Hybridized Parallel Genetic Algorithm for Facility Location Problem

Admi Syarif and Mitsuo Gen

Admi_syarif@yahoo.com

Abstract:
The facility location problem is known as one of the important problem faced in industry. There are many variations of this problem for different applications, however, they can be classified as a capacitated or uncapacitated problem according to its capacity constraint. In this paper, we consider an uncapacitated facility location problem which is known to be an NP-hard problem. A parallel genetic algorithm approach is proposed for solving this problem. To improve the performance of the algorithm, the fuzzy logic controller (FLC) approach is adopted to auto-tune the GA parameters. We test the proposed algorithm by using some standard test problems taken from literature. The computational results given by the proposed algorithm are compared with the known optimal solutions.

Keywords: Facility Location Problem, Parallel Genetic Algorithm, Fuzzy Logic Controller

1. INTRODUCTION

The facility location problem is known as one of the important problem faced in industry. There are many variations of this problem dealing different models, relevant to various situations. However, in most cases, this problem is classified according to the capacity of the facilities. When the facilities have certain capacity, the problem is referred as a capacitated location allocation problem. In this problem, a number of potential facilities with certain limit on capacity such as service centers, plants, distribution centers (DCs) are given and the problem is to assign facilities to the location in such a way that the sum of the fixed cost of opening facilities and variable cost of transporting the customer demand from facilities is minimized. On the other hand, when it is assumed that the facilities have no limit on capacity then the problem is referred as an uncapacitated location/allocation problem (uLAP).

In uLAP, some facilities are located among *n* possible sites and the objective is to satisfy all demand at *m* given location with least cost. The cost here usually consists of both the fixed cost for establishing the facilities and the cost for fulfilling the demand (transportation/

distribution cost). It has been shown that this problem is NP-hard problem [1].

The body of literature on various location/allocation problems is large, however, most of them deal with capacitated location allocation problem. Al-Sultan and Al-Fawzan [2] presented a Tabu search algorithm for solving uLAP. They developed a Net Benefid Heuristic (NBH) algorithm for the uLAP. The improvement of this algorithm is given by Al-Fawzan [3]. Another well known heuristic procedure for solving this problem is also given by Kuehn and Hamburger [4]. Sule [5] developed a heuristic method based on the idea of net saving resulting from the reallocation of facilities. This method has some similarities with Al-Fawzan's method.

Since it was introduced by Holland in 1975 [6], Genetic Algorithm (GA) approach has taken a great attention of researchers. It has been used to solve many difficult combinatorial optimization problems (see for example [7, 8]. When using GA, one of the important factors is a balance between exploitation and exploration in the search space [7]. To provide this balance, determination of design strategy for GA parameters such as population size, maximum generation, crossover probability and mutation probability is one of critical issues. To handle with this problem, several researchers have reported the use of FLC to automatically tune the GA parameters [9-12]. The main idea of FLC is to dynamically change the GA parameters based on the information in the previous generations such as the average fitness of the population.

In this paper, we propose a parallel genetic algorithm approach to solve uLAP. The fuzzy logic controller (FLC) approach is adopted to auto-tune the GA parameters. The proposed algorithm is tested by using some standard test problems taken from literature. The computational results of the proposed algorithm are compared with the known optimal solution taken from literature.

The rest of this paper is organized as follows: In the next Section 2. The Mathematical formulation of this problem is given. We describe the design of our algorithm including the chromosome representation, the GA process and FLC concept for auto-tuning the GA parameters in Section 3. In Section 4, Numerical experiments and comparison with the results of traditional algorithm are presented to demonstrate the efficiency of the proposed method. Finally, some concluding remarks are given in Section 5.

2. PROBLEM STATEMENT

In this section, we shall give the mathematical formulation of the uncapacitated facility location problem as follows: A homogeneous product to be produced in *n* possible sites, and given *m* customers, at known location, characterized by required level of demand. A cost function c_{ij} is associated with distribution cost of serving customer *j* from facility *i*. The fixed cost, $f_{i,}$ is the cost of establishing facility at site *i*. The problem can be mathematically stated as follows:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{i=1}^{n} f_i y_i \qquad (1)$$

s.t.

