2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Hybrid Genetic Algorithm for Obstacle Location-Allocation Problem

Jynichi Taniguchi*, Mitsuo Gen*, XiaoDong Wang** and Takao Yokota **

* Graduate School of Information, Production & Systems, Waseda University
taniguchi@po.miyazaki-u.ac.jp (J. Taniguchi)
gen@waseda.jp (M. Gen)
** Department of Industrial & Information Systems Engineering, Ashikaga Institute of Technology
wang1105@hotmail.com (X.D. Wang)

Abstract - **Location-allocation problem is known as one of the important problems faced in Industrial Engineering and Operations Research fields. There are many variations on this problem for different applications, however, most of them consider no obstacle existing. Since the location-allocation problem with obstacles is very complex and with many infeasible solutions, no direct method is effective to solve it. In this paper we propose a hybrid Genetic Algorithm (hGA) method for solving this problem. The proposed hGA is based on Lagrangian relaxation method and Dijkstra's shortest path algorithm. To enhance the proposed hGA, a Fuzzy Logic Controller (FLC) approach is also adopted to auto-tune the GA parameters.**

I. Introduction

Since Cooper presented location-allocation problem [1], this problem has been widely applied in many situations, such as material handling and logistic, design of traffic network, and various facility constructions. Many location-allocation models have been suggested in literatures, but most of them consider the ideal case without obstacles for facility location design. In practice, obstacle or forbidden region constraint usually need to be considered. For example, lakes, rivers, parks, buildings, cemeteries and so on, provide obstacles to facility location design. There are two kinds of obstacles: i) the prohibition of locations; ii) the prohibition of connecting paths. When an obstacle constraint is considered, the location-allocation problem becomes much more complex and very difficult. A primary concern is to reduce the size of the set of feasible locations and then is to locate the optimal solution.

Recently, genetic algorithm (GA) has been proved powerful and robust in finding global optimum and solving complex optimization problems such as location-allocation problem [2] [3]. However, a new trend in GA is to combine the GA approach with other conventional search techniques such as Tabu search, hill-climbing, etc.

In this research, we develop the hGA approach with Lagrangian relaxation, Dijkstra's shortest path algorithm and a FLC for solving the location-allocation problem with obstacles.

This paper is organized as follows: Section 2 gives the mathematical model of this problem. The hGA approach is discussed in Section 3. In Section 4 simulation results are presented and the last Section 5 is a conclusion.

II. Mathematical Model

There are n customers with known locations and m distribution centers (DCs) that must be built to supply some kind of service to all customers. There are also q obstacles representing some forbidden areas. The following assumptions are made to formulate the mathematical model:

- Customer j have service demand d_j $j=1,2,\ldots,n$
- DC i have service capacity q_i $i=1,2,\ldots,m$
- Each customer should be served by only one DC
- DCs should not be built within any obstacle Q_k $k=1,2,\ldots,q$
- Connecting paths between DCs and customers should not be allowed to pass through any of the obstacles.

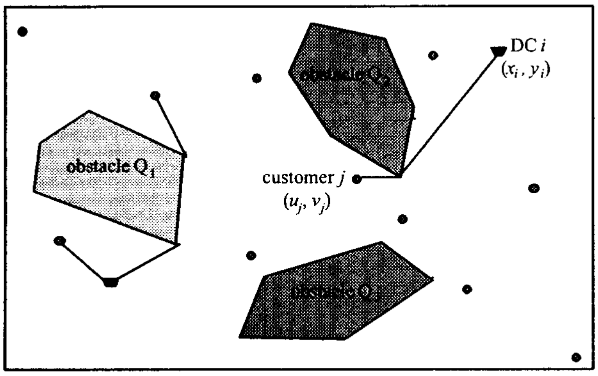

Figure 1: Illustration of the obstacle location-allocation problem

The problem is to choose the best locations for DCs so that the sum of distances among customers and their serving DCs is minimal. It is also assumed that all obstacles can be represented as a convex polygon. We illustrate this case problem with the following Figure 1.

