Title	**Global Optimization of Centroidal Voronoi Tessellation with Monte Carlo Approach**
Author(s)	**Lu, L; Sun, F; Pan, H; Wang, WP**
Citation	**IEEE Transactions on Visualization and Computer Graphics, 2012**
Issued Date	**2012**
URL	**http://hdl.handle.net/10722/165861**
Rights	**Creative Commons: Attribution 3.0 Hong Kong License**

Global Optimization of Centroidal Voronoi Tessellation with Monte Carlo Approach

Lin Lu, Feng Sun, Hao Pan, and Wenping Wang

Abstract—Centroidal Voronoi Tessellation (CVT) is a widely used geometric structure in applications including mesh generation, vector quantization and image processing. Global optimization of the CVT function is important in these applications. With numerical evidences, we show that the CVT function is highly nonconvex and has many local minima and therefore the global optimization of the CVT function is nontrivial. We apply the method of *Monte Carlo with Minimization* (MCM) to optimizing the CVT function globally and demonstrate its efficacy in producing much improved results compared with two other global optimization methods.

Index Terms—Centroidal Voronoi tessellation, global optimization, Monte Carlo with minimization

✦

1 INTRODUCTION

A Centroidal Voronoi Tessellation (CVT) is a special Voronoi tessellation of a domain in which every site coincides with the centroid of its Voronoi cell. Equivalently, a CVT is characterized by a critical point (i.e., a gradient vanishing point) of a certain CVT energy function, which we will discuss in detail shortly. Several methods, including Lloyd's method [1] and MacQueen's method [2] have been proposed for CVT computation. Liu et al. [3] prove that the CVT function is C^2 on a convex domain and present a quasi-Newton method to achieve significant acceleration over previous methods by one order of magnitude. It is noted, however, that the CVT function has many local minimizers. Most existing methods for CVT computation focus on fast computation of local minimizers. The problem of how to compute improved local minimizers via global minimization of the CVT function has not been investigated.

Global optimization of the CVT function can benefit many applications, including mesh generation, scientific computing, and image processing [4], [5]. We now use a simple example to show its application in mesh generation. Fig. 1 shows an example in mesh generation using the CVT. Starting from some random initialization, a locally minimal CVT (Fig. 1a) is generated by the quasi-Newton method in [3], and Fig. 1c shows its dual triangle mesh. With the global optimization scheme proposed in the present paper, an improved local minimizer (Fig. 1b) and its dual triangle mesh (Fig. 1d) are generated. Fig. 1e compares histograms of energies of all cells in the two CVTs in Figs. 1a and 1b; Fig. 1f compares histograms of the smallest angles of the triangles of the two corresponding meshes in Figs. 1c and 1d.

From this simple example, we can see that our global optimization method improves the CVT result greatly by eliminating most interior nonhexagonal cells, leading to a better triangle mesh with more regular vertices. The smallest angle in each triangle of the mesh generated by our method is closer to 60 degrees, indicating that the triangles have become more regular. According to the celebrated Gersho's conjecture [6], energy values of all cells in a globally optimal CVT are asymptotically equal. From Fig. 1e, we see that the variance of energy values of cells in our CVT result is much smaller than that of the local minimizer in Fig. 1a, an evidence that our result is closer to a globally optimal CVT.

The contribution of this paper is the development of a global optimization algorithm for minimizing the CVT function to obtain much improved locally optimal CVTs as compared with previous methods. As we will show by numerical evidences, the CVT function is highly nonconvex and hence it is quite challenging to find a global minimizer of the CVT function. Due to the nonconvexity of the CVT function and its large number of local minima, traditional global minimization approaches, such as direct Monte Carlo sampling approach, cannot be applied to the CVT minimization directly. The term "Monte Carlo with Minimization (MCM)" is used by Li and Scheraga [7] for their method for solving the protein folding problem. We adapt their algorithm for CVT computation and so also follow this name. The difference between this method and a conventional Monte Carlo method is that only local minimizer are considered as sampling points in the former, while any point can be used in the latter. The major issue in adapting the MCM approach and developing a practical algorithm for global CVT computation is the characterizing of several key parameters in the MCM approach. By investigating the specific properties of the CVT function, we have conducted a systematic study on how to choose these parameters to develop an efficient global optimization method for CVT computation.

2 LITERATURE REVIEW

2.1 Centroidal Voronoi Tessellation

Given a compact domain Ω in $\mathbb{R}^N$ and a set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ of n sites in Ω, a *Voronoi tessellation* of Ω is defined to be

- *L. Lu is with the School of Computer Science and Technology, Shandong University, Software College Campus, 1500 Shunhua Road, Jinan, Shandong 250101, P.R. China. E-mail: llu@sdu.edu.cn.*
- *F. Sun, H. Pan, and W. Wang are with the Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: {fsun, hpan, wenping}@cs.hku.hk.*

(a)   (b)

(c)   (d)

(e)   (f)
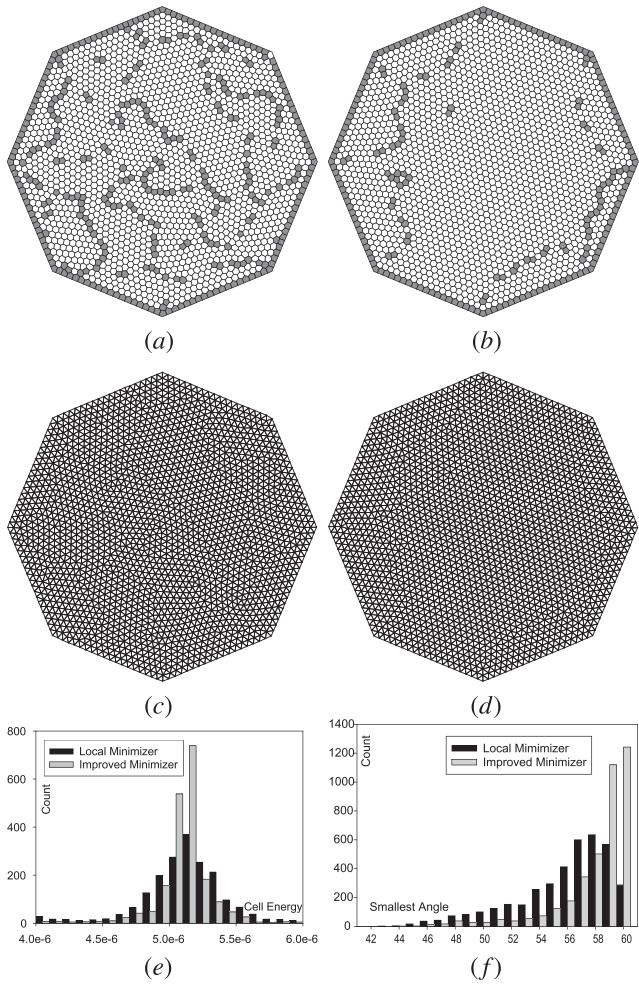
Fig. 1. Using the CVT to mesh a regular octagon domain with 2,000 sites. (Nonhexagonal cells highlighted in gray). (a) A locally minimal CVT and (c) the dual triangle mesh. (b) and (d) Show an improved local minimizer by our global optimization algorithm and its dual triangle mesh. The cell energy distributions of the two CVT's are shown in (e), and distributions of the smallest angles of the triangles of the two meshes are shown in (f). There are 1,849 interior vertices in both triangle meshes; among these, the mesh in (d) has 1,767 regular vertices (i.e., having valence six) and the mesh in (c) has only 1,631 regular vertices.

the collection of *Voronoi cells* $\{\Omega_i\}_{i=1}^n$ of the sites $\{\mathbf{x}_i\}_{i=1}^n$, where

$$\Omega_i = \{\mathbf{x} \in \Omega | \|\mathbf{x} - \mathbf{x}_i\| \le \|\mathbf{x} - \mathbf{x}_j\|, \forall j \ne i\},$$

and $\|\cdot\|$ denotes the Euclidean norm.

A Voronoi tessellation is called a *centroidal Voronoi tessellation* [4] if $\mathbf{x}_i = \mathbf{c}_i, i = 1, \dots, n$, where

$$\mathbf{c}_i = \int_{\mathbf{x}\in\Omega_i} \rho(\mathbf{x})\mathbf{x}\mathrm{d}\sigma \Big/ \int_{\mathbf{x}\in\Omega_i} \rho(\mathbf{x})\mathrm{d}\sigma$$

is the centroid of the Voronoi cell Ω_i of $\mathbf{x}_i$ and $\rho(\mathbf{x})$ is the density function defined over the domain Ω.

Equivalently, a CVT is also defined as a critical point of the following CVT function [4]:

$$F(\mathbf{X}) = \sum_{i=1}^n \int_{\mathbf{x}\in\Omega_i} \rho(\mathbf{x})\|\mathbf{x} - \mathbf{x}_i\|^2 \mathrm{d}\sigma.$$

In the following, $\int_{\mathbf{x}\in\Omega_i} \rho(\mathbf{x})\|\mathbf{x} - \mathbf{x}_i\|^2 \mathrm{d}\sigma$ is called the energy value of the Voronoi cell of the site $\mathbf{x}_i$.

