

INTERACTIVE LAND-USE OPTIMIZATION USING LAGUERRE VORONOI DIAGRAM WITH DYNAMIC GENERATING POINT ALLOCATION

S. Chaidee^a, P. Pakawanwong^b, V. Suppakitpaisarn^{b,c,*}, P. Teerasawat^d

^a Graduate School of Advanced Mathematical Sciences, Meiji University, Nakano, Tokyo, Japan, 164-8525 - schaidee@meiji.ac.jp

^b Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo, Tokyo, Japan, 113-0033
(prompong, vorapong)@is.s.u-tokyo.ac.jp

^c JST, ERATO, Kawarabayashi Large Graph Project, Japan

^d International Program in Architecture and Urban Design, Graduate School of Science and Technology, Meiji University,
Nakano, Tokyo, Japan, 164-8525 - p.teerasawat@gmail.com

Commission IV, WG IV/3

KEY WORDS: Spatiotemporal Data Services and Knowledge Services, Land-Use Optimization Problem, Laguerre Voronoi Diagram, Force-Directed Graph Drawing, Linear Programming, Interactive Design

ABSTRACT:

In this work, we devise an efficient method for the land-use optimization problem based on Laguerre Voronoi diagram. Previous Voronoi diagram-based methods are more efficient and more suitable for interactive design than discrete optimization-based method, but, in many cases, their outputs do not satisfy area constraints. To cope with the problem, we propose a force-directed graph drawing algorithm, which automatically allocates generating points of Voronoi diagram to appropriate positions. Then, we construct a Laguerre Voronoi diagram based on these generating points, use linear programs to adjust each cell, and reconstruct the diagram based on the adjustment. We adopt the proposed method to the practical case study of Chiang Mai University's allocated land for a mixed-use complex. For this case study, compared to other Voronoi diagram-based method, we decrease the land allocation error by 62.557%. Although our computation time is larger than the previous Voronoi-diagram-based method, it is still suitable for interactive design.

1. INTRODUCTION

Land-use optimization problem (LUOP) is one of the most important problems in computational landscape design. In this problem, we want to divide a land into several parts. Each part must have a specific area, and two adjacent parts must have a strong compatibility. There are many methods proposed to solve the problem. Most of them are based on combinatorial optimization techniques such as simulated annealing (Santé-Riveira et al., 2008), tabu search (Qi et al., 2008), genetic algorithms (Karakostas and Economou, 2014), and a hybrid of those methods (Mohammadi et al., 2016).

Those combinatorial optimization algorithms can output a very satisfiable output for the problem in less than 100 hours. When the problem inputs, which include many design issues, are decided beforehand, this computation time is acceptable. However, in many cases, the design process needs to be interactive (Michalek and Papalambros, 2002). Landscape architects usually cannot design the inputs beforehand. They have a rough idea and use that rough idea as an input of an optimization process. After that, they will iteratively improve their inputs based on the outputs they have got from the process. To have a successful interactive design, our optimization process cannot be longer than a few seconds. We have to consider a method that is much faster than those combinatorial optimization techniques.

For this propose, we can use techniques based on computational geometry techniques such as Voronoi diagram (Karimi et al., 2009), multiplicatively weighted Voronoi diagram (Karimi et al., 2009, Dong, 2008), adaptive multiplicatively weighted Voronoi diagram

(Reitsma et al., 2007), and LaguerreVoronoi diagram (Song et al., 2015). In this paper, we choose to develop our methods based on the Laguerre-Voronoi diagram because of two reasons. First, the boundaries of each cell in the diagram are straight lines (Imai et al., 1985), which are more desirable in practice than curves obtained from the multiplicatively weighted Voronoi diagrams. Second, we can adjust the weight of cells based on their area requirements. We can give a larger weight to a cell that requires a larger area, and give a small weight to the others.

1.1 Our Contribution

All Voronoi diagram-based approaches assume that generating points, which denote centers of all cells, are given as inputs. Indeed, those point positions are not given in many cases. Placing the points in inappropriate positions results in a partition of which cells do not have an expected area or pair of adjacent cells are not compatible with each other.

In this paper, we propose a method to place the generating points to the best positions as possible.

Our inputs include a network of cells, where two cells are adjacent in the network if they are compatible with each other. Then, we use a force-directed graph drawing algorithm (Bannister et al., 2012) to place the cell in a plane. By the algorithm, a pair of cells that is not compatible with each other is pushed to be apart from each other, and a pair of compatible cells is placed at a close position.

Based on the generating point positions obtained from the graph drawing algorithm, we construct a Laguerre Voronoi diagram in

*Corresponding author

a way that the area of each cell is not so different from the desired area, and the cells are adjacent as desired. We formulate the optimization problem to search the Laguerre Voronoi diagram satisfying the properties which is the multiobjective optimization problem.

We adopt our proposed method to the practical case study of Chiang Mai University's allocated land. The land is called as CMU square, and its area is 140,800 sq.m. (Prachachat Online, 2015). In 2015, the university plan to develop the land into a mixed-use complex, which contains areas of arts, culture, nature, and community. Although the land allocation has been finalized, we think that the compatibility between adjacent areas can be improved.

We use the center of each cell in the finalized land allocation as our baseline generating point positions. Compared to the baseline, we decrease the land allocation error, which indicates how much the areas of each cell are different from the desired areas, by 62.557%. In addition to that, the compatibility value is increased by 28.010%.

