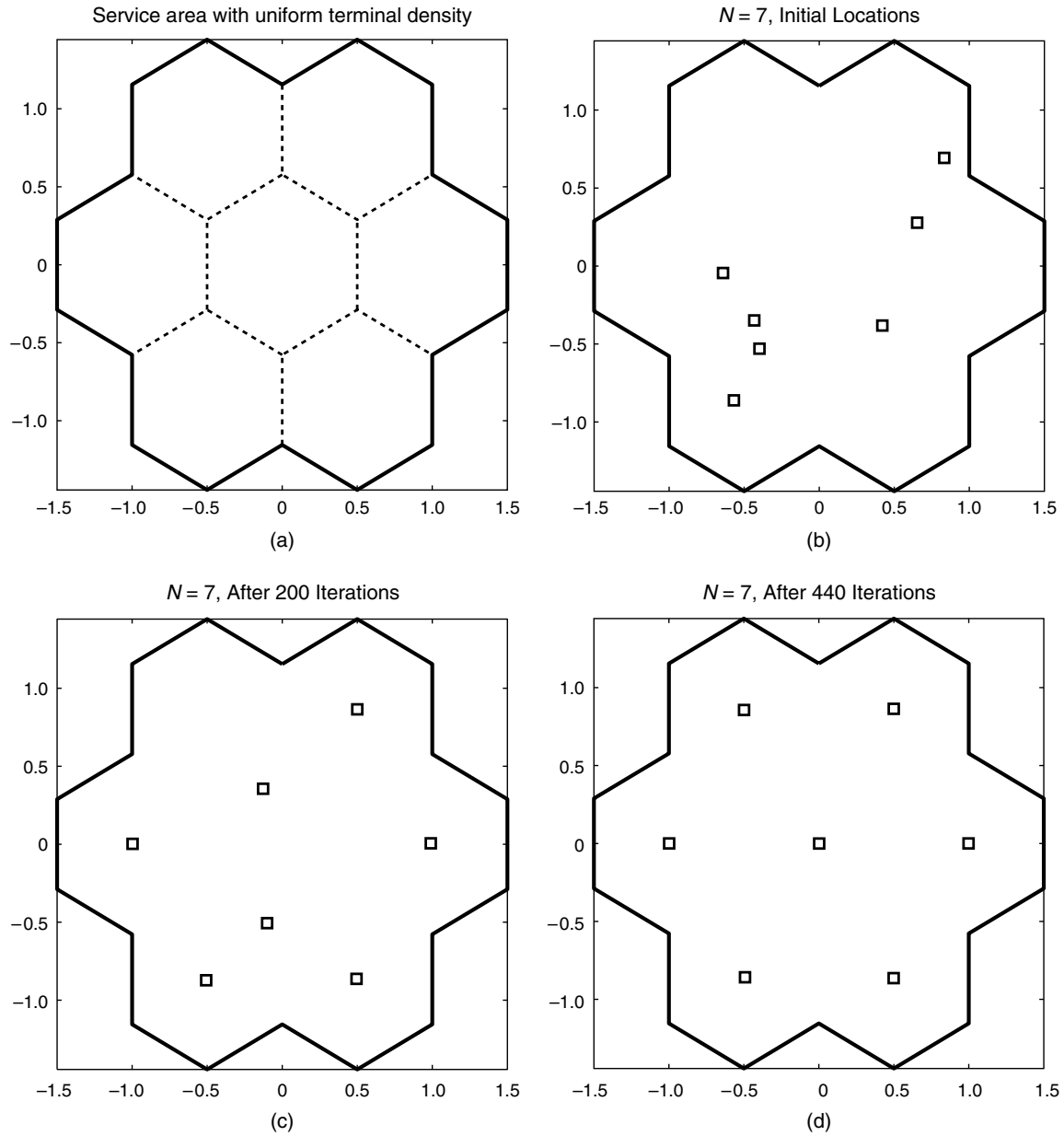


Figure 4 Verification of Convergence: (a) Area S ; (b) Initial Locations; (c) Locations After 200 Iterations; (d) Equilibrium Locations After 440 Iterations

3.46), using the polyhexagonal region S of Figure 4(a). The side of each hexagon in S equals $1/\sqrt{3}$. If $A^*(x) = \sqrt{3}/2$ and $N = 7$, then the partition in Figure 4(a) is optimal.

