VORONOI DIAGRAMS ON LINE NETWORKS AND THEIR APPLICATIONS

ZEHAN HU[†] AND ZEYU WANG[‡]

Abstract. The optimal mass transportation problem, proposed by Monge, is dominated by the Monge-Ampère equation. In general, as the Monge-Ampère equation is highly non-linear, this type of partial differential equations is beyond the solving ability of a conventional finite element method. This difficulty led Gu et al. alternatively to the discrete optimal mass transportation problem. They developed variational principles for this problem and reported the calculation for the optimal mapping, based on theorem that among all possible cell decompositions, with constrained measures, the transportation cost of the discrete mapping from cells to the corresponding discrete points is minimized by the decomposition induced by a power Voronoi diagram. Their research inspired us to consider a similar discrete optimal mass transportation problem. Here we replace the L2 Euclidean distance by the length of the shortest path connecting two points on a line network. To solve the optimal transportation problem on a line network, we study a type of Voronoi diagram on undirected and connected networks and propose an elegant construction algorithm. We further consider weighted distances in a network and develop a method to compute the centroidal Voronoi tessellation (CVT) for a network. By using real geographic data, the method proposed is proved efficient and effective in several practical applications, including charging station distribution in a traffic network, and trash can distribution in a park.

Key words. optimal mass transportation, Voronoi, CVT, line network, shortest path

1. Introduction

In 1781, the French mathematician, Monge, proposed the *optimal mass transmission problem* (a.k.a. *Monge's problem*). Let X and Y be two measure spaces with probability measures μ and ν respectively, and assume X and Y have equal total measure:

$$\int_X \mu(x)dx = \int_Y \nu(y)dy.$$

A map $T : X \to Y$ is measure preserving if for any measurable set $B \subseteq Y$, the following condition holds:

$$\int_{T^{-1}(B)} \mu(x)dx = \int_B \nu(y)dy.$$

[*]This work is advised by both Qingming Yang from Tsinghua University High School and Associate Professor Xianfeng Gu from State University of New York at Stony Brook.

[†] Tsinghua University High School, email(huzehandd@163.com)
[‡] Tsinghua University High School, email(wzy032679@sina.com)

Let $c(x,y)$ be the transportation cost for transporting $x \in X$ to $y \in Y$. Then the total transportation cost of T is given by:

$$\mathcal{E}(T) = \int_X c(x, T(x))\mu(x)dx.$$

The optimal mass transmission problem is to find a measure-preserving mapping such that the total transportation cost defined by the above formula is minimized.

In the 1940's, Kantorovich [1] relaxed Monge's problem and solved it using linear programming. At the end of the 1980's, Brenier [2] investigated the optimal transmission problem in the Euclidean space, and proved that there is a unique optimal transportation map $T:(X,\mu) \to (Y,\nu)$. Furthermore, there is a convex function $f: X \to \mathbb{R}$, unique up to a constant, such that the optimal mass transportation map is given by the gradient map $T: x \to \nabla f(x)$, if X is a convex domain and the transportation cost is the quadratic Euclidean distance.

Assuming the measures μ and ν are smooth, and f has second order continuity, then f satisfies the following Monge-Ampère equation if f is measure-preserving,

$$\det\left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right) = \frac{\mu}{\nu \circ \nabla f}.$$

In general, as the Monge-Ampère equation is highly non-linear, this type of partial differential equations are beyond the solving ability of conventional finite element methods. Alternatively, based on its geometric interpretation, we will try to solve it using a vibrational approach, by convex optimization.

The optimal mass transportation theorem holds in the discrete setting as well. Gu et al. [3] developed variational principles for the discrete case to compute the optimal mapping. Suppose Ω is a convex planar domain with a probability measure μ defined on it. Given a discrete point set $P = \{p_1, p_2, \cdots, p_n\}$, each point $p_i \in P$ is associated with a positive number A_i, such that $\sum_{i=1}^n A_i = \int_\Omega \mu(x)dx$. A cell decomposition of Ω is given by $\Omega = \bigcup_{i=1}^n D_i$, such that the probability of each cell D_i equals A_i. The discrete map induced by the decomposition is denoted by $f: D_i \to p_i$, then the transportation cost for f is given by:

$$\mathcal{E}(f) = \sum_{i=1}^n \int_{D_i} |x - p_i|^2 \mu(x)dx.$$

The power distance is defined as

$$Pow(x,p_i) = |x - p_i|^2 + h_i, h_i \in \mathbb{R}.$$

Then the power distance induces a power Voronoi diagram,

$$\Omega = \bigcup_{i=1}^{n} V_i, V_i = \{x \in \mathbb{R}^2 | Pow(x,p_i) \leq Pow(x,p_i), \forall j\}.$$

It's known that among all possible cell decompositions with the probability constraint $\int_{D_i} \mu(x)dx = A_i$, the transportation cost of the mapping $f: D_i \to p_i$ is minimized by the one induced by a power Voronoi diagram [4].

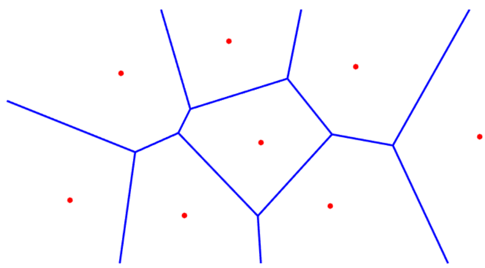

Fig. 1.1: A Voronoi diagram in the two-dimensional plane. Each red dot is the seed of the Voronoi cell, and the blue lines show the boundaries of Voronoi cells.

The Voronoi diagram is an important geometric structure in mathematics, with many important applications in physics, chemistry, biology, engineering and other fields [5]. In a Voronoi diagram, each point in the point set P is called a seed point, and the corresponding area of the seed point is called a Voronoi cell. The distance from any point in the cell to the associated seed point is less than the distance to any other seed point (seeing Figure 1.1). In 1850, Dirichlet used two-dimensional and three-dimensional Voronoi diagrams to study quadratic forms. In 1908, Georgy Voronoi defined and studied the general Voronoi diagram problem in n-dimensional space.

Moreover, the Voronoi diagram can be calculated for a set of discrete points on a given region. If the seed points are further allowed to move, the Voronoi cells are recalculated by choosing geometric centers of cells as the new seed points, and areas of the Voronoi cells can be made as equal as possible by iterations. This is the centroidal Voronoi tessellation (CVT) problem of the Voronoi diagram [6]. Wenping Wang of the University of Hong Kong and other researchers considered fast algorithms for computing

CVTs and their GPU-based acceleration [7, 8]. Wang et al. [9] investigated the CVT problem over triangular meshes.

The CVT method for a Voronoi diagram can also solve practical problems, such as electric vehicle charging stations' distribution in which people expect the coverage area of each charging station to be as equal as possible. However, the charging stations can only be distributed at the roadside, and electric cars can only run on the roads, so that the distribution problem cannot be solved by the traditional Voronoi diagram method in the Euclidean distance. After all, when we consider traffic congestion and speed limits, we do not necessarily expect the equal area of the geometric cell for each seed point, but the equal total weighted length of the streets dominated by each seed point. To optimize the distribution problem on road, we, therefore, propose a kind of Voronoi problem based on the undirected connected line network.

The problem we consider here is similar to the discrete optimal transmission problem, but we replace the L2 Euclidean distance with the shortest path distance between two points in the line network. In the following sections, based on an algorithm for the shortest paths in the network (connected graph), we first define Voronoi diagrams on undirected connected networks, and then present a method of calculating Voronoi cells. Then we extend CVT algorithm for Voronoi diagrams in the Euclidean space to the Voronoi diagram on the undirected connected line network. By iteration, thus we make the weighted lengths corresponding to seed points in the line network as equal as possible.

This paper is organized as follow. In Section 2, we define Voronoi diagrams on line networks and describe their properties, and an algorithm for calculating Voronoi cells. By applying the algorithm, we present an iterative method for finding the centroidal Voronoi in Section 3. Then Section 4 gives details on the algorithm design for Voronoi tessellation and centroidal Voronoi tessellation. Section 5 presents the experimental results with discussions, which are followed by the conclusions in Section 6.

2. Voronoi diagrams on line networks

2.1. Definition and properties

2.1.1. Line network

Line networks can represent a set of vertices and the connections between them. They have a wide range of applications, such as path planning. We first define a line

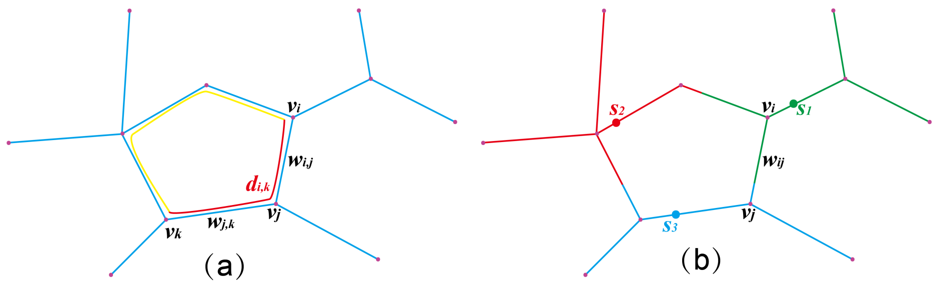

Fig. 2.1: Line networks and Voronoi diagrams over them. (a) a line network; (b) the Voronoi diagram over that line network. s_1, s_2 and s_3 are three seed points; the colors identify the corresponding Voronoi cells.

network $G=(V,E)$ in the N-dimensional space (seeing figure 2.1(a)). The set of vertices is given as,

$$V=\{v_1,v_2,\cdots,v_n\}, v_i\in\mathbb{R}^N, 1\le i\le n,$$

and the set of edges is,

$$\begin{aligned}&E=\{(v_i,v_j)|1\le i,j\le n,\\&v_i \textit{ and } v_j \textit{ are directly connected by a continous curve } \gamma_{ij} \textit{ of finite length.}\}\end{aligned}$$

In the shortest path problem, we usually assign a weight $w_{i,j}$ to each edge $(v_i,v_j)\in E$ of the line network. The weight of a path is the sum of weights of all edges on the path from vertex v_i to v_j. The path with the least total weight from vertex v_i to v_j is called the shortest path from vertex v_i to v_j. The corresponding total weight, called the shortest distance between vertices v_i and v_j, is denoted by $d_{i,j}$. If there is a path between any two points in the vertex set, then the line network is connected; otherwise the network is disconnected. If the edges are directional, that is, (v_i,v_j) and (v_j,v_i) represent different directed edges, then the network is called a directed network. In undirected networks, (v_i,v_j) and (v_j,v_i) represent the same edge and $w_{i,j}=w_{j,i}$. We also denote $w(p_i,p_j)$ as the weight between two points p_i and p_j on the same edge curve of the line network, which is proportional to the length of curve connecting p_i and p_j on the same edge curve.