We denote the set of sites $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ as an ordered set $\mathbf{X} = (\mathbf{x}_i)_{i=1}^n$, which is an n-dimensional vector consisting of the variables of the CVT function $F(\mathbf{X})$.

A critical point of the CVT function $F(\mathbf{X})$ may be a saddle point corresponding to an indefinite Hessian of $F(\mathbf{X})$ [3]. Saddle points are unstable and often undesirable in practice. In this paper, we focus only on the CVTs corresponding to local minimizers.

The computation of a local minimizer of the CVT function has been well studied. A thorough review of existing methods is out of the scope of this paper. We refer to [5, Section 3] for a survey on computing local minimizers. Here, we need to mention the following local-search algorithm invoked in our global optimization scheme. Liu et al. [3] prove that the CVT function $F(\mathbf{X})$ is C^2 in any convex domain with a C^2 smooth density function and is still C^2 when the domain is nonconvex in most cases except for some cases rarely encountered in practice and present an efficient quasi-Newton method for accelerating CVT computations. In practice, such local minimizers are often far from being globally optimal, as will be shown in our experiments later.

2.2 Globally Optimal CVT

The computation of a global minimizer of the CVT function is still outstanding in literature. However, there are some theoretical results on the global minimizers of the CVT function. Gersho [6] conjectures that, asymptotically, all Voronoi cells assume the same shape in an optimal CVT. Tóth [8] proves that, asymptotically, the Voronoi cells of an optimal 2D CVT in a convex domain assume a uniform regular hexagonal shape, and thus proves the conjecture in two dimensions. Du and Wang [9] present numerical evidence that supports the conjecture in three dimensions. According to the widely accepted Gersho's conjecture, all the sites $\mathbf{x}_i$ in $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ have asymptotically equal energy values $\int_{\mathbf{x}\in\Omega_i} \rho(\mathbf{x})\|\mathbf{x} - \mathbf{x}_i\|^2 \mathrm{d}\sigma$ in a globally optimal CVT; this is called the energy equidistribution principle in [4] and [10].

2.3 Global Optimization

A nonconvex function may have more than one local minimizer and algorithms based on local search converge only to a local minimizer. Finding the global minimizer or a good local minimizer within a reasonable amount of time is challenging. Existing global optimization algorithms can be classified into two categories: deterministic algorithms and stochastic algorithms. Deterministic algorithms are suitable when the characteristics of solutions are known. Then, the search space can be enumerated to find the global minimizer. Therefore, deterministic algorithms often become impractical when the size of the problem grows large, typically with several thousand variables [11], since the number of sample points in the search space is exponential in the number of variables. For a large-scale global optimization problem, stochastic algorithms are more efficient than deterministic algorithms.

Stochastic optimization algorithms have attracted extensive research in recent decades. Popular algorithms include Monte Carlo methods [12], Particle Swarm Optimization
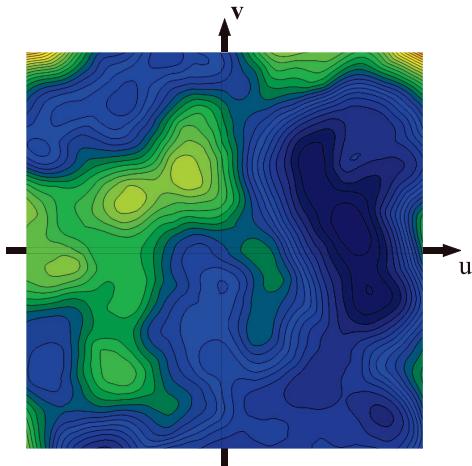
Fig. 2. An illustration of the neighboring landscape of $\mathbf{X}_0$.

(PSO) [13], evolutionary algorithms [14], the Monte Carlo with minimization method [7], [15], [16] and Simulated Annealing (SA) [17], [18]. Interested readers are referred to [19], [20]. As we will see the MCM method can be adapted for global optimization of a *continuous* nonconvex objective function with many local minima. In this paper, we apply the MCM approach to optimizing the CVT function.

3   PROPERTIES OF CVT FUNCTION

3.1   Nonconvexity of CVT Function

Before presenting our global optimization method, we shall show by numerical evidences that the CVT function is highly nonconvex and has many local minima. Consider a 2D example with $n = 100$ sites in the square $[-1, 1]^2$ with constant density $\rho(\mathbf{x}) \equiv 1$. Since it is difficult to visualize the complexity of the graph of $F(\mathbf{X})$ defined in such high dimensions, we instead show a clipped sectional view of $F(\mathbf{X})$ in Fig. 2. Starting from a set of sites corresponding to $\mathbf{X}_0$, we choose two linearly independent vectors $\mathbf{V}_0$ and $\mathbf{V}_1$ in $\mathbb{R}^{2n}$ to form a 2D subspace in $\mathbb{R}^{2n}$ and plot the graph of $f(u, v) = F(\mathbf{X}_0 + u \times \mathbf{V}_0 + v \times \mathbf{V}_1), u, v \in [-1, 1]$, where $|\mathbf{V}_0|_2 = 4.2510$ and $|\mathbf{V}_1|_2 = 4.2757$. Fig. 2 shows the contour of this 2D section of $F(\mathbf{X})$ centered at the point $\mathbf{X}_0$ in $\mathbb{R}^{2n}$. Here, darker color indicates a lower function value. It is easy to see the extraordinary complexity of the landscape of $F(\mathbf{X})$ and that $F(\mathbf{X})$ is highly nonconvex, suggesting the difficulty in global optimization of the CVT function $F(\mathbf{X})$.

3.2   Number of Local Minimizers of CVT Function

Next, we demonstrate that the CVT function can have a large number of local minima. Consider the square domain $\Omega : [-1, 1]^2$ with constant density $\rho(\mathbf{x}) \equiv 1$. Recalling that

the variables of the CVT function are considered as a vector $\mathbf{X}$ in $\mathbb{R}^{2n}$, a permutation of the sites corresponds to a permutation of the variables, therefore producing the same set of sites. Hence, the same CVT. We remove this redundancy by treating two vectors as the same if they are equal up to a permutation of the sites. Similarly, we remove the redundancy due to the symmetries of the square domain. In implementation, we regard two point sets as the same if their Hausdorff distance is less than 1e-12.

We define a *trial* to be the process of running a local search scheme, which is the L-BFGS method in the present case, to compute a local minimum from a randomly generated initial set of n sites in the square domain. With different numbers n of sites, we perform 5,000 trials and record the number of distinct local minimizers resulting from these 5,000 trials. Then, we do the same with 10,000 trials. The results are listed in Table 1. We see that when the number of sites is less than 20, the number of distinct minimizers observed is rather stable when the number of trials changes from 5,000 to 10,000, suggesting that they are nearly the set of all possible local minimizers. When the number of sites grows up to 100, almost all the 10,000 trials lead to distinct local minimizers. This suggests there are a large number of distinct local minimizers of the CVT function $F(\mathbf{X})$ in these cases.

4   GLOBAL ALGORITHM FOR CVT COMPUTATION

We shall first briefly introduce the Monte Carlo with minimization method [7], [15], explain why it is suitable for solving the CVT optimization problem, and discuss how to apply it to global optimization of the CVT function $F(\mathbf{X})$.

Global minimization of a nonconvex objective function with many local minima is a difficult problem. Applying conventional global optimization methods directly does not work well, since these methods often deal with discrete optimization problems. Given a continuous function, randomly sampling often finds only a point which is not a local minimizer, thus rendering these methods inefficient. Li and Scheraga [7] devise the MCM approach that applies the random sampling idea to global optimization and guarantees only local minimizers are considered. As a global optimization approach, MCM overcomes the inefficiency of the conventional Monte Carlo method, and has successfully been applied to solving various optimization problems in computational and applied chemistry, computational biology, and computational physics.

The main idea of the MCM method is to transform a continuous optimization problem to a discrete one by considering all local minimizers of the objective function as the feasible set. In each iteration, the method may jump from the current local minimizer to a new local minimizer.

TABLE 1
The Number of Observed Distinct Local Minimizers of the CVT Function

# of sites	10	11	12	13	14	15	16	17	18	19	20	30	40	50	100
# of different local minima observed in 5,000 trials	2	5	9	6	5	4	4	10	15	30	32	188	1,154	3,096	4,921
# of different local minima observed in 10,000 trials	2	5	9	6	5	4	4	10	15	30	33	226	1,584	4,903	9,742

If the new local minimizer is better than the current minimizer, then we accept the new minimizer as the current minimizer. Otherwise we accept it with a pre-defined probability. As the method runs to the converging stage, the probability drops gradually to zero and finally the best minimizer found so far is accepted as the final result.

