Inverting Laguerre Tessellations

Qibin Duan[1], Dirk P. Kroese[1], Tim Brereton[2], Aaron Spettl[2] and Volker Schmidt[2]

[1] *School of Mathematics and Physics, The University of Queensland, Brisbane 4072, Australia*
[2] *Institute of Stochastics, Ulm University 89069 Ulm, Germany*
Email: qbduan@gmail.com; kroese@maths.uq.edu.au

A Laguerre tessellation is a generalization of a Voronoi tessellation where the proximity between points is measured via a power distance rather than the Euclidean distance. Laguerre tessellations have found significant applications in materials science, providing improved modeling of (poly)crystalline microstructures and grain growth. There exist efficient algorithms to construct Laguerre tessellations from given sets of *weighted* generator points, similar to methods used for Voronoi tessellations. The purpose of this paper is to provide theory and methodology for the inverse construction; that is, to recover the weighted generator points from a given Laguerre tessellation. We show that, unlike the Voronoi case, the inverse problem is in general non-unique: different weighted generator points can create the same tessellation. To recover pertinent generator points we formulate the inversion problem as a multimodal optimization problem and apply the cross-entropy method to solve it.

Keywords: Laguerre tessellation, power diagram, power distance, inverse problem, generator points, multimodal optimization, cross-entropy method

1. INTRODUCTION

The *Voronoi tessellation* of a given set of *generator points* in the Euclidean plane divides the plane into disjoint cells (regions, tiles), such that, for all points in a cell, the Euclidean distance to the generator point within that cell is less than the distance to all other generator points. Applications of Voronoi tessellations may be found in fields as diverse as computational geometry, cell-biology, architecture, image analysis, ecology, and materials science. Various generalizations of the Voronoi tessellation have been proposed and investigated to better fit tessellations appearing in nature. These include tessellations that use different distance metrics and tessellations that use different geometric objects (e.g., lines and areas, instead of points); see, for example, [1].

An important generalization is the *Laguerre tessellation*, also called *power diagram*, which employs *weighted* generator points and uses the *power distance* to measure the proximity of points; see, e.g., [1, 2, 3]. It has been shown that many convex tilings in three or more dimensions are Laguerre tessellations (see [3]), and also in two dimensions Laguerre tessellations are common. In addition, Voronoi tessellations in a number of non-Euclidean geometries can be represented as Laguerre tessellations (see [4]). For these reasons, Laguerre tessellations have been a popular choice for modeling grain growth structures (see [5, 6]), foams (see [7, 8]), and boundaries of polycrystalline materials (see [9]).

In the statistical analysis of spatial data described by Laguerre tessellations (for example, in the study of microscopic materials and cell-tissue), it is often the case that the positions of the generator points are unknown. Knowledge of these sites is useful for statistical inference on the properties of random tessellations. In particular, the weighted generator points provide important information for constructing stochastic models and fitting them to experimental data. For example, stochastic models based on Laguerre tessellations can be used to investigate the effect of production parameters on stochastic realizations of material microstructures and their resulting functionality. This is done using computer simulations without the need to generate physical copies of the material. This process is known as virtual materials design (see, e.g., [10]). Another application where knowledge of the generating points is important is the modeling of grain growth via Laguerre tessellations. Here, it is useful to initialize computer simulations with exactly the same grains as are present in the real specimen. This makes it possible to compare and evaluate the evolution of individual grains.

The problem of *inverting* Voronoi tessellations — that is, recovering the generator points from the tessellation — has been well studied; see, for example, [11, 12, 13, 14, 15, 16]. However, the inversion problem for Laguerre tessellations seems to be much less studied. Recently, an approximate inversion method was given in [17], although with a different motivation, i.e., the possibility of reconstructing the complete tessellation using only cells' centers of mass and cell volumes. An alternative approach was presented in [18], where experimental 3D data was described by extraction of (parametric) cells using orthogonal regression — but without using Laguerre tessellations.

The goal of the current paper is to provide a better understanding of the inversion properties of Laguerre tessellations, and to formulate a simple and effective method for computing such inversions for 2D Laguerre tessellations. We find that, in contrast to Voronoi tessellations, the Laguerre inversion problem admits many solutions. We identify *pertinent* solutions with useful optimality properties. Our method involves a minimax optimization problem that is solved via the cross-entropy method ([19]).

The rest of this paper is organized as follows. In Section 2 we provide the mathematical background for Laguerre tessellations. In Section 3 we give the problem description and show that there may be multiple solutions to the inversion problem. We provide a simple algorithm to generate possible solutions, and discuss a minimization approach to obtain pertinent generator points. Section 4 details the cross-entropy algorithm for finding generator points that minimize the maximum weight (radius) of the Laguerre generator points. Numerical experiments in Section 5 demonstrate the effectiveness of the approach. Finally, Section 6 gives the conclusions and proposes directions for future research.

2. LAGUERRE TESSELLATIONS

A Laguerre tessellation, also called a power diagram or a Laguerre diagram, is a weighted version of the well-known Voronoi tessellation. In this section we introduce the mathematical notation and review some basic facts about these objects. See [1] and [2] for more details on Laguerre tessellations.

2.1. Definitions

Let $p \in \mathbb{R}^d$ be a fixed point and $w \in \mathbb{R}$ be a fixed value, called the *weight* of point p. We call the pair (p, w) a *weighted point.* For all $x \in \mathbb{R}^d$, we define the *power* of x with respect to (p, w) as

$$\operatorname{pow}(x, (p, w)) = \|x - p\|^2 - w.$$

Suppose $\mathbf{P} = \{(p_1, w_1), (p_2, w_1), \ldots, (p_n, w_n)\}$ is a (finite) set of weighted (generator) points in $\mathbb{R}^d$. The Laguerre tessellation of $\mathbf{P}$ divides $\mathbb{R}^d$ into cells, using the power of these points. The cell associated with the i-th generator point, C_i, is defined by

$$C_i = \left\{ x \in \mathbb{R}^d : \operatorname{pow}(x, (p_i, w_i)) \leqslant \operatorname{pow}(x, (p_j, w_j)),\ i \neq j \right\}.$$

Note that if all the weights are equal, the Laguerre tessellation reduces to the standard Voronoi tessellation.

