

서비스 위치 그룹핑을 위한
보로노이 다이어그램 기반의 유전자알고리즘

서정연¹, 박상민², 정인재¹, 김덕수¹

Regrouping Service Sites:
a Genetic Approach using a Voronoi Diagram

Jeong-Yeon Seo¹, Sang-Min Park², In-Jae Jeong¹ and Deok-Soo Kim¹

¹ Department of Industrial Engineering
Hanyang University
Sungdong-gu, Seoul 133-791
KOREA

² Voronoi Diagram Research Center
Hanyang University
Sungdong-gu, Seoul 133-791
KOREA

Abstract

In this paper, we consider the problem of regrouping a number of service sites into a smaller number of service sites called *centers*. Each service site is represented as a point in the plane and has an associated value of service demand. We aim to group the sites so that each group has the balanced service demand and the sum of distances from the sites in the group to their corresponding center is minimized.

To solve this problem, we propose a hybrid genetic algorithm that is combined with Voronoi diagrams. We provide a

variety of experimental results by changing the weights of the two factors: service demands and distances. Our hybrid algorithm finds better solutions in a shorter computation time in comparison with a pure genetic algorithm.

1. Introduction

Suppose there are n service sites spread over an area and each site has an associated value of service demand. We would like to partition the area into k disjoint regions. More precisely, we want to

select k service sites called centers each of which serves the corresponding region. Each selected center will replace a group of neighboring service sites. We aim to group the sites so that service demands are evenly distributed over the groups and the total sum of distances from the sites in the groups to their corresponding centers is minimized. We assume that every pair of sites are connected by an edge whose weight is the Euclidean distance between the two sites.

This kind of problems have been considered in numerous areas including operations research, pattern recognition, image processing and computer graphics. Grouping is usually called clustering or sometimes decomposition in terms of graphs or d -dimensional space. The problem of finding the optimal clustering is NP-hard in general [5] and approximate solutions have been proposed [3,12]. Similar problems can be found in facility location which is to locate a set of supply stations which optimizes a given objective function. The k -center problem and the k -median problem are well-known facility location problems. The k -center problem is to select k sites (centers) among the sites of a given set so that the maximum of the distances from each site to the closest center is minimized. In the k -median problem, the total sum of the distances should be minimized [1].

The objective of regrouping problem is to consolidate resources in general. The selected centers are expected to provide

higher quality services and at the same time reduce the cost of services. For example, suppose that a company plans to restructure the system of branch offices for efficient operation. You can merge a certain number of offices into one in order to cope with the decrease of overall service demand. This solution can be applied to various kinds of service sites such as medical service centers, educational sites, warehouses, shops, etc.

Regrouping problem was addressed by Mansour *et al.*[9] whose service sites are given as a graph. Thus the connection between sites are defined by a given set of edges and their weights. The goal of the problem is to locate the centers so that the total travel distance between service sites and their corresponding centers is minimized. At the same time, it should have balanced distribution of services over the different regions. They presented a two-phase method: first decompose the graph using a tuned hybrid genetic algorithm, and then find a suitable center for each region by a heuristic algorithm.

In this paper we present a new algorithm to solve the regrouping problem in a geometric setting. We decompose the $2D$ plane whereas Mansour *et al.* decompose a graph. A Voronoi diagram is a useful structure which provides the proximity information among points. It can be a good tool for minimizing the sum of distances. For balanced distribution of service demand, we adopt a genetic approach. Thus we suggest a hybrid genetic

algorithm that is combined with Voronoi diagrams. Voronoi diagram for n sites has n regions each of which consists of points closer to the corresponding site than any others. Using Voronoi diagrams, we obtain the topological information among the sites and based on this, we devise our algorithm.

To demonstrate the efficiency of our algorithm, we compare it with two other algorithms: One uses a pure genetic approach and the other uses Voronoi diagrams. The experimental results show improvements in running time as well as cost optimization. We also provide a variety of empirical results by changing the relative importance of the two criteria, which involve service demand and distances, respectively.

In the next section, we present the formal definition of the problem and describe the objective function. Then we give an overview of genetic algorithms. Section 3 gives the definition of the Voronoi diagram and explains how it works in our algorithm. In Section 4 & 5, we elaborate on our genetic algorithm and present experimental results.