The framework of the MCM method for a continuous optimization problem is as follows:

MCM Algorithm to minimize $F(\mathbf{X})$
1) Initialization—T_0, K, $\mathbf{X} \leftarrow \mathbf{X}_0$, $\overline{\mathbf{X}} \leftarrow \mathbf{X}_0$, $k \leftarrow 0$.
2) While the termination condition is not met, i.e., $k < K$:
   - Perturb $\mathbf{X}$ and minimize to $\mathbf{X}^*$;
   - $\Delta F = F(\mathbf{X}^*) - F(\mathbf{X})$. If $\Delta F < 0$, then $\mathbf{X} \leftarrow \mathbf{X}^*$; else if $\exp(-\Delta F/T_k) > random[0, 1]$, then $\mathbf{X} \leftarrow \mathbf{X}^*$;
   - If $F(\mathbf{X}^*) < F(\overline{\mathbf{X}})$, $\overline{\mathbf{X}} \leftarrow \mathbf{X}^*$;
   - Decrease T_k;
   - $k \leftarrow k + 1$.
3) Return $\overline{\mathbf{X}}$.

In this framework, each iteration includes three steps: Monte Carlo sampling, energy minimization, and examination by the Metropolis criterion. We call each iteration an *update* in the following. The probability of accepting a minimizer with higher energy value is $P = e^{-\frac{\Delta F}{T_k}}$, where ΔF is the increment of the energy function and T_k is a parameter, which is also called *system temperature* in the MCM method [15]. We use $\overline{\mathbf{X}}$ to store the best local minimizer found so far.

Besides the local search scheme invoked in the inner loop, there are several important parameters in the MCM method.

- The initial temperature T_0.
- The decrement function of T_k.
- The Perturbation magnitude.
- The termination condition, in terms of K, the total number of updates.

All these parameters are essential ones affecting the performance of the MCM method. We are going to adopt the MCM framework in global optimization of the CVT function $F(\mathbf{X})$ and show how to choose these parameters.

4.1 Local Search

We employ the L-BFGS method in [3], a quasi-Newton method, to compute a local minimizer $\mathbf{X}^*$. To find an accurate local minimum, one typically needs to set $\frac{\|g\|}{\|\mathbf{X}\|} \leq 10^{-12}$ as the termination condition, where $\|g\|$ is the norm of the gradient and $\|\mathbf{X}\|$ is the norm of variables. To improve efficiency, we may just compute sufficiently good local minimizers by setting a relatively larger threshold, say $\frac{\|g\|}{\|\mathbf{X}\|} \leq 10^{-7}$, as the termination condition of the L-BFGS method in each round. After the MCM process, the local minimizer will be minimized using the more accurate condition $\frac{\|g\|}{\|\mathbf{X}\|} \leq 10^{-12}$. We will show by experimental results, in Section 5.3, that this scheme of early termination of local search in the intermediate rounds saves much time while having little effect on the final minimization result.

4.2 Initial Temperature T_0

The initial temperature T_0 should be high enough so that MCM may successfully move to a neighboring minimizer even if the function value increased. If T_0 is too low, MCM
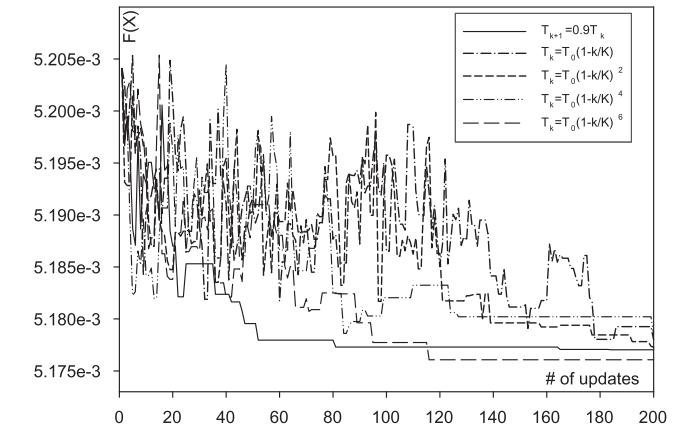

Fig. 3. $F(\mathbf{X})$ against the number of updates with different cooling functions.
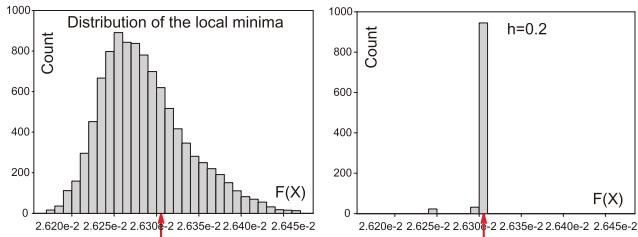
will not be able to jump out of the current basin, thus reducing to a local minimization approach. If T_0 is too high, MCM will accept all neighboring minimizers without selection, defeating the purpose of optimization. Meanwhile, since T_k is the denominator and ΔF is the numerator of the ratio in the exponent to control the probability, the choice of T_0 depends on the magnitude of function values and hence is problem-specific. We follow the idea in [21] to choose T_0: starting from an initial minimizer, we find some neighboring minimizers and compute the average increment of the energy function as $\overline{\Delta F}$ and then T_0 is set to $-\overline{\Delta F}/\ln(P_0)$, where $P_0 = 0.8$. In other words, T_0 is chosen in such a way that the probability of jumping out to a local minimizer of a higher value is $P_0 = 0.8$. Our experiments confirm that this value of T_0 works well.

4.3 Decrement Function of T_k
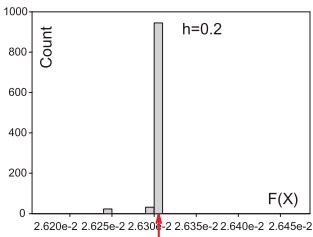
The temperature decrement function (also called *cooling function*) influences both the speed of the optimization and the *optimality* of the final result. If the temperature decreases too quickly, it leads to a fast search but the algorithm may easily get stuck at a poor local minimum. On the other hand, if the temperature decreases too slowly, the algorithm usually requires a long time to converge. There exists a wide range of selections for the decrement function [22]: one is the function $T_{k+1} = \alpha T_k$, where α is a constant, empirically between 0.8 and 0.9. Another is the variant of the above function $T_k = T_0(1 - k/K)^r$, where k is the number of updates so far and r is a constant, typically 1, 2, 4, or 6, and K is the total number of updates to be performed. These functions and parameters are tried in our experiments and we find that $T_k = T_0(1 - k/K)^6$ works best for both constant and nonconstant density function cases. Fig. 3 shows a result of 500 sites in a square domain $[-1, 1]^2$ with a constant density function $\rho(\mathbf{x}) \equiv 1$ with different settings of the cooling function. We use the function $T_k = T_0(1 - k/K)^6$ as the decrement function in all the experiments in this paper.

4.4 Perturbation Magnitude
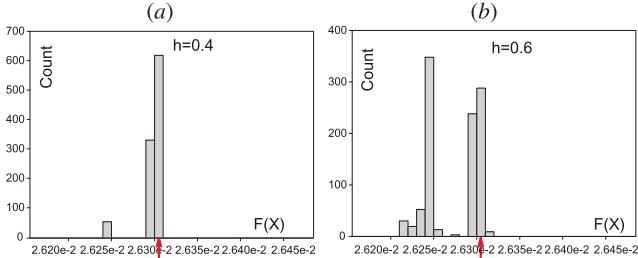
The magnitude of the perturbation is also critical to the effectiveness of the MCM algorithm. If the neighborhood size is too small, the algorithm may stop at a near and
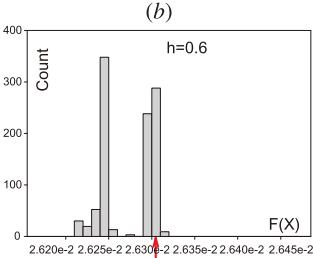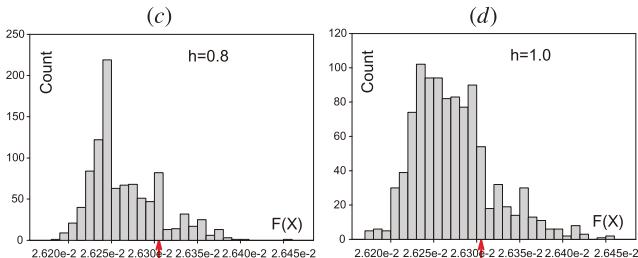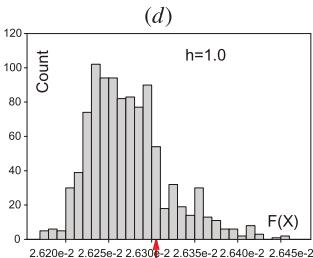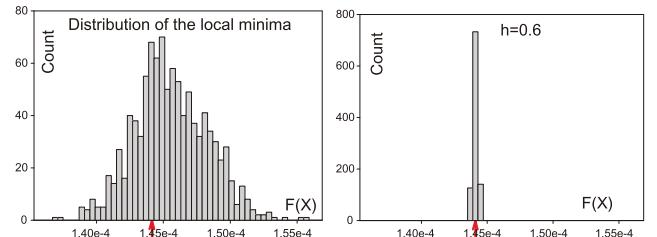
Fig. 4. An example of 100 sites in a square domain $[-1, 1]^2$ with density function $\rho(\mathbf{x}) \equiv 1$. (a) The distribution of 10,000 random local minima. (b)-(f) The distribution of generated neighbors from a local minimizer with function value $F(\mathbf{X}) = 2.6295e\text{-}2$ when $h = 0.2, 0.4, 0.6, 0.8, 1.0$, respectively. Nearby minimizers such generated are grouped into bins with the value 2.630e-2 marked by an arrow.