Remark 1. Laguerre tessellations on a locally finite but possibly infinite set of generator points are defined in the same way.

TABLE 1. Tessellation storage format

cell i	cell j	edge $e_{i,j}$			
		$v_1(x)$	$v_1(y)$	$v_2(x)$	$v_2(y)$
1	4	126.14	138.02	140.86	156.07
1	11	98.75	150.55	126.14	138.02
1	14	93.26	164.39	98.75	150.55
2	6	100.97	84.85	107.45	89.69
2	10	107.45	89.69	112.99	113.46
2	11	112.99	113.45	83.46	126.78
⋮	⋮		⋮		⋮

When all weights are positive, each weighted generator point $(p, w) \in \mathbf{P}$ can be interpreted and visualized as a sphere (denoted by $S(p, r)$) with radius $r = \sqrt{w} \geqslant 0$ centered at point p. The power of a point x with respect to the sphere $S(p, r)$ is thus given by

$$\operatorname{pow}(x, S(p, r)) = \|x - p\|^2 - r^2.$$

Geometrically, this means that for a point x outside the sphere $S(p, r)$, the value of $\operatorname{pow}(x, S(p, r))$ is equal to the squared length of the tangent line from x to $S(p, r)$.

The boundary between two adjacent cells generated by spheres $S_1 = S(p_1, r_1)$ and $S_2 = S(p_2, r_2)$, consists of all points $z \in \mathbb{R}^d$ such that $\operatorname{pow}(z, S_1) = \operatorname{pow}(z, S_2)$. These points form a hyperplane $H(S_1, S_2)$, where

$$H(S_1, S_2) = \{z \in \mathbb{R}^d : 2\langle z, p_1 - p_2 \rangle = \|p_1\|^2 - \|p_2\|^2 + r_2^2 - r_1^2\}.$$

This boundary is perpendicular to the line joining p_1 and p_2 and is called the *radical axis* of S_1 and S_2.

In this paper, for simplicity, we will only consider Laguerre tessellations in $\mathbb{R}^2$. In this case, the generators can be interpreted as circles.

2.2. Representations

A Laguerre tessellation can be represented mathematically as a *geometric graph* — a collection of vertices $\mathbf{V} = \{v_1, \ldots, v_m\}$ (also $\mathbf{V}(\mathbf{P})$) and edges $\{(v_i, v_j)\}$, where the vertices are assigned positions in space. Note that some cells are not only bounded by segments but also by rays that extend to infinity in a certain direction, because we consider finite sets of generators. Such edges are often represented using a 'dummy' vertex. That is, one vertex of the edge is set to be an arbitrary point on the ray extending to infinity, see [15]. Interior vertices of the tessellation have equal power with respect to at least three separate circles. In contrast, 'dummy' vertices only have equal power with respect to two circles.

Using the above representation, the vertices and edges of the Laguerre tessellations can be stored in the format given in Table 1. The cells are labeled from 1 to n. The first two columns of Table 1 correspond to the labels of adjacent cells. For example, cell 1 is adjacent to cells 4, 11, and 14. The coordinates of the vertices of the edge that separates two cells are given in columns 3–6. Note that a more compact representation can be achieved by storing the coordinates of each vertex, together with the indices of the adjacent vertices.

A *normal* tessellation is one in which adjacent cells are face-to-face, i.e., (in 2D) they share edges and vertices; furthermore, each edge borders exactly two cells, and each vertex is shared by exactly three cells. An example of a normal 2D Laguerre tessellation, generated by 15 circles, is given in Figure 1. The boundaries between the cells are the edges of the geometric graph. The powers of the points in each boundary are equal with respect to the two neighboring circles. Notice that the degree of each (interior) vertex is 3. The powers of the vertices are therefore equal with respect to three neighboring circles.

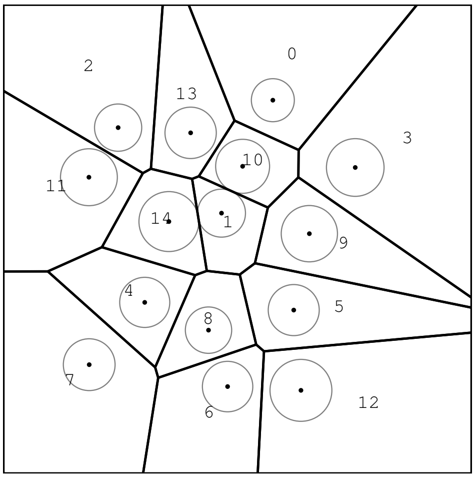

FIGURE 1. Laguerre tessellation for 15 circles

Typical 2D Laguerre tessellations are normal. For example, when the generator points are randomly and uniformly chosen within a bounded sampling window, the tessellation is normal with probability 1. From now on, we consider only normal tessellations.

2.3. Properties

We next discuss some properties of Laguerre tessellations which are important for the problem of inverting Laguerre tessellations.

Property 1. A Laguerre cell does not necessarily contain its generator and a generator does not necessarily generate a cell.