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j \in J \qquad (2)$$

$$x_{ij} \le y_i \qquad \forall i \in I,\ j \in J \qquad (3)$$

$$x_{ij},\ y_i \in \{0, 1\} \qquad \forall i \in I,\ j \in J \qquad (4)$$

where

x_{ij} = 1 If the customer *j* is served by facility *i*. Otherwise x_{ij} = 0.

y_i = 1 If facility *i*is established. Otherwise y_i = 0.

I = {1, 2, . . ., *n*}

J = {1, 2, . . ., *m*}

In the above model, the equation 1 represents the total cost of establishing the facilities and fulfilling the demand (transportation/distribution cost) to be minimized. The constraint 2 ensures that the demand of each customer is fulfilled by only one facility (i.e. no partial fulfillment of demand is allowed).

3. DESIGN OF ALGORITHM

The uLAP can mathematically be decomposed into two independent problems:

1. Location. Determination of facilities to be established

2. Allocation. For those established facilities, determine the distribution/allocation pattern

It was noted by Al-Fawzan [3] that when the established facilities are known, the distribution/allocation can easily be optimally determined. Following this argument, to solve this problem, the most important this is to determine which facilities to be established.

3.1. Parallel Genetic Algorithm

Since it was introduced by Holland in 1975, Genetic Algorithm (GA) approach has been attracted the attention of researchers and used to solve many difficult combinatorial optimization problems. The GA is an iterative procedure that maintains a number of candidate solutions, called population, over many simulated generations. Each chromosome is represented by a number of strings and undergoes genetic operation such as crossover, mutation and selection for improving the quality of the solution. At each iteration, called generation, each chromosome is evaluated and recombined with others on basis of its overall quality or fitness value in solving the problem. Recently, to improve the performance of the GA method, many variations of GA method are developed such as the concept of parallel GA.

In this paper, we proposed a hybridized parallel genetic algorithm (hp-GA) approach. We used the concept of sub-population to increase the diversity of the chromosome. The chromosome in the generation are divided into two sub-populations. In each sub-population, different kinds of genetic operators are used.

3.1.1 Representation and initialization

Each chromosome is represented by using *n* digit 0-1 variables inclusively. These decision variables represented the opened/closed facilities. In the following figure 1, we give an illustration of the chromosome representation used in this paper.

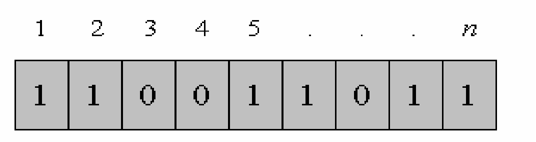

Figure 1. Chromosome representation

As the initial population, we randomly generated two sub-populations of chromosome randomly. Each sub-population here consist of *pop_size* chromosome.

3.1.2 Genetic Operations

Crossover

Crossover is known as the most important recombination operator in GA. We use one point crossover and two-point crossover for first subpopulation and second subpopulation respectively. The illustration of this crossover operations are given in the following Figure 2.

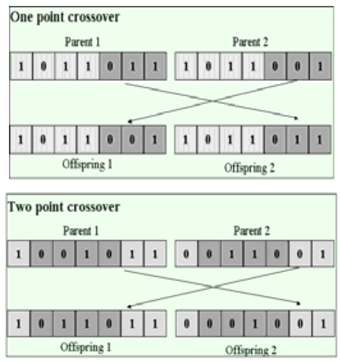

Figure 2. Illustration of crossover operations

Mutation

Mutation is usually used to prevent premature lost of information. It is done by exchanging the information within a chromosome. Inversion mutation operator is used for the first subpopulation and the displacement mutation operator is used for the second subpopulation. The inversion mutation is selects two positions within a chromosome at random and then inverts the sub-string between these two positions. The displacement mutation selects a sub-string at random and inserts it in a random position. In Figure 3, we show the illustration of these two mutation operators