Our computation time is 839.78 s in Matlab. Because computation times in Matlab is usually more than a hundred times larger than those in C or C++ (Andrews, 2012), we strongly believe that our computation time will reduce to a few seconds when our software is implemented in those languages. Hence, the algorithm is fast enough for the interactive design.

1.2 Paper Organization

This paper is organized as follows: In Section 2, we will explain the techniques which we use in this work, Laguerre Voronoi diagram and force-directed graph drawing. Then, in Section 3, we will formally state our problem definition. In Section 4, we will give our methods, and, in Section 5, we will show the experimental results of our case study. We then conclude the paper in Section 6.

2. PRELIMINARIES

2.1 Laguerre Voronoi Diagram

The concepts of Voronoi diagram have been widely studied in many areas as compiled in (Okabe et al., 2009). There are many generalizations of Voronoi diagram. One of the generalizations is to put weights to the generators of the Voronoi diagram such as additively and multiplicatively weighted Voronoi diagram. In this study, we mainly focus on the use of Laguerre Voronoi diagram, the special case of the additively weighted Voronoi diagram, which was introduced by (Imai et al., 1985, Aurenhammer, 1987). Let $S = \{c_1, \dots, c_n\}$ be a set of generating circles in \mathbb{R}^2 such that $c_i = (\mathbf{p}_i, r_i)$, where $\mathbf{p}_i = (x_i, y_i)$ is the position of generator and r_i is its weight, which is interpreted as a circle in the plane. The *Laguerre distance* (or recognized as *power distance*) is defined by the distance

$$d_L(\mathbf{p}, c_i) = \|\mathbf{p} - \mathbf{p}_i\|^2 - r_i^2. \quad (1)$$

For each adjacent circles c_i and c_j , Laguerre bisector is defined by

$$B_L(c_i, c_j) = \{\mathbf{p} : d_L(\mathbf{p}, c_i) = d_L(\mathbf{p}, c_j)\} \quad (2)$$

which is a straight line in \mathbb{R}^2 . The half space of p_i dominated by the generator p_j is

$$H(p_i, p_j) = \{\mathbf{p} \in \mathbb{R}^2 : d_L(\mathbf{p}, c_i) \leq d_L(\mathbf{p}, c_j)\},$$

and finally the Laguerre Voronoi region is defined by $L(p_i) = \bigcap_{j \in I_n \setminus \{i\}} H(p_i, p_j)$. Therefore, the Laguerre Voronoi diagram can be constructed from the set of region $\mathcal{L} = \{L(p_1), \dots, L(p_n)\}$ including their boundaries. The properties were widely studied as shown in (Okabe et al., 2009), and the robust algorithm was presented in (Sugihara, 2000).

In the case of Laguerre Voronoi diagram, remark that cells of the corresponding generators may be lost, or generating points may not be contained in the Voronoi cell as mentioned in (Reitsma et al., 2007). However, (Cheng et al., 2000) gave the necessary condition for keeping the generator positions lay inside the cell by the following theorem.

Theorem 1. (Cheng et al., 2000) For a set S of points, let $w(p) \geq 0$ be the weight of point $p \in S$. If, for all q ,

$$d(p, q) \geq 2 \cdot w(p),$$

then no cell of the Laguerre Voronoi diagram is lost, and each point of S lays inside its cell.

Remark that the Laguerre Voronoi diagram has its dual structure called the Laguerre triangulation, which can be considered as a planar graph. If a cell $L(p_i)$ is adjacent to cell $L(p_j)$, there exists an edge $\{p_i, p_j\}$ in the graph whose the vertex set is $\{p_1, \dots, p_n\}$.

2.2 Force-Directed Graph Drawing

Graph drawing is a technique to visualize a given network $(V = \{v_1, \dots, v_n\}, E)$ on a 2-dimensional plane by a map function $f : V \rightarrow \mathbb{R}^2$. To find the best visualization for each network, one can define energy for all map functions, and search for a function that minimizes the defined energy. To have generating point positions that satisfies Theorem 1, in this paper, we choose the energy function defined in (Kamps et al., 1995).

The energy of each function is defined based on two types of energy. Given a parameter α , we want the Euclidean distance between v_i and v_j to be $\alpha \cdot d(v_i, v_j)$ when $d(v_i, v_j)$ is a shortest path length between v_i and v_j in (V, E) . The energy between v_i and v_j , which is defined by the expected Euclidean distance, can be defined as follows:

$$\rho_{i,j}(f) := \left(1 - \frac{D(f(v_i), f(v_j))}{\alpha \cdot d(v_i, v_j)}\right)^2, \quad (3)$$

where $D(f(v_i), f(v_j))$ is a Euclidean distance between $f(v_i)$ and $f(v_j)$. The energy $\rho_{i,j}(f)$ is minimized when the distance $D(f(v_i), f(v_j))$ is equal to the expected distance $\alpha \cdot d(v_i, v_j)$, and the energy becomes large when the difference between two distances becomes large.