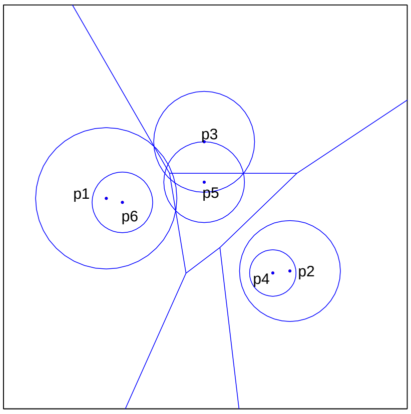

FIGURE 2. Laguerre tessellation for six circles

This property is well known (see, for example, [2]). Figure 2 (from [2, 3]), shows that the generator point of a Laguerre cell can lie outside its cell; in particular, p_4 lies outside cell 4. The same figure shows a generator circle, $S(p_6, r_6)$, for which the corresponding Laguerre cell is empty.

A consequence of Property 1 is that the generator set for a given tessellation is not unique. Namely, one can add circles that do not generate additional cells, and the new set will give the same Laguerre tessellation as the original one.

However, even when each circle generates a cell, the generators of a Laguerre tessellation are not necessarily unique.

Property 2. Two completely different sets of circles can generate the same Laguerre tessellation.

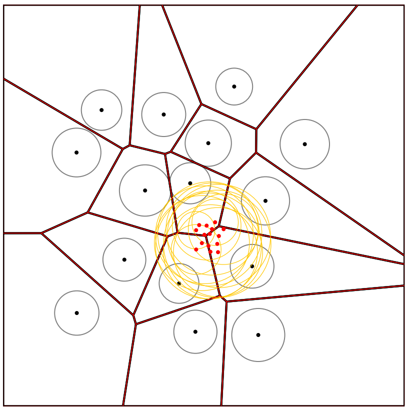

FIGURE 3. One Laguerre tessellation generated by two completely different sets of circles

An extreme example is given in Figure 3, where both the gray circles and yellow circles yield the same tessellation. The method for constructing such generating circles is discussed in Section 3. Property 2 has been mentioned in the literature (see [3, 20]). However, in our view, it has received surprisingly little attention. In particular, it has not been stressed that two entirely different (in both location and radii) sets of circles can generate the same tessellation.

3. PROBLEM DESCRIPTION

The problem of efficiently generating tessellations has been extensively studied in computer science and computational geometry. In this problem, the generators of the tessellation are given, but the tessellation itself is unknown. A less studied problem, but one with extensive applications, is the inverse problem. In this problem, the tessellation is given, but the generators are unknown.

In the case of Voronoi tessellations, the inverse problem has a number of elegant solutions. See, for example, [16, 15, 11, 13, 14]. Unfortunately, these approaches do not easily extend to the Laguerre case. One reason for this is that the inverse Voronoi problem has more structure than the inverse Laguerre problem. Another reason is that the inverse Voronoi problem has a unique solution, whereas the inverse Laguerre problem in general has an infinite number of solutions.

Finding a set of weighted generating points that generate a given Laguerre tessellation is not too difficult; see Algorithm 1 below. However, finding pertinent solutions — that is, more suitable, meaningful solutions — is considerably harder. Pertinent solutions possess extra structure that is imposed by the modeler. Typical requirements arising in materials science (see, e.g., [8, 17]), geometry (e.g., sphere packing [21]), molecular biology and biochemistry (see, e.g., [22]), include the following:

1. The weights of the generating points should be non-negative, so that the weighted points can be interpreted as circles.
2. The generating points should ideally lie within the cells they generate and, if they do not, they should not be too far away.
3. The maximum radius should be as small as possible.
4. The average radius should correspond to a circle whose average volume equals the average volume of a cell.
5. The generating points should be close to the centers of mass of the cells.

3.1. Weighted points that generate a given Laguerre tessellation

We begin by describing how, for a given (normal) tessellation, a set of weighted generator points can be determined by specifying only the coordinates and weight of one weighted generator point and one coordinate of the weighted generator point of a neighboring cell.

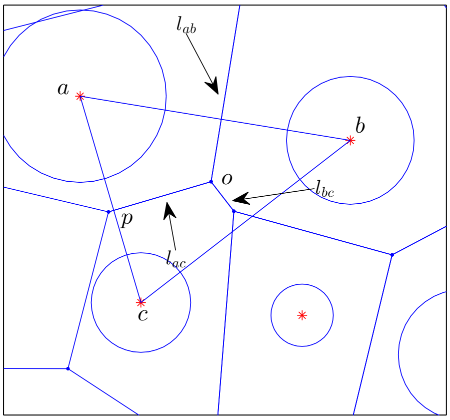

FIGURE 4. Two generators determine the third

Theorem 3.1. *The weighted generator points of a given normal 2D Laguerre tessellation can be entirely determined from the weighted generator point of one interior cell, and one coordinate of the weighted generator point of an adjacent cell.*

Proof. For simplicity we assume that the weights of the generator points are positive, although the proof does not use this assumption — the advantage is that the weighted generator points can be interpreted as circles; see Figure 4. Let $S_1(p_1, r_1)$ be the generator circle of some interior cell, C_1, and let $S_2(p_2, r_2)$ be the generator circle of an adjacent cell, C_2. Let the coordinates of p_1 and p_2 be (x_1, y_1) and (x_2, y_2), respectively. The assumption of the theorem is that x_1, y_1, r_1, and x_2 are given.

Let $e_{1,2}$ be the edge of the tessellation separating C_1 and C_2. For Laguerre tessellations, the line segment connecting p_1 and p_2 is perpendicular to $e_{1,2}$. Let $m_{1,2}$ be the slope of $e_{1,2}$. It follows that y_2 is determined by

$$\frac{y_2 - y_1}{x_2 - x_1} = -\frac{1}{m_{1,2}}. \tag{1}$$

(Note that while it is possible that $m_{1,2} = 0$, this is not relevant in practical applications. The problem can be solved, for example, by rotating the tessellation. Therefore, we assume the slope to be non-zero.) From this we can determine r_2 via

$$\|p_1 - q\|^2 - r_1^2 = \|p_2 - q\|^2 - r_2^2, \tag{2}$$

for any point q on the line containing segment $e_{1,2}$. In particular, we can take $q = p_{1,2}$, the intersection of the line through p_1 and p_2, and the line containing the edge $e_{1,2}$.

