

基于局部优化的重心 Voronoi 图计算

叶旼宇¹, 王逸群², 严冬明^{2*}, 雍俊海¹

(1. 清华大学软件学院, 北京 100084; 2. 中国科学院自动化研究所模式识别国家重点实验室, 北京 100190)

摘要: 重心 Voronoi 图(centroidal Voronoi tessellation, CVT)是一个重要的几何结构, 在地理信息系统, 信号处理, 网格生成/优化, 可视化等领域有着重要应用。针对传统全局生成、优化的方法的不足, 比如奇异点多、收敛速度较慢等问题, 提出了生成优化与随机扰动两种局部优化方法, 以及一个整合了层次生成、局部优化、蒙特卡罗优化的 CVT 生成算法框架。实验结果表明, 相比于已有算法, 本文方法在速度与质量上有综合的提升。

关键词: 重心 Voronoi 图; 局部极小; 优化; 奇异点

中图分类号: TP301

文献标识码: A

文章编号: 1004-731X (2019) 02-0218-09

DOI: 10.16182/j.issn1004731x.joss.17DEA-010

Centroidal Voronoi Tessellation with Local Optimization

Ye Tianyu¹, Wang Yiqun², Yan Dongming^{2*}, Yong Junhai¹

(1. School of Software, Tsinghua University, Beijing 100084, China;

2. NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Centroidal Voronoi tessellation is a special geometric structure, which has many applications in various fields such as geographical information system, signal processing, mesh generation/optimization, visualization and so on. Due to the highly non-convex nature of the CVT energy function, the existing methods for computing CVT have several drawbacks, which always trap into local minima. We propose generation optimization and stochastic optimization schemes for further reducing the CVT energy. Experimental results show that the proposed method improves both quality and efficiency compared to the recent approaches.

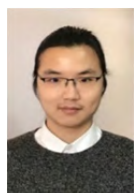
Keywords: centroidal Voronoi tessellation; local minimum; optimization; irregular point

引言

重心 Voronoi 图(Centroidal Voronoi Tessellation - CVT)是一种特殊的 Voronoi 图, 每个生成点与其对应的 Voronoi 单元的重心重合^[1]。在最优情况下, CVT 的每个 Voronoi 单元都是全等的正六边形, 其

对偶则是一个由等边三角形组成的三角网格, 该性质在网格优化、科学计算、图像处理等应用中被广泛使用^[2-3]。因而, CVT 的计算方法是一个重要研究课题。

传统的方法采用 Lloyd 迭代^[4]来计算 CVT, 但是该方法只有线性的收敛速度^[5]。Liu 等^[6]对 CVT 能量函数的光滑性进行了深入的研究, 给出 CVT 是二阶连续的理论证明, 并且提出了一个基于 L-BFGS 的快速优化框架计算 CVT, 比已有的算法加速一个数量级。但是, 由于 CVT 能量函数有着极强的非凸性质, 有非常多局部极小, 目前如何计



收稿日期: 2016-10-12 修回日期: 2017-01-30;
基金项目: 国家科技支撑计划(2015BAF23B03), 国家
国际科技合作专项(2013DFE13120), 国家自然科学基金
(61772523, 61272235, 61372168, 61672307);
作者简介: 叶旼宇(1990-), 男, 上海, 博士生, 研究
方向为计算机辅助几何设计。

算 CVT 的全局最优仍然是一个开放问题。

近期, Lu 等^[7]提出基于蒙特卡罗和模拟退火方法来进一步优化 CVT, Wang 等^[8]则采用层次化方法计算 CVT。但是前者的速度较慢、效率不高, 后者的生成质量有限。

针对当前 CVT 计算中的问题, 本文提出了基于局部优化的思想计算 CVT, 并给出了“生成优化”与“随机优化”等两种局部优化方法, 可以按照一定优先序对 CVT 进行局部优化。同时, 提出一个综合的算法框架, 有效地结合了层次生成法、局部优化、以及蒙特卡罗优化等算法, 有效的提高了 CVT 全局优化的计算效率以及质量。

1 相关工作

本节我们简要介绍关于 CVT 的研究历史, 并且讨论一些最近的相关工作。关于 CVT 的详细介绍可以参考综述文献[1,9]。

CVT 最早用于信号处理中的最优量化计算。给定一个初始采样集合, 通常采用 Lloyd 迭代^[4]进行优化, 直至收敛。关于 CVT 的理论最优解, 著名学者 Gersho 在一维和二维欧式空间给出了一个猜想, 既“在渐进意义下, 最优的 CVT 形成一个空间划分, 所有的 Voronoi 单元都全等于一个维数相关的基本单元”^[10]。在二维空间, 这个基本单元是一个正六边形。Tóth 于 2010 年在二维空间证明了这个猜想^[11]。