2.1.2. Voronoi diagrams on undirected networks

We only consider the Voronoi diagram problem on undirected connected networks. For Voronoi diagram problems on disconnected networks, we can divide the disconnected network into several connected sub-networks and then solve the Voronoi diagram problem on each connected sub-network.

We define the network space Ω_G on the N-dimensional connected network $G=(V,E)$ as:

$$\Omega_G=\{p\in\mathbb{R}^N|p \textit{ is on curve } \gamma_{ij} \textit{ and } (v_i,v_j)\in E\}.$$

Definition 2.1. *Let $d_G(p_i,p_j)$ be the distance metric between two points p_i and p_j on Ω_G. For a given point set $S=\{s_k\}_{k=1}^m$ on Ω_G, the Voronoi cell $\widehat{V}_k$ corresponding to a point s_k is defined as*

$$\widehat{V}_k=\{p\in\Omega_G|d_G(p,s_k)\leq d_G(p,s_j),\forall j\}. \tag{2.1}$$

Then $\{\widehat{V}_k\}_{k=1}^m$ is called a Voronoi tessellation of the network space Ω_G. Figure 2.1(b) shows the Voronoi cells are determined by three seed points on the line network.

In order to define the distance between seed points and points on Ω_G, we further define a new network $G_S=(V_S,E_S)$ by adding the set of discrete seed points S to the vertices of G, $V_S=V\cup S$, and E_S is updated according to the following rules: if the curve γ_{ij} corresponding to an edge $(v_i,v_j)\in E$ does not pass through any point in S, then (v_i,v_j) is added to E_S; otherwise, if the curve γ_{ij} passes through points in S in the order of $\{s_{k_t}\}_{t=0}^l, l\geq 0$, edges $(v_i,s_{k_0}),(s_{k_l},v_j),(s_{k_t},s_{k_{t+1}}),t=0,\cdots,l-1$ are added to E_S.

For a given discrete set $S=\{s_k\}_{k=1}^m$ and network $G=(V,E)$, we define the distance between any two points $p_1,p_2\in\Omega_{G_S}$ as the length of their shortest path on network G_S, denoted by $d_{G_S}(p_1,p_2)$. Then the Voronoi tessellation $\{\widehat{V}_k\}_{k=1}^m$ of Ω_{G_S} with respect to the discrete set S has the following properties.

Proposition 2.2 (Shortest Path). *$\forall v\in\Omega_{G_S}$, if $v\in\widehat{V}_k$, we have $p\in\widehat{V}_k$ for an arbitrary point p on the shortest path from v to the underlying seed point s_k of $\widehat{V}_k$.*

Proof: As shown in figure 2.2, if there exists a point p on the shortest path (the green solid curve) from v to s_k, such that $p\in\widehat{V}_l, p\notin\widehat{V}_k, l\neq k$, according to (2.1), we have,

$$d_{G_S}(p,s_l)<d_{G_S}(p,s_k). \tag{2.2}$$

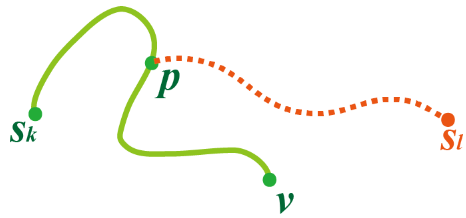

Fig. 2.2: Proof of Proposition 2.2. The green solid line represents the shortest path from v to the seed point s_k of the Voronoi cell v belongs to. p is an arbitrary point on the path and s_l is another seed. The red dotted line is the shortest path from p to s_l.

Immediately, (2.2) gives,

$$d_{G_S}(v,p)+d_{G_S}(p,s_l)<d_{G_S}(v,p)+d_{G_S}(p,s_k). \tag{2.3}$$

Since the sub-path of a shortest path is also a shortest path connecting the corresponding endpoints of the sub-path, we have,

$$d_{G_S}(v,s_k)=d_{G_S}(v,p)+d_{G_S}(p,s_k). \tag{2.4}$$

(2.3) and (2.4) together give that,

$$d_{G_S}(v,p)+d_{G_S}(p,s_l)<d_{G_S}(v,s_k). \tag{2.5}$$

This means that we find a closer seed point, s_l, for v, which is contrary to $v\in\widehat{V}_k$.■

We can see that if both endpoints of an edge belong to the same Voronoi cell, any points on the edge must also belong to that cell. Since a point may be the intersection of two or more cells, we have the following property for Ω_{G_S}.

Proposition 2.3 (Closure). *$\forall(v_i,v_j)\in E_S$, if $v_i\in\bigcap_{k=1}^{q}\widehat{V}_{i_k}, v_j\in\bigcap_{l=1}^{t}\widehat{V}_{j_l}, q\geq 1, t\geq 1$, for any point p on the underlying curve, γ_{ij} of the edge (v_i,v_j), we obtain $p\in(\bigcup_{k=1}^{q}\widehat{V}_{i_k})\cup(\bigcup_{l=1}^{t}\widehat{V}_{j_l})$.*

Proof: As shown in figure 2.3, assume that there exists a point p on curve γ_{ij}, such that $p\in\widehat{V}_r, r\neq i_k, r\neq j_l, k=1,\cdots,q, l=1,\cdots,t$. Without loss of generality, assume that the shortest path from p to s_r goes through vertex v_i. According to Proposition 2.2, we have $v_i\in\widehat{V}_r$. Since $r\neq i_k, k=1,\cdots,q$, it is contrary to $v_i\in\bigcap_{k=1}^{q}\widehat{V}_{i_k}$.■

Proposition 2.4 (Locality). *If $\{\widehat{V}_{c_i}\}_{i=1}^{k}, k\geq 1$ are all the adjacent Voronoi cells*

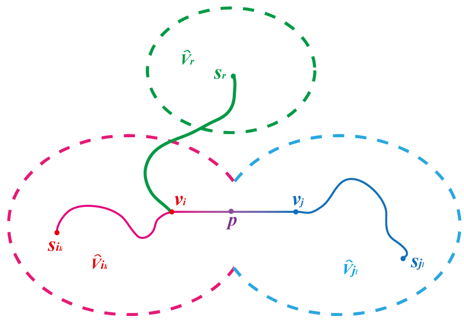

Fig. 2.3: Proof of Proposition 2.3. (v_i, v_j) is an edge of the line network, p is an arbitrary point on the underlying curve of the edge. $v_i \in \widehat{V}_{i_k}, v_j \in \widehat{V}_{j_l}$, $\widehat{V}_r$ is another Voronoi cell (shown as dotted regions). Assume that the shortest path (the green solid curve) from p to s_r goes through vertex v_i.

to Voronoi cell $\widehat{V}_c$*, for an arbitrary point* $p \in \widehat{V}_c$ *and an arbitrary non-adjacent cell* $\widehat{V}_f$ *of* $\widehat{V}_c$*, there exists an* $i \in \{1, \cdots, k\}$*, such that* $d_{G_S}(p, s_{c_i}) < d_{G_S}(p, s_f)$.

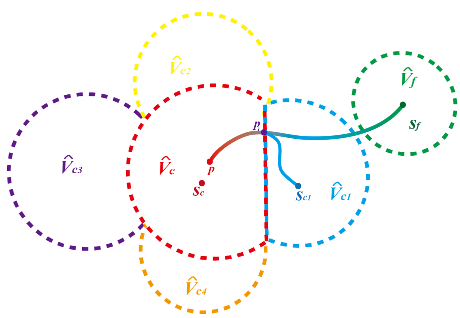

Fig. 2.4: Proof of Proposition 2.4. Cell $\widehat{V}_c$ is adjacent to $\widehat{V}_{c_1}, \widehat{V}_{c_2}, \widehat{V}_{c_3}, \cdots$, except $\widehat{V}_f$. Point p is an arbitrary point in $\widehat{V}_c$. Without loss of generality, assume that the shortest path from p to s_f goes through $\widehat{V}_{c_1}$, and the entrance point in $\widehat{V}_{c_1}$ for this path is p_i. The blue solid curve depicts the shortest path form p_i to s_{c_1}.