Because the tessellation is assumed to be normal and C_1 is an interior cell, there is a cell adjacent to both C_1 and C_2, say C_3. The generating circle of C_3, $S_3 = (p_3, r_3)$, is determined as follows. The point $p_3 = (x_3, y_3)$ is the intersection of (1) the line that passes through p_1 and is perpendicular to $e_{1,3}$ (the edge between C_1 and C_3) and (2) the line that passes through p_2 and is perpendicular to $e_{2,3}$ (the edge between C_2 and C_3). It follows that the coordinates x_3 and y_3 of p_3 satisfy

$$\frac{y_3 - y_1}{x_3 - x_1} = -\frac{1}{m_{1,3}} \quad \text{and} \quad \frac{y_3 - y_2}{x_3 - x_2} = -\frac{1}{m_{2,3}}, \tag{3}$$

where $m_{1,2}$ and $m_{2,3}$ are the slopes of $e_{1,2}$ and $e_{2,3}$, respectively. Hence,

$$\begin{aligned} x_3 &= \frac{m_{2,3}(m_{1,3}y_1 + x_1) - m_{1,3}(m_{2,3}y_2 + x_2)}{m_{2,3} - m_{1,3}} \\ y_3 &= \frac{m_{1,3}y_1 + x_1 - (m_{2,3}y_2 + x_2)}{m_{1,3} - m_{2,3}}. \end{aligned} \tag{4}$$

The radius, r_3, is then determined as in (2); that is,

$$\|p_1 - q\|^2 - r_1^2 = \|p_3 - q\|^2 - r_3^2, \tag{5}$$

where q is any point on the line containing the edge $e_{1,3}$.

It is also possible to determine r_3 by considering the pair C_2, C_3 rather than C_1, C_3, giving

$$\|p_2 - q\|^2 - r_2^2 = \|p_3 - q\|^2 - r_3^2, \tag{6}$$

where q is any point on the line containing the edge $e_{2,3}$. To prove that (5) and (6) give the same value for r_3^2, it suffices to show that

$$r_1^2 - \|p_1 - u\|^2 = r_2^2 - \|p_2 - u\|^2, \tag{7}$$

where u is the vertex on the intersection of the lines containing edges $e_{1,2}$ and $e_{2,3}$. But this follows directly from the definition of the Laguerre tessellation, and the fact that u lies also on the line through edge $e_{1,2}$.

Proceeding in this fashion, it is possible to iteratively determine the generator circle of each cell. □

Remark 2. Theorem 3.1 was given in 2D, but the idea of the proof is applicable for higher dimensions. Using normal vectors for hyperplanes separating cells, e.g., edges in 2D or planar faces in 3D, it is clear that generators have to lie on lines with the same orientation. See also [20], where a related non-unique construction is given for the orthogonal dual of a Laguerre tessellation (in the Voronoi case this corresponds to the Delaunay triangulation).

Remark 3. The proof of Theorem 3.1 shows that if we add the *same* constant to all squared radii, the tessellation does not change. Namely, if $\|p_i - q\|^2 - r_i^2 = \|p_j - q\|^2 - r_j^2$ for every q on an edge separating adjacent cells C_i and C_j, then the same is true if r_i^2 and r_j^2 are replaced by $r_i^2 + c$ and $r_j^2 + c$. As a consequence, it is always possible to find a set of generators that all have positive weights. If some weights are negative, we simply find the minimum of these and subtract this value from all weights.

Theorem 3.1 and Remark 3 suggest the following algorithm for determining the generator circles of a Laguerre tessellation, given x_1, y_1, r_1 and x_2. Note that in the algorithm we start with a pair of (internal) cells (C_1, C_2). This may be replaced with any pair of internal cells by relabeling.

Algorithm 1 Generator Construction

```
Input: $x_1, y_1, r_1 (\geqslant 0), x_2$ and tessellation data such as
    in Table 1.
Output: a set of generators $\mathbf{P} = \{(p_k, r_k^2)\}$ with non-
    negative minimum radius.
Initialize $\mathbf{P} = \{(p_1, r_1^2), \ldots, (p_n, r_n^2)\}$ to $\mathtt{NAN}_{n\times 3}$.
Compute $y_2$ and $r_2^2$ using (1) and (2). Flag that $C_1$
    and $C_2$ have been assigned generators.
while not all cells have generators do
    for $k = 1 : n$ do
        if $(p_k, r_k^2)$ has not been assigned and more than
        two of its adjacent cells have been assigned
        then
            Choose two of the adjacent cells of $C_k$ with
            assigned generators.
            Compute $p_k$ using equation (3).
            Compute $r_k^2$ using equation (2).
        end if
    end for
end while
if $\min\{r_k^2\} < 0$ then
    set $\{r_k^2\} = \{r_k^2\} - \min\{r_k^2\}$
end if
```

3.2. Choosing a pertinent solution

Algorithm 1 provides a method for recovering the generating circles of a tessellation given a small number of inputs, namely x_1, y_1, r_1, and x_2. However, depending on how these inputs are chosen, very different results can be obtained. Although the algorithm guarantees positive weights, in some cases the generating circles can lie far outside the cells they generate. An example is given in Figure 3. In other cases, the radii of the generating circle can be much bigger than the cells or even the observation window; see Figure 5.