从优化角度看, CVT 对应的能量函数希望极小化量化误差。早期日本学者 Iri 等认为, 由于连接关系复杂, CVT 能量函数是不光滑的^[12-13]。Du 等^[5]首先证明了 Lloyd 迭代具有线性的收敛速度。接着, Du 等假设当 Voronoi 图拓扑结构不再发生变化时, 提出了一个 CVT 的快速计算方法^{错误!未找到引用源。}, 但是他们的优化目标和 CVT 能量函数不同。随后, Liu 和 Wang 等^[6]证明 CVT 能量函数具有二阶光滑性质, 并且提出了一个拟牛顿框架来迭代计算 CVT。比较传统的 Lloyd 迭代, 将 CVT 的计算效率提高了一个数量级。但是, 由于 CVT 能量函数

的高度非凸性质, 很容易陷入局部极小。如何计算 CVT 全局最优仍然是没有解决的问题。

最近, Lu 等提出采用蒙特卡罗以及模拟退火方法对 CVT 进行全局优化^[7], 通过不断对 Voronoi 图中各种子点进行小幅度扰动并再以拟牛顿法得到局部最优解的方法来尽可能逼近一个全局最优解。由于该方法需要进行全局扰动, 其效率不是很高。

另外, Wang 等采用的层次化方法^[8]则通过对稀疏 CVT 进行插点与拟牛顿优化的方法来得到高密度 CVT。通过该方法得到的 CVT 更为结构化, 但是 CVT 的能量并没有显著降低, 尤其在边界处。

本文针对这些缺陷进一步进行研究。

2 CVT 定义

给定 n 个 R^N 空间中不同的种子点(或者生成点) $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, Voronoi 图定义为 R^N 空间的一个划分 $\mathbf{C} = \{\Omega_i\}_{i=1}^n$, 每个种子点对应一个 Voronoi 单元 $\Omega_i = \{\mathbf{x} \in R^N \mid \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{x}_j\|, \forall j \neq i\}$ 。其中 $\|\cdot\|$ 为欧式距离。

重心 Voronoi 图是一个特殊的 Voronoi 图: 所有的种子点 \mathbf{x}_i 和它对应的 Voronoi 单元 Ω_i 的重心 \mathbf{x}_i^* 重合, 其中

$$\mathbf{x}_i^* = \frac{\int_{\Omega_i} \rho(\mathbf{x}) \mathbf{x} d\sigma}{\int_{\Omega_i} \rho(\mathbf{x}) d\sigma} \quad (1)$$

这里 $\rho(\mathbf{x}) > 0$ 是一个用户指定的密度函数。如果 ρ 是一个常数, 则 CVT 为均匀密度。CVT 也可以定义为下面能量函数的一个局部极小值^[1]:

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

3 算法概述

本文首先提出了对 CVT 局部优化的概念, 即固定大部分采样点, 通过算法生成或随机生成的方式对局部采样点进行尝试优化。并提出了一个优先度函数的概念, 通过定义优先度函数 $f: \Omega \rightarrow \mathbb{R}$, 按照优先序由低到高的顺序依次对 CVT 进行局部

优化。在此之上，本文提出了一个算法整合框架，将局部优化与层次法、蒙特卡罗法进行了整合。

3.1 局部生成优化

给定输入点集 X_0 、待优化点 x_i 、优先级函数 f ，并设置合适的内半径 R_{inner} 与外半径 R_{outer} ，可按照以下步骤来进行局部生成优化，见图 1(从左至右分别为：1) 对输入点集进行分类，其中蓝色为已优化点、橙色为待处理点、灰色为边界点；2) 生成点；3) 若生成点过多，则删去其中优先序最高者(图中以圈出表示)；4) 局部迭代优化)。

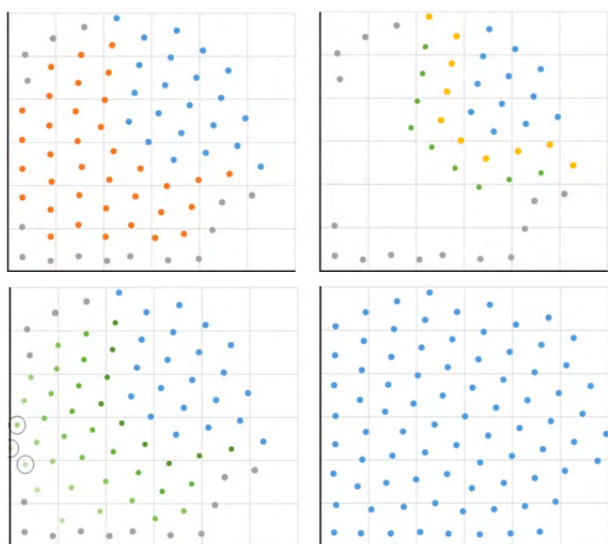


图 1 局部生成优化的几个步骤

Fig. 1 Steps of local generation optimization

1. 对 X_0 中的点 x_j 按照其优先序以及与 x_i 的距离 $d = \|x_i - x_j\|$ 进行如下分类：

- 已优化点： $d < R_{inner}$ 且 $f(x_j) < f(x_i)$ ；
- 待处理点： $d < R_{inner}$ 且 $f(x_j) > f(x_i)$ ；
- 边缘点： $R_{inner} < d < R_{outer}$ ；
- 外部点： $d > R_{outer}$ ；