The obstacle location-allocation problem can be formulated as the following model:

$$\min \quad f(\mathrm{D}, z) = \sum_{i=1}^{m}\sum_{j=1}^{n} t(\mathrm{D}_i, \mathrm{C}_j)\cdot z_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} d_j z_{ij} \le q_i, \quad i = 1,2,\ldots,m \tag{2}$$

$$\sum_{i=1}^{m} z_{ij} = 1, \quad j = 1,2,\ldots,n \tag{3}$$

$$\mathrm{D}_i = (x_i, y_i) \notin \{\mathrm{Q}_k \mid k = 1,2,\ldots,q\}, \quad i = 1,2,\ldots,m \tag{4}$$

$$(x_i, y_i) \in \mathrm{R}_\mathrm{T}, \quad i = 1,2,\ldots,m \tag{5}$$

$$z_{ij} = 1 \text{ or } 0, \quad i = 1,2,\ldots,m,\ j = 1,2,\ldots,n \tag{6}$$

where

$C_j = (u_j, v_j)$ the location of the jth customer.

$D_i = (x_i, y_i)$ decision variable, the location of the ith distribution center DC_i should not fall into the inside of any obstacles.

$t(D_i, C_j)$ the shortest connecting path consisted of the set of possible paths between the distribution center DC_i and customer C_j which avoid any obstacle.

R_T total area considering for location and allocation problem.

z_{ij} 0-1 decision variable; $z_{ij}=1$ indicates that the jth customer is served by DC_i, $z_{ij}=0$ otherwise.

Constraints (2) represent that service capacity of each facility should not be exceeded; Constraints (3) represent that each customer should be served by only one facility; Constraints (4) represent that locations of DCs should not be within any obstacle.

III. Design of the Algorithm

As shown above, there are two kinds of decision variables. One is a continuous location variable and another is a zero-one allocation variable. We can see that once the locations of DCs are fixed, we can calculate the investment cost by solving allocation problem with some more efficient traditional optimization techniques.

A. Chromosome Representation

As we know, the chromosome representation is very importantly for the success of GA. Since the location variables are continuous and with some restriction according to obstacles. We use the coordinate location of DC to represent them as follows:

$$c_t = [(x_1', y_1'), (x_2', y_2'), \cdots, (x_i', y_i'), \cdots, (x_m', y_m')]$$

where (x_i', y_i') represents the location of the i-th DC in the t-th chromosome, i=1,2,…,m.

B. Repairing chromosome

When a new position for a DC is generated, it should be known whether or not this position is feasible, that is, whether it at the inside of some obstacle or not. If it is infeasible, we use a simple way to repair it according to the characteristics of our problem. Unlike some other problems, the constraint functions on location variables in this problem are not mathematical formula but geometrical one. So a computationally efficient algorithm is important and necessary.

Proposition 1: Given three points $p_1(x_1, y_1)$, $p_2(x_2, y_2)$, and $p_3(x_3, y_3)$ on the coordinate plane, then the square of the triangle $\Delta p_1 p_2 p_3$ can be calculated as:

$$S_{\Delta p_1 p_2 p_3} = \frac{1}{2}\left|\det\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}\right|$$

Proposition 2: Let $P_{k1}, P_{k2}, \ldots, P_{kp_k}$ be vertices of a p_k-polygon and D_i be a point on the same place, S_{Q_k} be the square of the polygon $P_{k1}, P_{k2}, \ldots, P_{kp_k}$, $S_{\Delta D_i P_{kr} P_{k(r+1)}}$ be the square of triangle $\Delta D_i P_{kr} P_{k(r+1)}$ (r=1,2,..., p_k). Consisting of point D_i and the edge $\overline{P_{kr} P_{k(r+1)}}$ of the polygon (where $P_{k(p_k+1)}$?P_{k1}), then

(1) Point D_i is at the outside the polygon $P_{k1}, P_{k2}, \ldots, P_{kp_k}$ if and only if $S_{Q_k} < \sum_{r=1}^{p_k} S_{\Delta D_i P_{kr} P_{k(r+1)}}$