Proof: As shown in figure 2.4, without loss of generality, assume that the shortest path from p to seed point s_f of $\widehat{V}_f$ goes through $\widehat{V}_{c_1}$, and the entrance point in $\widehat{V}_{c_1}$ for this path is p_i. Since the sub-path of a shortest path is also a shortest path connecting the corresponding endpoints of the sub-path, we have,

$$d_{G_S}(p, s_f) = d_{G_S}(p, p_i) + d_{G_S}(p_i, s_f). \tag{2.6}$$

As $\widehat{V}_c$ is not adjacent to $\widehat{V}_f$, but adjacent to $\widehat{V}_{c_1}$, it is clear that, $p_i \in \widehat{V}_{c_1}$ and $p_i \notin \widehat{V}_f$.

According to (2.1), this means,

$$d_{G_S}(p_i, s_{c_1}) < d_{G_S}(p_i, s_f). \tag{2.7}$$

(2.6) and (2.7) yield,

$$d_{G_S}(p, p_i) + d_{G_S}(p_i, s_{c_1}) < d_{G_S}(p, p_i) + d_{G_S}(p_i, s_f) = d_{G_S}(p, s_f). \tag{2.8}$$

Since the sub-path from point p to p_i, and the shortest path between p_i and s_{c_1} together give a path from point p to seed point s_{c_1}, we have,

$$d_{G_S}(p, s_{c_1}) \leq d_{G_S}(p, p_i) + d_{G_S}(p_i, s_{c_1}). \tag{2.9}$$

Combining the inequalities in (2.8) and (2.9), we obtain $d_{G_S}(p, s_{c_1}) < d_{G_S}(p, s_f)$.■

From Proposition 2.4, we can conclude that adding or removing a seed point in the network only impacts the cells influenced by the seed, i.e., cells containing the seed and cells adjacent to those cells.

2.2. Computation of the Voronoi diagram on an undirected connected line network

According to the definition and analysis in Section 2.1, given an undirected connected line network $G = (V, E)$ and a seed point set $S = \{s_k\}_{k=1}^m$, the computation of the Voronoi tessellation of network space Ω_G with regard to S requires us to locate the nearest seed points in S for given vertices in V_S, or points on underlying curves of edges in E_S. So we have the following definition:

Definition 2.5. *Let $Vor : \Omega_{G_S} \to 2^S$ be a map from network space Ω_{G_S} to the power set of the seed point set S. $Vor(p)$ indicates the set of seed points whose seeds are nearest to a given point $p \in \Omega_{G_S}$.*

So, the computation of the Voronoi diagram on an a line network is equivalent to find out the map Vor.

2.2.1. Voronoi cells for vertices in V_S

Vertices in V_S can be divided into two kinds, i.e., the set of vertices V on the line network G and the set of seed points S. Since for any seed point $s \in S, d_{G_S}(s, s) = 0$, we easily obtain,

$$Vor(s) = \{s\}, \forall s \in S. \tag{2.10}$$

According to the definition in (2.1), for any vertex $v \in V$, we have,

$$Vor(v) = \{\bar{s} \in S | d_{G_S}(v, \bar{s}) \leq d_{G_S}(v, s), \forall s \in S\}, \forall v \in V. \tag{2.11}$$

2.2.2. Voronoi cells for points on edges in E_S

Proposition 2.2 tells that if the shortest path from a vertex to the seed point of the vertex's containing Voronoi cell includes the edges adjacent to the vertex, then all the points on that edge belong to the containing cell. Otherwise, the points on the edge may belong to different cells. Thus we have the following theorem.

THEOREM 2.6. *If $(v_i, v_j) \in E_S$, denote $S_{i,j} = Vor(v_i) \cap Vor(v_j)$, where d_i is the distance from v_i to the seed points in $Vor(v_i)$, and d_j is the distance from v_j to the seed points in $Vor(v_j)$.*

(i) if $|d_i - d_j| = w_{i,j}$, then for any point p on the underlying curve γ_{ij} of edge (v_i, v_j) (excluding vertices v_i and v_j), $Vor(p) = S_{i,j}$;

(ii) otherwise, there exists a cut point p_c on curve γ_{ij}, such that $Vor(p_c) = Vor(v_i) \cup Vor(v_j)$ and the weight from p_c to v_i on curve γ_{ij} is

$$w(p_c, v_i) = (w_{i,j} + d_j - d_i)/2. \tag{2.12}$$

Proof:*(i)* Without loss of generality, assume that,

$$d_j = d_i + w_{i,j}. \tag{2.13}$$

For any seed point $s' \in Vor(v_i)$, obviously we have $d_{G_S}(s', v_i) = d_i$. So, (2.13) gives,

$$d_{G_S}(s', v_i) + w_{i,j} = d_j. \tag{2.14}$$

This means there exists a path from v_j to seed point s' with the same distance d_j. Since d_j is the minimal length of the path from v_j to all seed points, we immediately conclude that $s' \in Vor(v_j)$. This means,

$$Vor(v_i) \subseteq Vor(v_j). \tag{2.15}$$

So, we have,

$$S_{i,j} = Vor(v_i). \tag{2.16}$$

Since the sub-path of a shortest path is also a shortest path connecting the corresponding endpoints of the sub-path, we conclude from (2.14) that *there exists a shortest path from* v_j *to* s' *which passes through edge* (v_i, v_j). According to Proposition 2.2, for any point p on curve γ_{ij} (excluding vertices v_i and v_j), p also belongs to the Voronoi cell containing s'. This gives,

$$S_{i,j} \subseteq Vor(p). \tag{2.17}$$

Consider any seed point $s'' \in Vor(v_j) \backslash S_{i,j}$, any seed point $s' \in S_{i,j}$, and any point p on curve γ_{ij} (vertices v_i and v_j are excluded). From (2.13), we obtain,

$$d_i = d_{G_S}(p, s') < d_{G_S}(p, s'') = d_j. \tag{2.18}$$

According to the definition of shortest path, we have,

$$d_{G_S}(p, s'') = \min(w(p, v_j) + d_{G_S}(v_j, s''), w(p, v_i) + d_{G_S}(v_i, s'')). \tag{2.19}$$

Since there exists a shortest path from v_j to s' which passes through edge (v_i, v_j), we also have,

$$d_{G_S}(p, s') = w(p, v_i) + d_{G_S}(v_i, s'). \tag{2.20}$$

Again, (2.13) gives,

$$d_j = d_{G_S}(v_j, s'') = d_{G_S}(v_j, s') = d_{G_S}(v_i, s') + w_{i,j}. \tag{2.21}$$

So, (2.18)–(2.21) together yield,

$$d_{G_S}(p,s') < d_{G_S}(p,s''). \tag{2.22}$$

This immediately implies that for any $s'' \in Vor(v_j) \backslash S_{i,j}$,

$$s'' \notin Vor(p). \tag{2.23}$$

We known from Proposition 2.3 that,

$$Vor(p) \subseteq Vor(v_i) \cup Vor(v_j). \tag{2.24}$$

Combining (2.16), (2.17), (2.23), (2.24), we have $Vor(p) = S_{i,j}$.

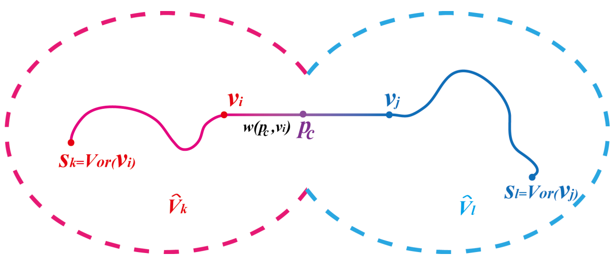

Fig. 2.5: Proof of Theorem 2.6(ii). p is an arbitrary point on edge (v_i,v_j), $s_k \in Vor(v_i), s_l \in Vor(v_j), s_k \notin Vor(v_j), s_l \notin Vor(v_i)$, $w(p,v_i)$ is the weight from p to v_i on the underlying curve $\gamma_{i,j}$. The solid curves depict shortest paths.

(ii). Without loss of generality, assume $d_i < d_j$, then we must have,

$$d_j < d_i + w_{i,j}. \tag{2.25}$$

Otherwise, if $d_j > d_i + w_{i,j}$, we can find a shorter path between v_j and the seed points. This is contrary to the fact that d_j is the minimal length of paths from v_j to all seed points. So,

$$0 < (w_{i,j} + d_j - d_i)/2 < w_{i,j}. \tag{2.26}$$

Proposition 2.3 tells that, for any point p on curve γ_{ij}, we have,

$$Vor(p) \subseteq Vor(v_i) \cup Vor(v_j). \tag{2.27}$$

If $Vor(v_i)$ is equal to $Vor(v_j)$, Obviously, (ii) holds from Propostion 2.3 and (2.26). Otherwise, as shown in figure 2.5, consider any seed point $s_k \in Vor(v_i)$ and any seed point $s_l \in Vor(v_j)$, $s_k \notin Vor(v_j), s_l \notin Vor(v_i)$. Without loss of generality, assume that

$$d_{G_S}(p, s_k) < d_{G_S}(p, s_l). \tag{2.28}$$

We claim that the shortest path from p to s_k must first meet vertex v_i and then reach seed point s_k. Otherwise, the shortest path first meets v_j and then reaches s_k. According to the definition of shortest path, we have

$$d_{G_S}(p, s_k) = w(p, v_j) + d_{G_S}(v_j, s_k). \tag{2.29}$$

Since d_j is the minimal length of paths from v_j to all seed points and $s_k \notin Vor(v_j)$, we obtain

$$d_{G_S}(v_j, s_k) > d_j = d_{G_S}(v_j, s_l). \tag{2.30}$$

Combining (2.29),(2.30) and the definition of shortest path, we have

$$d_{G_S}(p, s_k) > w(p, v_j) + d_{G_S}(v_j, s_l) \geq d_{G_S}(p, s_l). \tag{2.31}$$