2. Preliminary

In this section, we formally introduce the definition of regrouping problem. Next, we briefly describe the concept of genetic algorithm.

2.1 Problem definition

Assume that n sites are distributed in $2D$ plane R . Each site v_i has x - and y -

coordinates (x_i, y_i) and weight w_i . An edge e_{ij} between v_i and v_j also has the weight l_{ij} which is the Euclidean distance between (x_i, y_i) and (x_j, y_j) . More formally, we are given the following undirected graph $G \langle V, E \rangle$ with vertex set $V = \{v_i \mid i = 1, 2, \dots, n\}$ and edge set $E = \{e_{ij} = (v_i, v_j) \mid i, j = 1, 2, \dots, n\}$ where each $v_i = (x_i, y_i)$ is associated with w_i and each e_{ij} is associated with l_{ij} .

The regrouping problem is to partition the plane R into k disjoint regions r_1, r_2, \dots, r_k and select a center within each region so that the objective function O is minimized. We define the function O with regard to two criteria:

- The deviation of the total vertex weights in a region r_j is minimized.

$$W_{r_j} = \left| \sum_{v_i \in r_j} w_i - \sum_{v_i \in V} \frac{w_i}{k} \right| \quad (1)$$

- The sum of distances from the sites to the center within a region r_j is minimized.

$$L_{r_j} = \sum_{v_i \in r_j} l_{ic} \quad : v_c \text{ is the center of } r_j \quad (2)$$

The cost of the region r_j is the sum of the two terms and we use α as an experimental parameter.

$$C_{r_j} = \alpha W_{r_j} + (1 - \alpha) L_{r_j} \quad (3)$$

The objective function O to be minimized is given by the average of the region costs:

$$O = \frac{1}{k} \sum_{r_j \in R} C_{r_j} \quad (4)$$

To summarize, we want to find k centers among n sites so that the weights of vertices are distributed over the regions as evenly as possible and the total sum of

distances to the centers in all regions is minimized.

2.2 Overview of genetic algorithms

The motivation of the genetic algorithm is the biological paradigm of natural selection as first articulated by Holland [7]. A genetic algorithm starts with a population. This population consists of a set of possible solutions, called chromosome or individuals. This population then evolves into different populations for several iterations. For each iteration, the evolution process proceeds as selection, crossover, and mutation in concert with the value of objective function. For more details on genetic algorithms, refer to [4] and [6].

The main operations of genetic algorithms are selection, crossover and mutation. A selection operation decides whether each individual of current population survives or not according to its fitness value. Then, two individuals of population are randomly selected and combined through a crossover operation. The resulting offspring individuals are modified by a mutation operation. Now, a new population is created and the evolution process is repeated until the terminal condition is met.

3. Why Voronoi diagram?

Let $P = \{ v_1, v_2, \dots, v_n \}$ be a finite set of sites in the plane. A Voronoi diagram of P partitions the plane such that each region corresponds to one of the sites and all the points in one region are closer to the

corresponding site than to any other site. More formally, a Voronoi region $V(v_i)$ of a site v_i is defined as $V(v_i) = \{ P \mid d(p, v_i) < d(p, v_j) \text{ for any } j (\neq i) \}$ where $d(p, q)$ is the Euclidean distance between p and q . Note that a Voronoi region is always a bounded convex polygon as shown in Figure 1. The Voronoi diagram is a powerful tool in computational geometry which provides all spatial information among geometric objects in a system with efficient data structures[8]. For constructing the Voronoi diagram, it takes $O(n \log n)$ time in the worst case and $O(n)$ time on the average[10]. Voronoi diagrams have been extensively studied as a useful tool in various fields such as astronomy, biology, cartography, crystallography and marketing [8,11]. The survey on Voronoi Diagrams can be found in [10,2].

In the regrouping problem, we have two criteria to determine the quality of a solution. One is the balanced distribution of service demands and the other is the sum of distances to the centers. Noting that a Voronoi diagram provides the most compact and concise representation of the proximity information in Euclidean space, we adopt this structure in our algorithm. In fact, the Voronoi diagram can give the solution that minimizes the sum of distances provided that a set of centers is known. The regrouping problem consists of two parts—decomposing the plane into k regions and select a center for each region. In our algorithm, we first select centers with regard to the distribution of the service

demands. With this set of centers, we decompose the plane using the structure of Voronoi diagrams. These two steps are used as basic operations of our genetic algorithm to optimize the solution.