Fig. 5. An example of 2,000 sites in a 2D domain of a regular hexagon with the bounding box $[-2, 2] \times [-1.732, 1.732]$. The density function $\rho(\mathbf{x})$ is $e^{-20(x^2+y^2)} + 0.05 \sin^2(\pi x) \sin^2(\pi y)$. (a) The distribution of 1,000 random local minima. (b)-(f) The distribution of generated neighbors from the start local minimum $F(\mathbf{X}) = 1.4408e\text{-}4$ (marked with the arrows) with $h = 0.6, 0.8, 1.0, 1.2, 1.4$, respectively.

probably bad local minimizer. On the other hand, a too large neighborhood size would amount to restarting from a random initialization.
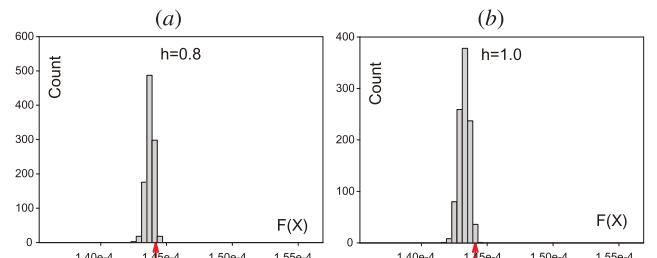
For our objective function, which is the CVT function $F(\mathbf{X})$, we use the size of Voronoi cells as a reference to determine the perturbation magnitude for each site. The figure on the right illustrates the Voronoi cell of a site in 2D. The black point is the site and gray points are Voronoi vertices. Let w_i denote the average of the edge length of the edges (i.e., dashed edges) incident to the site $\mathbf{x}_i$. Clearly, w_i depends on its corresponding Voronoi cell, and therefore is different for different sites. Let h be a constant to be determined for all sites. Then, we designate that $\mathbf{X} = (\mathbf{x}_i)_{i=1}^n$ is perturbed to $(\mathbf{x}_i + hw_i\mathbf{r}_i)_{i=1}^n$, where $\mathbf{r}_i$ is a random vector in $[-1, 1]^2$. Here, hw_i controls the magnitude of the perturbation, that is, the neighborhood size for basin hopping. As $\{w_i\}_{i=1}^n$ is computed from the Voronoi diagram, we will discuss how to choose the suitable h in Section 5.1.

4.5 Termination Condition (K)
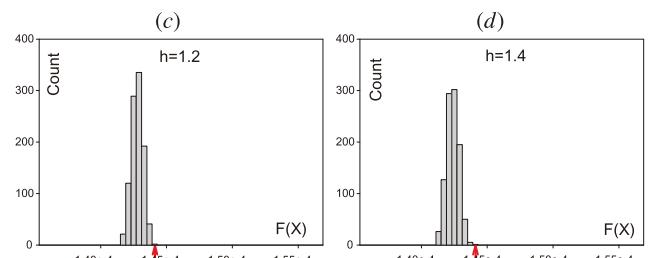
It is known that the MCM method converges to a global minimizer with probability 1 if given infinite amount of time [7]. However, in practice, we must stop running the algorithm within a reasonable amount of time. We use the total number of updates, K, as the termination condition. We will discuss how to choose a suitable K in Section 5.2.

5 EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we present experimental results to demonstrate the selection of some key parameters of the MCM method, validate the effectiveness of our algorithm and compare it with two other methods. All the experiments are run on a workstation with an Intel Xeon 3.33 GHz CPU.

5.1 Selection of Neighborhood Size

We first show how different neighborhood sizes affect the performance of the MCM method. In our first experiment, we use 100 sites in a square domain $[-1, 1]^2$ with a constant density function $\rho(\mathbf{x}) \equiv 1$. Let h change from 0.2 to 1.0 by an increment of 0.2 in each step. The result is shown in Fig. 4. The histogram of 10,000 local minima computed from random initializations is shown in (a), appearing in a normal distribution. For each h, we perform 1,000 tests with the same minimizer with the initial function value $F(\mathbf{X}) = 2.6295e\text{-}2$
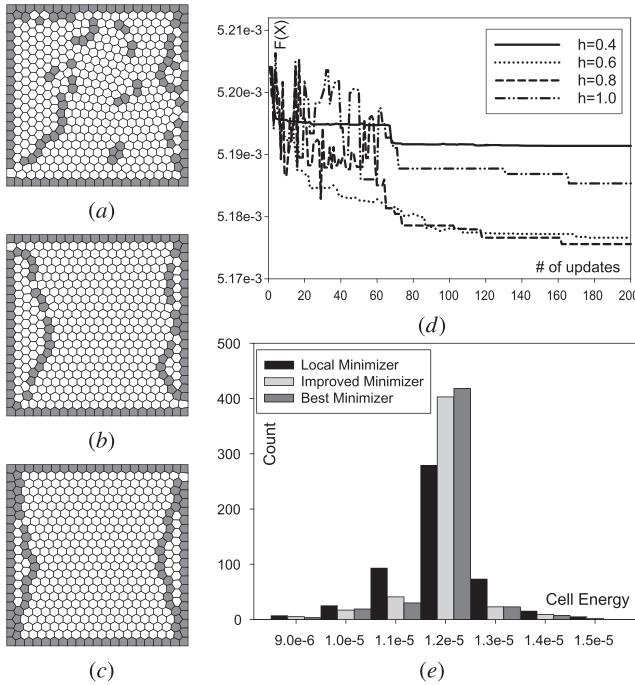
Fig. 6. The result of 500 sites in a square domain $[-1,1]^2$ with density function $\rho(\mathbf{x}) \equiv 1$. (a) The tessellation of the starting local minimum, $F(\mathbf{X}) = 5.2041e\text{-}3$. (b) A typical improved local minimum with $h = 0.8$, $F(\mathbf{X}) = 5.1771e\text{-}3$. (c) The best minimum observed with $h = 0.8$, $F(\mathbf{X}) = 5.1736e\text{-}3$. (d) $F(\mathbf{X})$ against the number of updates of different h values. (e) The cell energy distributions in (a), (b), and (c). Nonhexagonal cells are highlighted in gray.


Fig. 7. The result of 444 sites in a square domain with the bounding box $[-2,2] \times [-2,2]$ with density function $\rho(\mathbf{x}) = e^{-20(x^2+y^2)} + 0.05 \sin^2(\pi x) \sin^2(\pi y)$. (a) The tessellation of the starting local minimum, $F(\mathbf{X}) = 4.0323e\text{-}4$. (b) A typical improved local minimum with $h = 0.8$, $F(\mathbf{X}) = 3.3833e\text{-}4$. (c) The best minimum observed with $h = 0.8$, $F(\mathbf{X}) = 3.3774e\text{-}4$. (d) $F(\mathbf{X})$ against the number of updates of different h values. (e) The cell energy distributions in (a), (b), and (c).

and record the function values of nearby minimizers. These values are plotted in (b) to (f). When h is 0.2 or 0.4, nearby minimizers have very similar function values to that of the initial minimizer. When h is 0.6, these nearby minimizers lie in two intervals, with similar distribution. When h is 0.8, we have a larger range and most of the function values have smaller values than the initial minimizer. Finally, when h is 1.0, the range is nearly as large as that of the randomly generated local minima. In this context, we conclude that the best h is 0.8, because given one chance to perturb the current local minimizer to a both new and nearby minimizer, the probability is highest when h is 0.8. Our experiments also verify this conclusion.

When the density function is not constant, it is more difficult to find a universally acceptable h. Although $\{w_i\}_{i=1}^n$ is dependent on the Voronoi diagrams, which reflects the influence of the density function already, our experiments show that different density functions need different values of h to perturb the current minimizer to a new one. It is nontrivial to find the relations between the density functions and the magnitude of perturbation.