This is contrary to the assumption in (2.28). Based on (2.27) and (2.28), we conclude that,

$$s_k \in Vor(p), s_l \notin Vor(p). \tag{2.32}$$

For any point p satisfying (2.28), based on the arbitrariness of s_k and s_l, (2.27) and (2.32) give,

$$Vor(p) = Vor(v_i). \tag{2.33}$$

Since curve γ_{ij} is continuous, as point p continuously varies from one endpoint to another on the curve, the minimal length of the paths from p to all the seed points is also continuous. Obviously, this minimal length is bounded and thus the maximum of this minimal length exists. Assume that the maximum is reached at point p_c on the curve γ_{ij}. We claim that for any seed point $s_k \in Vor(v_i)$ and any seed point $s_l \in Vor(v_j)$, $s_k \notin Vor(v_j), s_l \notin Vor(v_i)$,

$$d_{G_S}(p_c, s_k) = d_{G_S}(p_c, s_l). \tag{2.34}$$

Otherwise, without loss of generality, we assume $d_{G_S}(p_c, s_k) < d_{G_S}(p_c, s_l)$. According to (2.33), we know that minimal length of the paths from p_c to all the seed points is $d_{G_S}(p_c, s_k)$. Since the minimal length of the paths from p to all the seed points is continuous *w.r.t.* p on curve γ_{ij}, we can always find a change $|\delta p| > 0$ small enough toward v_j, such that $d_{G_S}(p_c + \delta p, s_k) < d_{G_S}(p_c + \delta p, s_l)$. Thus, the shortest path from $p_c + \delta p$ to s_k also first goes through v_i. So, we have $d_{G_S}(p_c + \delta p, s_k) = d_{G_S}(p_c, s_k) + |\delta p| > d_{G_S}(p_c, s_k)$, which is contrary to the maximality of $d_{G_S}(p_c, s_k)$.

Immediately, (2.34) implies

$$Vor(p_c) = Vor(v_i) \cup Vor(v_j). \tag{2.35}$$

Since the shortest path from p_c to s_k first goes through v_i, we have

$$d_{G_S}(p_c, s_k) = w(p_c, v_i) + d_i. \tag{2.36}$$

Similarly, we have

$$d_{G_S}(p_c, s_l) = w(p_c, v_j) + d_j. \tag{2.37}$$

We also know

$$w_{i,j} = w(p_c, v_i) + w(p_c, v_j). \tag{2.38}$$

Substituting (2.36)–(2.38) into (2.34), we finally obtain (2.12).■

For any edge $(v_i, v_j) \in E_S$, we can judge whether a cut point exists on the edge or not, based on Theorem 2.6. If it exists, (2.12) locates the cut point.

According to the definition of Voronoi cells, we can also define Voronoi cells by means of the map Vor,

$$\widehat{V}_k = \{p \in \Omega_{G_S} | s_k \in Vor(p)\}, k = 1, \cdots, m. \tag{2.39}$$

Consider the sub-network $G_k=(V_k,E_k)$ formed by all vertices and cut points in Voronoi cell $\widehat{V}_k$. We can easily obtain

$$\widehat{V}_k=\Omega_{G_k},k=1,\cdots,m. \tag{2.40}$$

This indicates that the Voronoi tessellation $\{\widehat{V}_k\}_{k=1}^m$ of the network space Ω_{G_S} implies a partition $\{G_k\}_{k=1}^m$ of the underlying line network G_S.

3. CVT on a line network

3.1. CVT on $\mathbb{R}^N$

Given an open set $\Omega\subset\mathbb{R}^N$ and a probability density function ρ defined on Ω, the mass centroid s^* of Ω is usually defined as

$$s^*=\frac{\int_\Omega \mathbf{y}\rho(\mathbf{y})d\mathbf{y}}{\int_\Omega \rho(\mathbf{y})d\mathbf{y}} \tag{3.1}$$

Given m seed points $\{s_k\}_{k=1}^m$ on Ω, we can determine their associated Voronoi cells $\{\widehat{V}_k\}_{k=1}^m$. In turn, given regions $D_k,k=1,\cdots,m$, we can also calculate their mass centers $s_k^*,k=1,\cdots,m$ according to (3.1). Given a Voronoi tessellation $\{\widehat{V}_k\}_{k=1}^m$ of Ω, if the seed points for the Voronoi cells are themselves the mass centers of those cells, i.e., $s_k=s_k^*,k=1,\cdots,m$, we call such a tessellation a centroidal Voronoi tessellation. Obviously, an arbitrary set of seed points will unlikely be the mass centroids of the associated Voronoi cells.

Next, to calculate the CVT for a line network, we introduce an iterative algorithm, called *Lloyd's method* [10]. Lloyds method updates the Voronoi tessellation and seed points in each iteration. Given the number of seed points, m, an open region $\Omega\subset\mathbb{R}^N$ and a probability density function on Ω, the main procedures of Lloyd's method are as follows:

1. Randomly select m seed points $\{s_k\}_{k=1}^m$;
2. Calculate the associated Voronoi tessellation $\{\widehat{V}_k\}_{k=1}^m$ of Ω with seed points $\{s_k\}_{k=1}^m$;
3. Calculate the mass centroids, s_k^*, for each Voronoi cell $\widehat{V}_k,k=1,\cdots,m$;
4. If centroids and seed points meet the given convergence criterion, the algorithm terminates; otherwise, take the centroids as new seed points and go to Step 2.

This algorithm has been proved convergent [6, 11].

3.2. CVT on a line network

Given a Voronoi tessellation on a line network, we also expect the average distances to be equal between points and their associated seed points for all Voronoi cells. This is common in real scenarios. For example, we hope to set charging stations in a city such that electric automobile drivers can go to their nearest charging station in nearly equal time in average. Again, we hope to place garbage bins in a park to ensure that people travel roughly equally average distance to walk to the nearest bin when they need to drop litter.

For this end, given an undirected connected line network $G=(V,E)$, a probability density function ρ on network space Ω_G, we define the mass centroid $\tilde{s}$ for Ω_G as

$$\tilde{s}=\arg\min_{p\in\Omega_G}\sum_{(v_i,v_j)\in E}\int_{\gamma_{ij}} d_G^2(p,y)\rho(y)dy. \tag{3.2}$$

Though line network space is usually not convex, one can prove that (3.2) and (3.1) are essentially the same when used in a convex space.

Now, we generalize CVT on N-dimensional to line networks. Given an undirected connected line network $G=(V,E)$, a probability density function ρ on network space Ω_G and m seed points $S=\{s_k\}_{k=1}^m$, if the mass centroids of Voronoi cells $\{\widehat{V}_k\}_{k=1}^m$ corresponding to S are also the same as their seed points, we refer to this Voronoi tessellation as the centroidal Voronoi tessellation of Ω_G.

Following the cost function used in discrete optimal transportation problem, given any seed points set $S=\{s_k\}_{k=1}^m$ on Ω_G and any tessellation $T=\{T_k\}_{k=1}^m$ of Ω_G, we generalize the cost function to line networks and have

$$\mathcal{E}(S,T)=\sum_{k=1}^{m}\sum_{(v_i,v_j)\in\widetilde{E}_k}\int_{\widetilde{\gamma}_{ij}} d_G^2(y,s_k)\rho(y)dy, \tag{3.3}$$

where $\widetilde{E}_k$ is edge set of T_k. Denote by $V(S)=\{\widehat{V}_k\}_{k=1}^m$ the Voronoi tessellation of Ω_G corresponding to seed points set $S=\{s_k\}_{k=1}^m$. We have the following results.

THEOREM 3.1. *Denote by $\{S_n\}$ the seed points sequence during iterations of Lloy's algorithm used to calculate CVT on a line network G. We have*

$$\mathcal{E}(S_n,V(S_n))\le\mathcal{E}(S_{n-1},V(S_{n-1})). \tag{3.4}$$

Proof: According to the definition in (3.2), given T, when $\mathcal{E}(S,T)$ is minimized, s_k is the mass centroid of T_k, $k=1,\cdots,m$. So, we have

$$\mathcal{E}(S_{n+1},V(S_n))=\min_{S}\mathcal{E}(S,V(S_n)). \tag{3.5}$$

Following [6], we fix the seed point set S. Consider any tessellation T which is not the Voronoi tessellation $\{\widehat{V}_j\}_{j=1}^m$ corresponding to S. We compare (3.3) with the following

$$\mathcal{E}(S,\{\widehat{V}_j\}_{j=1}^m)=\sum_{j=1}^{m}\sum_{(v_k,v_l)\in E_j}\int_{\gamma_{kl}} d_G^2(y,s_j)\rho(y)dy, \tag{3.6}$$

where E_j is edge set of $\widehat{V}_j$.