The first type of energy for mapping function f , defined as $\rho(f)$, can be defined as follows:

$$\rho(f) = \sum_{i=1}^n \sum_{j=1}^n \frac{\rho_{i,j}(f)}{d(v_i, v_j)^2} \quad (4)$$

The second type of energy comes from the fact that we draw each node v_i as a rectangle on the plane, and we want to have an overlap area between two rectangles to be as small as possible. We assume that the size of a rectangle for v_i is given as s_i , and $Rec(f(v_i), s_i)$ be a rectangle centered at $f(v_i)$ with size

s_i . Given a parameter ω , the energy between v_i and v_j , which is defined by the overlap area, can be defined as follows:

$$\gamma_{i,j}(f) = \omega \cdot |Rec(f(v_i), s_i) \cap Rec(f(v_j), s_j)| \quad (5)$$

The second type of energy, $\gamma(f)$, is defined as follows:

$$\gamma(f) = \sum_{i=1}^n \sum_{j=1}^n \gamma_{i,j}(f) \quad (6)$$

We are now ready to define the energy function, denoted by $E(f)$, in the following equation:

$$E(f) = \rho(f) + \gamma(f) \quad (7)$$

In most graph drawing settings, there are some points with given fixed positions. Let $V' \subseteq V$ be the set, and $f : V' \rightarrow \mathbb{R}^2$ be a function that give those fixed positions. The set of possible function, denoted by \mathcal{F} , is

$$\mathcal{F} = \{f : V \rightarrow \mathbb{R}^2 : f(v) = f'(v) \text{ for all } v \in V'\}.$$

Because the function E is differentiable at most points, to find $f \in \mathcal{F}$ such that $E(f)$ is minimized, it is shown in the paper that the gradient descent method can be used.

2.3 Nelder-Mead Method for Optimization

The Nelder-Mead method is a technique for solving the optimization problem proposed by Nelder and Mead (Nelder and Mead, 1965), which is known as a derivative-free optimization method. In brief, for a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ which is supposed to find the minimum of $f(\mathbf{X})$ where $\mathbf{X} \in \mathbb{R}^d$, a simplex consisting of $d + 1$ vertices of dimension d is chosen. In the optimization processes, the simplex is expected to move and converge to the local minimum.

From a set of simplex vertices $\{\mathbf{x}_1, \dots, \mathbf{x}_{d+1}\}$, the set $\{f(\mathbf{x}_i)\}_{i=1}^{d+1}$ is ordered and relabelled with respect to the sequence $f(\mathbf{x}_1) \leq \dots \leq f(\mathbf{x}_{d+1})$. After ordering and relabelling, \mathbf{x}_1 is said to be the *best* vertex and \mathbf{x}_{d+1} the *worst* vertex. From the worst vertex, we try to replace the worst vertex with a new vertex which is acquired from the *reflection*, *expansion*, or *contraction* through the direction of the centroid among the remaining points, or *shrinkage* all vertices (except for the best vertex) to the direction of the best vertex. The parameters for reflection, expansion, contraction, and shrinkage are written as $\{\alpha, \beta, \gamma, \delta\}$, and normally set as $\{\alpha, \beta, \gamma, \delta\} = \{1, 2, 1/2, 1/2\}$.

It is remarkable that the Nelder-Mead method is easy to implement. The convergent properties of the Nelder-Mead in low dimension was proved in (Lagarias et al., 1998).

3. PROBLEM DEFINITION

In this section, we will give a formal definition for our LUOP. We assume that a designer gives us a landscape object of region (V, E) . Each $v_i \in V$ denotes an object, and, for any $v_1, v_2 \in V$, $\{v_1, v_2\} \in E$ if v_1 and v_2 are compatible with each other. For each v_i , the designer also gives us an expected area of the region v_i , denoted by \mathcal{A}_i^* , and a land before an allocation R .

Our input defined in the previous paragraph are a particular case of the input of the LUOP in (Mohammadi et al., 2016). In that paper, instead of defining if there is a compatibility between two areas, they give a compatibility value between all pairs of areas. Although their input provides more flexibilities to designers, putting values for all pairs of areas is a tedious task, and is not possible for the interactive design. Because of that, we limit the values to only 0 and 1, and make the inputs be a network, which can be entered interactively.

We aim to find a partition of regions $\mathcal{R} = \{R_1, \dots, R_n\}$, where R_i is a region corresponding to object v_i . The regions must cover a whole land area, i.e. $R = \bigcup_{i=1}^n R_i$. Two regions are *adjacent* to each other if they share some boundary lines.

Definition 1 (Compatibility of Partition). *Let $\mathcal{R}_i \subseteq \mathcal{R}$ be a set of regions that is adjacent to R_i . A compatibility of \mathcal{R} , denoted by $C(\mathcal{R})$, can be defined as follows:*

$$C(\mathcal{R}) := \sum_{i=1}^n \frac{|\{R_j \in \mathcal{R}_i : \{v_i, v_j\} \in E\}|}{|\mathcal{R}_i|}. \quad (8)$$

The previous compatibility definition is slightly different from the definition given in (Mohammadi et al., 2016). In the previous definition, the geographic properties of the regions, such as lengths of lines that two regions share each other, are also considered. We do not consider those geographic properties because calculating those properties is time-consuming. If we have to calculate their compatibility value, our method might not be efficient enough for interactive design. Also, we strongly believe that, when two landscape objects are allocated adjacent to each other, users of the objects can easily walk between them. Having them sharing a long line does not make the walk between two regions more convenient.

Next, we will define the land allocation error of partition $\mathcal{R} := \{R_1, \dots, R_n\}$ in the following definition. The definition is same as that defined in (Karimi et al., 2009).

Definition 2 (Land Allocation Error). *Recall that the area of R_i is $|R_i|$. The land allocation error of \mathcal{R} , denoted by $\text{err}(\mathcal{R})$, as follows:*

$$\text{err}(\mathcal{R}) := \sum_{i=1}^n \frac{||R_i| - \mathcal{A}_i^*|}{\mathcal{A}_i^*} \quad (9)$$

In this paper, we aim to find a partition of regions that maximize the compatibility and minimize the land allocation error.

