Location of an Obnoxious Facility with Rectilinear Distance: A Genetic Algorithm Approach

Chih-Chin Lai

Department of Information Management
Shu-Te University, Yen Chau, Kaohsiung County, Taiwan 824
E-mail: cclai@mail.stu.edu.tw

Abstract

Undesirable or obnoxious facility location problems are the most active research areas with location theory in recent years. In this paper, we consider the problem of locating a single obnoxious facility in the continuous plane, where the location of the facility is restricted to be outside a specified forbidden region around each demand point. The objective of this problem is to minimize the sum of the weighted rectilinear distances from the demand points to the facility. We propose an efficient approach for finding the near-optimal facility location based on a well-known optimization procedure, genetic algorithms. Experimental results are presented to illustrate the feasibility of the proposed approach.

Keywords**:** Obnoxious facility, location, genetic algorithms

1. Introduction

One of the most active research areas within location theory in recent years is to deal with the location of undesirable or obnoxious facilities [1, 3, 5, 15, 17]. A so-called undesirable or obnoxious facility is that if it may cause lower quality of life or pose a serious danger to the individuals living nearby. Examples of obnoxious facilities include nuclear power plants, solid waste repositories, chemical incinerator, etc. For the location of a single obnoxious facility, the most frequently used objective is to find a location within a feasible region that maximizes its minimum distance with respect to all existing facilities. This is referred to as the maximin criterion [2, 10]. On the other hand, the maxisum criterion involves the maximization of a weighted sum of the distances from the obnoxious facility to all the demand points [2, 10]. For a general overview of the maximin and maxisum criteria for locating single or multiple facilities, the interesting reader may refer to Erkut and Neuman [7].

Brimberg and Wesolowsky [2] described a mathematical model for locating a single obnoxious facility on a continuous plane, which considers transportation costs between the facility and a set of demand points, as well as social costs arising from the undesirable characteristics of the facility. Two main features of the model include the following: (1) A standard minisum objective function is used to measure the transportation costs, while the social costs are included implicitly in the lower bound constraints associated with the model that force the facility location to be outside a specified forbidden region around each demand point; (2) The model adopts the rectilinear norm to measure the distances. The distance metric is applicable since the travel between facilities can be approximated by the rectilinear paths. Moon and Chaudhry [14] also considered the above type of formulation, but only applied it to a discrete setting. Examples of location in the presence of forbidden regions were discussed in Hamacher and Nickel [9], and Buchanan and Wesolowsky [4], etc.

Although varieties of approaches have been proposed for solving the problem of locating obnoxious facilities in the plane, the best selection of a site is still a complex problem. In this paper, we adopt the assumption and model proposed by the Brimberg and Wesolowsky [2], and we use genetic algorithms to find the best facility locations. Genetic algorithms (*GAs*) [6, 8, 12, 13] are robust computational and stochastic search procedures modeled on the mechanics of natural genetic systems. *GAs* act as a biological metaphor and try to simulate some of the processes observed in natural evaluation. *GAs* are well known for their ability by efficiently exploiting the historical information to improve search performance and *GAs* have the following advantages over traditional search methods: (1) *GAs* directly work with a coding of the parameter set; (2) search is carried out from a population of points; (3) payoff information is used instead of derivatives or auxiliary knowledge; and (4) probabilistic transition rules are used instead of deterministic ones [8]. *GAs* are gradually finding applications in various fields, such as combinational optimization [6], machine learning [16], and image processing [11].

The paper is organized as follows: In Section 2, we formulate the obnoxious facility location problem in the literature and present the *GA* approach. The example and computational results are given in Section 3. The conclusions are summarized in Section 4.