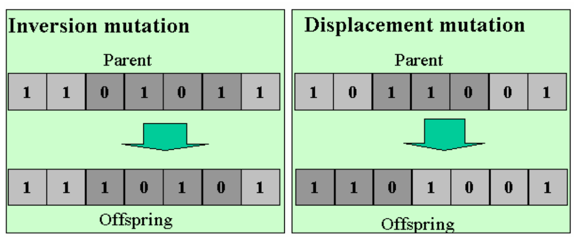

Figure 3. Illustration of mutation operations

3.1.3 Evaluation

As in nature, it is necessary to provide driving mechanism for better individuals to survive. Evaluation is to associate each chromosome with a fitness value that shows how good it is based on its achievement of the objective function. The higher fitness value of an individual, the higher its chances for survival for the next generation. So the evaluation plays a very important role in the evolutionary process. For this problem, we used the objective function as the fitness value. This fitness value is computed during the decoding of the chromosome. The decoding procedure of the chromosome is given as follows:

Procedure: Decoding

Step 1: Determine the opened/closed facilities.
Step 2: Each customer is assigned to the opened facility that gives the least cost
Step 3: Repeat the procedure for all individuals..

3.1.4 Selection

The chromosomes are selected for each subpopulation in the next generation based on their fitness value. Before doing the selection, all chromosome (parent and offspring) in the current generation in combined together. The first and the second subpopulations of chromosome for the next generation are then generated by using elitist selection method and roulette wheel respectively.

3.2. Automatic Fine Tuning for GA Parameters using Fuzzy Logic Controller

It has been noted before that GA has been proven to a versatile approach for searching the global optimality. However, it also has a disheartening weakness in taking too much time to reach a fine satisfactory solution. In order to overcame this weakness, some authors even proposed to combine GA with local search method (*i.e.* hill climbing method) to have a rapid convergence. The pioneer work in extending the fuzzy logic controller (FLC) to dynamically control the GA parameter was done by Lee and Takagi [9] Xu and Vukovich [10], Zeng and Rabenasolo [11] and Wang *et. al*. [12].]. The main idea of using FLC is to automatically adjust the GA parameters (*i.e.* crossover and mutation) during the evolutionary process.

In our implementation of fuzzy logic controller, we make modification on Wang *et al.*'s concepts [12] to regulate automatically the GA parameters, crossover ratio p_C and mutation ratio p_M. The heuristic updating principles for the crossover and mutation ratio are to consider changes in the average fitness of the populations.

As the inputs to the crossover fuzzy logic controller are changes in average fitness at consecutive two generations and the output is the change in crossover ratio $\Delta c(t)$. Based on a number of experiment and domain expert opinion, the input values are respectively normalized into the range [-4.0,4.0] according to their corresponding maximum/minimum values. The membership function μ of fuzzy all input and output linguistic variables are illustrated in Figure 4.

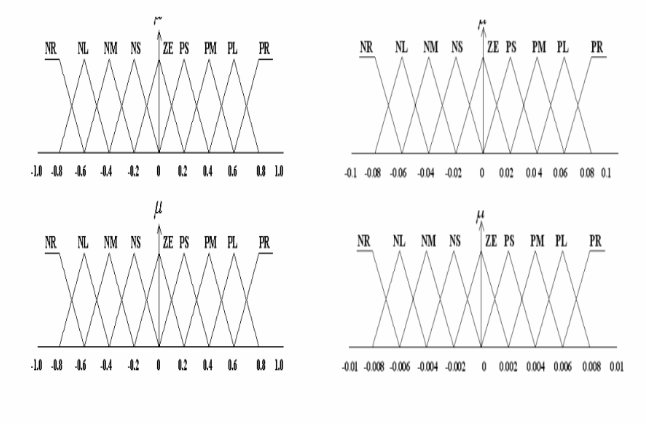

Figure 4. Membership function for $\Delta f(v,t-1), \Delta f(v,t), \Delta c(t), \Delta m(t)$

where:

NR--Negative larger, NL ---Negative large,
NM --- Negative medium NS -- Negative Small,
ZE --- Zero, PS ---- Positive small,
PM -- Positive medium, PL --- Positive large,
PR --- Positive large

For simplicity, we set up a look-up table for fuzzy logic controller action as shown in Table 1. Using look-up table in the Table 1, the value of $Z(i,j)$ is found.