2. 删去待处理点，并找到与待处理点相邻的已优化点作为初始种子点集；

3. 不断通过种子点集得到新一批的生成点，并将该批生成点作为新一轮的种子点进行下一轮生成，直到填满该局部为止；

4. 通过删除高优先序生成点或在生成点与边

缘点之间添加新点来保证该局部的点数与优化前保持一致；

5. 采用 Newton-Lloyd 迭代法对生成点进行局部迭代优化。

对于均匀密度 CVT，若采样点之间的平均最小距离为 $2r$ ，则此处 R_{inner} 与 R_{outer} 可分别取 $7r$ 与 $10r$ 。两者差值取 $3r$ 以保证边缘点的选取；若两者取值过小，则难以生成有效局部优化；若取值过大，则单步耗时会较长。对于非均匀密度 CVT，则应根据该局部的密度变化情况来确定 R_{inner} 与 R_{outer} 的取值。

对于步骤 3，在二维空间中可以采用如图 2 所示的方法来生成候选点，其中， O 为种子点， P 、 P' 为相邻点， Q_1 、 Q_2 、 Q_2' 、 Q_2'' 为生成点。对每个种子点，从已优化点、外部点、生成点中找到与其相邻的点，通过判断相邻点之间的夹角来均匀等角度、等半径变化地生成适量的候选点，使其相邻点之间的夹角接近 60° ；然后，将各个种子点产生的候选点进行合并，从而得到生成点。

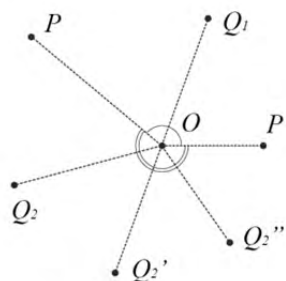


图 2 通过相邻点生成候选点

Fig. 2 Generating candidate points from adjacent points.

另外值得注意的是，通过上述步骤得到的点集并没有达到全局的 CVT 局部极小值，需要再进行全局的 Newton-Lloyd 迭代优化。当点数较大时，为了提升算法效率，可以采用以 x_i 为中心、半径较大的局部迭代优化来近似代替全局迭代优化。

3.2 局部随机扰动

与局部生成优化类似，给定输入点集 X_0 、待优化点 x_i 、优先级函数 f ，并设置合适的内半径

R_{inner} 、外半径 R_{outer} 与扰动半径 r_{rd} , 可以按照以下几个步骤来进行局部随机扰动:

1. 对 X_0 中的点 x_j 按照其优先序以及 x_i 的距离 $d = \|x_i - x_j\|$ 进行如下分类:

- 已优化点: $d < R_{inner}$ 且 $f(x_j) < f(x_i)$;
- 待处理点: $d < R_{inner}$ 且 $f(x_j) \geq f(x_i)$;
- 边缘点: $R_{inner} \leq d < R_{outer}$;
- 外部点: $d \geq R_{outer}$;

2. 对待处理点进行最大距离为 r_{rd} 的随机扰动;

3. 对扰动后的点进行局部迭代优化。

此处 R_{inner} 与 R_{outer} 的取值限制影响与局部生成优化相同; r_{rd} , 根据 Lu 的研究^[7], 可取 $0.6r$ 。另外同样, 在得到局部随机优化的结果后, 还应在更大范围内进行 Newton-Lloyd 迭代优化。

相比于局部生成优化, 随机扰动优化有更大的灵活性, 可以应对各种复杂情形。但是同时, 由于局部随机扰动具有不确定性, 需要尝试多次才能得到一个更优的结果, 故而一般随机扰动更为费时。

3.3 局部优化整体算法框架

考虑到局部生成优化效率较高, 局部随机扰动能应对各种复杂情况, 我们提出一个局部优化的算法框架来逐个局部地对给定 CVT 进行优化。给定空间 Ω 、密度函数 $\rho: \Omega \rightarrow R$ 、以及初始点集 X_0 , 并设置合适的优先序函数 $f: \Omega \rightarrow R$ 、终止优先序 f_{max} 、以及随机优化次数 m_{rd} , 算法框架步骤如下:

1. 令难以优化点集 $P \leftarrow \emptyset$, 结果点集 $X \leftarrow X_0$;

2. 寻找到不在集合 P 中且优先序最小的待优化点 x_i , 若找不到则算法结束;

3. 以 x_i 为中心进行生成优化, 然后从得到的结果点集 X' 中找出未优化点构成点集 P' , 并将其与 P 进行最大匹配, 并判断 X' 是否优于 X ;

4. 若 X' 未优于 X , 则对 x_i 不断进行局部随机扰动优化、从扰动结果 X' 中得到未优化点集 P' 并与 P 进行最大匹配, 直到 X' 优于 X 或达到最大随

机扰动次数 m_{rd} 为止;

5. 若得到了更优的 X' , 则将 X' 与 P' 赋予 X 与 P ; 否则, 令 $P \leftarrow P \cup x_i$ 。

6. 跳转步骤 2。

该算法对输入的初始点集 X_0 没有硬性要求。如果 X_0 已经采用 Newton-Lloyd 迭代法达到一个极小值, 会有助于后续算法的进行; 如果采用随机点集作为输入, 也会随着算法的进行、逐个局部的优化, 得到一个优化后的结果。

