

# Solving Capacitated P-Median Problem using Genetic Algorithm

K. Ghoseiri<sup>1</sup>, S. F. Ghannadpour<sup>2</sup>

<sup>1</sup>School of Railway Engineering, Iran University of Science and Technology, Tehran, Iran

<sup>2</sup>School of Railway Engineering, Iran University of Science and Technology, Tehran, Iran

**Abstract** – Capacitated p-median problem (CPMP) is an important variation of facility location problem in which  $p$  capacitated medians are economically selected to serve a set of demand vertices so that the total assigned demand to each of the candidate medians must not exceed its capacity. This paper presents a genetic algorithm to solve the CPMP. Two different assignment techniques namely, classical assignment method and assignment through urgencies are used to assign the demand points to the  $p$  selected medians. The behavior and efficiency of the assignment scenarios are examined and compared on CPMP. According to the results, the classical scenario shows superiority in time consuming, whereas the assignment through urgencies scenario is absolutely superior in quality of the obtained solutions over the classical one. In order to check for quality and validity of the suggestive method, we compare the final solution produced over the 10 test problems of Osman and Christofides [1]. Comparison of the results indicates good quality and solutions.

**Keywords** - capacitated p-median problem, facility location, genetic algorithm

## I. INTRODUCTION

Facility-location problems have several applications in telecommunication, transportation, scheduling and distribution problems. One important way to measure the effectiveness of facility location is by evaluating the average (total) distance decreases, the accessibility and effectiveness of the facility increases. This relationship applies to both private and public facilities such as supermarkets, post office, as well as emergency service centers, for which proximity is desirable. The p-median problem that is one of the most well-known facility location problems takes this measure into account and is defined as: minimize the average (total) distance between the demands and the selected facilities. We illustrate a typical p-median model in Fig. 1. The total cost of the solution presented is the sum of the distance between demand points and selected location represented by a black line.

In the non-capacitated p-median problem, one considers that each facility candidate to median can satisfy an unlimited number of demand points. By contrast, in the capacitated p-median problem each candidate facility has a fixed capacity, i.e. a maximum number of demand points that it can satisfy. The p-median problem is *NP-hard* [2]. Therefore, even heuristic methods specialized in solving this problem require a considerable computational effort.

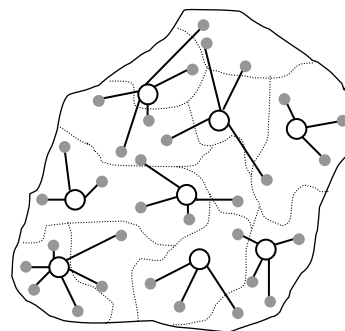


Fig.1. Typical output for the p-median problem

In this work we solve the capacitated p-median problem (CPMP) using genetic algorithm. Two different assignment techniques namely, classical assignment method and assignment through urgencies are used to assign the demand points to the  $p$  selected medians and to compare their behavior and efficiency on this problem. In order to check the quality and validity of the overall approach we compare the final solution produced over the test problems of [1].

The remaining parts of paper are organized as follows. Section 2 reviews literature of p-median problem and its recent developments in solution approaches. Section 3 states the p-median problem and its mathematical formulation. First, in section 4 we present the genetic solution approach and two assignment methods for CPMP and then in section 5, we check the validity of this solution approach and compare the efficiency of two assignment techniques on the test problems. Section 6 provides the concluding remarks.

## II. LITERATURE REVIEW

The p-median problem is well known in the OR literature and has received much attention in the past decades (see e.g. [3]). An extensive bibliography of related problems, and also a set of test problems were presented in [2]. Some earlier approaches applying Lagrangean heuristics to the CPMP are proposed in [4] and in [5]. But recent approaches tend to use metaheuristics, such as simulated annealing and tabu search ([6], [7] and [8]). To solve the p-median problem using genetic algorithm, [9] applied cluster seed points and used the genetic algorithm to select the most suitable cluster seeds (medians). The remaining vertices are

assigned to clusters according to their similarity to the cluster seeds or their ability to improve the objective function. A genetic algorithm for the capacitated p-median problem is developed by [10] that assigned vertices to the nearest median that is not already full. Also it introduced a new genetic operator called heuristic hypermutation that improved the fitness of certain percentage of genes. A fixed-length subset genetic algorithm for the p-median problem is presented in [11]. Although, actual computational results are not presented, but the authors claim that their method outperforms the traditional genetic algorithm and is able to find solutions very close to optimal for most problems. The constructive genetic algorithm that it is different from the traditional genetic algorithm is presented in [12] that it used a dynamic population. Two separate fitness functions are described and the clustering problems are formulated as bi-objective optimization problems. The literature has shown richness of usage of the genetic algorithm.