(2) Point D_i is strictly at the inside of the polygon $P_{k1}, P_{k2}, \ldots, P_{kp_k}$, if and only if

$$S_{Q_k} = \sum_{r=1}^{p_k} S_{\Delta D_i P_{kr} P_{k(r+1)}}, S_{\Delta D_i P_{kr} P_{k(r+1)}} > 0, \quad r = 1,2,\ldots,p_k$$

(3) Point D_i is on an edge of the polygon $P_{k1}, P_{k2}, \ldots, P_{kp_k}$, if and only if

$$S_{Q_k} = \sum_{r=1}^{p_k} S_{\Delta D_i P_{kr} P_{k(r+1)}}, S_{\Delta D_i P_{kr} P_{k(r+1)}} = 0, \quad \exists r \in \{1,2,\ldots, p_k\}$$

According to above Propositions, a simple algorithm is developed which can check whether a point is within a polygon. If location of a DC in a solution is infeasible, *i.e.*, the point falls at the inside of some obstacles, it is replace by one vertex of the obstacle according to move the infeasible location to its nearest vertices of the obstacle. This is shown in following Figure 2.

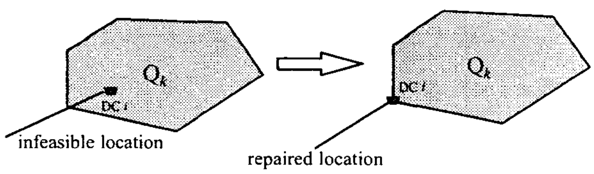

Figure 2: Repairing infeasibility

C. Initialization

We select a point randomly from the square region containing all customers as the initial location of some distribution center and repeat this procedure until all population members are selected.

D. Evaluation of the Chromosome

To determine the fitness value of the chromosome, we use the objective function value that is total distance of the customers to the DCs. To effectively evaluate fitness, efficient solution procedures are necessary to solve the allocation and path selection subproblems under satisfy the obstacle constraints. Once the locations of DCs are fixed and the obstacles are ignored, the allocation subproblem can be solved efficiently by using Lagrangian relaxation method. Since some DCs are overloaded, some customers belonging to overloaded DCs should be assigned to other unsaturated ones in such a way: the distance increment is as small as possible. The path avoiding obstacles can be found on the constructed visible network whose nodes consist of the location points corresponding to the DC and the customer and all vertices of the obstacles, and edges are all line segment between all pairs of points which does not break through any obstacle [4]. For example as shown in Figure 3:

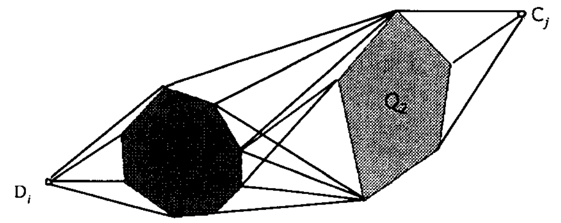

Figure 3: Illustration of the visible network

After the visible network has been constructed, using Dijkstra's shortest algorithm to find the shortest path. It is very easy to find the shortest path between the two points while at the same time avoiding obstacles.

E. Genetic Operators

1) *Crossover Operator*: Suppose that two parents are selected for crossover operation, where genes in the chromosomes of children are decided by following equations:

$$\bar{x}_i^1 = \alpha_i \cdot x_i^1 + (1-\alpha_i) \cdot x_i^2 \qquad \bar{y}_i^1 = \alpha_i \cdot y_i^1 + (1-\alpha_i) \cdot y_i^2$$
$$\bar{x}_i^2 = (1-\alpha_i) \cdot x_i^1 + \alpha_i \cdot x_i^2 \qquad \bar{y}_i^2 = (1-\alpha_i) \cdot y_i^1 + \alpha_i \cdot y_i^2$$

where α_i is a random numbers in (0, 1) (i=1,2,...,m).

2) *Mutation Operator*: We suggest two kinds of mutation operators. One is *subtle mutation* which only gives a small random disturbance to a chromosome to form a new child chromosome. Another is *violent mutation* which give a new child chromosome totally randomly the same as the

initialization.