关于优先序函数 f , 该函数定义了空间 Ω 中各点的优化先后顺序, 可以根据空间 Ω 的形状、框架中采用的局部生成算法的性质、以及对生成曲面的要求等来进行选取。考虑到本文所采用的局部生成算法通常在 Ω 内部远离边界处得到较好的结果, 对于单位方域, 此处可以采用 $f_{square}(x) = \|x - (0.5, 0.5)^T\|$ 作为一个基本的优先序函数。

关于终止优先序 f_{max} , 则可以根据需要进行设置与调整。如后文中章节 4.4 所需, 若调用者希望该局部优化算法能对 Ω 中距离边界 d_{bd} 以上的部分进行优化, 则可选取合适的过度距离 d_{tr} 来构造边界函数 $f_{boarder}$, 并将其与基本优先序函数 f_0 进行调和得到:

$$f(x) = \begin{cases} f_0(x), & d_{tr} + d_{bd} < d \\ \alpha \cdot f_0(x) + (1 - \alpha) \cdot f_m, & d_{bd} \leq d < d_{tr} + d_{bd} \\ f_m + d_{bd} - d, & d \leq d_{bd} \end{cases} \quad (3)$$

式中: d 为 x 到采样区域边界 $\partial\Omega$ 的最小距离; $\alpha = (d - d_{bd}) / d_{tr}$; $f_m = \max_x \{f(x) | d = d_{bd}\}$ 。当 x 远离边界时, 该函数即为基本优先序函数; 当 x 靠近边界时, 该函数即为边界函数, 以与边界的距离来线性减少优先序; 当 x 位于过渡区时, 该函数为两者的线性插值。在单位方域中, 若采样点之间的平均最小距离为 $2r$, 则此处 b_{tr} 的取值范围可为 $[4r, 6r]$ (见图 3, 其中, 左图为基函数 f_{square} ; 中图为边界函数 $f_{boarder}$; 右图为两者的调和结果)。

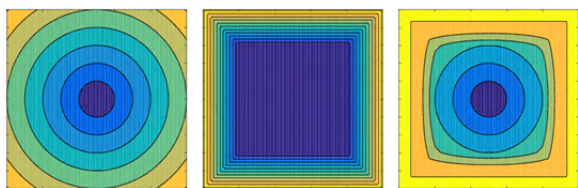


图 3 单位方域上的一种优先序函数
Fig. 3 A kind of priority on unit square field.

关于一个点是否还需优化的判断,在二维平面上,可以简单采用点的度数(相邻点的个数)是否为 6 来进行衡量。对于变密度以及三维 CVT 的情况,可以结合密度变化以及曲率变化来进行判断。

考虑到会存在靠近 Ω 边界附近、或是在优先序函数的鞍点或最大值附近会存在一些点难以被局部优化,我们在算法中设置 P 来存储难以优化的点集合。若生成优化与 m_{rd} 次随机扰动均无法对某个局部点进行优化,则将该点加入 P ,不再对该点进行优化。

关于未优化点的匹配,这里并不需要进行精确匹配,关注点在于优先序小于 X_i 的未优化点是增加或减少了。匹配的作用在于,考虑到局部优化会对全局未优化点的位置造成一定影响,匹配能帮助判断在与 X_i 优先序接近的未优化点中,哪些局部是已经尝试过进行优化而没有成功的,从而避免因扰动造成对一些难以优化的局部进行反复优化。

关于点集 X' 是否优于 X 的判断,先从 P' 中找到最小优先序的未匹配点 x_p ,并在 x_i 的给定半径 R_{detect} 范围内找到局部最小优先序的未匹配点 x_p' ,若满足以下条件之一,则认为 X' 优于 X :

- 最小未优化点外移: $f(x_p) > f(x_i) + df_{\text{toler}}$;
- 未优化点减少: $f(x_p) > f(x_i) - df_{\text{toler}}$ 且 $|P_{\text{match}}'| < |P|$;
- 局部已优化,远处发生的劣化在可接受范围内: $f(x_{p'}) > f(x_i) + df_{\text{toler}}$ 且 $f(x_p) > f(x_i) - df_{\text{worse}}$ 。

此处 df_{toler} 与 df_{worse} 为给定的经验参数,若采样点之间的平均最小距离为 $2r$,则 df_{toler} 与 df_{worse} 分别可取 $r/6$ 与 $r/3$ 。设置条件三接受局部优化,

是因为在优化过程中可能会产生不稳定的优化局部,远处的些许改变都会导致该局部的劣化,从而阻止了其他区域的优化。有了条件三,使得其他地方的优化可以继续,之后再重新对该劣化后的不稳定局部进行优化。

3.4 多算法整合框架

局部生成算法在靠近采样区 Ω 边界附近通常表现不佳,需要通过随机扰动来进行优化,从而导致算法效率低下。故本文将局部优化法与层次法、蒙特卡罗法进行结合,提出了一个混合框架(下文简称混合法, Hybrid CVT)。