4. Regrouping using genetic approach & Voronoi diagram

In this section, we propose a genetic algorithm using a Voronoi diagram, which we will call VGA, for the problem of regrouping service sites. To verify the performance of VGA, we implement two more algorithms as well as VGA : a pure genetic algorithm(GA) and an algorithm that only uses Voronoi diagrams(VD). In the followings, we describe each of the three algorithms.

4.1. GA algorithm

We first give GA algorithm which is loyal to the original concept of genetic algorithms. Each chromosome in our problem is represented by n - elements row $[C(v_1), C(v_2), C(v_3), \dots, C(v_n)]$. The i -th site v_i takes a value $C(v_i)$ from $1, 2, \dots, k$ (k is the number of regions) which represents a center whose region contains the site. In general, the initial population of possible solutions is randomly generated. We use the roulette-wheel selection and the fitness value of the individuals to be minimized is the objective function O . The crossover operation used is the double-point crossover and the crossover rate is set to be 0.7. It is applied to a randomly selected pair of individuals. The probability of

mutation is 0.2. We perform a mutation operation by switching the regions of two randomly selected sites. The following gives the outline of GA algorithm.

```

Program GA
{ Input : graph  $G < V, E >$ 
{ Output: Subgraphs :  $R_1, R_2, \dots, R_k$ 

Calculate the Probability of each site :
 $Pr(V)$ ;
Generate the initial population by randomly
assigning sites to subgraphs;
Evaluate fitness for each individual;
Repeat (100 times)
{
Rank individuals;
Give probability to each individual;
Apply selection, crossover, and mutation;
Evaluate fitness of individuals;
}
    
```

4.2 VD algorithm

VD algorithm solves the regrouping problem using a Voronoi diagram only. This method is simple and easy to implement. Since a Voronoi diagram is constructed in linear time, VD algorithm has a short computation time. Instead of generating random values which form the individuals of the initial population of GA algorithm, we use a Voronoi diagram for the initial population. We first assign the probability to each site according to its weight. Then, we choose k centers with high probability as the generators of Voronoi diagram and then construct the Voronoi diagram of the k

centers. The sites that are contained a Voronoi region belong to the same group. The outline of generating the regrouping solution using a Voronoi Diagram is given as follows.

Program VD
 { Input : graph $G < V, E >$ }
 { Output: Subgraphs : R_1, R_2, \dots, R_k }

Assign probability to each site according to its weight : $Pr(V)$;
 Choose k sites (for each individual);
 Compute VD of k sites;
 Classify v_i into one of k Voronoi regions for each i ;
 Evaluate O ;
 Consider the individual with the minimum O as the solution;

4.3. VGA algorithm

In VGA algorithm, the basic method is a genetic algorithm and at the generation of the initial population and the mutation, we employ the Voronoi diagram structure that gives topological information among the sites.

First we start with the same initial population as VD algorithm. By giving the probability of each site according to its weight, we select k sites as centers. In order to assign the sites to one of the k regions, we construct the Voronoi diagram for the k centers, which we call VD_k . VD_k provides a grouping which minimizes the distances from the sites in each group to their corresponding center.

To take advantage of the topology and geometry information of VD_k , we define a mutation operator which is applied only on the sites on the boundaries of VD_k . But it is not clear which sites are close to the boundaries. Thus we construct the Voronoi diagram of all the sites, $VD(V)$ and find sites whose neighboring site belongs to a different group. If a site is selected as a mutation site, the site moves to one of the neighboring regions.

Refer to Figure1: site A mutates into Region 1 and site B arbitrarily chooses one of Region 1 and Region 2 and moves to it.