Table 1. Fuzzy Decision

$Z(i,j)$		i								
		-4	-3	-2	-1	0	1	2	3	4
j	-4	-4	-3	-3	-2	-2	-1	-1	0	0
	-3	-3	-3	-2	-2	-1	-1	0	0	1
	-2	-3	-2	-2	-1	-1	0	0	1	1
	-1	-2	-2	-1	-1	0	0	1	1	2
	0	-2	-1	-1	0	2	1	1	2	2
	1	-1	-1	0	0	1	1	2	2	3
	2	-1	0	0	1	1	2	2	3	3
	3	0	0	1	1	2	2	3	3	4
	4	0	1	1	2	2	3	3	4	4

To get the output of this fuzzy logic scheme, we use the following formula:

$$\Delta c(t) = 0.02 \times Z(i, j) \qquad (5)$$

where $i, j \in \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$.

After getting the output of this fuzzy logic controller system, then the crossover ratio for the next generation is regulated as follows:

$$p_C\ (t+1) = p_C(t) + \Delta c(t) \qquad (6)$$

We use the similar concept to regulate the mutation ratio. The input is the same as in the fuzzy logic controller for crossover ratio. The output is the change in the mutation value. The membership function μ of $\overline{f}(v; t-1)$, $\overline{f}(v; t)$, $\Delta c(t)$ and $\Delta m(t)$ are also given in Figure 4. The fuzzy decision table and look-up table of this mutation fuzzy controller are the same concept as in crossover fuzzy logic controller. However the control action is done using the following formula:

$$\Delta m(t) = 0.002 \times Z(i, j) \qquad (7)$$

where $i, j \in \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$.

The mutation ratio for the next generation is regulated as follows:

$$p_M\ (t+1) = p_M\ (t) + \Delta m(t) \qquad (8)$$

1.1. Procedure: Auto-tuning for GA parameters

Step 1: Calculate the change in average fitness value at the current generation and the previous generation.

Step 2: Determine the control action value of the current generation and the previous generation by using Fuzzy decision tables in [14]

Step 3: After scaling the control action value, calculate the changes of crossover probability and mutation probability.

Step 4: Update the crossover and mutation ratio

Step 5: Return to GA loop

3.3. Overall Procedure

Let $SP_1(t)$ and $SP_2(t)$ be the first and the second subpopulations of chromosomes for iteration t respectively, $O_1(t)$ and $O_2(t)$ be the generated chromosomes at iteration t. The overall procedure of hybrid genetic algorithm based on the spanning tree encoding is summarized as follows:

Procedure: hp-GA
begin
  t =0
  initialize $SP_1(t)$, $SP_2(t)$
evaluate $SP_1(t)$, $SP_2(t)$
  while (not termination condition) **do**
    recombine $SP_1(t)$, $SP_2(t)$;to generate $O_1(t)$, $O_2(t)$
    evaluate $O_1(t)$ and $O_2(t)$
    combine $SP_1(t)$, $SP_2(t)$, $O_1(t)$ and $O_2(t)$
      t =t+1
      select $SP_1(t+1)$, $SP_2(t+1)$
      determine the GA parameters using FLC;
    end
end

4. NUMERICAL EXPERIMENTS

The proposed algorithm was implemented in visual C language and run on PC Pentium 700. We tested our algorithm using a set of standard algorithm taken from OR-libraries [13, 14]. The GA parameters are set as follows: crossover probability (p_C) =0.5 and mutation probability (p_M) =0.3. To confirm the effectiveness of this algorithm, for each test problem, we run our proposed algorithm 10 times. We compared our results with those of known optimal solutions. We summarize the results of the experiments in Table 3.