We show one example with nonconstant density function in Fig. 5: 2,000 sites in a regular hexagon domain with bounding box $[-2,2] \times [-1.732, 1.732]$. The density function $\rho(\mathbf{x})$ is $e^{-20(x^2+y^2)} + 0.05 \sin^2(\pi x) \sin^2(\pi y)$. The value of h increases from 0.6 to 1.4 by 0.2 in each step. The initialization is a local minimizer with the function value $F(\mathbf{X}) = 1.4408e\text{-}4$. (a) Shows the histogram of function values of 1,000 random minimizers. And (b) through (f) shows, respectively, the distribution of 1,000 tests with h from 0.6 to 1.4, increased by 0.2 each time. From the neighborhood distribution, we infer that setting h to a value

from 1.0 to 1.4 may all work well. Our experiments show the slight differences caused by different values and suggest 1.2 as the best value. Hence, we use $h = 1.2$ for this density function in the later examples.

Figs. 6, 7, and 8 show three examples with different density functions. In each figure, we show a typical local minimizer, a typical improved minimizer by our algorithm in one run and the best minimizer observed in 100 runs. Energy curves of different values of h are plotted in the subfigure (d). The subfigure (e) shows the comparison of (a), (b), and (c). From these figures, we see that a typical improved minimizer has similar distributions with the best minimizer observed in 100 runs, both of which are much better than a typical local minimizer, in terms of the equal distribution of cell energies. ($K = 200$ for all the three examples.)

5.2 Selection of Termination Condition

Then, we show the influence of the values of K, the termination condition, in some scenarios. As listed in Table 2, we test three examples by increasing K gradually from 100 to 3,000. The result is the average value of 100 runs of our MCM algorithm. It shows that a larger K leads to better results as expected and the results become stable after K reaches some certain number.

Due to the relatively small number of sites and the boundary effect, the Gersho's conjecture is not applicable to the above instances and the global minimizers are unknown to us, which makes it difficult to evaluate the gap between the improved minimizers obtained by some reasonable expense, i.e., K, and the true global minimizers. To circumvent this difficulty, we construct an example having hexagonal patterns with constant density function, which has the
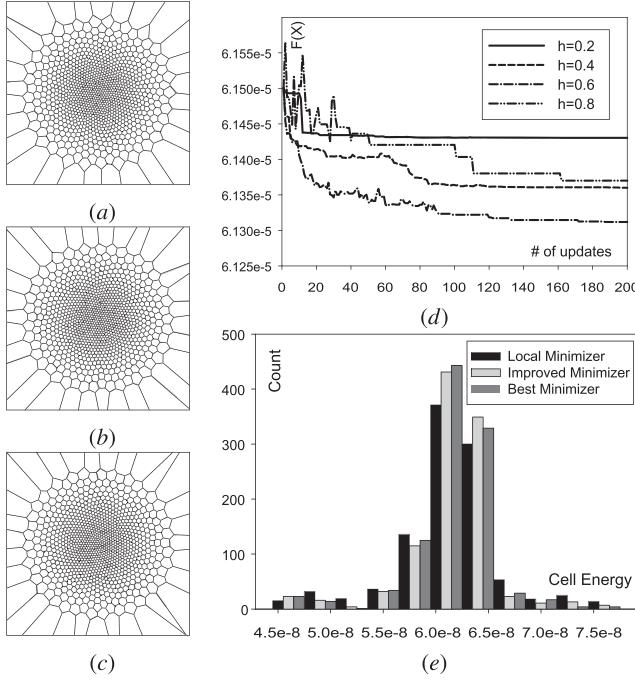
Fig. 8. The result of 1,024 sites in a square domain $[-1,1]^2$ with density function $\rho(\mathbf{x}) = e^{-10(x^2+y^2)}$. (a) The tessellation of the starting local minimum, $F(\mathbf{X}) = 6.1509e\text{-}5$. (b) A typical improved local minimum with $h = 0.6$, $F(\mathbf{X}) = 6.1317e\text{-}5$. (c) The best minimum observed with $h = 0.6$, $F(\mathbf{X}) = 6.1293e\text{-}4$. (d) $F(\mathbf{X})$ against the number of updates of different h values. (e) The cell energy distributions in (a), (b), and (c).

Fig. 9. Comparison between our improved local minimum and the global minimum. (a) A random local minimum with 400 sites in a 20×20 hexagonal pattern domain with the bounding box $[-17.754, 17.754] \times [-15.25, 15.25]$, $F(\mathbf{X}) = 4.3658e\text{+}2$. (b) The improved local minimum of (a), $F(\mathbf{X}) = 4.3301e\text{+}2$. (c) A random local minimum with 900 sites in a 30×30 hexagonal pattern domain with the bounding box $[-26.414, 26.414] \times [-22.75, 22.75]$, $F(\mathbf{X}) = 9.8210e\text{+}2$. (d) The improved local minimum of (c), $F(\mathbf{X}) = 9.7428e\text{+}2$. Nonhexagonal cells are highlighted in gray.
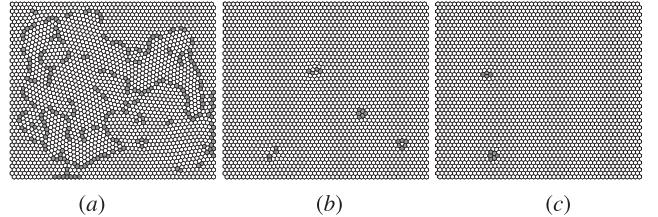
Fig. 10. Comparison between our improved local minimum and the global minimum. (a) A random local minimum with 2,500 sites in a 50×50 hexagonal pattern domain with the bounding box $[-43.734, 43.734] \times [-37.75, 37.75]$, $F(\mathbf{X}) = 2.7256e\text{+}3$. (b) The improved local minimum of (a) with $K = 200$, $F(\mathbf{X}) = 2.7080e\text{+}3$. (c) The improved local minimum of (a) with $K = 400$, $F(\mathbf{X}) = 2.7071e\text{+}3$. Nonhexagonal cells are highlighted in gray.

known global minimizer. As shown in Fig. 9, after applying our MCM algorithm on the local minimum (a) (resp. (c)), we always achieve the global minimum (b) (resp. (d)) with $K = 200$. However, when the number of sites becomes large, the global minimum is hard to reach with the same K. As shown in Fig. 10, given the local minimum (a) as the input, (b) is the result with $K = 200$, and (c) is the result with $K = 400$.

5.3 Local Search

Early termination when computing a local minimizer in the intermediate rounds of MCM saves much time while having little effect on the final result. We set $\frac{\|g\|}{\|\mathbf{X}\|} < 1e\text{-}7$ as the

stopping criterion of the L-BFGS method in the intermediate rounds. Now, we show how this setting does accelerate computation as shown by the experimental results in Table 3. Here, each result is the average value of 100 runs of the algorithm. Column F_{local} is the average energy value of the local minimizers computed from randomly initialized generators, with the time cost in column t_{local}. The value of h used is listed in column h. Columns F_{MCM200} and t_{MCM200} show the average energy value and time for our MCM algorithm to improve the local minima in F_{local}. Columns $F_{MCM200Fast}$ and $t_{MCM200Fast}$ list the results of the algorithm which finds only an approximate minimizer in each iteration (i.e., using the stopping criterion $\frac{\|g\|}{\|\mathbf{X}\|} < 1e\text{-}7$). We can see such

TABLE 2
Results of Some Examples with Different K of the MCM Method

n	h	$F_{K=100}$	$F_{K=200}$	$F_{K=400}$	$F_{K=800}$	$F_{K=1000}$	$F_{K=1500}$	$F_{K=2000}$	$F_{K=2500}$	$F_{K=3000}$
100	0.8	2.6182e-2	2.6178e-2	2.6175e-2	2.6173e-2	2.6172e-2	2.6172e-2	2.6171e-2	2.6171e-2	2.6171e-2
444	1.2	3.3850e-4	3.3833e-4	3.3824e-4	3.3814e-4	3.3811e-4	3.3809e-4	3.3805e-4	3.3804e-4	3.3804e-4
256	0.6	2.4169e-4	2.4160e-4	2.4156e-4	2.4153e-4	2.4151e-4	2.4148e-4	2.4147e-4	2.4145e-4	2.4145e-4

The examples on the first two rows are for the one in Figs. 4 and 7, respectively. The example on the third row has the same setting with the one in Fig. 8 except that the number of sites used here is 256.