给定有限空间 Ω 、采样点数 n 、密度函数 $\rho: \Omega \rightarrow R$ 、优先序函数 $f: \Omega \rightarrow R$ 、局部优化最大优先序 f_{max} 、以及其他各方法中用到的相关参数,则如图 4 所示,从左至右依次为:初始随机采样点(31 个点);第一次细分并局部优化(125 个点);第二次细分并局部优化(500 个点);蒙特卡罗法进行周边优化(500 个点)。填充色表示该单元单位面积 CVT 能量值(红高蓝低)。可以按如下算法框架生成结果点集 X :

- 1: 通过 n 计算层次法层数 h 与每层点数 $n_i, i = 0 \cdots h$;
- 2: 生成初始 n_0 个采样点的 CVT 点集 X ;
- 3: for $l \in [1, h]$ do
- 4: 以层次法插入 $n_l - n_{l-1}$ 个点,并更新点集 X ;
- 5: 以局部优化法按一定优先序优化 $f(x_i) < f_{\text{max}}$ 的点,并更新点集 X ;
- 6: end for
- 7: 用蒙特卡罗法优化 X 中 $f(x) > f_{\text{max}}$ 部分区域。

步骤 1 中, $n_h = n, n_i = n_{i-1} \times 4 + c_i, c_i \in [0, 4]^{[8]}$, 并根据 Ω 的形状,选择合适的 n_0 。对于单位方域, n_0 可在 $[25, 100]$ 内进行选取。步骤 2 中,本文简单采用随机法产生 n_0 个采样点,并用 Newton-Lloyd 法进行全局迭代优化。步骤 7 中,若采用章节 3.3 所述调和优先序函数,则此处可给定边距 d_{influ} ,

简单对与边界距离小于 d_{bd} 的点进行随机扰动, 并对边距小于 d_{inflow} 的点进行局部 Newton-Lloyd 迭代优化, 以替代全局迭代优化, 近似得到全局 CVT 极小值。

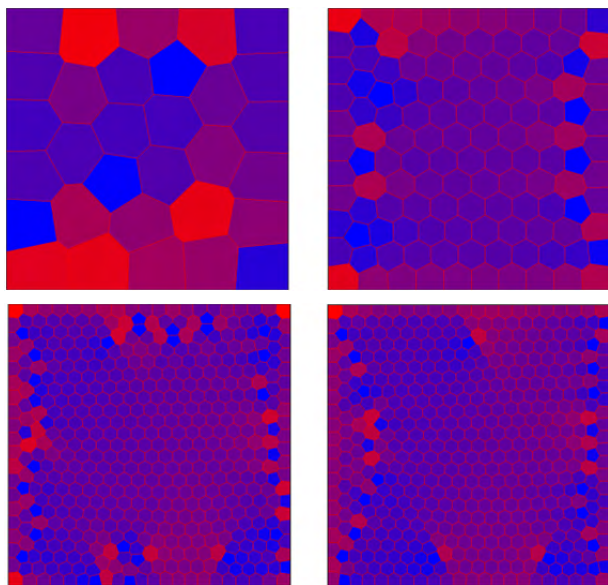


图 4 混合算法框架步骤说明
Fig. 4 Hybrid CVT framework.

关于算法中蒙特卡罗法的迭代次数问题, 考虑到蒙特卡罗法较为耗时且优化效果有限, 为了平衡算法用时与优化效果, 可以设置该算法在整个算法中的相对用时系数 β 。通常, β 可取 0.2–0.8。即, 在步骤 7 中估算算法其他步骤所耗费的时间 t_{other} 与蒙特卡罗法一次迭代耗时 t_{one} , 并以蒙特卡罗法迭代 $a_{time} = \left\lceil \frac{\beta \cdot t_{other}}{(1-\beta) \cdot t_{one}} \right\rceil$ 次。另外, 为了避免当 n 较小时 t_{other} 几乎为零、无法执行蒙特卡罗法的问题, 可以设置一个最小迭代次数 a_{min} 。

4 实验结果

在本章节中, 我们对传统 CVT、全局蒙特卡罗法、层次法、以及本文所述混合法等在等密度单位方域上进行了实验与比较, 并对局部生成优化与局部随机扰动的运行效率进行了分析。

在我们的实现中, 采用开源的计算几何算法库

CGAL^[15] 计算 Delaunay 三角化以及 Voronoi 图, 采用 Yan 等^[16] 的周期 CVT 计算框架。对于蒙特卡罗法, 我们采用 Liu 等^[7] 所推荐的参数, 并采用 200 次迭代与其他算法进行比较, 与 Wang 等^[8] 保持一致。对于混合法, 我们根据采样点个数 n , 计算采样点之间最小平均距离 $2r$, 取参数局部优化内半径 $R_{inner} = 7r$ 、外半径 $R_{outer} = 10r$, 随机优化次数 $m_{rd} = 20$, 蒙特卡罗优化系数 $\beta = 0.5$, 最小迭代次数 $a_{min} = 20$ 。优先序函数 f 采用 3 式, 其中 $d_{tr} = 4r$, $d_{bd} = 5.5r$ 。所有的实验均在 windows 10、Intel i7 3.9 Hz 环境下进行。