4. METHODOLOGY

4.1 Graph Drawing-Based Point Placement

Our point placement method is based on the graph drawing techniques explained in Section 2.2. In the following subsection, we construct Laguerre Voronoi diagram from $f(v_1), \dots, f(v_n)$ obtained from the techniques. Let w_i be the weight corresponding to v_i in the construction, and R_i be the region obtained. Because we do not want R_i to be an empty region, by Theorem 1, we require that the distance from $f(v_i)$ to any other points have to be at least $2 \cdot w_i$. We can have that requirement, when no rectangle overlap with $Rec(f(v_i), 4 \cdot w_i)$. Because of that, we choose s_i in Section 2.2 to $4 \cdot w_i$, and, to guarantee that the overlapping does not happen, we set the parameter ω to a very large real number.

On the other hand, during the interactive design, we ask designers to choose the appropriate α for their design. A larger α will make the generating points spread more on the plane.

While there is not a drawing boundary in the graph drawing technique, we have to place all points in a given boundary. We update the energy function $E(f)$ to include the constraint in our point placement. Let ℓ_1, \dots, ℓ_m be line segments that define the boundary, and let $D(f(v_i), \ell_j)$ be a shortest distance from $f(v_i)$ to ℓ_j . Given a parameter θ , we define another type of energy for v_i , denoted by $\phi_i(f)$, as follows:

$$\phi_i(f) = \theta \cdot \sum_{j=1}^m \frac{1}{(D(f(v_i), \ell_j))^2}. \quad (10)$$

The energy of $\phi_i(f)$ is very large when the point $f(v_i)$ is very close to some boundary. The third type of energy $\phi(f)$ and the updated energy function E' is defined as in the following equation:

$$\phi(f) = \sum_{i=1}^n \phi_i(f). \quad (11)$$

$$E'(f) = E(f) + \phi(f) \quad (12)$$

Similar to function E , function E' is also derivable, and we can use the gradient descent method to search for $f \in \mathcal{F}$ that minimize E' .

4.2 Laguerre Approximation

Let $\mathcal{S} = \{c_1, \dots, c_n\}$ be a set of circles such that $c_i = (f(v_i), w_i)$, where $f(v_i)$ is a point which is determined by the graph drawing in section 4.1, and w_i is a weight corresponding to the position $f(v_i)$.

In this process, we would like to find the Laguerre Voronoi diagram $\mathcal{R} = \{R_1, \dots, R_n\}$ satisfying the following properties.

- P1 In the set E denoted from a designer, the Laguerre triangulation of \mathcal{R} contains edges of E as much as possible.
- P2 Area of R_i is not much different to the expected area \mathcal{A}_i^* ;

Therefore, the search of the Laguerre Voronoi diagram which satisfies the mentioned properties is necessary. Therefore, we formulate this issue to the optimization problem.

For satisfying the property [P1], we use the compatibility as defined in Definition 1 for determining the connection between any two cells as the desired structure in the graph $\{V, E\}$ from the designer. Also, we consider the property [P2] when the i -th area of Laguerre Voronoi diagram is computed as $|R_i|$ which is the area of the convex polygon. Then we use the criterion of land allocation error as defined in Definition 2 to the Laguerre Voronoi diagram \mathcal{R} .

Since the Laguerre Voronoi diagram is constructed with respect to the set of generators \mathcal{S} , we can consider the compatibility and land allocation error as a function of generator set, i.e. $C(\mathcal{S})$ and $\text{err}(\mathcal{S})$ is a function from \mathbb{R}^{3d} to \mathbb{R} defined by.

$$C(\mathcal{S}) := \sum_{i=1}^n \frac{|\{R_j \in \mathcal{R}_i : \{v_i, v_j\} \in E\}|}{|\mathcal{R}_i|}. \quad (13)$$

and

$$\text{err}(\mathcal{S}) := \sum_{i=1}^n \frac{||R_i| - \mathcal{A}_i^*|}{\mathcal{A}_i^*}. \quad (14)$$

We would like to find the set \mathcal{S}^* which maximizes $C(\mathcal{S})$ which is equivalent to the minimization problem $\min -C(\mathcal{S})$, and minimizes $\text{err}(\mathcal{S})$. This problem leads to the multi-objective optimization, that is, to find

$$\min_{\mathcal{S} \in \mathbb{R}^{3d}} \{-C(\mathcal{S}), \text{err}(\mathcal{S})\}. \quad (15)$$

In this study, we solve the multi-optimization problem (15) using the weighting method. Therefore, we define a function

$$F(\mathcal{S}) = \omega C(\mathcal{S}) - (1 - \omega) \text{err}(\mathcal{S}), \quad (16)$$

and the minimization is done by solving $\min F(\mathcal{S})$.

We treat the function $F(\mathcal{S})$ defined in (16) as the black-box function, i.e. we can implicitly compute the value of the function without knowing the explicit formula. Therefore, we solve the optimization problem using the derivative-free method. In this study, we will use the Nelder-Mead method for finding the minimum of $F(\mathcal{S})$.

5. CASE STUDY

Our application of the experiment will be conducted on the plot of land in Chiang Mai, Thailand, owned by Chiang Mai University. Originally, the land is utilized by the Faculty of Agriculture for education and experiments, however, will later be allocated as an area to promote education and Lanna's historical heritages. This project is called *CMU Square* as announced in (Prachachat Online, 2015).