As stated above, many applications based on Laguerre tessellations ascribe meaning to the generating circles. For this reason, it is important that we have a method that chooses a solution that satisfies a number of criteria such as those listed above. Some of these criteria are model dependent. Others, however, are fairly universal. In particular, it is almost always desirable to have generating points lie within the cells they generate and it is almost always desirable to have real-valued radii.

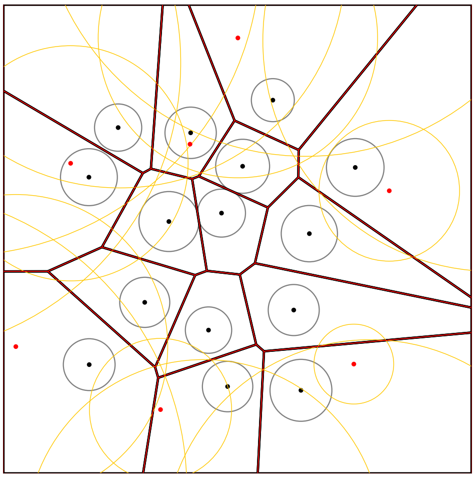

FIGURE 5. Another example of a Laguerre tessellation that is generated by two completely different sets of circles.

We encode these universal criteria into our algorithm by requiring that the generating circles produced all lie within their cells and all have real-valued radii. We call the set of points that satisfy these two criteria $\mathcal{P}$. We then choose the inputs to Algorithm 1, (x_1, y_1, r_1, x_2), so that the resulting set of generators, $\mathbf{P}$, belongs to $\mathcal{P}$.

In addition to requiring that the set of generator points lies in $\mathcal{P}$, we also require that the generators satisfy an additional optimality criterion that is determined by the application. In the following, we choose to solve

$$\min_{\mathbf{P}\in\mathcal{P}} \max_{r\in\mathbf{P}} r. \qquad (8)$$

That is, we wish to find the generator set $\mathbf{P} \in \mathcal{P}$ for which the maximum radius is minimal.

However, the method we propose is more general. We could, for example, seek to minimize the difference between the average circle area and the average cell area, or minimize the average distance of a generating point from the center of mass of its cell.

Problem (8) is an optimization problem with complicated constraints and many local optima. As a result, it is a difficult problem to solve numerically. In particular, numerical solutions based on local search are very dependent on the choice of initial conditions. Small changes in initial conditions can result in very different generator sets.

Stochastic algorithms tend to outperform deterministic approaches in such settings. In particular, these algorithms are able to escape many local optima and are not so sensitive to initial conditions. We use a stochastic global optimization technique, the cross-entropy (CE) method, which is both effective and straightforward to implement.

4. CE METHOD FOR INVERTING LAGUERRE TESSELLATIONS

The CE method has been successfully applied to many complicated integer non-linear programming and continuous multi-extremal optimization (see, for example, [23]). In this section we describe how the method can be used to determine pertinent (meaningful) Laguerre generators.

The idea behind the CE method is that the global optimum can be described by a degenerate probability density (that is, a density that ascribes all its mass to a single point). The algorithm iteratively generates a sequence of probability densities that converge to this degenerate density. In general, the densities are chosen to be normal densities. This is because normal densities allow for the whole parameter space to be explored, are easily updated, and converge to degenerate densities as their variances go to zero. For more details on the method we refer to [19] and [24]. The convergence properties of the Cross-Entropy method are discussed in [23] and the references given therein.

For the minimization problem (8), we wish to find a generator set $\mathbf{P}^*$ in the collection all feasible sets $\mathcal{P}$, that attains the minimum maximum radius:

$$\gamma^* = \min_{\mathbf{P}\in\mathcal{P}} \max_{r\in\mathbf{P}} r,$$

where $\mathbf{P}$ is generated via Algorithm 1. Since we are interested in the minimizer $\mathbf{P}^*$ rather than the minimum γ^*, we can, in view of Remark 3, reduce the dimension of this constrained minimization problem from four to three, by setting $r_1 = 0$ in the input of Algorithm 1.

The CE algorithm now comprises the following iterative steps:

1. Generate a random sample of starting values (x_1, y_1, x_2) according to a multi-variate normal distribution.
2. For each of these starting values generate the set of generators via Algorithm 1 and compute the corresponding maximum radii.
3. Update the parameters of the sampling distribution, based on the best performing samples (the so-called *elite samples*), using cross-entropy minimization. The best performing samples are those with the smallest maximum radii.

The CE method produces a sequence of parameters $\{(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2)\}$ for the multivariate normal sampling distribution and a sequence of levels $\{\gamma_t\}$ decreasing to γ^*. To run the CE method, we need to specify the initial sampling parameters, the sample size N, a rarity parameter ϱ, and a stopping criterion.

At iteration t we generate N independent samples $(x_1^{(k)}, y_1^{(k)}, x_2^{(k)})$, $k = 1, \ldots, N$ from a three-dimensional normal distribution, where each component is sampled independently. We write $(x_1^{(k)}, y_1^{(k)}, x_2^{(k)}) \sim_{\text{iid}} \mathsf{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2)$, where $\boldsymbol{\mu}_t$ is the three-dimensional vector of expectations and $\boldsymbol{\sigma}_t^2$ is the three-dimensional vector of variances. For the normal distribution the CE updating rule in Step 3 above is particularly easy (see [24]): the expectation vector becomes the vector of sample means of the elite samples, and the vector of variances becomes the vector of sample variances of the elite samples.

The detailed steps of the CE method for inverting Laguerre tessellations are as follows.