For any $y\in\widehat{V}_j$, we have

$$\rho(y)d_G^2(y,s_j)\leq\rho(y)d_G^2(y,s_k), \tag{3.7}$$

because $y\in\widehat{V}_j$ but y may not belong to the Voronoi cell of s_k, i.e., $y\in T_k$ but T_k may not be the Voronoi cell of s_k. Since T is not a Voronoi tessellation, (3.7) must hold with strict inequality over some measurable set of Ω_G. So, we obtain

$$\mathcal{E}(S,\{\widehat{V}_j\}_{j=1}^m)<\mathcal{E}(S,T). \tag{3.8}$$

Immediately, (3.8) implies

$$\mathcal{E}(S_n,V(S_n))=\min_{T}\mathcal{E}(S_n,T). \tag{3.9}$$

Substitute n into $n-1$ in (3.5), we obtain

$$\mathcal{E}(S_n,V(S_{n-1}))\leq\mathcal{E}(S_{n-1},V(S_{n-1})). \tag{3.10}$$

Additionally, (3.9) gives

$$\mathcal{E}(S_n,V(S_n))\leq\mathcal{E}(S_n,V(S_{n-1})). \tag{3.11}$$

Finally, (3.10) and (3.11) together give the expected result (3.4).■

Algorithm 1 Define distance metric on line network

Input: Line network $G=(V,E)$,and weights W
Output: Distance $Dist$

```
function ShortestPath(G, W)
    for all v ∈ V do
        Dist(v) ← Dijkstra(G,W,v)
    end for
end function
```

According to Theorem 3.1, when we use Lloyd's algorithm to iteratively find the CVT of a line network, the cost function defined in (3.3) declines monotonically. Since the cost function has a lower boundary (not negative), the sequence $\{\mathcal{E}(S_n,V(S_n))\}$ converges, resulting in a corresponding CVT of line network G.

4. Algorithms

In this section, we develop algorithms for finding the Voronoi diagram and centroidal Voronoi tessellation of line networks.

4.1. Voronoi diagram on a line network

For a given undirected network $G=(V,E)$ with weight W and initial seeds $S=\{s_k\}_{k=1}^m$ on network space Ω_G, to obtain a Voronoi tessellation on network G, we must compute Voronoi cells $\{\widehat{V}_k\}_{k=1}^m$ for corresponding seeds. According to Section 2.2, calculating the Voronoi diagram on the undirected network is to calculate the length of the shortest path between any two points in the network, that is, to define the distance between any two points on the network space Ω_{G_S}. So, we need to calculate the shortest path length between vertices on the line network G_S. For more details about the shortest path algorithms, please refer to [12].

We consider only connected networks with non-negative weights. We could traverse each vertex of the network and use *Dijkstra's algorithm* [12] to get the shortest path between any two vertices.

The function for calculating the shortest path between any two vertices in a network G is denoted by *ShortestPath*,shown in Algorithm 1, and results are stored in a data structure *Dist*. We use $Dist(v)$ to hold the shortest path lengths from vertex v to the remaining vertices, and $Dist(v_i,v_j)$ to hold the shortest path length between any two vertices v_i and v_j. $Dijkstra(G,W,v)$ represents Dijkstra's algorithm to calculate the shortest path lengths from vertex v to the rest of the vertices on a given line network G and corresponding weights W. The pseudo-code for the *ShortestPath* is as follows:

Algorithm 2 Add seed points into line network and update distances

Input: Line network $G=(V,E)$, weights W, distance $Dist$ and seed points $S=\{s_k\}_{k=1}^m$
Output: New line network $G_S=(V_S,E_S)$, weights W_S and distance $Dist_S$

function UPADATEGRAPH(G, W, $Dist$, S)
  $V_S \leftarrow V$, $E_S \leftarrow E$, $W_S \leftarrow W$, $Dist_S \leftarrow Dist$;
  for all $s \in S$, suppose s is on edge $(v_i,v_j) \in E_S$ **do**
    $E_S \leftarrow E_S \cup (\{(s,v_i),(s,v_j)\})$;
    $W_S(s,v_i) \leftarrow w(s,v_i)$,$W_S(s,v_j) \leftarrow w(s,v_j)$,$W_S(v_i,v_j) \leftarrow +\infty$, $W_S(s,s) \leftarrow 0$;
    for all $v \in V_S$ **do**
      $Dist_S(s,v) \leftarrow \min(Dist_S(v,v_i)+W_S(s,v_i), Dist_S(v,v_j)+W_S(s,v_j))$;
    end for
    $V_S \leftarrow V_S \cup \{s\}$,$Dist_S(s,s) \leftarrow 0$;
  end for
end function

Algorithm 3 Compute mapping Vor

Input: Network $G_S=(V_S,E_S)$, seeds set $S=\{s_k\}_{k=1}^m$weights $W_S$and distance $Dist_S$
Output: Mapping Vor, and cut points set C

function VORONOICUT(G_S, W_S, $Dist_S$, S)
  for all $v \in V_S$ **do**
    Compute $Vor(v)$ based on (2.10) and (2.11);
  end for
  $C \leftarrow \emptyset$;
  for all $(v_i,v_j) \in E_S$ **do**
    if there exists a cut point p_c on (v_i,v_j) based on Theorem 2.6 **then**
      Calculate p_c's position according to (2.12);
      $Vor(p_c) = Vor(v_i) \cup Vor(v_j)$;
      $C \leftarrow C \cup \{p_c\}$;
    end if
  end for
end function

In the above algorithm, we add initial seeds $S=\{s_k\}_{k=1}^m$ into the vertex set of the network G giving an updated network $G_S=(V_S,E_S)$ with corresponding weights. In Algorithm 2, *UpdateGraph* finds the shortest path distances between the vertices on the updated network G_S.

Here, the function *UpdateGraph* adds seeds in S into the original network, updates the corresponding vertices and edges, and calculates the shortest path lengths from those newly joined points to the original vertices. In this algorithm, $\min(a,b)$ is the smaller of a and b, $w(v_i,v_j)$ denotes the weight of the curve connected directly between v_i and v_j (without passing through any other vertices); otherwise, $w(v_i,v_j)=+\infty$.

In the next step, we can compute the Voronoi cells of $G_S=(V_S,E_S)$ and determine cut points on edges. Function *VoronoiCut* performs this task, as shown in Algorithm 3.

Algorithm 4 Compute Voronoi tessellation for line networks

Input: Line network $G=(V,E)$,weights W,distance $Dist$,and seed points set $S=\{s_k\}_{k=1}^m$

Output: Voronoi tessellation $\{\widehat{V}_k\}_{k=1}^m$ on network space Ω_G

1: **function** GraphVoronoi(G, W, $Dist$, S)

2: $(G_S, W_S, Dist_S) \leftarrow UpdateGraph(G, W, Dist, S)$;

3: $(Vor, C) \leftarrow VoronoiCut(G_S, W_S, Dist_S, S)$;

4: Construct $\{\widehat{V}_k\}_{k=1}^m$ from Vor and C;

5: **end function**

Algorithm 5 Compute centroidal Voronoi tessellation for line networks

Input: Line network $G=(V,E)$,weights W,and the number of seed points m

Output: centroidal Voronoi tessellation $\{\widehat{V}_k\}_{k=1}^m$ on network space Ω_G

1: **function** GraphCVT(G, W, m)

2: Generate m random seed points $S=\{s_k\}_{k=1}^m$;

3: $Dist \leftarrow ShortestPath(G, W)$

4: $\{\widehat{V}_k\}_{k=1}^m \leftarrow GraphVoronoi(G, W, Dist, S)$;

5: Compute the centers, $\{\widetilde{s}_k\}_{k=1}^m$, of all Voronoi cells by (3.2);

6: **if** $\{s_k\}_{k=1}^m$ and $\{\widetilde{s}_k\}_{k=1}^m$ satisfy the given convergence criterion **then**

7: go to Step 12;

8: **else**

9: $S \leftarrow \{\widetilde{s}_k\}_{k=1}^m$;

10: go to Step 4;

11: **end if**

12: **end function**

Finally, Algorithm 4 presents the Voronoi tessellation algorithm *GraphVoronoi* for a line network by combining above algorithms.

4.2. Centroidal Voronoi tessellation algorithm on networks

We use Lloyd iteration to solve the problem of centroidal Voronoi tessellation. Combining the previous algorithms, we can design a centroidal Voronoi tessellation algorithm *GraphCVT* for line networks as listed in Algorithm 5.

In all experiments, we adopted the convergence criterion that the sum of the shortest path lengths between the corresponding points of $\{s_k\}_{k=1}^m$ and $\{\widetilde{s}_k\}_{k=1}^m$ is less than a given tolerance ϵ, i.e., $\sum_{k=1}^m d_G^2(s_k, \widetilde{s}_k) < \epsilon$.

5. Experimental results and discussions

In this section, we illustrate the effectiveness of the proposed method for typical rectangular and triangular grid examples, and provide several practical applications. In each case, the probability density of the points in the network space is evenly distributed according to the weight of the edge.

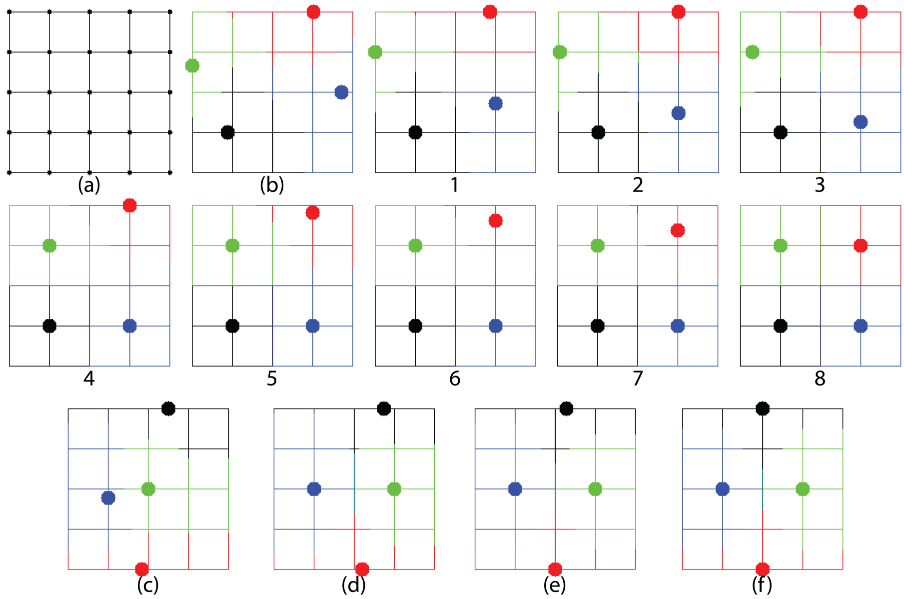

Fig. 5.1: Centroidal Voronoi tessellation on a rectangular network with 4 initial seed points.