Along with the original programmatic scheme design, the proposed master plan seems like the design went through without any regards to the surrounding context as shown in Figure 1. Hence, this is where the external anchor points on the graph come to play. To create a contextually related master plan, one must be mindful of what the surrounding contents are. The selected site is located between various important landmarks of Chiang Mai, whether it be the Chiang Mai University to the west, the university's hall to the east, and even the famous Nimmanhaemin road towards the north. With multiple characters surrounding the site, it is safe to assume the local demographic of the users, as well as where and how will they access the area.

5.1 Our Assumptions

Our method's objective is to maximize the usable spaces while minimizing hard to reach areas resulting from a long clean-cut zoning plan. While the original design portrays a solid and organized, there are certain portions of the zoning that could be underutilized. On the other hand, our experiment method will allow us to optimize these space into polygons surrounding the center point of the zoning which would be the structure, meaning that the circulation distance within the zoning would be equally distributed rather than taking a long distance walk from a corner of a rectangle to another.

Using our proposed method, we define the graph $G = (V, E)$ consisting of the vertices represented as the facilities and edges

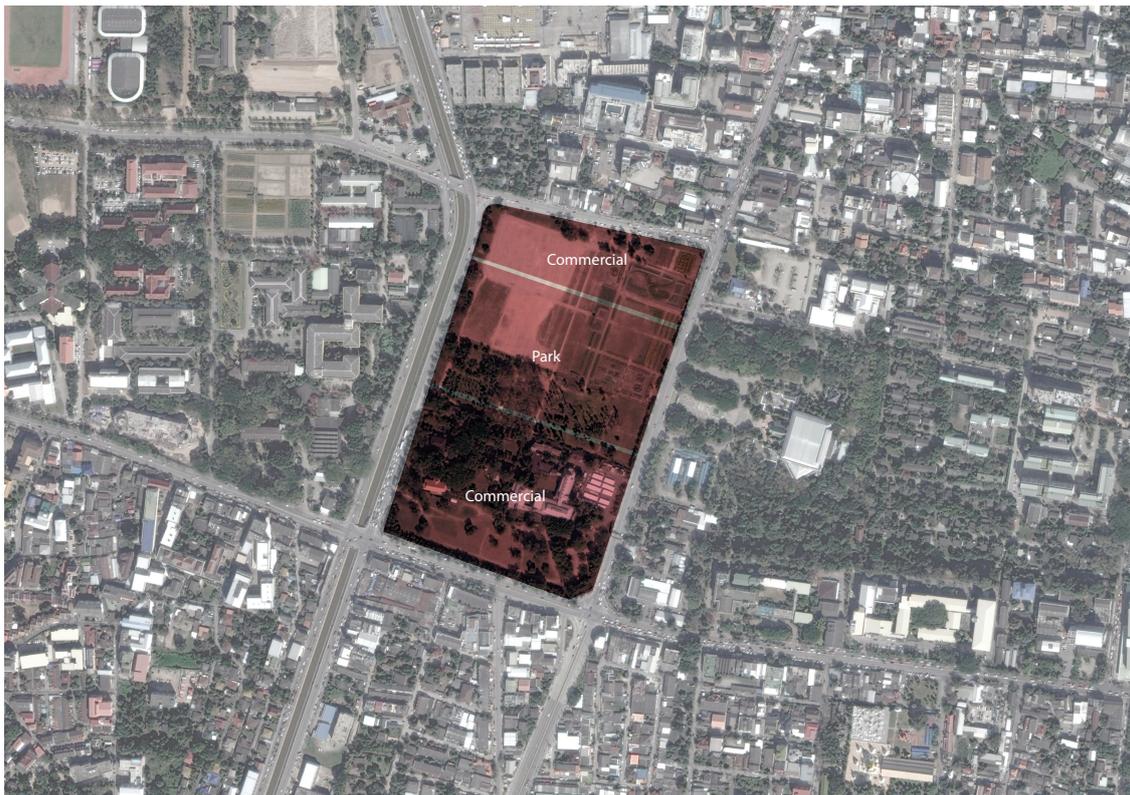


Figure 1. The master plan of CMU square from a newspaper. The figure is adapted from Google®map.