2. The Problem Formulation and Proposed Approach

2.1. Mathematical model

The minisum criterion of an obnoxious facility location problem using rectilinear distances is stated as follows: given a two-dimensional region $S \subset \boldsymbol{R}^2$, and a set of n demand points representing the customers $P_i(x_i, y_i),\ i = 1, 2, \ldots, n$, located in S, find a point $(u, v) \in S$ that minimizes the total transportation cost between (u, v) and the n-points:

$$\min_{(u,v)\in S} \left\{ w = \sum_{i=1}^{n} v_i d_i(u,v) \right\} \tag{1}$$

subject to

$$d_i(u,v) \geq r_i, \quad \text{for } i = 1, 2, \ldots, n,$$

where

- w is the minisum objective function,
- $d_i(u,v)$ is the rectilinear distance between the new facility to be located at (u, v) and existing demand point i located at $P_i(x_i, y_i)$, i.e.,

$$d_i(u,v) = |u - x_i| + |v - y_i|, \quad i = 1, 2, \ldots, n, \tag{2}$$

- v_i is a specified nonnegative weight which converts the distance traveled from demand point i to the new facility into a transportation cost, and
- $r_i > 0$, for $i = 1, 2, \ldots, n$, represents a specified lower bound on the distance separating demand point i from the new facility.

Alternatively, by using the sign function

$$\delta(s,t) = \begin{cases} 1 & \text{if } s \geq t \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

for all real values of the scalar variables s and t, the Eq.(1) can be rewritten as follows:

$$\min_{(u,v)\in S} \left\{ w = \sum_{i=1}^{n} v_i \left[(u - x_i)\delta(u, x_i) + (v - y_i)\delta(v, y_i) \right] \right\} \tag{4}$$

subject to

$$(u - x_i)\delta(u, x_i) + (v - y_i)\delta(v, y_i) \geq r_i, \quad \text{for } i = 1, 2, \ldots, n.$$

Using this model, Brimberg and Wesolowsky [2] presented an $O(n^3)$ algorithm to solve the rectilinear distance minisum problem with infeasible regions. Konforty and Tamir [10] also focused on the same model and proposed an $O(n \log n)$ algorithm to solve the problem. Here, we provide an alternative way and adopt the robust search method of genetic algorithm to solve this problem.

2.2 Genetic Algorithm

Genetic algorithms (*GAs*) are randomized search and optimization techniques guided by the principles of evolution and natural genetics, and have a large amount of implicit parallelism. They provide near optimal solutions of an objective or fitness function in complex, large, and multimodal landscapes. In general, a *GA* contains a fixed-size population of potential solutions over the search space. These potential solutions of the search space are encoded as binary or floating-point strings and called *individuals* or *chromosomes*. The initial population can be created randomly or based on the problem-specific knowledge. In each iteration, called a *generation*, a new population is created based on a preceding one through the following three steps: (1) evaluation: each individual of the old population is evaluated using a fitness function and given a value to denote its merit, (2) (2)
selection: individuals with better fitness are selected to generate next population, and (3) mating: genetic operators such as crossover and mutation are applied to the selected individuals to produce new individuals for the next generation. The above three steps are iterated for many

generations until a satisfactory solution is found or a terminated criterion is met. The standard *GA* procedure is described in the following pseudocode:

$t \leftarrow 0$
Randomly generate a population *P*(*t*)
Evaluate each member in *P*(*t*)
while (*the termination criterion is not met*) do
  begin
    $t \leftarrow t+1$
    select *P*(*t*) from *P*(*t*-1)
    recombine *P*(*t*)
    evaluate each member in *P*(*t*)
  end

When we use a *GA* to solve a problem, we must consider the following components: (1) a genetic representation of solutions to the problem, (2) one way to create the initial population of solutions, (3) an evaluation function that rates all candidate solutions according to their "fitness", (4) genetic operators that alter genetic composition of children during reproduction, and (5) control parameters (e.g., population size, crossover and mutation rates) [13].

A. Solution representation

A real-coded *GA* is a *GA* that uses floating-point numbers to represent genes [13]. In our utilization of *GA* to find proper facility location, the coordinates of a facility are real-coded and represent an individual. Since each location has two coordinates, the length of the individual is 2*p*, where *p* is the number of facilities.

B. Initial population

A *GA* requires a population of potential solutions to be initialized at the beginning of the *GA* process. Usually, the initialization process varies with the applications. Here, we randomly select a point from the rectangular region contains all demand points and associated forbidden areas as the facility location until all population members are created.

C. Fitness function

A fitness function is the survival arbiter for individuals. Since the objective of the problem is to minimize the sum of the weighted rectilinear distances from the demand points to the facility, the fitness function is just defined as the same as Eq.(4),

$$F=\sum_{i=1}^{n} v_i\left[(u-x_i)\delta(u,x_i)+(v-y_i)\delta(v,y_i)\right], \quad (5)$$

where (u,v) and (x_i, y_i) represent the location of the facility and the demand point, respectively, and v_i is a specified nonnegative weight.