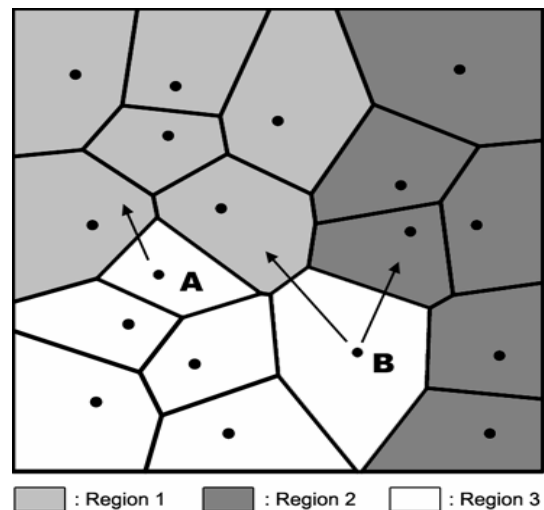


Figure. 1 Mutation on the boundaries of Voronoi regions

Program VGA
 { Input : graph $G < V, E >$ }
 { Output: set of subgraphs: R_1, R_2, \dots, R_k }

Calculate the Probability of each site:
 $Pr(V)$;
 Compute $VD(V)$;
 Choose k sites (for each individual) ;

```

Compute VD of  $k$  sites;
Classify  $v_i$  into one of  $k$  Voronoi regions
for each  $i$ ;
Evaluate  $O$  for each individual;
For( 50 times )
{
    For( each individual )
    {
        Make a boundary site list for each
region;
        Apply a mutation operation on the
list;
    }
}
Evaluate  $O$  for each individual ;
Apply selection;
    
```

We set the probability of mutation to be 0.2 as in GA algorithm. Unlike general genetic algorithms, a crossover operation turned out to be unsuitable for these populations. Since it is applied to a randomly-selected pair of individuals, some individuals created by crossover may have either more than or less than k centers. Besides, we may lose the advantage from using Voronoi diagrams. Hence, the crossover operation is avoided in VGA algorithm. We give the outline of VGA algorithm as above.

5. Experimental results

In our implementation, we choose the size of the population to be equal to $2 * n * k$. And x - and y - coordinates and weight of site are randomly generated between 10

and 100. The average, min and max values are computed over 10 problem instances. These instances are derived from graphs with $n= 50$ that are decomposed into 5 different regions ($k = 5$). Experiments were performed on a PC with Pentium IV 2.4 GHz CPU and 512 MB of RAM. The implementation was done using visual C++ . Recall that the objective function is computed for j -th group as follows (refer to Equation (3)).

$$C_{r_j} = \alpha W_{r_j} + (1 - \alpha)L_{r_j}$$

Note that the first term is related to the service demand and the second term is related to the distances. To give the diversity of the simulation, we scale the value of α from 0 to 1 range in our experiment. Hence α is used to investigate how the results of three algorithms are influenced by the change of relative importance of the two criteria.

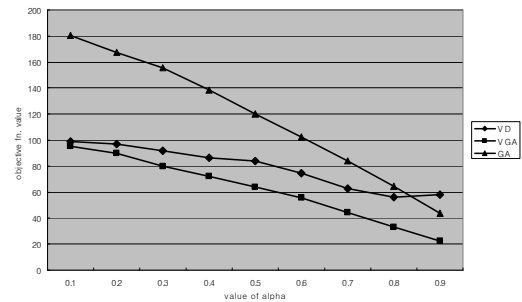


Figure. 2 Comparison three algorithms

The results obtained in our experiments are shown as a graph in Figure 2. The overall performance of VGA algorithm is superior to the other two. When α is small, VD algorithm is good because it is mainly focused on minimizing

the sum of distances. When α is close to 1, GA wins VD which makes little efforts to optimize the service demand term. VGA algorithm which uses both Voronoi diagrams and genetic approaches produces high quality solutions for the regrouping instances compared to GA algorithm (pure GA approach) and VD algorithm (pure geometric approach). The value of the objective function of the solutions of VGA algorithm is 47.5% better than that of GA algorithm on the average. In terms of running time, the simple VD algorithm is the fastest. GA algorithm is able to obtain smaller values of the objective function than those in Figure 2, but it takes much longer than VGA algorithm. We set the number of iteration for VGA algorithm and GA algorithm to be 50 and 100 respectively to complete both in a similar time. On that condition, VGA algorithm produces a better objective function value than GA algorithm. Therefore, the suggested VGA algorithm gives good results in terms of both quality of grouping as well as running time. To summarize these considerations, the followings can be said:

- VGA algorithm is distinguished in the overall performance.
- VD algorithm gives good solution when α is small because it is mainly focused on minimizing the sum of distances.
- GA algorithm gives good solution when α is close to 1 because it makes efforts to optimize the service demand term compared to VD.
- Between two algorithms using genetic

approach, VGA algorithm is much faster than GA algorithm when the same quality of the solution is required.