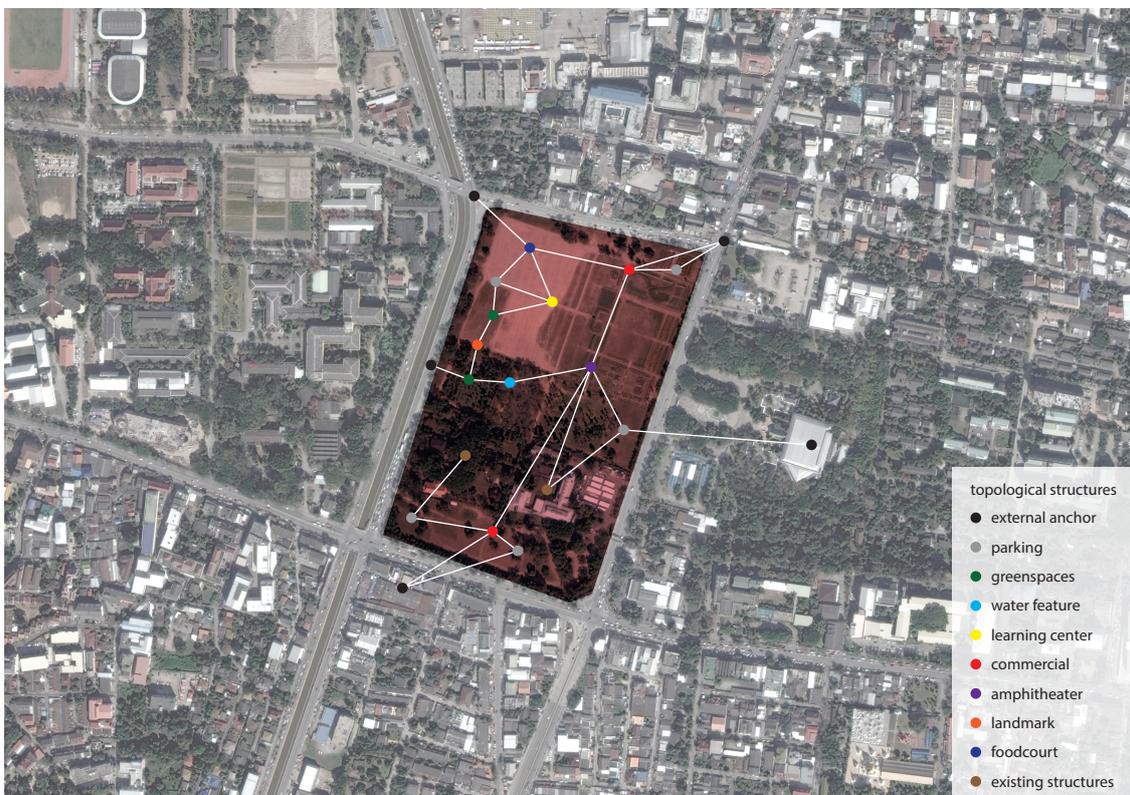


Figure 2. The graph G which reflects the facilities and their connection. The figure is adapted from Google®map.

meant the connected parts between two facilities as shown in Figure 2 where the shaded area is the area for allocation. We assume

that the area for allocation is a convex polygon written as P .

In this experiment, there were 21 vertices in the graph whose seven vertices represented as the existing buildings. Five vertices among all fixed vertices are fixed points laid outside the shaded area. The weight w_i for point i is computed by $w_i = \sqrt{A_i^*}/\pi$. In the case of outer fixed points, we assume that weights are equal to zero.

We follow the process discussed in Section 2.2 and 4.1 to find an appropriate generator positions. After that, we used the resulting positions in the optimization processes using the Nelder-Mead method. In this case, the optimizations were done among all non-fixed points with parameters of generator positions and their weights. In the case of fixed points, we only optimized radii of the inner fixed-points. In the step of the Laguerre Voronoi diagram construction, since the Laguerre Voronoi regions of outer fixed points may affect the Laguerre Voronoi diagram cell inside the polygon P , the construction of Laguerre Voronoi diagram were done only the vertices of the polygon P .

Hence, this experiment considered the function $F(S)$ as a function from \mathbb{R}^{44} to \mathbb{R} when the parameter ω was chosen. In this sense, ω was determined to show the priority between the compatibility and land allocation error.

We optimized $F(S)$ using Nelder-Mead method with a chosen parameter ω . Remark that the simplex contained 45 vertices. For the initial simplex, the first vertex $S_0 \in \mathbb{R}^{44}$ was chosen from the resulting parameters of graph drawing. The remaining vertices were chosen from by $S_i = S_0 + kS_R$, where each coordinate s_j of S_R was a random number from 0 to 1, and k was chosen for scaling the size of the simplex. In this study, the Nelder-Mead method was terminated when the standard deviation of computed function values of all simplex vertices was smaller than a fixed threshold ϵ .

5.2 Experimental Results

5.2.1 Experiments using the proposed framework In the experiment, coordinate of points in the initial step were located using GeoGebra 5.0. The graph drawing was implemented using Java. The construction of the Laguerre Voronoi diagram and Nelder-Mead optimization were done using the software MATLAB®R2017a with libraries 'Bounded Power Diagram' and 'Nelder-Mead Simplex Optimization' published in file exchange system of MATLAB®.

During the graph drawing process, we set $\alpha = 0.5$, $\omega = 20.0$, and $\theta = 0.5$. For the Nelder-Mead optimization to minimize $F(S)$, we set $\omega = 1/8$ with the simplex size $k = 0.1$. The initial function value from the initial point S_0 is 15.606. The land allocation error value was 18.170, and the compatibility was 2.3417.

After 4,915 iterations with threshold $\epsilon = 10^{-5}$, the function value of $F(S)$ converges to 5.5773, as shown in Figure 4. The land allocation error was 6.8033, which decreased by 62.557%, and the compatibility was 2.9976 which increased by 28.010%. The optimization with Nelder-Mead method took 839.78 seconds. The resulting diagram plotted from the converged value of the function is shown in Figure 3.

The experimental result shows that we can automatically locate the facilities using graph drawing schemes. Also, we can allocate the area to facilities using the Laguerre Voronoi diagram in a way that land allocation error is minimized, and the compatibility is maximized.

Remark that the computational time can be shortened when we use other compilers such as C or C++, as discussed in (Andrews, 2012). In many cases, the computations in C or C++ is more than a hundred time faster than those in Matlab. Hence, we strongly believe that our computation time will decrease to a few seconds if we implement our software in those languages.