4.1 单位方域上的 CVT

我们使用随机点作为初始点集, 分别采用传统 CVT、全局蒙特卡罗法(Monte-Carlo CVT)^[7]、层次法(Hierarchical CVT)^[8]、以及本文的混合法(Hybrid CVT) 对 CVT 进行优化。

如图 5 与表 1 所示, 混合法在点数较少时优于传统 CVT 与层次法, 得到与蒙特卡罗法相似的优化结果; 在点数较多时, 混合法的结果优于其他 3 种方法, 蒙特卡罗法 200 次迭代已不足以进行充分优化。图 5 中所采用的算法从左至右依次为: 传统 PCVT、蒙特卡罗、层次法、混合法; 采样点的个数从上至下依次为: 250, 1000, 4000。填充色表示该单元单位面积 CVT 能量值。

蒙特卡罗法在迭代优化时, 会使得奇异点合并、形成几大块无奇异点的区域, 区域与区域之间以奇异点链相连。当点数较少时, 奇异点链的曲率半径相对于单元半径的相对系数较小, 故而在进行蒙特卡罗尝试时能较容易地使得奇异点链向凹陷方向移动, 从而完成奇异点的合并优化。随着点数的增多, 单元半径减小, 奇异点链的曲率半径相对于单元半径的相对系数增大, 即奇异点链变得更为平坦, 从而不易发生有效移动, 导致优化效率降低。

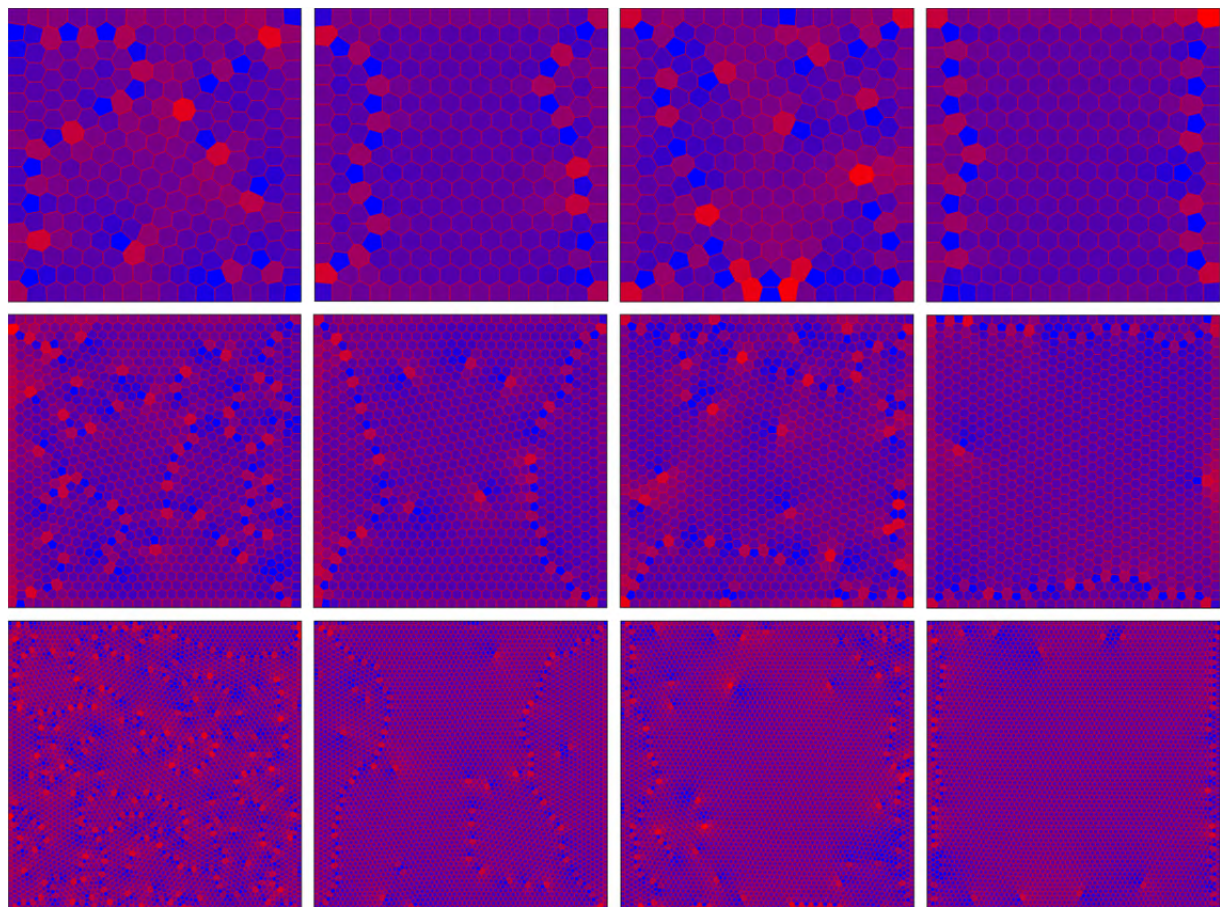


图 5 单位方域上不同算法的比较结果
Fig. 5 Results of different algorithms on a unit square.