Main contribution of this work includes replacement of the traditional assignment method with the assignment through urgencies to assign the demand points to the p selected medians. Although the assignment through urgencies has previously been used in [13] on a Multi-Depot vehicle routing Problem with Time Windows, but it is for the first time that it is used in capacitated p-median problem.

### III. P-MEDIAN PROBLEM

Informally, the goal of the p-median problem is to determine  $p$  facilities in a predefined set with  $n$  ( $n > p$ ) candidate facilities in order to satisfy a set of demands, so that the total sum of distances between each demand point and its selected facility is minimized. Formally, assuming all vertexes of a graph are potential medians, the p-median problem can be formally defined as follows. Let  $G = (V, A)$  an undirected graph where  $V$  are the vertexes and  $A$  are the edges. The goal is to find a set of vertexes  $V_p \subset V$  (median set) with cardinality  $p$ , such that: (1) the sum of the distance between each remaining vertex in  $\{V - V_p\}$  (demand set) and its selected vertex in  $V_p$  be minimized; and (2) all demand points are satisfied without violating the capacity restrictions of the median facilities. By comparison with the p-median problem, the capacitated p-median problem has the following additional constraints: (1) each facility can satisfy only a limited number of demands (capacity restriction); and (2) all demand points must be satisfied by respecting the capacities of the facilities selected as medians. We present below a formulation of the capacitated p-median in terms of Integer Programming.

$$v(p) = \text{Min} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} \quad (1)$$

Subject to :

$$\sum_{j \in V} x_{ij} = 1 \quad ; \quad i \in V \quad (2)$$

$$\sum_{j \in V} x_{jj} = p \quad (3)$$

$$\sum_{i \in V} q_i x_{ij} \leq Q_j x_{jj} \quad ; \quad j \in V \quad (4)$$

$$x_{ij} \in \{0,1\}; i \in V, j \in V \quad (5)$$

Where:  $V = \{1, 2, \dots, n\}$  is the set of vertexes to allocate and also of possible medians, where  $p$  medians will be located,  $q_i$  is the demand of vertex  $i$ ,  $Q_j$  is capacity of median  $j$ ,  $[d_{ij}]_{n \times n}$  is a distance matrix,  $[x_{ij}]_{n \times n}$  is the allocation matrix, with  $x_{ij}=1$  if demand point  $i$  is allocated to median  $j$ , and  $x_{ij}=0$ , otherwise;  $x_{jj}=1$  if median  $j$  is selected and  $x_{jj}=0$ , otherwise.

Constraint (2) imposes that each demand point is allocated to only one median. Constraint (4) ensures that  $p$  medians are selected. Constraint (4) imposes that a total median capacity must be respected, and (5) provides the integer conditions.

### IV. GENETIC ALGORITHM FOR CPMP

The genetic algorithm (GA) is a class of adaptive heuristics based on the drawing concept of evaluation – "survival of the fitness", and it has been developed by [14] at the University of Michigan in 1975. The genetic algorithm that used in this paper is as follows.

#### A. Individual representation

Each individual (chromosome) has exactly  $p$  genes, where  $p$  is the number of medians, and the allele of each gene represents the index (a unique ID number) of a facility selected as median. In this GA the genome is interpreted as a set of facility indices, in the mathematical sense of set, there are no duplicated indexes and there is no ordering among the indices.

#### B. Fitness evaluation

Once the medians are selected, each vertex is assigned to the medians (as it is presented in the next two section two different assignment are used) and once each vertex is assigned, the fitness of a chromosome can be computed by calculating the sum of Euclidean distances between each vertex and its assigned median. The minimum sum is the optimal solution.