TABLE 3
Results of Some 2D Examples

n	domain	$\rho(\mathbf{x})$	F_{local}	t_{local}(seconds)	h	F_{MCM200}	t_{MCM200}(seconds)	$F_{MCM200Fast}$	$t_{MCM200Fast}$(seconds)
100	square	1	2.6282e-2	0.20	0.8	2.6178e-2	33.27	2.6180e-2	19.27
500	square	1	5.2076e-3	1.14	0.8	5.1773e-3	180.81	5.1777e-3	76.48
2000	octagon	1	1.0365e-2	5.10	0.8	1.0321e-2	781.58	1.0322e-2	202.79
444	square	ρ_1	4.0242e-4	5.95	1.2	3.3833e-4	1123.42	3.3843e-4	280.86
2000	hexagon	ρ_1	1.4536e-4	63.29	1.2	1.2113e-4	9198.97	1.2147e-4	380.06
256	square	ρ_2	2.4242e-4	1.14	0.6	2.4160e-4	133.86	2.4162e-4	104.01
1024	square	ρ_2	6.1509e-5	8.89	0.6	6.1319e-5	939.59	6.1361e-5	139.25

$\rho_1(\mathbf{x}) = e^{-20(x^2+y^2)} + 0.05\sin^2(\pi x)\sin^2(\pi y)$ *and* $\rho_2(\mathbf{x}) = e^{-10(x^2+y^2)}$.
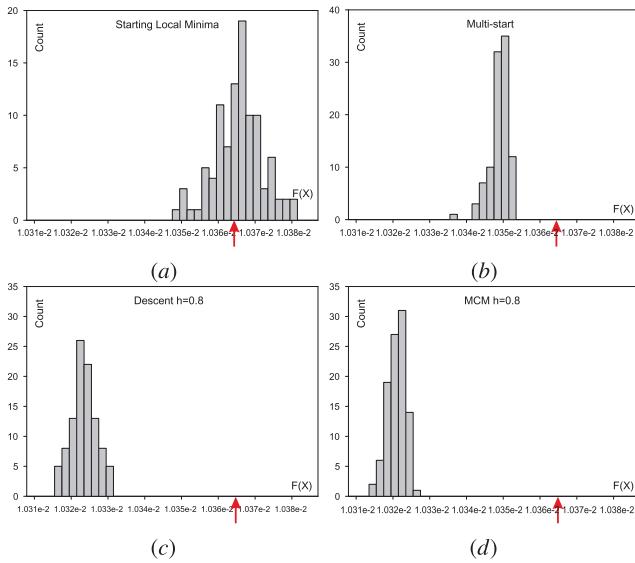
Fig. 11. Comparison of different methods. The example has 2,000 sites in an octagon domain as shown in Fig. 1. (a) The distribution of 100 local minimizers from randomly sampled generators. (b)-(d) The distribution of the results starting from the local minima $F(\mathbf{X}) = 1.0364e\text{-}2$ (marked with the arrows) with the multistart method, descent method, and our MCM method by $h = 0.8$ and the average energy value of three methods are 1.0348e-2, 1.0322e-2, and 1.0320e-2, respectively.

treatment saves a lot of time, especially when the number of sites is large, with only small sacrifice on the quality of the final result. ($K = 200$ for all the listed examples.)

5.4 Comparisons with Other Methods

We point out that the problem of customizing different global optimization techniques for CVT computation is wide open and the MCM method may not be the most effective one.

In this paper, we only compare our MCM algorithm with two other methods: the multistart method [23] and the descent method [24]. In a *local* phase of the multistart method, one starts from a random initialization and minimizes it to a local minimizer. In the *global* phase, the multistart method runs local phases K times, where K is a user-specified parameter and returns the best result of them. The descent method is similar to the MCM method, but with the difference that the descent method jumps to a new minimizer only if $F(\mathbf{X})$ decreases.

We choose three examples with different density functions and first randomly generate 100 local minima, respectively. Then, we carry out two types of experiments. Fig. 11 shows the results of 100 times of tests starting from one local minimizer for the three methods, respectively. Figs. 12 and 13 show the results of 100 times of tests starting from random local minimizer each time for the three methods, respectively. For consistency of comparison, we set K to 200 for all the three methods. The computational time of the three compared methods is nearly the same because the major computational expense is all local search, which follows the same procedure, i.e., L-BFGS. The difference is the MCM method needs to examine the local minimizer by the Metropolis criterion and compute the cooling function in each update that costs negligible time. It is not hard to see that our MCM method outperforms the
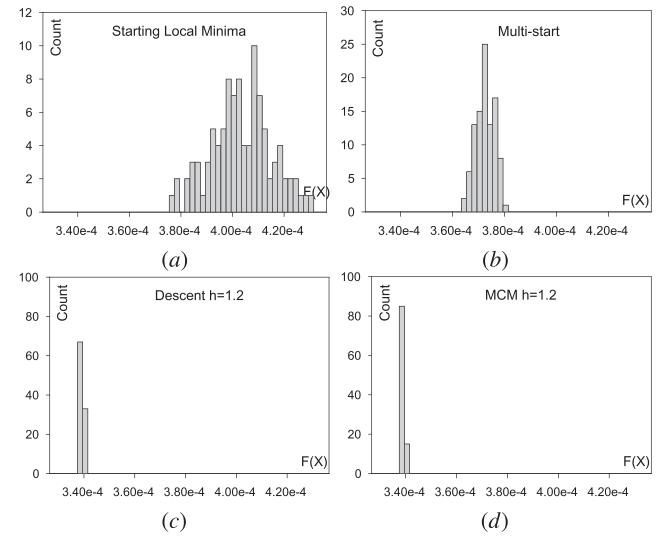
Fig. 12. Comparison of different methods. The example has 444 sites in a square domain as shown in Fig. 7. (a) The distribution of 100 local minimizers from randomly sampled generators. (b)-(d) The distribution of the results starting from a local minima in (a) with the multistart method, descent method, and our MCM method by $h = 1.2$ and the average energy value of three methods are 3.7168e-4, 3.3845e-4, and 3.3833e-4, respectively.

descent method, and our method and the descent one are both better than the multistart method.

5.5 Results in Mesh Generation and Optimization

We give an example of unconstrained mesh generation of a general 2D domain in Fig. 14. Since the boundary constraint is specific to 2D mesh generation [25], we do not use a specific strategy to handle the boundary and simply take the dual triangle mesh of the minimal CVT to evaluate our global algorithm. As the result shows, the variance of the energy values of cells in our CVT result is much less than
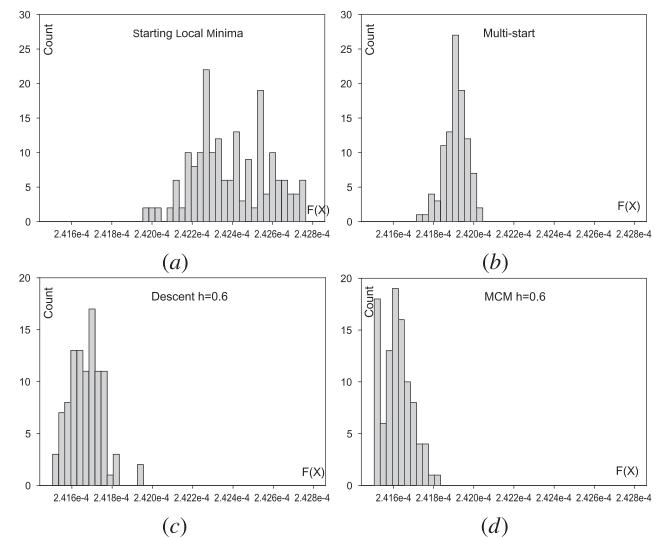
Fig. 13. Comparison of different methods. The example has 256 sites in a square domain $[-1, 1]^2$ with density function $\rho(\mathbf{x}) = e^{-10(x^2+y^2)}$. (a) The distribution of 100 local minimizers from randomly sampled generators. (b)-(d) The distribution of the results starting from the local minima in (a) with the multistart method, descent method, and our MCM method by $h = 0.6$ and the average energy value of three methods are 2.4190e-4, 2.4166e-4, and 2.4160e-4, respectively.

(a)

(b)

(c)
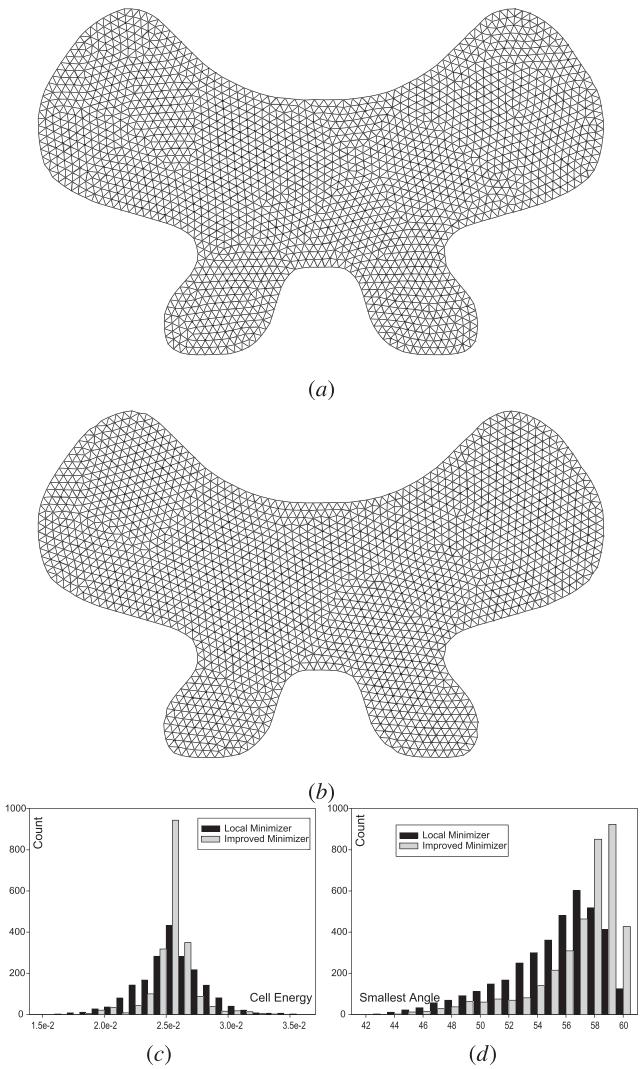
Fig. 14. Using the CVT to mesh a general 2D domain with 2,000 sites. (a) The dual triangle mesh of a local minimizer. (b) The dual triangle mesh of the improved local minimizer by our global optimization algorithm. The cell energy distributions of the two CVTs in (a) and (b) are shown in (c). The distributions of the smallest angle of the triangles of the two meshes are shown in (d). The mesh in (b) has 1,669 regular vertices and the mesh in (a) has 1,594 regular vertices. $K = 100$ and $h = 1.2$.

that of the local minimizer and the dual mesh quality is much better concerning the smallest angle of each triangle.