Algorithm 2 Generators of minimax radius

```
Input: Sample size $N$, rarity parameter $\varrho$, and initial
    sampling parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\sigma}_0^2$.
Output: The minimizer, $\mathbf{P}^*$, of (8).
Set the iteration counter to $t = 1$.
while stopping criterion is not met do
    For $k = 1, \ldots, N$, set $r_1^{(k)} = 0$ and draw $(x_1^{(k)}, y_1^{(k)}, x_2^{(k)}) \sim_{\text{iid}} \mathsf{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\sigma}_{t-1}^2)$ using the following
        acceptance–rejection step: compute $y_2^{(k)}$, $x_3^{(k)}$, $y_3^{(k)}$, $x_4^{(k)}$, and $y_4^{(k)}$. If the four generators lie in
        their cells, then accept $(x_1^{(k)}, y_1^{(k)}, x_2^{(k)})$, otherwise reject.
    Compute $\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \ldots, \mathbf{P}^{(N)}$ via Algorithm 1. Find the maximum radius $l^{(k)}$ for each set $\mathbf{P}^{(k)}$.
    Order the maximum radii from smallest to largest. Let $\gamma_t$ be the largest of the $\lceil \varrho N \rceil$ smallest
        radii. Let the elite set be the collection of samples $(x_1^{(k)}, y_2^{(k)}, x_2^{(k)})$ for which the corresponding
        generator set $\mathbf{P}^{(k)}$ has a maximum radius less than or equal to $\gamma_t$.
    Update the parameters $\boldsymbol{\mu}_t$ and $\boldsymbol{\sigma}_t^2$ as the sample mean and sample variance (component-wise) of
        the elite samples, and set $t = t + 1$.
end while
```

REMARK 4. Algorithm 2 uses an acceptance–rejection step to constrain the first four generators to lie within their cells. We found experimentally that this is sufficient to ensure that eventually all generators end up within their cells. It is difficult to initially constrain *all* generators inside their cells, as the search space will become very small. As far as we are aware, it remains an open theoretical problem whether every Laguerre tessellation admits a solution in which every cell contains one generator point.

A possible stopping criterion (which we have used in Section 5) is to stop when the maximum standard deviation, $\max(\boldsymbol{\sigma}_t)$, is less than some small tolerance $\varepsilon > 0$.

5. NUMERICAL EXPERIMENTS

In this section, two numerical examples are given to indicate how Algorithm 2 works for inverting Laguerre tessellations. The tessellation data is of the format in Table 1 and can be downloaded from `http://www.maths.uq.edu.au/~kroese/Laguerre`. Matlab and Java code is available on request.

TABLE 2. Example 1: Convergence of parameters

		$\boldsymbol{\mu}_t$			
t	γ_t				$\max\{\boldsymbol{\sigma}_t\}$
1	31.7198	93.4043	109.6592	122.2072	4.6120
2	21.6147	93.0674	110.9199	123.5051	1.3520
3	13.5177	93.0501	111.1407	123.7959	0.4514
4	9.9260	93.1264	111.1727	123.9915	0.1348
5	8.8108	93.1707	111.1562	124.0233	0.0553
6	8.5782	93.1934	111.1419	124.0357	0.0368
7	8.5237	93.2111	111.1350	124.0412	0.0221
8	8.4993	93.2210	111.1309	124.0441	0.0108
9	8.4857	93.2271	111.1286	124.0456	0.0062
10	8.4782	93.2301	111.1274	124.0466	0.0034
11	8.4742	93.2316	111.1267	124.0469	0.0020
12	8.4719	93.2325	111.1263	124.0471	0.0012
13	8.4706	93.2331	111.1260	124.0472	0.0007
14	8.4699	93.2334	111.1259	124.0473	0.0004
15	8.4694	93.2335	111.1258	124.0473	0.0002
16	8.4692	93.2336	111.1257	124.0473	0.0001
17	8.4690	93.2336	111.1257	124.0473	0.0001

Example 1

The first example is the running example, whose original generators are shown in Figure 1. In Figures 3 and 5, we saw examples of poorly behaved generators, obtained via Algorithm 1. We now employ Algorithm 2 to obtain a good generator set, whose maximum radius is as small as possible.

In this example, we use cells 2 and 6 as our starting cells (instead of 1 and 2; note that cell 1 is not internal here). We choose $\boldsymbol{\mu}_0 = (\mu_0^{x_1}, \mu_0^{y_1}, \mu_0^{x_2})$ as follows, where $\mu_0^{x_1}$ is the average of the x-coordinates of all vertices of C_1, $\mu_0^{y_1}$ is the average of the y-coordinates of all vertices of C_1, and $\mu_0^{x_2}$ is the average of the x-coordinates of all vertices of C_2. The three initial variances are set to 100 (standard deviation 10), so that most of samples lie inside the cells C_2 and C_6.

Table 2 shows the progress of the CE method, where the sample size $N = 4000$, rarity parameter $\varrho = 0.05$ and stopping criterion $\texttt{MAX}(\boldsymbol{\sigma}) < \varepsilon = 10^{-4}$.

In Table 2, γ_t is the $\lceil N\varrho \rceil$-th largest radius over all maximum radii of each set in t-th iteration. Note that the minimum radius of each set is 0.

Figure 6 indicates the difference between our generators and the original ones. While the generator points are very similar, the radii are significantly smaller for the CE generators. The maximum radius reduces from 9.5034 to 8.4690. Also, the minimum radius for the CE case is (always) 0.