表 1 不同算法的 CVT 能量比较
Tab. 1 CVT energy comparison between algorithms

点数	传统 CVT	蒙特卡罗法	层次法	混合法
100	0.019 782	0.019 660	0.019 683	0.019 674
250	0.007 830	0.007 799	0.007 824	0.007 808
500	0.003 901	0.003 886	0.003 897	0.003 891
1 000	0.001 947 2	0.001 938 9	0.001 944 3	0.001 940 5
2 000	0.000 972 3	0.000 967 7	0.000 970 7	0.000 968 0
4 000	0.000 485 8	0.000 483 5	0.000 484 5	0.000 483 2
8 000	0.000 242 70	0.000 241 57	0.000 242 01	0.000 241 34
16 000	0.000 121 34	0.000 120 75	0.000 120 91	0.000 120 59

如表 2 所示，传统 CVT 与层次法运算效率最快，本文所述方法能在一个可接受时间内得到较优解，而蒙特卡罗法则耗时巨大。

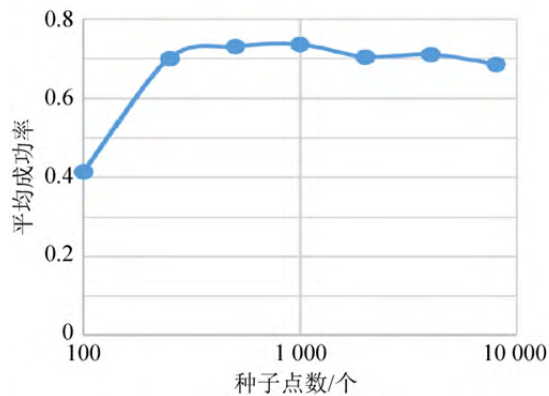
同时，如图 6 所示，我们对在不同种子点数下局部生成优化算法的优化平均成功率、以及随机优化中平均所需随机次数进行了统计。当种子点数较

少时，大多数种子点的优先序皆大于终止优先序，算法仅对中央少数种子点进行局部优化，而最初优化的几个种子点由于缺少有效的参考，故而其优化难度较大。随着种子点数的进一步增加，对局部进行优化时外部点的增加会导致 CVT 的局部最优能量势阱加深，从而对局部的修改将更难以影响全局、

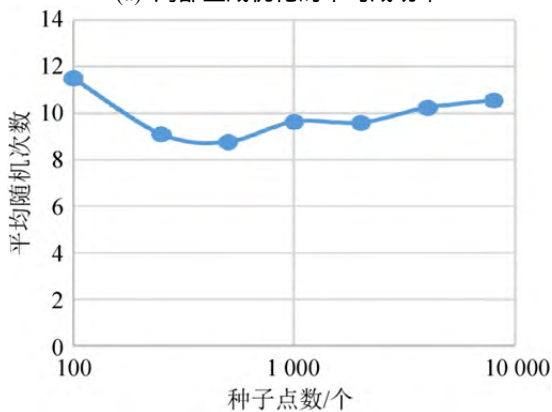
跳出该局部最优解。这一点表现在数据上即为当种子点数较大时生成优化的平均成功率逐步降低、随机优化的平均随机次数逐步增高。

表 2 不同算法的耗时比较
Tab. 2 Time consumption comparison /s

点数	传统 CVT	蒙特卡罗法	层次法	混合法
100	0.205	24.9	0.261	1.74
250	0.474	62.2	0.634	4.98
500	0.758	111.9	1.140	7.02
1000	1.230	178.1	2.045	15.84
2000	1.914	302.2	3.329	49.56
4000	3.077	529.3	5.525	111.21
8000	5.352	998.2	9.493	293.1
16000	9.475	1913.8	15.893	761.0