#### C. Classical Assignment method

In this approach, each vertex from the demand set that selected randomly should assign to its nearest median. However, since each median has a fixed capacity, some vertices will have to be assigned to the second (or third, fourth ...) nearest median to it. Fig. 2 shows the procedure of this method. In Fig. 2,  $S_k$  is a chromosome that in which  $V_m$  is the median set corresponding to alleles of  $S_k$ ,  $Q_m$  is the set of median capacity,  $V_D$  is the demand point set,  $q_{D_j}$  is the amount of demand in node  $D_j$ ,  $q_{M_i}$  is the amount of demand in median  $M_i$  and  $C_r$  is the cluster of median  $r$  and its demand points.

```

Read
     $S_k$ ,
     $V_m(S_k) = \{M_1, M_2, \dots, M_p\}$ ,
     $Q_m(S_k) = \{Q_1, Q_2, \dots, Q_p\}$ ,
     $V_D(S_k) = \{D_1, D_2, \dots, D_{|V_D(S_k)|}\}$ ,
     $q_{D_j}$ ,  $j = 1, \dots, |V_D(S_k)|$ ,
     $q_{M_i}$ ,  $i = 1, \dots, p$ ,
for  $i = 1 : p$ 
     $Q_i := Q_i - q_{M_i}$ ,
     $C_i(S_k) := \{M_i\}$ ,
end - for
for  $j = 1 : p$ 
     $r := \text{index} \left[ \text{Min}_{i=1, \dots, p} \{d_{M_i D_j} \mid Q_i - q_{D_j} \geq 0\} \right]$ ,
     $Q_r := Q_r - q_{D_j}$ ,
     $C_r(S_k) := C_r(S_k) \cup \{D_j\}$ ,
end - for
    
```

Fig.2. Procedure of Classical Assignment Method

#### D. Assignment through urgencies

The urgency is a way to define a precedence relationship between vertexes; the urgency to be assigned could also be viewed as a priority. This precedence relationship determines the order in which vertexes are assigned to medians. The vertexes with the highest urgency are assigned first. One of important way to calculate the urgencies; is the Simplified parallel assignment (see [15]) that in which according to relation (6), two medians are involved in the evaluation of the urgency:

$$\mu(c) = d(c, d''(c)) - d(c, d'(c)) \quad (6)$$

Where  $d(c, d''(c))$  is the distance between customer  $c$  and its second closest depot and  $d(c, d'(c))$  is the distance between customer  $c$  and its closest depot.

Fig. 3 gives an example of urgency for one vertex, represented by the solid black line, because in this algorithm the urgency is calculated as the difference between the distances to a vertex's closest and second closest median. The dotted circle shows the distance to the vertex's closest median.

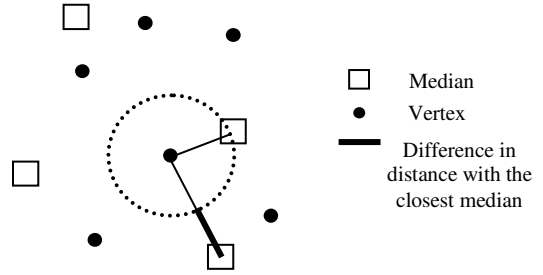


Fig.3. Calculate the urgency of one vertex

#### E. Selection

Genetic algorithm consists of three primary operators: 1) Reproduction, 2) Crossover and 3) Mutation.

Reproduction is the process through which solution characteristics are passed from one generation to the next. The solution chosen for reproduction can be selected in a manner that is considered biased random. One way to think about this is in terms of a roulette wheel. Each solution in the set makes up a different section of the wheel. The best solutions each respectively take up a larger piece of the pie. When the ball is spun, the outcome is random, but the better solutions will be chosen more often as they have a larger probability.

#### F. Crossover

The crossover that used in this work, initially computes two exchange vectors, one for each parent, as follows. For each gene of parent 1, operator checks whether the allele (facility index) of the gene is also present (in any position) at the genome of parent 2. if not, that facility index is copied to the exchange vector of parent 1. This means that facility index may be transferred to parent 2 as a result of crossover, since this transfer would not create any duplicate facility indices in parent 2's genotype. The procedure is performed for each facility index in the genotype of parent 2. Once the facility indices that can be exchanged have been identified, the crossover operator can be applied as follows.

No fixed crossover probability is used in here and crossover is performed whenever the two parents are not equal to each other, i.e. if the two parents are equal to each other, one of the parents is reproduced unaltered for the next generation and the other parent is deleted, to avoid that duplicate individuals be inserted into the population.

#### G. Mutation

Mutation is performed as follows. The gene being mutated has its current allele replaced by another randomly – generated allele (a facility index). Subject to the restriction that the new facility index is not present in the current genotype of the individual.

### H. Accepting

In this work the offspring that produced by crossover and mutation is inserted into the population only if they have a better (smaller) fitness than the worst individual of the current population.

### I. Improvement heuristic method

We also use improvement heuristic method in order to improve the each individual obtained through crossover and mutation. So this operator is applied with a fixed probability to each iteration and starts by randomly selecting a percentage of the individuals of the population. Then it tries to improve the fitness of each of the selected individuals as follows. For each gene of the individual, it tries to replace its facility index by each facility index that is not currently present in the genotype of the individual. For each gene, the replacement that most improves the individual's fitness is performed. In brief the flowchart of solution procedure is shown in Fig. 4.