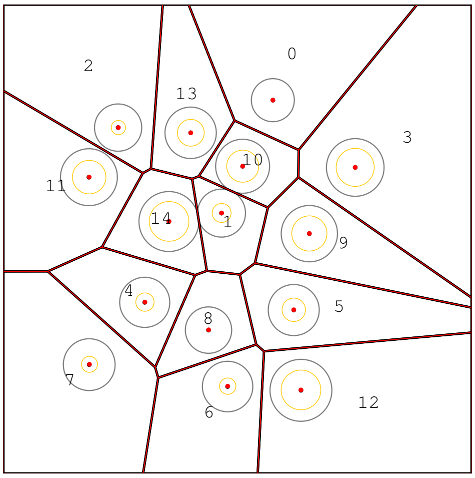

FIGURE 6. Result of Example 1

To show the dependence of the CE algorithm on the parameters, we ran Algorithm 2 with different sets of parameters, such as $N = 4000, \varrho = 0.1$, $N = 2000, \varrho = 0.05$ and $N = 2000, \varrho = 0.1$. The progress of γ_t as a function of t is shown in Figure 7. We see a robust behavior with respect to the choice of CE parameters.

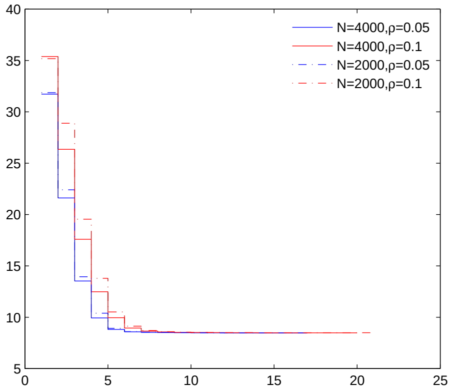

FIGURE 7. CE method for Example 1

Example 2

In Figure 8, we give a relatively complicated Laguerre tessellation, with 88 cells for which the generators need to be recovered. The original generators are also given in Figure 8.

TABLE 3. Example 2: Convergence of parameters

t	γ_t	$\boldsymbol{\mu}_t$			$\max\{\boldsymbol{\sigma}_t\}$
1	31.9997	142.7428	57.1691	131.5233	2.2070
2	17.7478	142.3349	57.4951	131.2341	0.5303
3	11.8451	142.2002	57.5020	131.1021	0.2092
4	9.6158	142.1381	57.5472	131.0550	0.0712
5	9.1285	142.1171	57.6062	131.0436	0.0319
6	9.0075	142.1075	57.6289	131.0378	0.0149
7	8.9754	142.1051	57.6437	131.0365	0.0065
8	8.9624	142.1044	57.6500	131.0362	0.0031
9	8.9568	142.1041	57.6535	131.0362	0.0015
10	8.9539	142.1041	57.6553	131.0363	0.0008

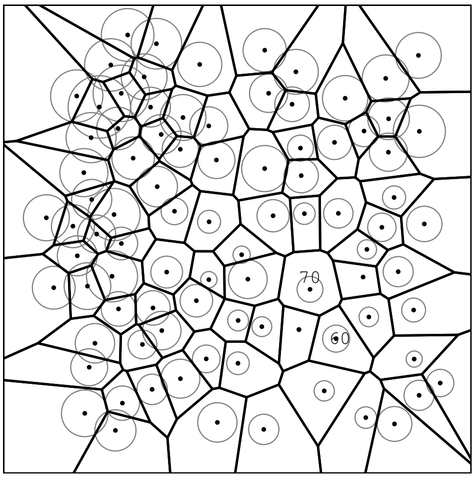

FIGURE 8. Laguerre tessellation for 88 generators

We use cells 60 and 70 as starting cells and set the initial distribution parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\sigma}_0$ in the same way as Example 1. As for the parameters of the CE method, we set sample size $N = 5000$, rarity parameter $\varrho = 0.01$ and stopping criterion $\max(\boldsymbol{\sigma}) < \varepsilon = 10^{-3}$. The progress of the algorithm is shown in Table 3.

Figure 9 shows the behavior of the method under different sets of CE parameters. Even though the processes are a little different, the results are almost the same.

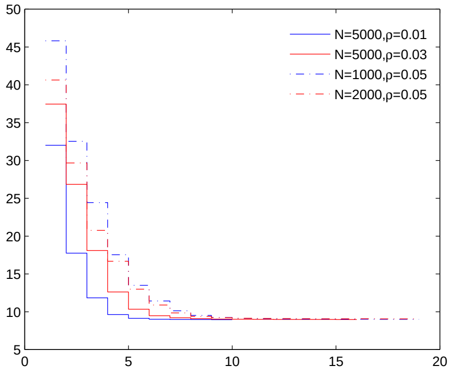

FIGURE 9. CE method for Example 2

The original and CE generators are given in Figure 10, showing very good agreement in location, with again slightly smaller radii for the CE case.

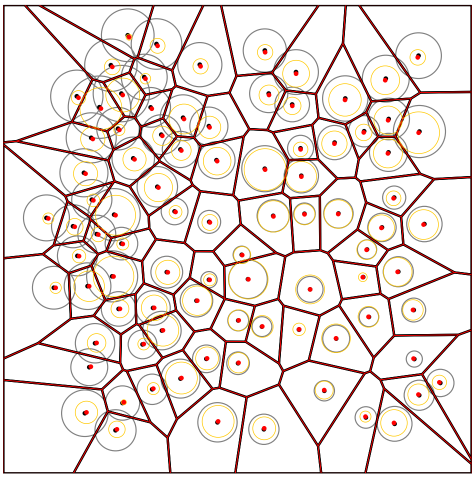

FIGURE 10. Result of Example 2

Remark 5. The Voronoi tessellation is a special case of the Laguerre tessellation. In the Voronoi case, all the generator weights are equal (and can be assumed to be 0). Algorithm 2 is easily adapted to invert 2D Voronoi tessellations.

As in Algorithm 2, we start from two initial cells, C_1 and C_2. We generate sample points in C_1 and find the corresponding points in C_2 using the perpendicular bisector. We set the radii of the generating circles in C_1 and C_2 to be 0. We then use the CE method to find points such that the maximum radius of a generating circle is zero. Note, however, that [15] gives a more efficient method for inverting Voronoi tessellations.