Table 3. Computational results for the problem taken from OR library

Data file	*M*	*n*	*Pop_size*	*max_gen*	best	Average	%Err.	Appearance	ACT[a]	Optimal[b]
Cap 71	16	50	100	3000	932615.75	932615.75	0.00	10	8.68	932615.75
Cap 72	16	50	100	3000	977799.40	977799.40	0.00	10	9.03	977799.40
Cap 73	16	50	100	3000	1010641.45	1010641.45	0.00	10	7.89	1010641.45
Cap 74	16	50	100	3000	1034976.98	1034976.98	0.00	10	8.45	1034976.98
Cap 101	25	50	120	4000	796648.44	796734.52	0.09	9	49.37	796648.44
Cap 102	25	50	120	4000	854704.20	855801.20	0.11	9	51.28	854704.20
Cap 103	25	50	120	4000	893782.11	894937.58	0.13	8	58.76	893782.11
Cap 104	25	50	120	4000	928941.75	928941.75	0.00	10	38.90	928941.75
Cap 131	50	50	150	5000	793439.56	794217.63	0.10	4	203.34	793439.56
Cap 132	50	50	150	5000	851495.33	853362.75	0.21	5	186.45	851495.33
Cap 133	50	50	150	5000	893076.71	894252.26	0.13	3	247.83	893076.71
Cap 133	50	50	150	5000	928941.75	930432.32	0.16	4	192.67	928941.75

[a] in second on IBM PC Pentium III 700 MHz.

[b] taken from OR –library

In that table, the following notations are used:

data file : Name of data file in OR-Lib
m : Number of facilities
n : Number of customers
best : The best result using our algorithm
Average : The average result of our algorithm
%Error: ((Optimal-Average)/Optimal)*100%
Appearance: The number of appearance of the optimal solution
ACT : Average computational time
Optimal : Optimal solution from OR-Library

From Table 3, one can notice the impressive performance of our algorithm in searching the optimal solution. It is clear here that our propose algorithm can gives the optimal solution in almost all of the time. Moreover, it is also shown that the proposed algorithm is fairly fast.

V. Conclusion

In this paper, we proposed a new approach by using genetic algorithm to solve uncapacitated location-allocation problem (uLAP). In order to improve the efficiency of genetic algorithm, Fuzzy Logic Controller was hybridized to the evolutionary process for making auto-tuning of the GA parameters. We carried out the numerical experiments by using several standard test problems taken from literature. We have shown that the proposed method can search the optimal solution in almost all of the time within a reasonable computational time.

Acknowledgement

This research was supported by the International Scientific Research Program, the Grant-in-Aid for Scientific Research (No.10044173:) by the Ministry of Education, Science and Culture, the Japanese Government. This paper have also been presented at Belman Continunium, Beijing (2002). The work of the first author was also supported by Development Undergraduate Education Project, (Indonesian Government and World Bank Project).

References

1. Shmoys, D. B., Tardos, E. and Aarday, K. I. (1997) Approximation algorithm for facility location problem, *Proceedings of 29th Annual ACM Symposium on Theory of Computing*, pp. 265-274.
2. Al-sultan, K. S. and Al-Fawzan, M. A. (1999) A new approach to the Uncapacitated Facility Location Problem, *Annal. Of Operations Research*, 86, pp. 91-103
3. Al-Fawzan, M. A., (2001) An improved heuristic for the uncapacitated facility location problem, *International Journal of Industrial Engineering*, 8 (2), pp. 115-121.
4. Kuehn, A. and Hamburger, M. J. (1963). A Heuristic program for locating warehouse, *Management Science*, 9 (4), pp. 643-666.
5. Sule, D. R. (1988), *Manufacturing Facilities*, PWS-KENT Pub. Co., Boston Massachusetts.