.D. Genetic operators

Three primary genetic operators: selection, crossover, and mutation are generally used in *GAs*.

- *Selection*

The selection operator determines which individuals are chosen for mating and how many offspring each selected individual produces. Two reproductive strategies are commonly used. *Generational* reproduction replaces the whole population in each generation, but *steady-state* reproduction only replaces the less-fitted members in a generation.

There are several schemes for the selection process. Baker compared various selection methods comprehensively, and presented an improved version called *stochastic universal sampling* (*SUS*) method [13]. The *SUS* method is an optimal sequential selection algorithm. All surviving individuals are simultaneously determined in a single traverse of the population. A *SUS* procedure is described by the following *C* code:

```
ptr = Rand( );
for(sum = i =0; i < N; i++)
   for(sum += ExpVal[i]; sum > ptr; ptr++)
      Selection_individual(i);
```

The Rand( ) returns a random real number between 0 and 1, *N* is the number of individuals in a population, and ExpVal[*i*] represents the expected value of individual *i* and the value is used to indicate the average number of offspring that individual should receive [13]. Here we adopt the steady-state reproduction and the *SUS* method in the proposed approach.

- *Crossover and Mutation*

Crossover and mutation operators are applied with different probabilities and play different roles in the *GA*. Crossover is aimed to increase the average quality of the population. On the other hand, mutation is needed to explore new areas of the search space and helps the algorithm avoid sticking in local optima. The One-point crossover and Gaussian mutation schemes [13] are adopted in the proposed approach.

The Gaussian mutation works as follows:

$$\hat{p} = p \pm \alpha \times p, \quad (6)$$

where p is the existing gene value in an individual, $\hat{p}$ is the mutated gene, and α is a real number selected from a Gaussian distribution with zero mean and standard deviation 0.1. If the mutation process generates gene values outside the valid range of the gene, the gene value will be reset to the lower or upper value of the initialization range of that gene.

E. Control parameters

The population size influences the performance of *GAs*. A small-sized population reduces the evaluation cost but results in premature convergence, because the population provides insufficient samples in the search space. For a large-sized population, the *GA* can gain more information to search better solutions because the population contains more representative solutions over the search space. However, more computations are needed in a large-sized population, and this situation possibly results in an unacceptably slow rate of convergence.

Both crossover and mutation probabilities also influence the performance of *GAs*. In order to get better performance, a few additional tries are performed to find more appropriate values for these desired *GA* control parameters in the proposed approach.

3. Experiments

In order to test the feasibility of the proposed approach, we use it to solve the following example. The problem deals with five demand points. The locations of each demand point, the specified weight and lower bound on the distance separating demand point from the facility are shown in Table 1. The parameters of *GA* used in the experiments are as follows: (1) the generation number is 3000, (2) the population size is 50, and (3) the probability of crossover is 0.85.

The five demand points and the near-optimal solution are indicated by the circle and square as shown in Fig.1. The forbidden region associated each demand point is also shown in Fig.1. We can find that the facility location is at (7.0005, 2.9988), and the total distance is 16.2549. Although the proposed approach achieved similar result that obtained via the branch-and-bound method [2]; however, the former is more efficient than the latter. Since the constraint associated Eq.(1) defines a nonconvex feasible region, it results in Eq.(1) cannot be expressed as a linear program [2]. Brimberg and Wesolowsky divided the plane into many rectangular cells by drawing horizontal and vertical lines through each demand point, and then to find the optimal solution. On the contrary, the proposed approach does not need any additional process.