6. CONCLUSIONS AND PROPOSED EXTENSIONS

In this paper, we have provided a general method for inverting normal Laguerre tessellations. We have demonstrated that there are often a number of different solutions to this inversion problem. Many of these solutions have limited explanatory power. We proposed a method, based on randomized optimization, for finding pertinent solutions. That is, solutions that satisfy certain criteria imposed by the application. We have given a number of numerical results that demonstrate the effectiveness of this approach.

There are many possible extensions to our work. Our method can be extended to higher dimension. Our method can be applied in a number of fields that use Laguerre tessellations as models, for example materials science and biology. We also believe that the cross-entropy method could also be applied to other tessellations, including those with non-convex cells.

ACKNOWLEDGEMENTS

This work was supported by the Australian Research Council under grant number DP140101956.

REFERENCES

[1] Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. (2000) *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd edition. J. Wiley & Sons, Chichester.

[2] Aurenhammer, F. (1987) Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing*, **16**, 78–96.

[3] Lautensack, C. (2007) Random Laguerre Tessellations. PhD thesis Karlsruhe Institue of Technology.

[4] Tanuma, T., Imai, H., and Moriyama, S. (2011) Revisiting hyperbolic Voronoi diagrams in two and higher dimensions from theoretical, applied and generalized viewpoints. *Transactions on Computational Science XIV: Special Issue on Voronoi Diagrams and Delaunay Triangulation*, pp. 1–30. Springer, New York.

[5] Telley, H., Liebling, T. M., and Mocellin, A. (1996) The Laguerre model of grain growth in two dimensions I. Cellular structures viewed as dynamical Laguerre tessellations. *Philosophical Magazine B*, **73**, 395–408.

[6] Telley, H., Libeling, T. M., and Mocellin, A. (1996) The Laguerre model of grain growth in two dimensions II. Examples of coarsening simulations. *Philosophical Magazine B*, **73**, 409–427.

[7] Lautensack, C. (2008) Fitting three-dimensional Laguerre tessellations to foam structures. *Journal of Applied Statistics*, **35**, 985–995.

[8] Wejrzanowski, T., Skibinski, J., Szumbarski, J., and Kurzydlowski, K. (2013) Structure of foams modeled by Laguerre-Voronoi tessellations. *Computational Materials Science*, **67**, 216–221.

[9] Ogawa, H., Wakai, F., and Waseda, Y. (1996) Molecular dynamics simulation of the model grain boundary structure of polycrystalline materials. *Molecular Simulation*, **18**, 179–192.

[10] Redenbach, C., Shyklar, I., and Andrä, H. (2012) Laguerre tessellations for elastic stiffness simulations of closed foams with strongly varying cell sizes. *International Journal of Engineering Science*, **50**, 70–78.

[11] Ash, P. F. and Bolker, E. D. (1985) Recognizing Dirichlet tessellations. *Geometriae Dedicata*, **19**, 175–206.

[12] Evans, D. G. and Jones, S. M. (1987) Detecting Voronoi (area-of-influence) polygons. *Mathematical geology*, **19**, 523–537.

[13] Aurenhammer, F. (1987) Recognising polytopical cell complexes and constructing projection polyhedra. *Journal of Symbolic Computation*, **3**, 249–255.

[14] Hartvigsen, D. (1992) Recognizing Voronoi diagrams with linear programming. *ORSA Journal on Computing*, **4**, 369–374.

[15] Schoenberg, F. P., Ferguson, T., and Li, C. (2003) Inverting Dirichlet tessellations. *The Computer Journal*, **46**, 76–83.

[16] Adamatzky, A. (1993) Massively parallel algorithm for inverting Voronoi diagram. *Neural Netw. World*, **5**, 385–92.

[17] Lyckegaard, A., Lauridsen, E. M., Ludwig, W., Fonda, R. W., and Poulsen, H. F. (2011) On the use of Laguerre tessellations for representations of 3D grain structures. *Advanced Engineering Materials*, **13**, 165–170.

[18] Spettl, A., Werz, T., Krill, C. E., and Schmidt, V. (2014) Parametric representation of 3D grain ensembles in polycrystalline microstructures. *Journal of Statistical Physics*, **154**, 913–928.

[19] Rubinstein, R. Y. and Kroese, D. P. (2004) *The Cross-Entropy method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning.* Springer, New York.

[20] Aurenhammer, F. (1987) A criterion for the affine equivalence of cell complexes in $\mathbb{R}^d$ and convex polyhedra in $\mathbb{R}^{d+1}$. *Discrete and Computational Geometry*, **2**, 49–64.

[21] Gervois, A., Oger, L., Richard, P., and Troadec, J. P. (2002) Voronoi and radical tessellations of packings of spheres. *Computational Science – ICCS 2002,* Amsterdam, 21-24 August, pp. 95–104. Springer, Berlin.

[22] Sadoc, J., Jullien, R., and Rivier, N. (2003) The Laguerre polyhedral decomposition: application to protein folds. *The European Physical Journal B-Condensed Matter and Complex Systems*, **33**, 355–363.

[23] Botev, Z. I., Kroese, D. P., Rubinstein, R. Y., and L'Ecuyer, P. (2013) The Cross-Entropy method for optimization. *Machine Learning: Theory and Applications, V. Govindaraju and C.R. Rao, Eds, Chennai: Elsevier B.V.*, **31**, 35–59.

[24] Kroese, D. P., Porotsky, S., and Rubinstein, R. Y. (2006) The Cross-Entropy method for continuous multi-extremal optimization. *Methodology and Computing in Applied Probability*, **8**, 383–407.