(a) 局部生成优化的平均成功率



(b) 局部随机优化的平均随机次数

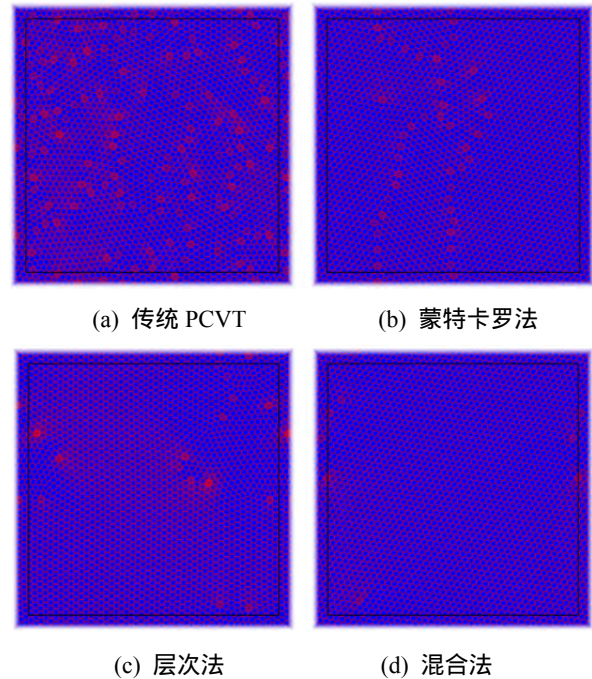
图 6 运行结果统计

Fig. 6 Statistics of operation results

4.2 单位方域上的周期 CVT 结果

如图 7 所示,在周期域上,蒙特卡罗法容易陷入难以优化的状态。相比而言,层次法除了初始第

一层随机 CVT 时产生的奇异点(度数非 6 的点)、以及每一轮插点时因点数不能被 4 整除而产生的个别奇异点之外,可以保证每一次插入的其他点的度数皆为 6,因而在周期域上效果较好。而本文所述混合法产生的 CVT 质量最高,由于在周期域上不受边界的影响,能去除绝大部分奇异点。



(a) 传统 PCVT

(b) 蒙特卡罗法

(c) 层次法

(d) 混合法

图 7 周期单位方域上 2 000 个点的不同算法结果比较
Fig. 7 Result comparison between different algorithms with 2 000 points

5 结论

本文针对 CVT 问题提出了局部优化的思路,并将其与层次法、蒙特卡罗法进行了整合。实验结果表明,本文所述方法能够得到更优的 CVT 结果。算法中,主要时间消耗在随机扰动生成上,若能改进局部优化生成算法,提升其优化成功率,则可进一步提升算法性能。

另外,在本文中,局部优化生成算法采用的是等角度生成相邻点,局部优化框架中通过采样点的度数是否为 6 来判断该点是否被优化,这两者均不能很好地适应变密度的 CVT 生成。在未来的研究中,可以针对变密度 CVT 进一步改进该算法和框架。

参考文献:

- [1] Du Q, Faber V, Gunzburger M. Centroidal Voronoi tessellations: Applications and algorithms[J]. SIAM review, (S0036-1445), 1999, 41(4): 637-676.
- [2] Du Q, Gunzburger M. Grid generation and optimization based on centroidal Voronoi tessellations[J]. Applied Mathematics and Computation (S0096-3003), 2002, 133(2/3): 591-607.
- [3] Du Q, Gunzburger M D, Ju L. Constrained centroidal Voronoi tessellations for surfaces[J]. SIAM Journal on Scientific Computing (S1064-8275), 2003, 24(5): 1488-1506.
- [4] Lloyd S. Least squares quantization in PCM[J]. IEEE transactions on information theory (S0018-9448), 1982, 28(2): 129-137.
- [5] Du Q, Emelianenko M, Ju L. Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations[J]. SIAM journal on numerical analysis (S0036-1429), 2006, 44(1): 102-119.
- [6] Liu Y, Wang W, Lévy B, et al. On centroidal Voronoi tessellation—energy smoothness and fast computation[J]. ACM Transactions on Graphics (ToG) (S0730-0301), 2009, 28(4): 101.
- [7] Lu L, Sun F, Pan H, et al. Global optimization of centroidal voronoi tessellation with monte carlo approach[J]. IEEE transactions on visualization and computer graphics (S1077-2626), 2012, 18(11): 1880-1890.
- [8] Wang L, Héty-Wheeler F, Boyer E. A hierarchical approach for regular centroidal Voronoi tessellations[J]. Computer Graphics Forum (S0167-7055), 2016, 35(1): 152-165.
- [9] Du Q, Gunzburger M, Ju L. Advances in studies and applications of centroidal Voronoi tessellations[J]. Numerical Mathematics: Theory, Methods and Applications (S1004-8979), 2010, 3(2): 119-142.
- [10] Gersho A. Asymptotically optimal block quantization[J]. IEEE Transactions on information theory, (S0018-9448), 1979, 25(4): 373-380.
- [11] Fejes Tóth G. A stability criterion to the moment theorem[J]. Studia Scientiarum Mathematicarum Hungarica (S0081-6906), 2001, 38(1/2/3/4): 209-224.
- [12] Asami, Y. "A Note on the Derivation of the First and Second Derivatives of Objective Functions in Geographical Optimization Problems", Journal of Faculty of Engineering, University of Tokyo[J], 1991, 61(1), 1-13.
- [13] Iri M, Murota K, Ohya T. A fast Voronoi-diagram algorithm with applications to geographical optimization problems[M]. Berlin, Heidelberg: System modelling and optimization. Springer, 1984: 273-288.
- [14] Du Q, Emelianenko M. Acceleration schemes for computing centroidal Voronoi tessellations[J]. Numerical linear algebra with applications (S1070-5325), 2006, 13(2/3): 173-192.
- [15] CGAL, Computational Geometry Algorithms Library. <http://www.cgal.org>.
- [16] Yan D M, Wang K, Lévy B, et al. Computing 2D periodic centroidal Voronoi tessellation[C]// Voronoi Diagrams in Science and Engineering (ISVD), 2011 Eighth International Symposium on. IEEE, 2011: 177-184.