We also show some 3D remeshing results. The remeshing result of a given mesh is the dual triangle mesh extracted from a critical point of the *Constrained* CVT (CCVT) function [26]. We employ the fast and exact method proposed by Yan et al. [27] to compute the Restricted Voronoi Diagram (RVD) and L-BFGS method to minimize the CCVT function and then compare the local minimum with the improved one by our global optimization algorithm. The parameters in the MCM framework such as the initial temperature and the cooling function are chosen following the earlier discussions. Particularly, the parameters $\{w_i\}_{i=1}^n$ that indicate the perturbation magnitude are specified to be the square root of the area of the constrained Voronoi cell of each site. The parameter $\mathbf{r}_i$ is changed into a random vector in $[-1,1]^3$ accordingly. The parameter h is set to 0.8, which is empirically good. (Refer to Section 4.4.) Since the RVD computation in 3D is more time consuming

(a)

(b)



(c)   (d)
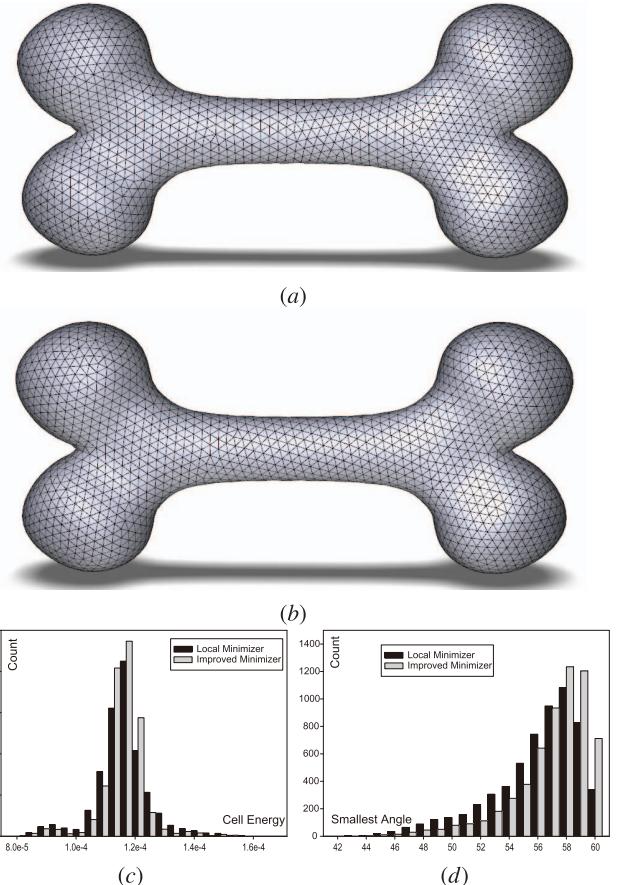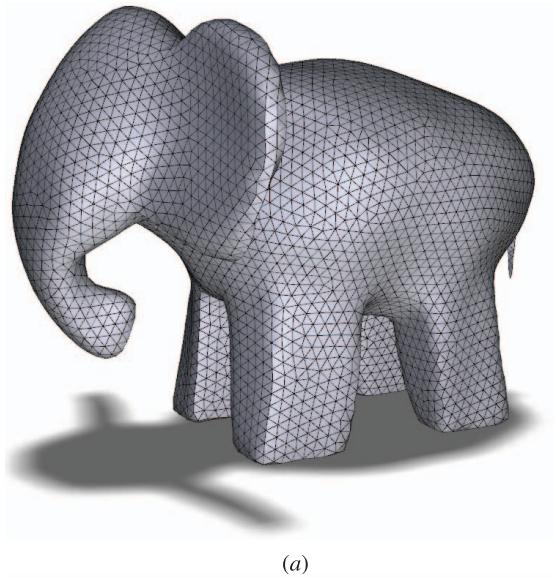
Fig. 15. Remeshing result of the bone model with 3,000 sites. (a) The dual triangle mesh of a local minimizer. (b) The dual triangle mesh of the improved local minimizer by our global optimization algorithm. The cell energy distributions of the two CCVTs in (a) and (b) are shown in (c). The distributions of the smallest angle of the triangles of the two meshes are shown in (d). The mesh in (b) has 2,806 regular vertices and the mesh in (a) has 2,664 regular vertices. $h = 0.8$.

than 2D cases, we set K to 100 in the experiments to get results in a reasonable amount of time. Empirically, our global method needs about 30 times of the time used for a single run to optimize the sites to a local minimizer. Figs. 15 and 16 show two examples with constant density. Fig. 17 shows an example with the density function $\rho(\mathbf{x}) = \mathcal{K}(\mathbf{x})^3$, where $\mathcal{K}(\mathbf{x})$ is the Gaussian curvature at $\mathbf{x}$. The subfigures (c) and (d) give the comparison of the cell energy of the Voronoi tessellations and the smallest angle of the remeshing result from a local minimizer and the improved minimizer by our algorithm, respectively. We can see that our global optimization achieves better results.
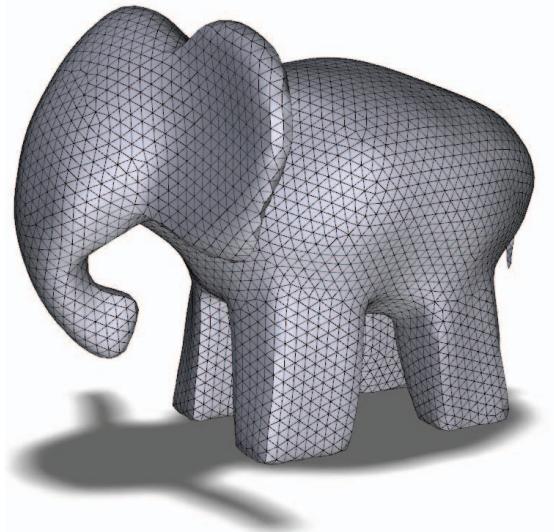
6   CONCLUSION

We have proposed a global optimization framework based on the MCM method for the CVT function, which is highly nonlinear and nonconvex. The framework starts from a local minimizer of the CVT function and improves it greatly. Experiments show our algorithm is effective.

Further study is needed to improve the selection of perturbation neighborhood size in the case of nonconstant density functions. Investigating more effective global optimization methods for computing CVT and making a comprehensive comparison is also regarded as the future work.

(a)