F. Selection: ES-(μ+λ) selection is adopted to select the better individuals among parents and their children to form the next generation. However, the strategy usually leads to degeneration of the genetic process. In order to avoid this degeneration, a new selection strategy called relative prohibition is suggested.

G. GA Parameter Adaptation Using FLC

The use fuzzy logic controller (FLC) to dynamically control the GA parameters were given in. It has been shown that the fuzzy controlled genetic algorithm gives better results than traditional based genetic algorithm. The main idea of this concept is to determine the value of the GA parameters (i.e., crossover ratio and/or mutation ratio) in the current generation by using the information of previous generation.

IV. Numerical Experiment

To confirm the effectiveness of the proposed method, implemented it in visual Basic language and run on PC Pentium 700. We develop a randomly generated test problem. There are 14 customers whose coordinates of locations and demands are shown in Table 1. There are four obstacles: two small towns, one large lake, and one forbidden region as shown in Table 2. The capacities of each DC are shown in Table 3. The environment parameters of GA are: *pop_size* =20, p_C=0.5, p_M=0.5, *max_gen*=1000. The comparative results are given in Table 4.

Table 1: Coordinates and Demands of Customers

j	(u_j, v_j)	d_j	j	(u_j, v_j)	d_j
1	(4.0, 2.0)	1.0	8	(28.0, 22.0)	1.0
2	(8.0, 6.0)	1.0	9	(32.0, 32.0)	1.0
3	(14.0, 4.0)	1.0	10	(26.0, 40.0)	1.0
4	(11.0, 14.0)	1.0	11	(16.0, 38.0)	1.0
5	(18.0, 16.0)	1.0	12	(12.0, 42.0)	1.0
6	(30.0, 18.0)	1.0	13	(8.0, 38.0)	1.0
7	(36.0, 10.0)	1.0	14	(2.0, 14.0)	1.0

Table 2: Locations of Vertices of Obstacles

k	p_k	Vertex coordinates of obstacles (a_{kr}, b_{kr}) r=1,2,…,p_k
1	5	(8.0, 3.0) (12.0, 3.0) (14.0, 6.0) (10.0, 8.0) (7.0, 5.0)
2	3	(14.0, 8.0) (16.0, 5.0) (15.0, 14.0)
3	4	(6.0, 10.0) (9.0, 9.0) (11.0, 16.0) (7.0, 18.0)
4	5	(13.0, 17.0) (22.0, 20.0) (20.0, 34.0) (18.0, 36.0) (11.0, 25.0)

Table 3: Capacity of DCs

DC i	Capacity q_i
1	4.0
2	6.0
3	4.0

Table 4: Comparison of Heuristic and hGA Methods

Method	Total length Sum	DC locations	Customer allocations
Heuristic [5]	107.0858	1.(15.5, 39.5) 2.(7.0, 6.59) 3.(25.8333, 18.6667)	10,11,12,13 1,2,3,14 4,5,6,7,8,9
hGA	97.422	1.(15.2, 38.9) 2.(9.1, 8.5) 3.(30.1, 18.7)	10,11,12,13 1,2,3,4,5,14 6,7,8,9

V. Conclusion

In this paper, a hybrid Genetic Algorithm (hGA) method for solving obstacle location-allocation problem is explored. We have shown that the computational result by hGA implementation is better than that of the Heuristic method in the numerical experiment.

References

[1] Cooper, L., "Location-Allocation Problems", *Operations Research*, Vol.11, No.3, pp. 331-344, 1963.

[2] Gen, M. and Cheng R., *Genetic Algorithms and Engineering Design*, 1997.

[3] Gen, M. and Cheng, R., *Genetic Algorithms and Engineering Optimization*, 2000.

[4] Gong, D., Evolutionary Computation and Artificial Neural Networks for Optimization Problems and Their Applications, PhD's dissertation, Tokyo Metropolitan Institute of Technology, 1998.

[5] Gong, D., M. Gen, G. Yammazaki, and W. Xu, Hybrid evolutionary method for capacitated location-allocation, *Engineering Design and Automation*, 1997.