An example of the output of our VGA algorithm is shown in Figure 3. Each region of k groups is filled with different patterns. The center of each region is represented as a black point while the other sites are shown as white ones.

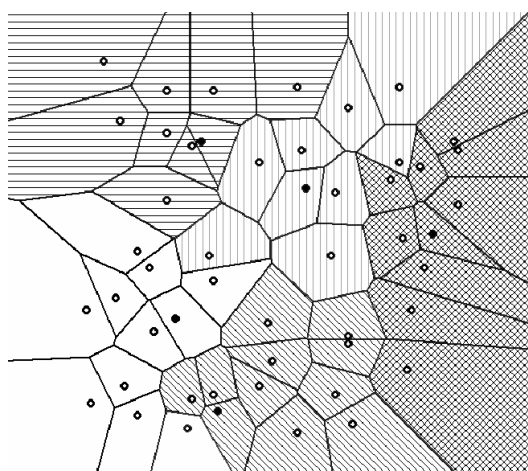


Figure. 3 VGA solution (objective function value=65.37, $\alpha = 0.5$)

6. Conclusion

We proposed a new genetic algorithm which produces good and efficient solutions for the regrouping problem. The idea of combining the genetic algorithm with the proximity information provided by Voronoi diagram has led to good solutions as shown in the empirical results. Possible extension of our work would be grouping problems with other kinds of quality measure such as minimizing the maximum radius, diameter or area of the group regions. We expect the Voronoi diagram to be a useful method for those

variants as well. In addition, algorithms employing Voronoi diagrams provide the grouping whose elements are concentrated around their center rather than being spread around, which is significant in an aesthetic sense.

Acknowledgement

This research was supported by Creative Research Initiatives from the Ministry of Science and Technology, Korea.

References

- [1] Arora, S., Raghavan, P., Rao, S.: "Approximation schemes for Euclidean k-medians and related problems." *In Proceedings of the 30th Annual ACM Symposium on Theory of Computing*. ACM, New York (1998) 106-113.
- [2] Aurenhammer, F.: "Voronoi diagrams - a survey of a fundamental geometric data structure." *ACM Computing Surveys*, Vol. 23. (1991)345-405.
- [3] Bui, T.N., Moon, B.R.: "Genetic algorithm and Graph partitioning." *IEEE Transactions on computers*, Vol.45. No.7. (1996) 841-855.
- [4] Chambers, L.D.: *Practical handbook of genetic algorithms. Vols. I and II* CRC Press, Boca Raton, FL (1995).
- [5] Garey, M.R., Johnson, D.S.: *Computers and intractability: a guide to the theory of NP-completeness*. Freeman and Co., New York (1979).
- [6] Goldberg, D.E.: *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, Reading, MA (1989)
- [7] Holland, J.: *Adaptation In Natural and Artificial Systems*. The University of Michigan Press, Ann Arbour (1975).
- [8] Kim, D.S., Chung, Y.C., Kim, J.J., Kim, D., Yu, K.: "Voronoi diagram as an analysis tool for spatial properties for ceramics." *Journal of Ceramic Processing Research*, Vol. 3, No. 3, PartII. (2002) 150-152.
- [9] Mansour, N., Tabbara, H., Dana, T.: "A genetic algorithm approach for regrouping service sites". *Computers & Operations Research*, Vol. 31. (2004) 1317-1333.
- [10] Okabe, A., Boots, B., Sugihara, K.: *Spatial Tessellations Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons (1992).
- [11] Seo, D.I., Moon, B.R.: "Voronoi Quantized Crossover for Traveling Salesman Problem." *In Genetic and Evolutionary Computation Conference*. (2002) 544-552.
- [12] Vazirani, V.: *Approximation Algorithms*. Springer, Berlin (2001).