The initial locations are arbitrarily generated and shown in Figure 4(b). Because k is known, we initialize the algorithm with this value and use a constant step size $\mu = 0.01$. Figure 4(c) shows an intermediate result, and Figure 4(d) the final equilibrium, which was achieved after 440 iterations. Note that the weighted-Voronoi tessellation corresponding to Figure 4(d) matches that in Figure 4(a). Thus, the algorithm performs as expected.

3.2. Practical Examples

In this section we use practical examples to further illustrate how the algorithm translates $A^*(x)$ into discrete designs $\{x_C, I_C\}$. The exact costs of the design, $Z_D(x_C, I_C)$, are then compared to the estimated costs $Z_C^*(A^*)$.

Daganzo (1999, §5.3.5) gives an example of terminal system design, in which customers are uniformly distributed in an $L \times L$ square area S . They are served with one transshipment from a depot at one corner of S . Line-haul vehicles with infinite capacity shuttle between the depot and the terminals. Local delivery

vehicles have a small capacity v_{\max} , travel full, and visit only one customer per delivery.

If we only consider inventory and transportation costs (both inbound and outbound) and ignore fixed costs such as terminal facility rents, the formula for $z_D(x, x_i, I_i)$ in (1) is (Daganzo 1999)

$$z_D(x, x_i, I_i) = \underbrace{2 \left(\frac{a'b'R(x_i)}{\int_{I_i} \lambda(x) dx} \right)^{1/2}}_{\text{Inbound costs}} + \underbrace{\frac{av_{\max}}{\lambda(x)} + \frac{1.5\sqrt{\pi}b}{v_{\max}} s(x, x_i)}_{\text{Outbound costs}}. \quad (5)$$

In (5), a' , b' , a , b are cost parameters, $R(x)$ is the distance from point x to the depot, and $s(x, x_i)$ is the outbound delivery distance from x_i to x .

On the other hand, the expression for $z_C(x, A(x))$ in (2), as shown in Daganzo (1999), is

$$z_C(x, A(x)) = 2 \left(\frac{a'b'R(x)}{\lambda(x)A(x)} \right)^{1/2} + \frac{av_{\max}}{\lambda(x)} + \frac{b}{v_{\max}} A^{1/2}(x). \quad (6)$$

Formula (6) is derived from (5) by approximating the terminal throughput $\int_{I_i} \lambda(x) dx$ appearing in the first term with $\lambda(x)A(x)$, and $s(x, x_i)$ with the average a delivery distance $(2/(3\sqrt{\pi}))\sqrt{A(x)}$ in a hypothetical circular influence area of size $A(x) \approx |I_i|$. The idea is to express every item in (5) as a local property of point x . This *local approximation* device can be used with more general forms of (5). Experience shows that it works well when $\lambda(x)$ and I_i vary slowly with position, as mentioned in §1. Two scenarios with different demand density functions $\lambda(x)$ are now used to demonstrate this idea. The results are then formalized in §4.

Scenario 1. Consider homogeneous demand $\lambda(x) = 1$, $\forall x \in S$, and also assume that $v_{\max} = b = b' = a' = b' = 1$. Then $A^*(x)$ is obtained by minimizing (6), and the result is

$$A^*(x) = \left(\frac{2v_{\max}}{b} \right) \left(\frac{a'b'R(x)}{\lambda} \right)^{1/2} = 2R^{1/2}(x). \quad (7)$$

Substituting (7) into (6) and (6) into (2), we then find

$$Z_C^*(A^*) = \int_S \lambda(x) \cdot z_C(x, A^*(x)) dx = \int_S (1 + 2\sqrt{2} \cdot R^{1/4}(x)) dx. \quad (8)$$

If we now combine (1) and (5), the result is

$$Z_D(\mathbf{x}, \mathbf{I}) = \sum_{i=1}^N (2R^{1/2}(x_i) \cdot |I_i|^{1/2} + |I_i|) + 1.5\sqrt{\pi} \cdot \sum_{i=1}^N \left(\int_{I_i} s(x, x_i) dx \right). \quad (9)$$

Our algorithm uses (7) as an input. The set of discrete designs $\{\mathbf{x}_C, \mathbf{I}_C\}$ obtained with it, and the associated values of $Z_C^*(A^*)$, and $Z_D(\mathbf{x}_C, \mathbf{I}_C)$ given by (8) and (9) for various L are shown in Figures 5(a)–(d).