Table 1. Input Data for Sample Problem

i	(x_i, y_i)	r_i	v_i
1	(2, 3)	2	1
2	(4, 4)	1.5	1
3	(5.5, 3.75)	2.25	1
4	(7, 6)	1.5	1
5	(8.25, 2.25)	1.25	1

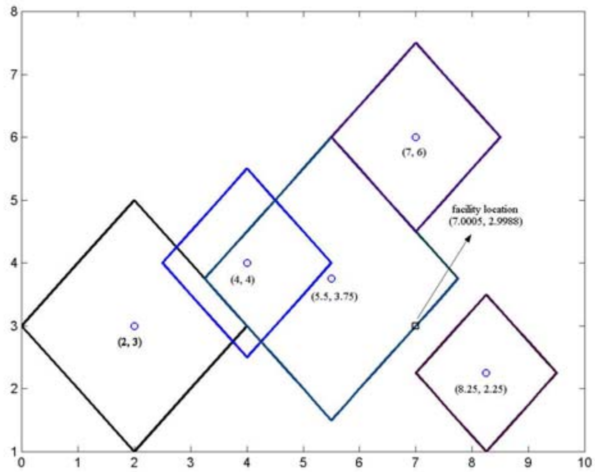

Fig. 1. Five demand points, forbidden regions, and facility location is obtained by the proposed approach.

4. Conclusions

In this paper, an obnoxious facility location problem was solved by the proposed *GA* approach. The problem deals with locating a single facility in the plane in order to minimize a weighted sum of distances between the facility and a set of demand points. The problem possesses a constraint that the location of the facility is restricted to be outside a specified forbidden region around each demand point. The experimental results reveal that the proposed approach provides a simple but effective way to solve the obnoxious facility location problem.

Further work of applying the proposed approach to the Euclidean versions of these problems is in progress.

Acknowledgment

This work was partially supported by National Science Council, Taiwan, R.O.C., under grant NSC 90-2213-E-366-004.

References

[1] Berman, O. and Z. Drezner, "A note on the location of an obnoxious facility on a network," *European Journal of Operational Research*, 120, pp.215-217, 2000.

[2] Brimberg, J. and G. O. Wesolowsky, "The rectilinear distance minisum problem with minimum distance constraints," *Location Science*, 3(3), pp.203-215, 1995.

[3] Brimberg, J. and H. Juel, "A bicriteria model for locating a semi-desirable facility in the plane," *European Journal of Operational Research*, 106, pp.144-151, 1998.

[4] Buchanan, D. J. and G. O. Wesolowsky, "Locating a noxious facility with respect to several polygonal regions using asymmetric distance," *IIE Transactions*, 25, pp.77-88, 1993.

[5] Butt, S. E. and T. M. Cavalier, "An efficient algorithm for facility location in the presence of forbidden regions," *European Journal of Operational Research*, 90, pp.56-70, 1996.

[6] Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[7] Erkut, E. and S. Neuman, "Analytical models for locating undesirable facilities," *European Journal of Operational Research*, 40, pp.275-291, 1989.

[8] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*,, Addison-Wesley, Reading, MA., 1989.

[9] Hamacher, H. W. and S. Nickel, "Restricted planar location problems and applications," *Preprint no. 227*, Universität Kaiserlauter, Fachbereich Mathematik, Erwin-Schrödinger-Strasse, D-6750 Kaiserlautern, 1992.

[10] Konforty, Y. and A. Tamir, "The single facility location problem with minimum distance constraints," *Location Science*, 5(3), pp.147-163, 1997.

[11] Lai, C.-C. and D.-C. Tseng, "An optimal *L*-filter for reducing blocking artifacts using genetic algorithms," *Signal Processing*, 81, pp.1525-1535, 2001.

[12] Man, K. F., K. S. Tang, and S. Kwong, *Genetic Algorithms*: *Concepts and Designs*, Springer-Verlag, London, 1999.

[13] Michalewicz, M., *Genetic Algorithms + Data Structure = Evolution Program*, Springer-Verlag, Berlin, 1992.

[14] Moon, I. D. and S. S. Chaudhry, "An analysis of network location problems with distance constraints," *Management Science*, 30, pp.290-307, 1984.

[15] Muñoz-Pérez, J. and Saameño-Rodríguez, J. J., "Location of an undesirable facility in a polygonal region with forbidden zones," *European Journal of Operational Research*, 114, pp.372-379, 1999.

[16] Pal, S. K. and P. P. Wang, eds., *Genetic Algorithms for Pattern Recognition*, CRC Press, Boca Raton, 1996.

[17] Plastria, F. and E. Carrizosa, "Undesirable facility location with minimal covering objectives," *European Journal of Operational Research*, 119, pp.158-180, 1999.