5.1. Rectangular and triangular grid networks

We have demonstrated the effectiveness of the proposed method by illustrating rectangular and triangular grid networks. Figure 5.1(a) shows a 5×5 rectangular mesh with 25 vertices, with equal weights on each edge. As shown in Figure 5.1(b), after four seed points have been randomly generated, a Voronoi tessellation is obtained. After eight iterations, it converges to a final centroidal Voronoi tessellation. Figure 5.1(c) shows another set of initial seed points and converges after only three iterations as shown in Figure 5.1(d)–(f). Both of these results are reasonable.Please note that the dots in all subfigures ,except subfigure (a), denote seed points. The Voronoi cells are identified by lines with the same color as the seed points. The color of an edge shared by cells is the average of the color of corresponding seeds in the cells.

Figure 5.2 shows an example of a planar triangular grid mesh. In Figure 5.2(a), each edge has the same weight. Figure 5.2(b) shows 3 randomly distributed seeds and their Voronoi cells. The seeds will algorithm converges after four iterations, and the average distances from each point in a cell to its seed point are equal. According to the symmetry, we can choose another vertex in the same edge as the seed point, which is different from the convergence result in Figure 5.2, and it is also a solution to this

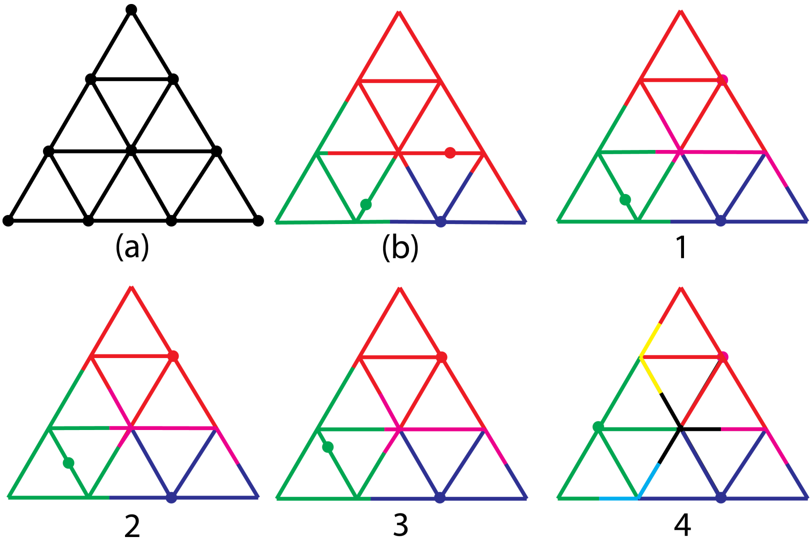

Fig. 5.2: Centroidal Voronoi tessellation on a triangular grid network with 3 initial seed points.

example. Similar with to the example in Figure 5.1, the Voronoi cells are identified by lines with the same color as the seed points, the color of the edge shared by neighboring cells is the average of the color of corresponding seeds in the cells and edges shared by all seeds are colored in black.

5.2. Practical applications

We apply the proposed centroidal Voronoi tessellation method to three practical situations: the trash cans in the Beijing Olympic Forest Park, the distribution of the electric car charging stations in Beijing, and optimization of the distribution of logistics or express depot around the expressways in China.

5.2.1. Distribution of the trash cans in the Beijing Olympic Forest Park

In our first application is to the Olympic Forest Park, we aim to put trash cans along the roadside, so that visitors in any places can reach to the nearest trash with almost the same time. For this objective, we use the initial locations of the trash cans as seed points, and compute the optimal locations by centroidal Voronoi tessellation on the line network of the park's roads.

Figure 5.3 maps out the Olympic Forest Park and the extracted line network (276 vertices, 434 edges, and the edge weights are the corresponding road lengths). Figure 5.4 shows the results of the proposed algorithm. The left column of Figure 5.4 shows a case

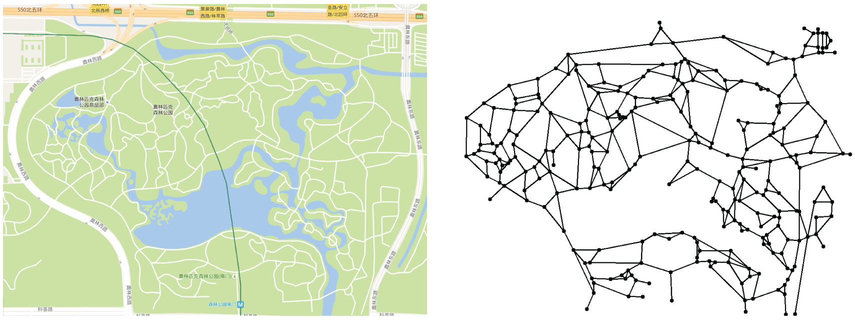

Fig. 5.3: Map of the Olympic Forest Park (left) and the extracted line network (right). Black points are road intersections, and lines are network connections.

with 10 trash cans. From top to bottom are randomly distributed initial seeds and locations after 1, 4, and 7 iterations respectively. The right column of Figure 5.4 shows a case for 20 trash cans. From top to bottom are randomly distributed initial seeds and locations after 1, 14, and 27 iterations respectively. Please note the algorithm converges after 7 and 27 iterations, which is very efficient. Different colors identify different Voronoi cells, and the seed points in different cells are represented by dots of corresponding colors.

5.2.2. Distribution of charging stations for electric cars in Beijing

As electric vehicles become more and more popular, the demand is increasing for electric vehicle charging stations. How to plan the locations of charging stations becomes an interesting problem, so as to allow users to quickly find their respective nearest charging station and improve the utilization of each charging station. Specifically, taking the main traffic roads within Beijing as an example, we use the proposed method to demonstrate how to determine the locations of 50 charging stations.

The right figure in Figure 5.5 shows the network of Beijing map with 1034 vertices and 1801 edge, with edge weights being the corresponding lengths of the roads.

Figure 5.6 and Figure 5.7 show that 50 charging stations converge to different locations under two different initial seed settings, demonstrating that the iterative method in this paper is locally convergent (discussed further in Section 5.3). From the convergent locations of charging stations, it can be found that some locations in two cases are highly consistent with each other, especially in the outer layer.

5.2.3. Optimization of depot locations for express or logistics

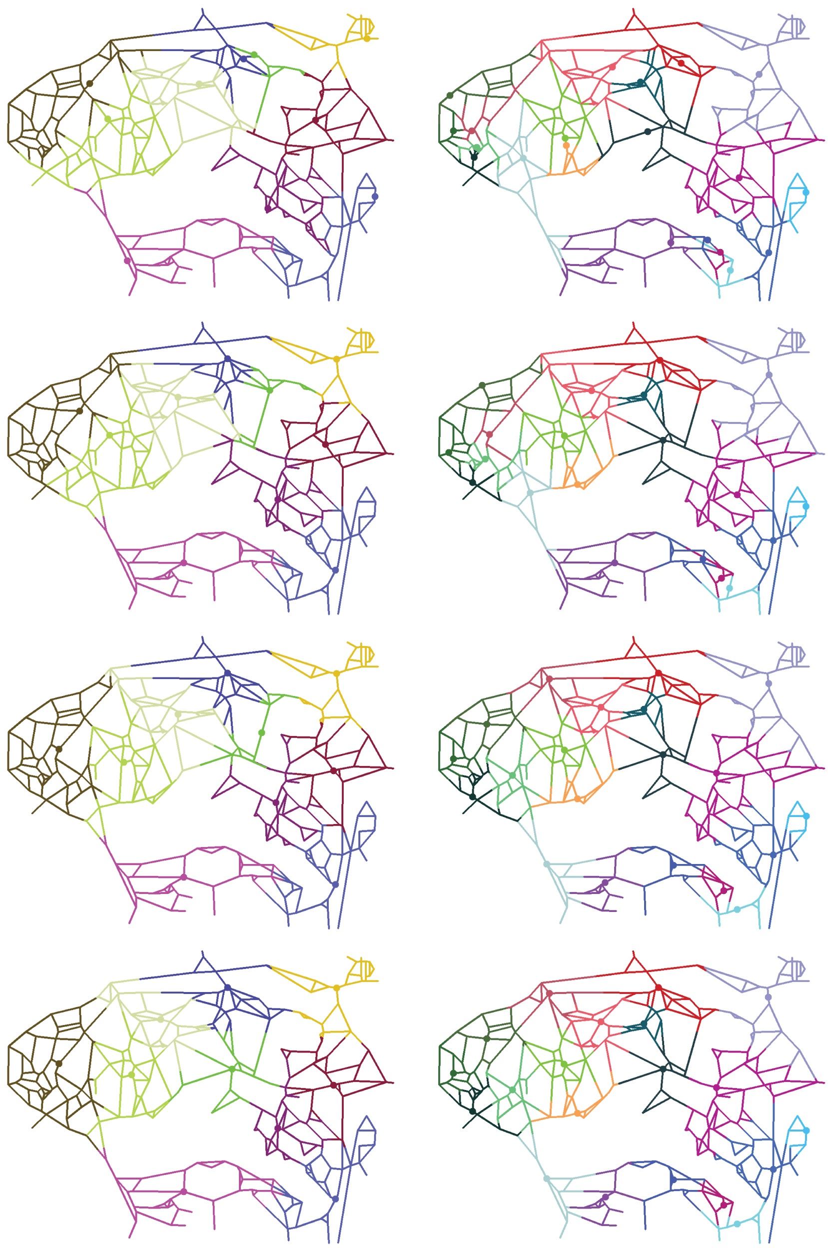

Fig. 5.4: Results of placing 10 (left) and 20 (right) trash cans, respectively. Top to bottom: random initial seeds, after one iteration, intermediate results(after 4 or 14 iterations) and final convergence results (after 7 or 27 iterations).