The difference between $Z_C^*(A^*)$, and $Z_D(\mathbf{x}_C, \mathbf{I}_C)$ is quite small: 2.4% for $L = 5$, 0.8% for $L = 7$, 0.9% for $L = 10$, and 0.9% for $L = 25$. These relative differences would be even smaller if other fixed costs were also included in our cost expressions.

Scenario 2. Assume now an inhomogeneous demand such that $\lambda(x) = R^{-1/2}(x)$, $\forall x \in S$. All other parameters remain the same. Now we have

$$A^*(x) = 2R^{3/4}(x), \quad (10)$$

$$Z_C^*(A^*) = \int_S \lambda(x) \cdot z_C(x, A^*(x)) dx = \int_S (1 + 2\sqrt{2}R^{-1/8}(x)) dx, \quad (11)$$

and

$$Z_D(\mathbf{x}, \mathbf{I}) = 2 \sum_{i=1}^N \left(R^{1/2}(x_i) \cdot \sqrt{\int_{I_i} \lambda(x) dx} \right) + \sum_{i=1}^N \left(|I_i| + 1.5\sqrt{\pi} \int_{I_i} \lambda(x) s(x, x_i) dx \right). \quad (12)$$

The set of designs and associated costs are now shown in Figures 6(a)–(d).

The cost differences are 2.6%, 2.3%, 1.6%, and 0.7%, respectively. They are approximately the same as those in Scenario 1. This shows that the cost differences are insensitive to gradual demand variations.

Note that if $Z_C^*(A^*)$ is a lower bound for the costs of a design $\{\mathbf{x}, \mathbf{I}\}$, then the cost difference is an optimality gap. Section 4 below gives conditions for this to happen. They are approximately satisfied by our examples. This explains why in all the examples, $Z_C^*(A^*)$ is slightly smaller than $Z_D(\mathbf{x}_C, \mathbf{I}_C)$. Our percentage differences can be interpreted as approximate optimality gaps.

In both scenarios, and in agreement with theory, accuracy of the CA formulae improves with problem size quite dramatically. This is fortuitous. It means both that the CA formulae describe the optimum costs of large complex problems well, and that the CA discretization algorithm can complement conventional methods when they would have the most difficulty. In all the examples, the algorithm produced the solution in less than 30 minutes on a 1.7 GHz PC with our choice of parameters.