## V. COMPUTATIONAL ANALYSIS

To check for quality of the solutions and the GA algorithm, we solve 10 test problems that are introduced in [1] and then compare the results with the best solution of those problems. The problems include 50 nodes in the network and the capacity of each median is equal to 120. Also the amount of  $p$  for all problems is considered to 5 and the demand amount of each vertex is different in each problem. Data sets for these instances can be obtained from <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>.

It is notable that the best computational results for these 10 test problems are calculated using Euclidean distances and reported in [1] after they are rounded down to the nearest integer.

The solution method for these problems in two different scenarios is programmed and implemented by MATLAB software. In the first scenario, it is assumed that the classical assignment method is used to allocate the demands points to medians whereas the Assignment through urgencies methods is used in second scenario. The parameters that used in this model are empirically set as follows.

- Population size=100.
- Number of iterations= 500.
- Probability of mutation= 1%.
- Probability of improvement operator= 0.5%.
- Number of chromosomes that are selected for undergoing heuristic improvement=10% of population size

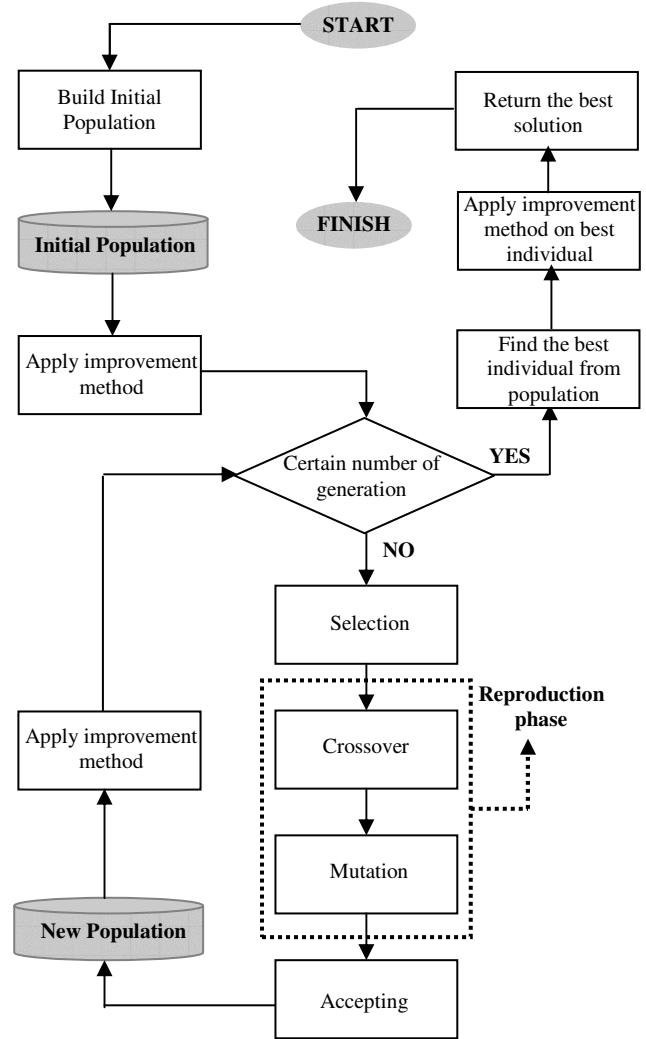


Fig.4. Flowchart of the solution method

The results, which are the best solution with 500 generations at the fifth running of the algorithm, are reported in Table 1.

TABLE I  
THE RESULT OF THE ALGORITHM

ID #	Best value	Scenario 1		Scenario 2	
		Fitness-value	Error %	Fitness-value	Error %
1	713	736	3.1	728	2.1
2	740	758	2.4	758	2.4
3	751	789	5	768	2.2
4	651	671	3	669	2.7
5	664	717	7.9	687	3.4
6	778	814	4.6	796	2.3
7	787	824	4.7	811	3
8	820	858	4.6	842	2.6
9	715	739	3.3	732	2.3
10	829	872	5.1	860	3.7