6. Holland, J. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.
7. Gen, M. and R. Cheng, (1997) *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York.
8. Gen, M. and Cheng, R., (2000) *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, New York.
9. Lee, M., and Takagi H., (1993**)** Dynamic Control of Genetic Algorithm Using Fuzzy Logic Techniques, *Proceedings of the 5th International Conference on Genetic Algorithms,* pp.76-83.
10. Xu, H., and Vukovich, G., (1994) Fuzzy Evolutionary Algorithm and Automatic Robot Trajectory Generation, *Proceedings of the First IEEE Conference on Evolutionary Computation*, pp 595-600.
11. Zeng, X., and Rabenasolo, B., (1997) A Fuzzy Logic Based Design for Adaptive Genetic Algorithms, *Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing*, pp. 660-664.
12. Wang, P. T., Wang, G. S., and Hu, Z. G., (1997) Speeding Up the Search Process of Genetic Algorithm by Fuzzy Logic, *Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing*, pp. 665-671.
13. Beasley, J. E.m (1990). OR Library: distribution Test Problem by Electronic Mail. *Journal of Operational Research Society*, 41 (11). Pp/ 1069-1072.
14. http://www.ms.ic.ac.uk/jub/pub

BIOGRAFI PENULIS

Admi Syarif is a Ph.D. candidate in Industrial and Information System Engineering, Ashikaga Institute of Technology, Japan. Since 1992, he has been a lecturer at Dept. of Mathematics, Faculty of Mathematics and Sciences, Lampung University, Indonesia. His research interests include mathematical programming, combinatorial optimization, Genetic Algorithm and supply chain management. He has been one of the referees for *European Journal of Operational Research* and *Central European Journal of Operational Research.* His papers have been published and accepted in *International Journal of Inverse Problems, International Journal of Applied Mathematics, Journal for Analysis and its Applications, Journal of Plant Engineer Society of Japan, International Journal of Computer and Industrial Engineering, International Journal of Intelligent Manufacturing, International Journal of Smart Engineering, International Journal of Knowledge Engineering and System Design and so on.* He also has presented his papers in several international conferences in Japan, USA (Los Angeles, San Francisco, Las Vegas, Florida, Hawaii), Hong Kong, Australia, Singapore, Beijing, Greece, Canada, South Korea, Taiwan, Ireland and so on. One of his papers also appeared in the book *Fuzzy Set and System* by Springer Verlag. Now he is the member of Indonesian Mathematical Society, Japan Industrial and Management Association, Japan Fuzzy System Society and Electronic Engineer Society of Japan.

Mitsuo Gen a professor at Graduate School of Information Production and System, Waseda university, Japan. He was a visiting professor at the Department of Industrial Engineering and Operations Research, University of California at Berkeley, USA. His research interests include Reliability Design, Fuzzy Operations Research, Network Optimization, Neural Networks, and Genetic Algorithms. He has published several books including the books of *Genetic Algorithms and Engineering Design*, 1997 and *Genetic Algorithms and Engineering Optimization*, 2000 with Dr. R. Cheng by John Wiley & Sons, Inc., respectively. He has published dozens of articles in several international journals such as *IEEE Transaction on Reliability, European Journal of Operational Research, Computers & Industrial Engineering, Soft Computing, International Journal of System Science, Fuzzy Sets & Systems, Networks, International Journal of Logistics*, and so on. He is a member of the editorial board of *International Journal of Computers & Industrial Engineering* and the editor of *OR Spectrums* and *International Journal of Logistics*. He is also a member of IEEE, IIE, Japan Industrial Management Association, Japan Society for Fuzzy Theory and Systems, Information Processing Society of Japan, Institute of Electronics, Information, and Communications Society, Operations Research Society of Japan, Japan Society for Artificial Intelligence, Reliability Engineering Assoc. of Japan, and Personal Computer User Technology Association.