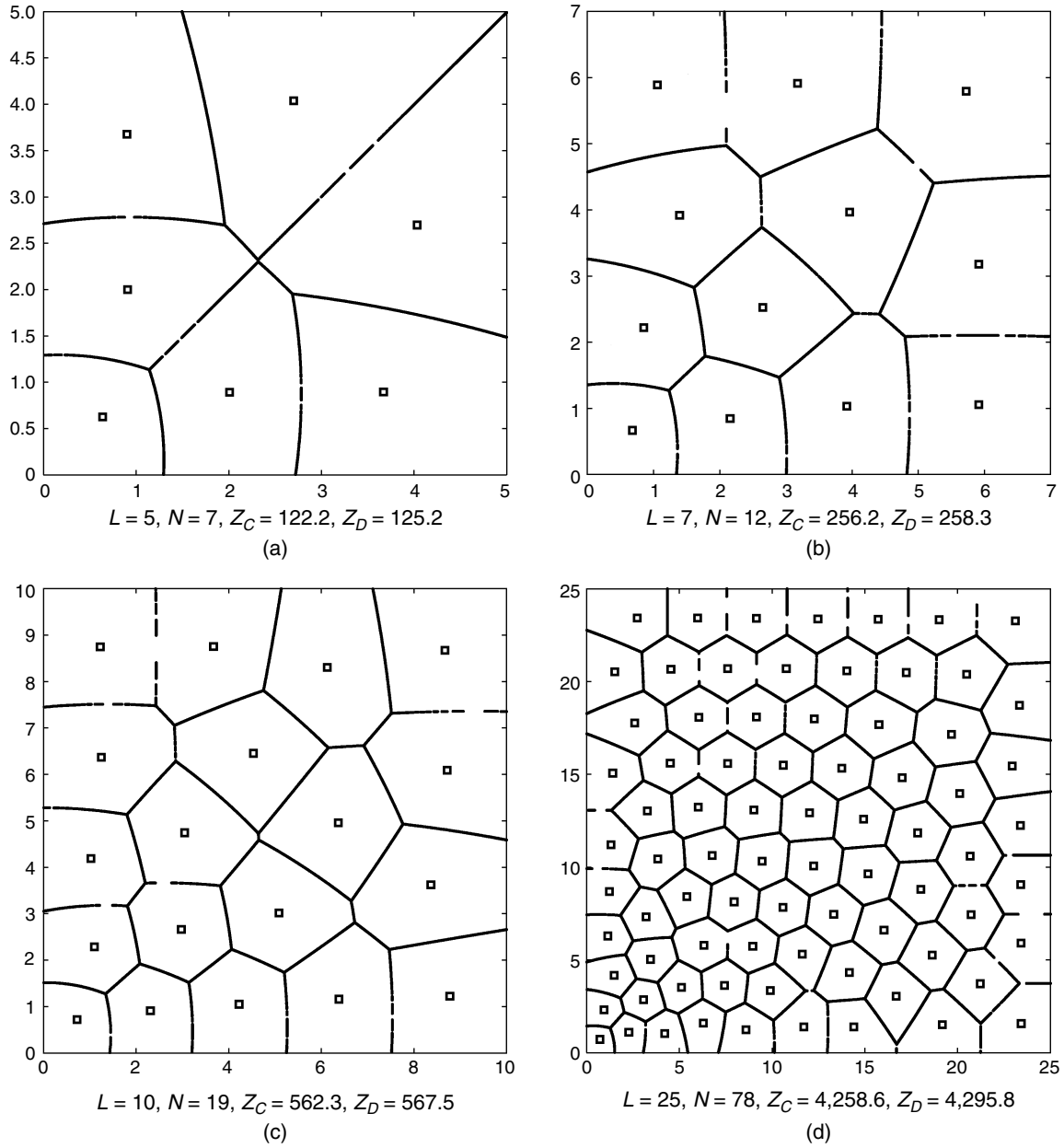


Figure 5 Terminal Designs for Homogeneous Customer Demand: (a) $L = 5$; (b) $L = 7$; (c) $L = 10$; (d) $L = 25$

4. A Lower Bound

We consider in this section a generalization of (5) of the following form:¹

$$z_D(x, x_i, I_i) = z^i \left(\underbrace{R(x_i), \int_{I_i} \lambda(x) dx}_{\text{Inbound costs}} \right) + z^o \left(\underbrace{s(x, x_i), \lambda(x)}_{\text{Outbound costs}} \right), \quad (13)$$

¹ If terminal costs, including fixed set-up costs, depend only on demand, they can be included in the first term of (13).

where z^i and z^o are ordinary functions of two arguments. For this case, the local approximation device yields

$$z_C(x, A(x)) = z^i(R(x), \lambda(x)A(x)) + z^o \left(\frac{2}{3\sqrt{\pi}} \sqrt{A(x)}, \lambda(x) \right). \quad (14)$$

We can now prove the following theorem.

THEOREM. $Z_C^*(A^*) \leq Z_D(\mathbf{x}, \mathbf{I})$, if: (a) Locations \mathbf{x} are centroids of the influence areas; (b) the demand density $\lambda(x)$ is a constant, λ_i , within each influence area; (c) the