(b)
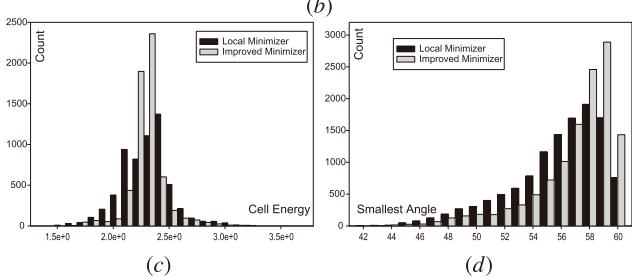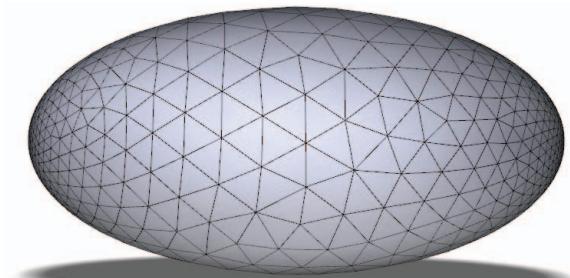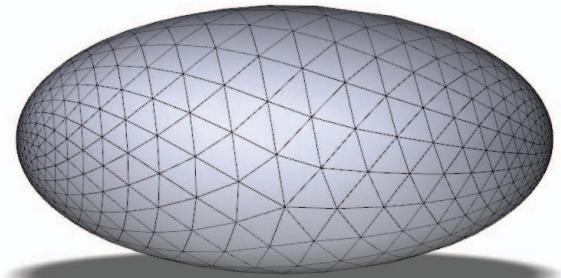
Fig. 16. Remeshing result of the elephant model with 6,000 sites. (a) The dual triangle mesh of a local minimizer. (b) The dual triangle mesh of the improved local minimizer by our global optimization algorithm. The cell energy distributions of the two CCVTs in (a) and (b) are shown in (c). The distributions of the smallest angle of the triangles of the two meshes are shown in (d). The mesh in (b) has 5,592 regular vertices and the mesh in (a) has 5,359 regular vertices. $h = 0.8$. Notice the elephant legs.
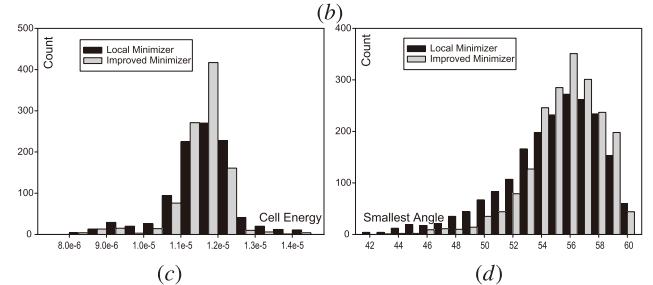
Fig. 17. Remeshing result of the ellipsoid model with the density function $\rho(\mathbf{x}) = \mathcal{K}(\mathbf{x})^3$ of 1,000 sites. (a) The dual triangle mesh of a local minimizer. (b) The dual triangle mesh of the improved local minimizer by our global optimization algorithm. The cell energy distributions of the two CCVTs in (a) and (b) are shown in (c). The distributions of the smallest angle of the triangles of the two meshes are shown in (d). The mesh in (b) has 942 regular vertices and the mesh in (a) has 878 regular vertices. $h = 0.8$.

ACKNOWLEDGMENTS

REFERENCES

[1] S.P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Information Theory,* vol. TIT-28, no. 2, pp. 129-136, Mar. 1982.
[2] J.B. Macqueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math., Statistics, and Probability,* pp. 281-297, 1967.
[3] Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Lu, and C. Yang, "On Centroidal Voronoi Tessellation—Energy Smoothness and Fast Computation," *ACM Trans. Graphic,* vol. 28, no. 4, pp. 1-17, 2009.
[4] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi Tessellations: Applications and Algorithms," *SIAM Rev.,* vol. 41, no. 4, pp. 637-676, http://link.aip.org/link/?SIR/41/637/1, 1999.
[5] Q. Du, M. Gunzburger, and L. Ju, "Advances in Studies and Applications of Centroidal Voronoi Tessellations," *Numerical Math.: Theory, Methods and Applications,* vol. 3, no. 2, pp. 119-142, 2010.
[6] A. Gersho, "Asymptotically Optimal Block Quantization," *IEEE Trans. Information Theory,* vol. TIT-25, no. 4, pp. 373-380, July 1979.

[7]   Z. Li and H.A. Scheraga, "Monte Carlo-Minimization Approach to the Multiple-Minima Problem in Protein Folding," *Proc. Nat'l Academy of Sciences of USA,* vol. 84, no. 19, pp. 6611-6615, Oct. 1987.

[8]   G.F. Tòth, "A Stability Criterion to the Moment Theorem," *Studia Scientiarum Mathematicarum Hungarica,* vol. 34, pp. 209-224, 2001.

[9]   Q. Du and D. Wang, "The Optimal Centroidal Voronoi Tessellations and the Gersho's Conjecture in the Three-Dimensional Space," *Computers and Math. with Applications,* vol. 49, nos. 9/10, pp. 1355-1373, 2005.

[10] R. Gray and D. Neuhoff, "Quantization," *IEEE Trans. Information Theory,* vol. 44, no. 6, pp. 2325-2383, Oct. 1998.

[11] E. Hendrix and B.G. Tóth, *Introduction to Nonlinear and Global Optimization.* Springer, 2010.

[12] N. Metropolis and S. Ulam, "The Monte Carlo method," *J. Am. Statistical Assoc.,* vol. 44, no. 247, pp. 335-341, 1949.

[13] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proc. IEEE Int'l Conf. Neural Networks,* pp. 1942-1948, 1995.

[14] *Handbook of Evolutionary Computation,* T. Baeck, D.B. Fogel, and Z. Michalewicz, eds. IOP Publishing Ltd., 1997.

[15] A. Nayeem, J. Vila, and H.A. Scheraga, "A Comparative Study of the Simulated-Annealing and Monte Carlo-with-Minimization Approaches to the Minimum-Energy Structures of Polypeptides: (met)-Enkephalin," *J. Computational Chemistry,* vol. 12, no. 5, pp. 594-605, 1991.

[16] S.B. Ozkan and H. Meirovitch, "Conformational Search of Peptides and Proteins: Monte Carlo Minimization with an Adaptive Bias Method Applied to the Heptapeptide Deltorphin," *J. Computational Chemistry,* vol. 25, no. 4, pp. 565-572, 2004.

[17] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chemical Physics,* vol. 21, pp. 1087-1092, http://dx.doi.org/10.1063/1.1699114, June 1953.

[18] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, no. 4598, pp. 671-680, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.4175, 1983.

[19] R. Horst, P.M. Pardalos, and N.V. Thoai, "Nonconvex Optimization and Its Applications," *Introduction to Global Optimization,* second ed., vol. 48, Kluwer Academic Publishers, http://www.ams.org/mathscinet-getitem?mr=1799654, 2000.

[20] P. Pardalos and E. Romeijn, *Handbook of Global Optimization,* vol. 2, Kluwer Academic Publishers, 2002.

[21] S. Kirkpatrick, "Optimization by Simulated Annealing: Quantitative Studies," *J. Statistical Physics,* vol. 34, no. 5, pp. 975-986, http://www.springerlink.com/content/R8316332T1U15773, 1984.

[22] T. Weise, *Global Optimization Algorithms—Theory and Application,* second ed., http://www.it-weise.de/, June 2009.

[23] J.A. Snyman and L.P. Fatti, "A Multi-Start Global Minimization Algorithm with Dynamic Search Trajectories," *J. Optimization Theory and Applications,* vol. 54, pp. 121-141, http://dx.doi.org/10.1007/BF00940408, 1987, doi: 10.1007/BF00940408.

[24] R.W. Eglese, "Simulated Annealing: A Tool for Operational Research," *European J. Operational Research,* vol. 46, no. 3, pp. 271-281, http://ideas.repec.org/a/eee/ejores/v46y1990i3p271-281.html, June 1990.

[25] Q. Du and M. Gunzburger, "Grid Generation and Optimization Based on Centroidal Voronoi Tessellations," *Applied Math. Computation,* vol. 133, nos. 2/3, pp. 591-607, 2002.

[26] Q. Du, M.D. Gunzburger, and L. Ju, "Constrained Centroidal Voronoi Tessellations for Surfaces," *SIAM J. Scientific Computing,* vol. 24, pp. 1488-1506, http://portal.acm.org/citation.cfm?id=767408.767429, May 2002.

[27] D.-M. Yan, B. Lévy, Y. Liu, F. Sun, and W. Wang, "Isotropic Remeshing with Fast and Exact Computation of Restricted Voronoi Diagram," *Computer Graphics Forum,* vol. 28, no. 5, pp. 1445-1454, 2009.
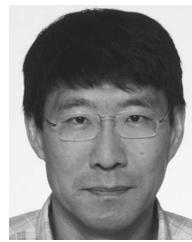
Lin Lu received the BEng and MEng degrees at Shandong University, and the PhD degree at The University of Hong Kong, all in computer science, in 2002, 2005, and 2011, respectively. Currently, she is working as an assistant professor of computer science at Shandong University, China. Her research interests include computer graphics and computational geometry.

Feng Sun received the BEng and MEng degrees from Shandong University and the PhD degree from The University of Hong Kong, all in computer science. His research interests include mesh generation, computational geometry, and computational topology.

Hao Pan received the BEng degree in software engineering from Shandong University. Currently, he is working toward the PhD degree in computer science at The University of Hong Kong. His research interests include computer graphics and computational geometry.

Wenping Wang received the BSc and MEng degrees at Shandong University, China, and the PhD degree at the University of Alberta, Canada, all in computer science, in 1983, 1986, and 1992, respectively. Currently, he is working as a professor of computer science at The University of Hong Kong and also an associate editor of the Springer journal *Computer Aided Geometric Design* and *IEEE Transactions on Visualization and Computer Graphics.* He is a program cochair of several international conferences, including Geometric Modeling and Processing (GMP 2000), Pacific Graphics 2003, ACM Symposium on Physical and Solid Modeling (SPM 2006), and International Conference on Shape Modeling (SMI 2009). His research interests include computer graphics, visualization, and geometric computing.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.