5.2.2 Comparison with the previous studies We validate how useful our point allocation method by comparing our result with point positions extracted from the master plan of CMU square (Prachachat Online, 2015), which is designed by a professional architect. The point positions are shown in Figure 5. Then, we generate Laguerre Voronoi diagram and multiplicative Voronoi diagram from the point positions in the master plan, and compare their land allocation error and compatibility with our results.

When we construct Laguerre Voronoi diagram from the point positions in the master plan, the land allocation error was 17.1373, which was worse than the result from the framework by 86.3037%. On the other hand, the compatibility was 3.5881, which was better than the optimal result 17.9328% difference. Although the compatibility obtained from the professional architect is slightly larger than our automatically generated point positions, their land allocation error is much larger than ours. Because the whole area is about 42.21, we strongly believe that the land allocation error of 17.14, more than 30% of the whole area, is not acceptable.

The results are even worse in the multiplicative Voronoi diagram. The compatibility obtained in the setting is 4.0, 28.6498% better than ours and slight better than Laguerre Voronoi diagram's. However, the land allocation error is as unacceptable as 33.0. By the multiplicative Voronoi diagram, many areas are significantly larger than the others. For example, we found that the region for Mall1 occupies almost all areas in the upper half of the space. We obtain a large compatibility. because those large regions connect to almost every regions, not because the large value do not indicate that the point positions follows the expected topology.

Although the experimental results cannot be used to judge whether or not our proposed method is better than the previous methods, the tendency of the results shows that the dynamic generating point allocation method together with the optimization can provide the smaller land allocation error as well.

6. CONCLUSION

Ultimately, the experiment is conducted to acquire the balance between maximizing the functionality of the land and the aesthetics of the design as a whole. Tackling the design project with computer-aided design software would reduce the time required for creating a complex design while still maintaining human errors to a minimum.

The allocation of functions, initially done by hand, is gradually optimized through the process by a design tool to maximize the compatibility for the adjacent land use.

For a further experiment, our focus is the development of simple but useful software to instantly allocate land uses according to their contributing factors such as the external context, internal relationship, and size. In parallel, a survey shall be conducted with a selected target group to acquire inputs for further development.

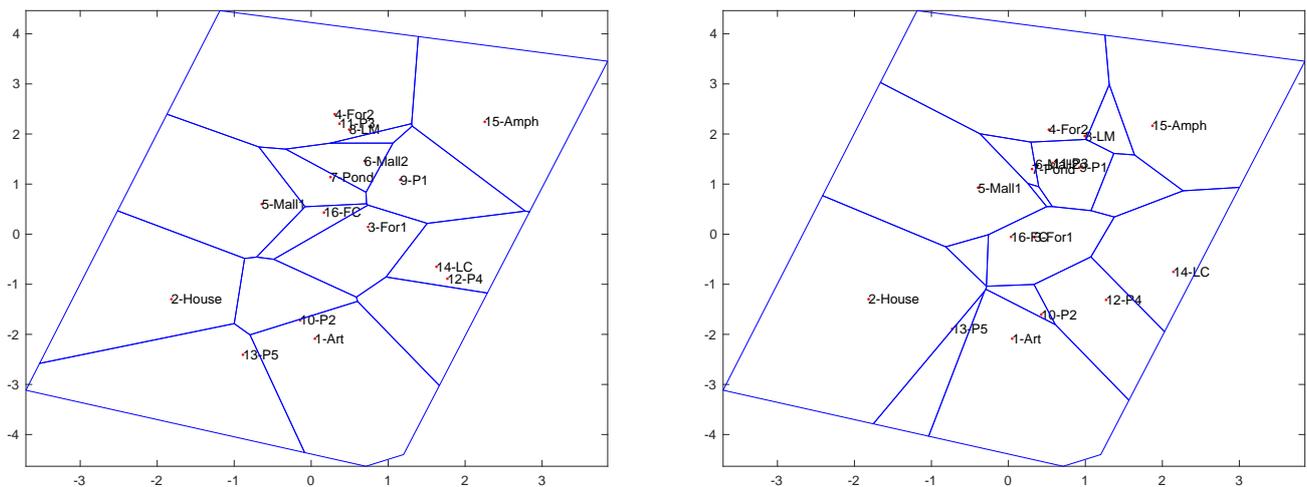


Figure 3. (Left) The Laguerre Voronoi diagram of the generator set generated from the graph drawing-based point placement; (Right) The resulting Laguerre Voronoi diagram after employing the optimization with parameter $\omega = 7/8$

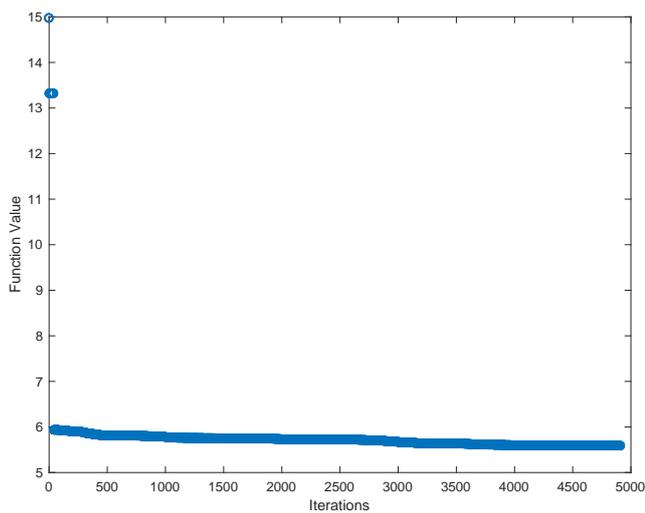


Figure 4. The graph shows the convergence of the function $F(S)$ using the Nelder-Mead method.