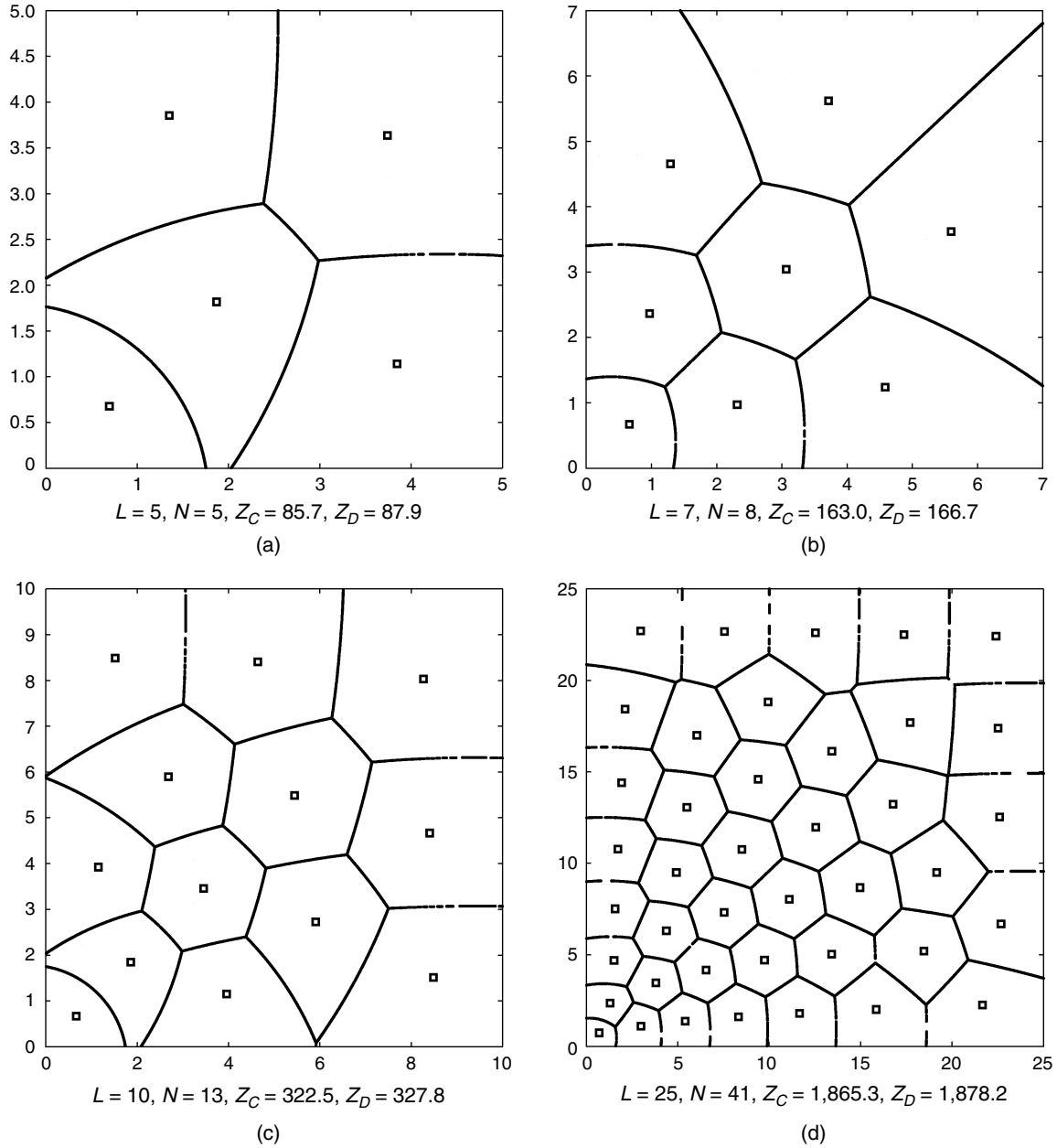


Figure 6 Terminal Designs for Inhomogeneous Customer Demand: (a) $L = 5$; (b) $L = 7$; (c) $L = 10$; (d) $L = 25$

inbound transportation cost is a concave function of distance; (d) the outbound transportation cost is a convex; and (e) increasing function of distance.

PROOF. Consider an arbitrarily shaped influence area, $I_i \in \mathbf{I}$, with a terminal i located at its centroid x_i ; see Figure 7. Let $Z_{D,i}(\mathbf{x}, \mathbf{I})$ and $Z_{C,i}(A)$ represent the parts of (1) and (2) corresponding to influence area i , and denote $s = s(x, x_i)$ for simplicity.