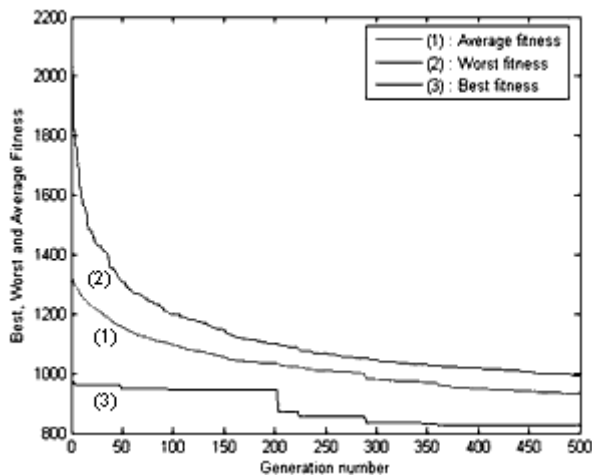


Fig.5. Convergence diagram for the 7<sup>th</sup> test problem

Concerning to Table 1, the proposed method in scenario 2 has produced appropriate solutions with an acceptable percentage of error. Comparison of results shows that the second scenario is thoroughly superior to the first one and has generated acceptable solutions. But, in respect of the CPU time, the maximum solving time spent for finding the solution among the ten problems in scenario 1 is 37s whereas in scenario 2 is 250s which denotes the superiority of scenario 1 to 2 in time consuming. Therefore we can say that the proposed algorithm has generated acceptable solutions in respect of both the quality and CPU time.

To demonstrate the effectiveness of recombination operators and proposed method, the convergence diagram of the best, worst and average fitness value for the 7<sup>th</sup> test problem in each generation are plotted in Fig. 5.

## VI. CONCLUSION

Capacitated p-median problem (CPMP) is an important variation of facility location problem in which p capacitated medians are economically selected to serve a set of demand vertices so that the total assigned demand to each of the candidate medians must not exceed its capacity. This paper presented a genetic algorithm to solve the CPMP. Two different assignment techniques namely, classical assignment method and assignment through urgencies were used to assign the demand points to the p selected medians. The behavior and efficiency of the assignment scenarios were examined and compared on CPMP. According to the results, the classical scenario showed superiority in time consuming, whereas the assignment through urgencies scenario was absolutely superior in quality of the obtained solutions over the classical one. In order to check for quality and validity of the suggestive method, we compared the final solution produced over the 10 test problems of Osman and

Christofides [1]. Comparison of the results indicated good quality and solutions.

## REFERENCES

- [1] I.H. Osman, and N. Christofides, "Capacitated clustering problems by hybrid simulated annealing and tabu search", *International Transactions in Operational Research*, Vol. 1, No. 3, pp317-336, 1994.
- [2] O. Karive, and S.L. Hakimi, "The p-median problem. In: *An Algorithmic approach to network location problems*", SIAM Journal on Applied Mathematic, Philadelphia, 37, 539-560, 1979.
- [3] P. Mirchandani, and R. Francis, "Discrete location theory", Wiley-Interscience, 1990.
- [4] Y.A. Koskosidis, and Powell, "Clustering algorithm for consolidation of customers orders into vehicle shipments", *Transportation Research B*, 26, 365-379, 1992.
- [5] J.M. Mulvey, and M.P. Beck, "Solving capacitated clustering problems", *European Journal of Operational Research* 18, 339-348, 1984.
- [6] K.E. Rosing, C.S. Revelle, E. Rolland, D.A. Schilling, and J.R. Current, "Heuristic concentration and tabu search: A head to head comparison", *European Journal of Operational Research* 104(1), 93-99, 1998.
- [7] S. Salhi, "Defining tabu list size and aspiration criterion within tabu search methods", *Computer and Operation Research* 29(1), 67-89, 2002.
- [8] T.V. Levanova, and M.A. Loresh, "Algorithm of ant system and simulated annealing for the p-median problem", *Automation and Remote Control*, 65(3), 431-438, 2004.
- [9] Y. Chiou, and L.W. Lan, "Genetic clustering algorithm", *European Journal of Operational Research* 135(2), 413-427, 2001.
- [10] E.S. Correa, M.T.A. Steiner, A.A. Freitas, and C. Carnieri, "A genetic algorithm for solving a capacitated p-median problem", *Numerical Algorithms*, 35, 373-388, 2004.
- [11] A. Lim, and Z. Xu, "A fixed-length subset genetic algorithm for the p-median problem", *Lecture Notes in Computer Science*, 2724, 1596-1597, 2003.
- [12] L.A.N. Lorena, and J.C. Furtado, "Constrictive genetic algorithm for clustering problems", *Evolutionary Computation*, 9(3), 309-328, 2001.
- [13] O. Viera, and L. Tansini, "Adapted Clustering Algorithms for The assignment Problem in the MDVRPTW", Working Paper, 2004.
- [14] J.H. Holland, "Adaptation in natural and artificial system", Ann Arbor, Michigan: The University of Michigan Press, 1975.
- [15] J. Potvin, and J. Rousseau, "A parallel route building algorithm for the VRPTW", *EJOR* 66, 331-340, 1993.