Our approach can also optimize the distribution of depot locations for express or logistics companies. Figure 5.8 shows the national expressway network. We took the national roads and built a corresponding network with 287 vertices and 508 edges;

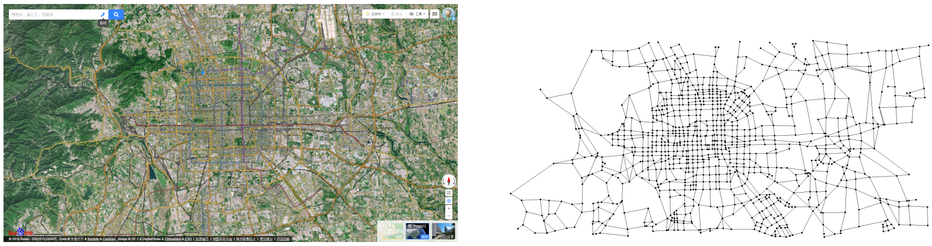

Fig. 5.5: A traffic satellite map of Beijing city (left, Baidu map ©Baidu.com) and the main traffic roads within the Sixth Ring Road (right). Black points depict road intersections, and lines locate the network connections.

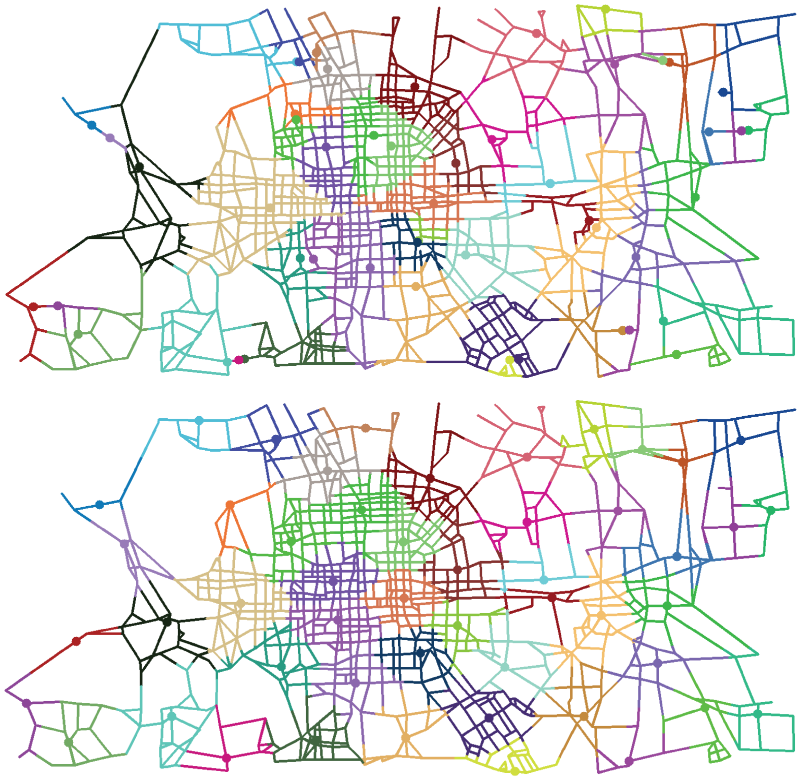

Fig. 5.6: Locations of charging stations at convergence after 28 iterations. Initial seeds (above) and converged results (below).

the weight of each edge is the length of the corresponding road. Express or logistics companies usually build depots beside the expressway. In order to make effective use of each depot, the average length of the road covered by each depot should be equal as far as possible. We optimize the locations of the depots by computing a centroidal

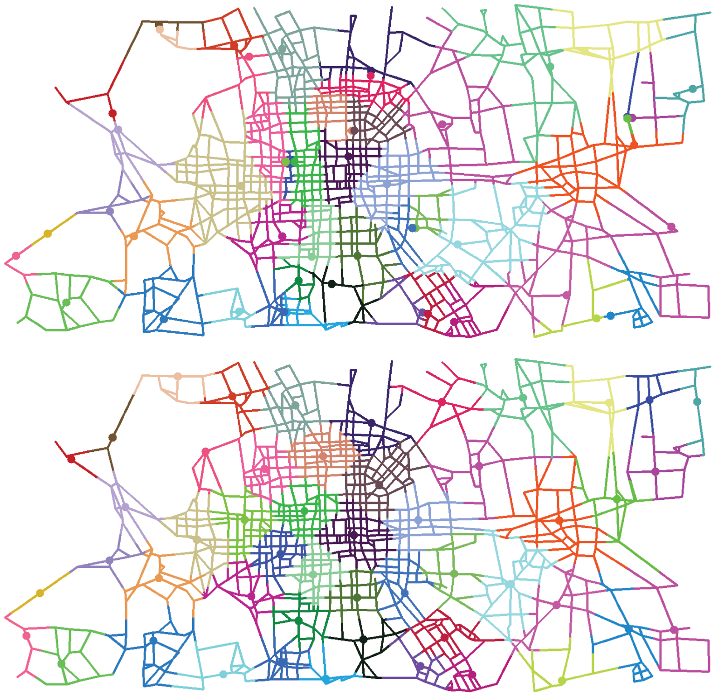

Fig. 5.7: Locations of charging stations with different initial seeds. Initial seeds (above) and converged results after 33 iterations (below).

Voronoi tessellation. Figure 5.9 shows a result for 35 depots on the national expressway network.

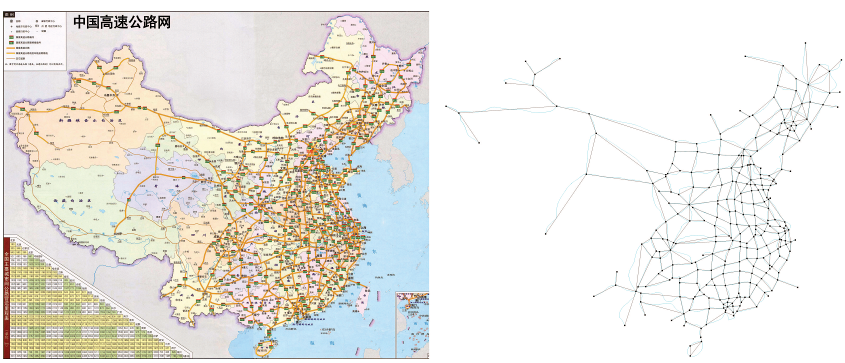

Fig. 5.8: The national expressway network and corresponding line network. Right: black points are vertices, the straight lines are edges of the line network, and the blue-green curves are the real highways.

5.3. Discussions

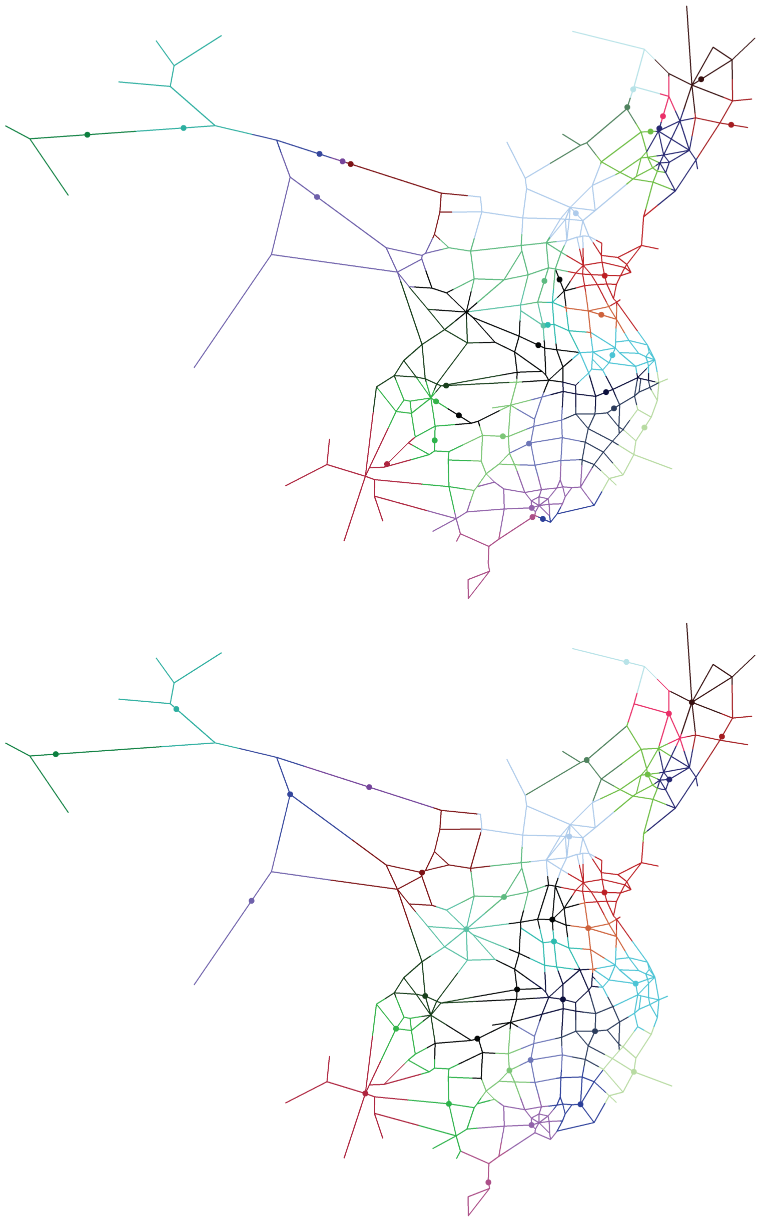

Fig. 5.9: Results of building 35 depots on the national expressway. The upper figure shows random seeds, and the figure below is the converged result after 38 iterations. Dots indicate the locations of the depots, and the corresponding color indicates the Voronoi cell of each depot.