Because the demand density is constant, substitution of (13) into (1) yields

$$Z_{D,i}(\mathbf{x}, \mathbf{I}) = \int_{I_i} z^i(R(x_i), \lambda_i |I_i|) \lambda_i dx + \int_{I_i} z^0(s, \lambda_i) \lambda_i dx. \quad (15)$$

Likewise, substitution of (14) into (2) yields

$$Z_{C,i}(A) = \int_{I_i} z^i(R(x), \lambda_i A(x)) \lambda_i dx + \int_{I_i} z^0\left(\frac{2}{3\sqrt{\pi}} \sqrt{A(x)}, \lambda_i\right) \lambda_i dx. \quad (16)$$

If we can prove that

$$Z_{D,i}(\mathbf{x}, \mathbf{I}) \geq Z_{C,i}(A_s), \quad (17)$$

where $A_s(x)$ is constrained to be a step function—i.e., $A_s(x) = |I_i|$, if $x \in I_i$ —then (17) would establish that $Z_D(\mathbf{x}, \mathbf{I}) \geq Z_C(A_s)$. This would prove the theorem because $Z_C^*(A^*)$ is the optimum of $Z_C(A)$ without any constraint; therefore $Z_C^*(A^*) \leq Z_C(A_s) \leq Z_D(\mathbf{x}, \mathbf{I})$.

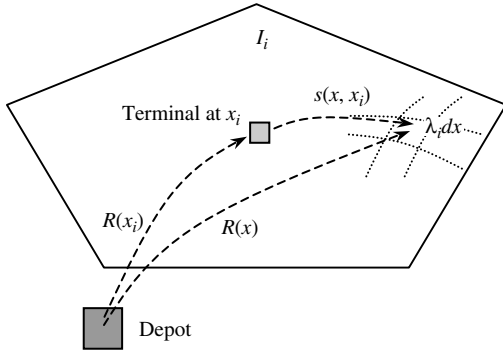


Figure 7 Logistic Operations in I_i

Note that $Z_{C,i}(A_s)$ can be expressed as

$$Z_{C,i}(A_s) = \int_{I_i} z^i(R(x), \lambda_i |I_i|) \lambda_i dx + \int_{I_i} z^o\left(\frac{2}{3\sqrt{\pi}}\sqrt{|I_i|}, \lambda_i\right) \lambda_i dx. \quad (18)$$

To prove (17), we first show that the first term of $Z_{D,i}(\mathbf{x}, \mathbf{I})$ bounds from above the first term of $Z_{C,i}(A_s)$. This is clear if we compare the first terms of (15) and (18), because $R(x_i)$ is the average of $R(x)$ by assumption (a), and Jensen’s inequality suggests (assumption (c)) that

$$\int_{I_i} z^i(R(x_i), \lambda_i |I_i|) \lambda_i dx \geq \int_{I_i} z^i(R(x), \lambda_i |I_i|) \lambda_i dx. \quad (19)$$

Thus, to prove (17) we only have to show that the second term of (15) bounds from above the second term of (18); i.e., that

$$\int_{I_i} z^o(s, \lambda_i) \lambda_i dx \geq \int_{I_i} z^o\left(\frac{2}{3\sqrt{\pi}}\sqrt{|I_i|}, \lambda_i\right) \lambda_i dx = \lambda_i |I_i| \cdot z^o\left(\frac{2}{3\sqrt{\pi}}\sqrt{|I_i|}, \lambda_i\right). \quad (20)$$

Note as a preliminary step that

$$\int_{I_i} z^o(s, \lambda_i) \lambda_i dx \geq \int_{I_i} z^o(\bar{s}, \lambda_i) \lambda_i dx, \quad (21)$$

where \bar{s} is the average outbound delivery distance in I_i . This is true, again, by virtue of assumption (d) and Jensen’s inequality.

We now define a point-to-point mapping $\{M: y = M(x), x \in I_i, y \in I'_i\}$ that transforms I_i into a round area I'_i with the same centroid and the same area, and such that $s'(y, x_i) \leq s(x, x_i)$ for $\forall y = M(x)$; see Figure 8. (This last condition is trivially satisfied by specifying that all points in $I_i \cap I'_i$ should be fixed points; i.e., $y = x$.) We consider now the cost of serving the transformed region if the demand density in it is still λ_i . Clearly, the inbound costs stay the same. Obviously,

$$\bar{s}' \leq \bar{s}, \quad (22)$$

where $\bar{s}' = (2/(3\sqrt{\pi}))\sqrt{|I'_i|} = (2/(3\sqrt{\pi}))\sqrt{|I_i|}$ is the average outbound delivery distance in I'_i . We can now write

$$\int_{I_i} z^o(s, \lambda_i) \lambda_i dx \geq \int_{I_i} z^o(\bar{s}, \lambda_i) \lambda_i dx \geq \int_{I'_i} z^o(\bar{s}', \lambda_i) \lambda_i dx = \lambda_i |I_i| \cdot z^o\left(\frac{2}{3\sqrt{\pi}}\sqrt{|I_i|}, \lambda_i\right), \quad (23)$$

where the first inequality is (21), the second inequality follows from (22) and assumption (e), and the final equality follows from the fact that $|I_i| = |I'_i|$. This completes the proof. \square