ACKNOWLEDGEMENT

This work was supported by JST ERATO Grant Number JPM-JER1201, Japan.

REFERENCES

Andrews, T., 2012. Computation time comparison between Matlab and C++ using launch windows.

Aurenhammer, F., 1987. Power diagrams: Properties, algorithms and applications. *SIAM Journal on Computing* 16(1), pp. 78–96.

Bannister, M. J., Eppstein, D., Goodrich, M. T. and Trott, L., 2012. Force-directed graph drawing using social gravity and scaling. In: *Proc. GD'12*, Springer, pp. 414–425.

Cheng, S.-W., Dey, T. K., Edelsbrunner, H., Facello, M. A. and Teng, S.-H., 2000. Silver exudation. *Journal of the ACM (JACM)* 47(5), pp. 883–904.

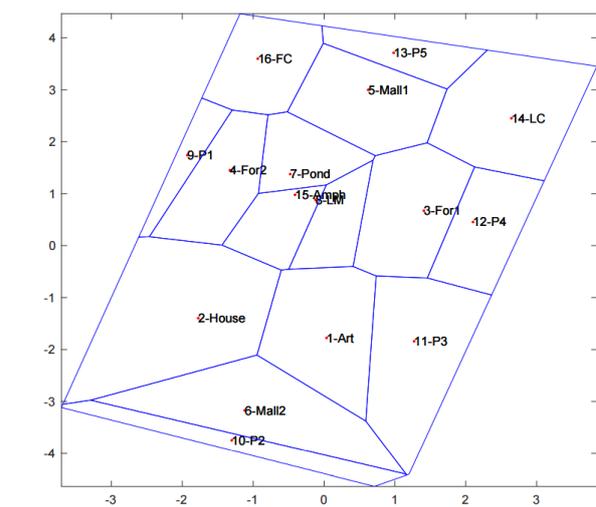


Figure 5. Laguerre Voronoi diagram obtained point positions in the master plan of CMU square

Dong, P., 2008. Generating and updating multiplicatively weighted Voronoi diagrams for point, line and polygon features in GIS. *Computers & Geosciences* 34(4), pp. 411–421.

Imai, H., Iri, M. and Murota, K., 1985. Voronoi diagram in the Laguerre geometry and its applications. *SIAM Journal on Computing* 14(1), pp. 93–105.

Kamps, T., Kleinz, J. and Read, J., 1995. Constraint-based spring-model algorithm for graph layout. In: *GD'95*, Springer, pp. 349–360.

Karakostas, S. and Economou, D., 2014. Enhanced multi-objective optimization algorithm for renewable energy sources: optimal spatial development of wind farms. *International Journal of Geographical Information Science* 28(1), pp. 83–103.

Karimi, F., Delavar, M. and Mostafavi, M., 2009. Space allocation of educational centers using multiplicatively weighted

Voronoi diagram. In: *Workshop on Quality, Scale & Analysis Aspects of City Models*.

Lagarias, J. C., Reeds, J. A., Wright, M. H. and Wright, P. E., 1998. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization* 9(1), pp. 112–147.

Michalek, J. and Papalambros, P., 2002. Interactive design optimization of architectural layouts. *Engineering Optimization* 34(5), pp. 485–501.

Mohammadi, M., Nastaran, M. and Sahebgharani, A., 2016. Development, application, and comparison of hybrid meta-heuristics for urban land-use allocation optimization: Tabu search, genetic, GRASP, and simulated annealing algorithms. *Computers, Environment and Urban Systems* 60, pp. 23–36.

Nelder, J. A. and Mead, R., 1965. A simplex method for function minimization. *The computer journal* 7(4), pp. 308–313.

Okabe, A., Boots, B., Sugihara, K. and Chiu, S., 2009. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics, Wiley.

Prachachat Online, 2015. Chiang mai university plans to develop CMU Square: The last high-rise area in nimmanhaemin. http://www.prachachat.net/news_detail.php?newsid=1450767410. Accessed: 2017-03-01 (In Thai).

Qi, H., Altınakar, M. S., Vieira, D. A. and Alidaee, B., 2008. Application of Tabu search algorithm with a coupled AnnAGNPS-CCHE1D Model to optimize agricultural land use. *JAWRA Journal of the American Water Resources Association* 44(4), pp. 866–878.

Reitsma, R., Trubin, S. and Mortensen, E., 2007. Weight-proportional space partitioning using adaptive Voronoi diagrams. *Geoinformatica* 11(3), pp. 383–405.

Santé-Riveira, I., Boullón-Magán, M., Crecente-Maseda, R. and Miranda-Barrós, D., 2008. Algorithm based on simulated annealing for land-use allocation. *Computers & Geosciences* 34(3), pp. 259–268.

Song, T., Cui, X. and Yu, G., 2015. A general vector-based algorithm to generate weighted Voronoi diagrams based on ArcGIS engine. In: *Proc. ICMA'15*, IEEE, pp. 941–946.

Sugihara, K., 2000. Three-dimensional convex hull as a fruitful source of diagrams. *Theoretical Computer Science* 235(2), pp. 325–337.