For any given connected networks, the proposed algorithm will converge to a solution. However, this method does not necessarily guarantee the optimal solution(i.e.,

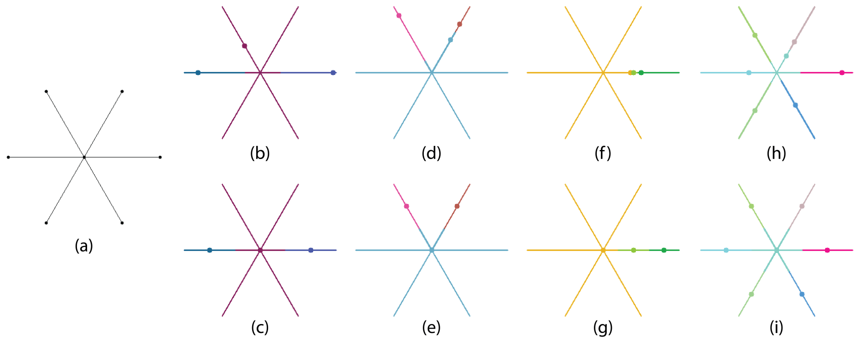

Fig. 5.10: Star-like networks with tree structure: Voronoi tessellations with 3 seed points (b-g) and 7 seed points (h-i). The dots represent the seed points, with the corresponding Voronoi Cells represented in consistent color.

the average distance from each point to the seed point is equal), due to the structural characteristics of the networks. We now illustrate various networks with their respective structures.

5.3.1. Line networks with a tree structure

When the network has a tree structure, all the sub-networks corresponding to the Voronoi tessellation of the network also have tree structures. When the centroid of the tree happens to be the seed point of the partition, a (locally) optimal solution will be obtained. Figure 5.10(a) shows a "star" like line network with six edges, where each edge weight is equal. A Voronoi tessellation is obtained by randomly spreading three seed points(see Figure 5.10(b)), and a locally optimal solution is reached after 7 iterations(see Figure 5.10(c)). Figure 5.10(d) and (f) show two other initializations, converging to the Voronoi tessellations in Figure 5.10(e) and (g) after 8 iterations and 16 iterations respectively. If the average distance from the points in each cell to the corresponding seed point is required to be as uniform as possible, Figure 5.10(c) and (e) are better than the tessellation shown in Figure 5.10(g). However, in this example, a tessellation with 3 cells cannot be found so that the average distance from the point to the seed point is equal for each cell. If the number of seed points is set to 7, a tessellation with equal average distances can be obtained for a specific choice of initial seeds, as shown in Figure 5.10(h) and (i).

Therefore, for a general tree network, a centroidal Voronoi tessellation whose average distances are equal in each Voronoi cell does not necessarily exist, but depends on the number and locations of the seeds.

5.3.2. Line networks with a loop

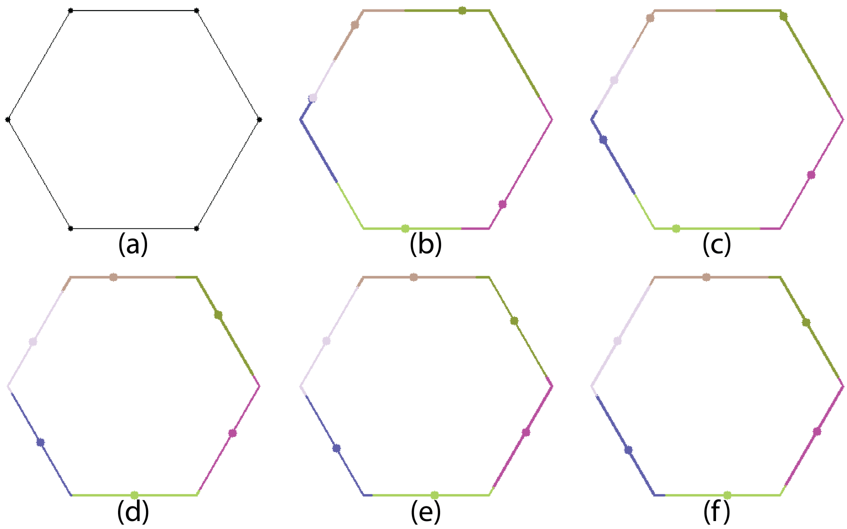

Fig. 5.11: Centroidal Voronoi tessellation of a hexagonal network with 6 seed points. (a) the network; (b) 6 random seed points and corresponding Voronoi tessellation(c)–(e) tessellations after 1, 7 and 12 iterations(f) the converged tessellation after 32 iterations.

We next consider the networks with a loop. Figure 5.11(a) shows a hexagonal network in which the weight of each edge is equal. We initialize six seed points randomly on the network and obtain the initial Voronoi partition in Figure 5.11(b). After 32 iterations, the tessellation converges to a final result shown in Figure 5.11(f). For a network with a single loop, the centroidal Voronoi tessellation on the loop can be transformed into a one-dimensional Voronoi problem. Although the centroidal Voronoi tessellation on networks with a loop varies with the initial seed points, the mean distances from the points in each cell to their respective seed points are equal.

Obviously, for a network with a single loop, a centroidal Voronoi tessellation exists with equal mean distances from points in each cell to the corresponding seed point, but this tessellation is not unique.

6. Conclusion

This paper has studied a special discrete optimal mass transportation problem on the line networks. Previous researches on discrete optimal mass transportation problem usually require the space to have the property of convexity. This is not the case in real scenarios where the space is constrained on curves. We, therefore, propose the concept of Voronoi diagrams on undirected connected line networks and develop an algorithm for calculating such Voronoi diagrams based on the shortest path distance. Considering practical applications, we extend the centroidal Voronoi tessellation of Euclidean space to line networks so that the average distance is as uniform as possible between the points in each Voronoi cell and the corresponding seed points. To achieve this objective, we have computed the centroidal Voronoi partition of line networks by using an iterative method. The method is proved effective in typical line network examples and practical applications.

The increasing online taxi booking service poses a new problem, namely, how to match drivers and passengers in a city, given the density distribution of the drivers and the probability distribution of the passengers, to meet the service demands and to minimize the total cost for drivers picking up passengers. This problem is associated with the method proposed that deserves further further research.

REFERENCES

[1] L. Kantorovich, *On a problem of Monge*, Uspekhi Mat. Nauk, 3, 225–226, 1948.

[2] Y. Brenier, *Polar Factorization and Monotone Rearrangement of Vector-Valued Functions*, Communications on Pure and Applied Mathematics, 1991, 44(4), 375–417, 1991.

[3] Xianfeng Gu, Feng Luo, Jian Sun and Shingtung Yau, *Variational principles for minkowski type problems, discrete optimal transport, and discrete Monge-Ampère equations*, Asian Journal of Mathematics, 3(20), 383-C398, 2016.

[4] Franz Aurenhammer, *Power diagrams: properties, algorithms and applications*, SIAM Journal of Computing, 16(1), 78–96, 1987.

[5] Franz Aurenhammer, *Voronoi diagramsa survey of a fundamental geometric data structure*, ACM Computing Surveys, 23(3), 345–405, 1991.

[6] Qiang Du, Vance Faber and Max Gunzburger, *Centroidal Voronoi Tessellations: Applications and Algorithms*, SIAM Review, 41(4), 637C-676, 1999.

[7] Yang Liu, Wenping Wang, Bruno Lévy, Feng Sun, Dong-Ming Yan,Lin Lu and Chenglei Yang, *On centroidal voronoi tessellation - energy smoothness and fast computation*, ACM Transactions on Graphics, 28(2),101:1–101:17, 2009.

[8] Guodong Rong, Yang Liu, Wenping Wang and Xiaohu Guo, *GPU-Assisted Computation of Centroidal Voronoi Tessellation*, IEEE Transactions on Visualization and Computer Graphics, 17(3), 345–56, 2001.

[9] Xiaoning Wang, Xiang Ying, Yong-Jin Liu, Shi-Qing Xin, Wenping Wang, Xianfeng Gu, Wolfgang Mueller-Wittig and Ying He, *Intrinsic computation of centroidal Voronoi tessellation (CVT) on meshes*, Computer-Aided Design, 58(1), 51-C61, 2015.

[10] Stuart P. Lloyd, *Least squares quantization in PCM*, IEEE Transactions on Information Theory, 28(2), 129–137, 1982.

[11] John C. Kieffer, *Uniqueness of locally optimal quantizer for log-concave density and convex error weighting function*, IEEE Transactions on Information Theory, 29(1), 42–47, 1983.

[12] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms (Third Edition)*, The MIT Press, Cambridge, Massachusetts and London, England.