This theorem is valid for any N and any partition of S . Of course, it is based on idealized conditions that are quite unrealistic if strictly enforced—because the cost conditions may not apply in many cases, and demand density will rarely be constant within each influence area. However, we are often faced with problems for which these conditions are approximately true, such as our examples.

5. Conclusion

This paper proposed an automated algorithm to obtain discrete designs out of the continuum approximation recipes for location problems. It can be easily extended to other logistics problems. Numerical results show that the algorithm systematically finds feasible discrete terminal designs with costs very close to those predicted.

The algorithm was illustrated with Euclidean metrics and circular disks. However, it can easily be extended to other metrics and/or applications that require elongated influence areas. For example, by constructing square rather than round disks, the algorithm can be applied with L_1 metrics. The distances are then measured on the L_1 grid, while all the forces still go through the disk centers. An example is shown in Figure 9. Recall too that our algorithm looks for

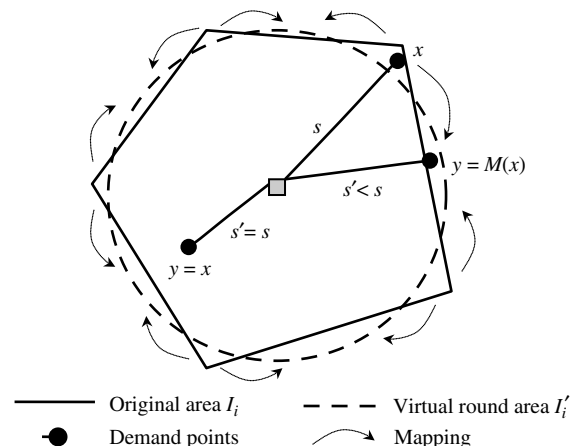


Figure 8 Mapping Points from I_i into a Round Area I'_i

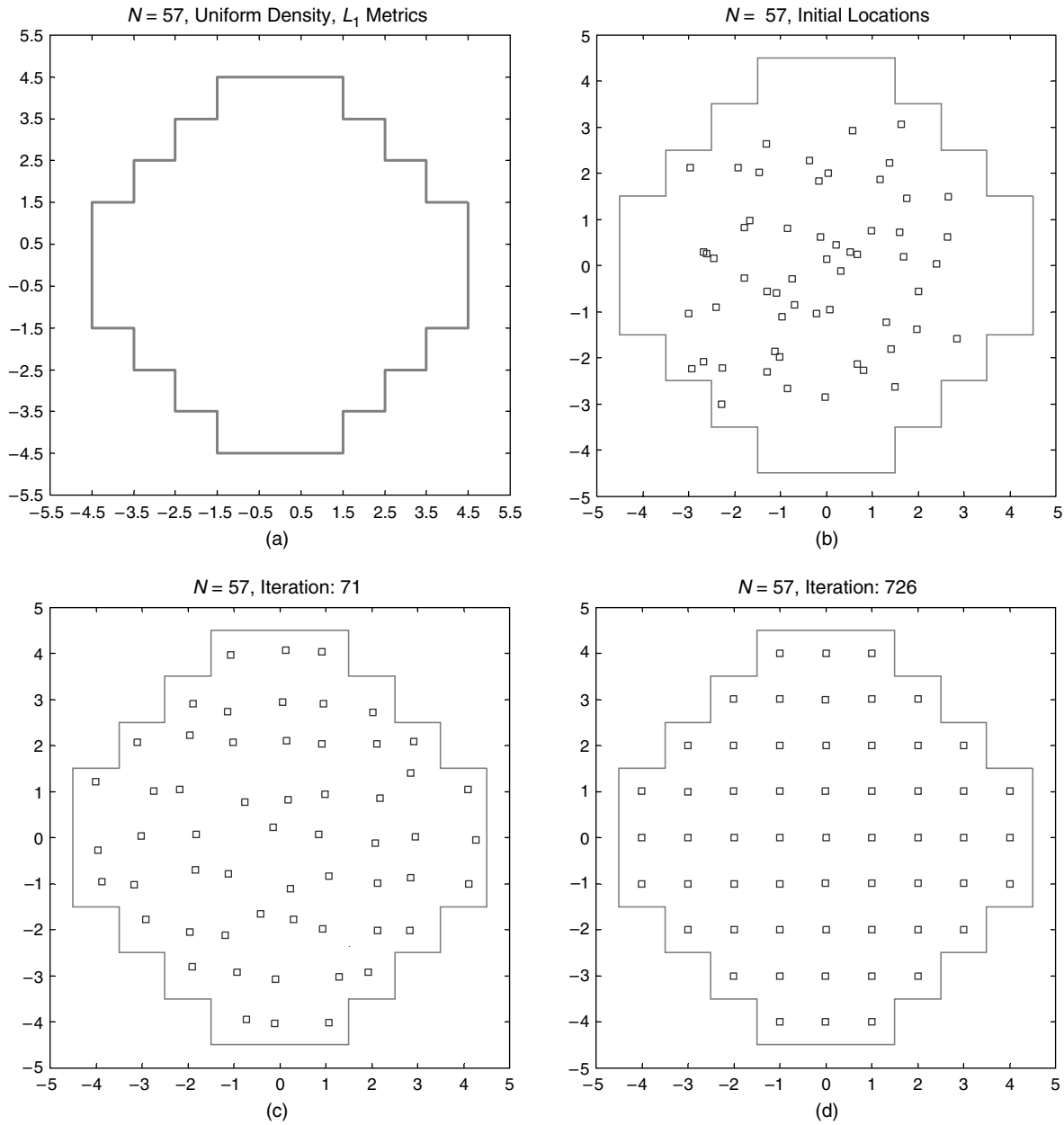


Figure 9 Algorithm Convergence in L_1 Metrics: (a) Area S ; (b) Initial Locations; (c) Locations After 71 Iterations; (d) Equilibrium Locations After 726 Iterations

Note. Tolerance $\varepsilon = 0.001$.

centrally located terminals. There are systems, however, for which terminals should not be at the center of their influence areas; e.g., newspaper distribution systems, where it is advantageous to locate drop-off spots on the edge of their delivery districts (see Daganzo 1984). In these cases the algorithm should be modified too.

The study also validates the CA cost predictions by comparing them with the costs for actual designs. The approximation improves for large problems. The CA prediction is also shown to be an approximate lower bound of the true optimum under certain conditions; and to be quite close to the costs of feasible designs.

In these cases the CA formula is a good predictor of optimum system costs. The CA discretization algorithm can be used to obtain efficient system designs for very large problems.

Acknowledgments

This research is supported in part by a research grant from the University of California Transportation Center (UCTC).

References

Daganzo, C. F. 1984. The distance traveled to visit N points with a maximum of C stops per vehicle: An analytic model and an application. *Transportation Sci.* 18(4) 331–350.

- Daganzo, C. F. 1999. *Logistics System Analysis*, 3rd ed. Springer, Berlin, Germany.
- Daganzo, C. F., G. F. Newell. 1986. Configuration of physical distribution networks. *Networks* **16** 113–132.
- Daskin, M. S. 1995. *Network and Discrete Location: Models, Algorithms and Applications*. Wiley, New York.
- Drezner, Z., H. W. Hamacher. 2002. *Facility Location: Applications and Theory*. Springer, Berlin, Germany.
- Du, Q., V. Faber, M. Gunzburger. 1999. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Rev.* **41**(4) 637–676.
- Hori, H., M. Nagata. 1985. Examples of optimization methods for environment monitoring systems. Report B-266-R-53-2, Environmental Sciences, Ministry of Education, Japan, 18–29 (in Japanese).
- Newell, G. F. 1971. Dispatching policies for a transportation route. *Transportation Sci.* **5** 91–105.
- Newell, G. F. 1973. Scheduling, location, transportation and continuum mechanics: Some simple approximations to optimization problems. *SIAM J. Appl. Math.* **25**(3) 346–360.
- Okabe, A., B. Boots, K. Sugihara. 1992. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, Chichester, UK.