ISSN 2047-3338

Optimizing Facility Location Problem Using Genetic Algorithm

[1]Rajni Pasricha and [2]Vaishali Wadhwa
[1]Lecturer in NC College of Engineering, India
[2]Professor in NCCE, India

***Abstract*– Facility location problem have several application in telecommunication, transportation, distribution etc. One of the most well-known facility location problems is the p-median problem. We use genetic algorithm to solve the capacitated p-median problem. In this paper we solve a real problem and give their computational results.**

***Index Terms*– Optimizing, Solution, Algorithm and Problem**

I. INTRODUCTION

FACILITY-location problems have several applications in telecommunications, industrial transportation and distribution, etc. One of the most well-known facility location problems is the p-median problem. This problem consists of locating *p* facilities in a given space (e.g. Euclidean space) which satisfy *n* demand points in such a way that the total sum of distances between each demand point and its nearest facility is minimized. In the no 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* (Kariv and Hakimi, 1979). Therefore, even heuristic methods specialized in solving this problem require a considerable computational effort.

II. THE 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 nearest facility is minimized. The *p* facilities composing a solution for the problem are called medians.

Formally, assuming all vertexes of a graph are potential medians, the p-median problem can be 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 $Vp \in V$ (median set) with cardinality *p*, such that the sum of the distance between each remaining vertex in $\{V - Vp\}$ (demand set) and its nearest vertex in *Vp* be minimized.

We present below a formulation of the p-median problem in terms of Integer Programming proposed by Revelle and Swain in 1970. This formulation allows that each vertex be considered, at the same time, as demand and facility (potential median), but in many cases (including our real world application) demand and facilities belong to disjoint sets.

$$\text{Min} \sum_{i=1}^{n}\sum_{j=1}^{n} aidijxij \quad (2.1)$$

Subject to:

$$\sum_{j=1}^{n} xij \quad (2.2)$$

$$xij \le yj, \quad i, j = 1,2 \dots n \quad (2.3)$$

$$\sum_{j=1}^{n} yj = p \quad (2.4)$$

$$xij, yj \in \{0,1\}, i, j = 1,2 \dots . n \quad (2.5)$$

Where,

n = total number of vertexes in the graph

ai = demand of vertex j

dij = distance from vertex i to vertex j

p = number of facilities used as medians

$$xij = \begin{cases} 1, \text{if the vertex } i \text{ is assigned to facility } j \\ 0, \text{otherwise} \end{cases}$$

$$yj = \begin{cases} 1, \text{if the vertex } j \text{ is used as a median} \\ 0, \text{otherwise} \end{cases}$$

The objective function (2.1) minimizes the sum of the (weighted) distances between the demand vertexes and the median set. The constraint set (2.2) guarantees that all demand vertexes are assigned to exactly one median. The constraint set (2.3) forbids that a demand vertex be assigned to a facility that was not selected as a median.

The total number of median vertexes is defined by (2.4) as equal to *p*. Constraint (2.5) guarantees that the values of the decision variables x and y are binary (0 or 1).

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 $Vp \in V$ (median set) with cardinality p, such that: (a) the sum of the distance between each remaining vertex in $\{V - Vp\}$ (demand set) and its nearest vertex in Vp be minimized; and (b) 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: (i) Each facility can satisfy only a limited number of demands (capacity restrictions), and (ii) All demand points must be satisfied by respecting the capacities of the facilities selected as medians.

III. A REAL-WORLD APPLICATION

Suppose we have an exam for the B.Tech students. The goal was to assign 19710 candidate students to facilities as close as possible to their corresponding homes. In order to obtain the distance between each candidate student's home and each facility, all the addresses in question have been precisely located in a digitized map of the city.

It was previously determined, for operational and economic reasons that an algorithm should select 26 facilities to satisfy all 19710 candidate students, among a set of 43 candidate facilities. We cast this problem as a capacitated p-median problem, as follows:

1. The set of 43 facilities (potential exam locations) is the set V ($|V| = 43$) of all facilities candidate to median (actual exam locations).
2. Let $Vp \in V$ ($|Vp| = 26$) be the set of the 26 selected exam locations.
3. Each of the 43 potential exam locations can satisfy only a limited number of candidate students.
4. The goal is to select a set $Vp \in V$ that minimizes the total sum of distances between each candidate student's home and its nearest exam location (median).

IV. GENETIC ALGORITHM FOR CAP-P-MED-GA

This section describes our proposed GA for the capacitated p-median problem, Cap-p-Med-GA.

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. For instance, consider a problem with 15 facilities (potential medians) represented by the indexes 1, 2,..., 15. Suppose one wants to select 5 medians.

In our GA, the individual [2, 7, 5, 15, 10] represents a candidate solution for the problem where facilities 2, 5, 7, 10 and 15 have been selected as medians. In Cap-p-Med-GA the genome is interpreted as a set of facility indexes, in the mathematical sense of set - i.e. there are no duplicated indexes and there is no ordering among the indexes.

B. Fitness Evaluation

In essence, the fitness of an individual is given by the value of the objective function for the solution represented by the individual - as measured by formula (2.1). However, there is a caution in the computation of the fitness of an individual. Note that Capacitated-p-Median-GA is used only to optimize the choice of the 26 medians, out of the 43 facilities. However, the computation of formula (2.1) requires that each of the 19710 candidate students be assigned to exactly one of the selected medians (i.e. the facility where the student will take the admission exam). This assignment is done by a procedure that is used by Cap-p-Med-GA as a black box.

Here we just mention the basic idea of this procedure. Once the 26 medians are selected, this procedure tries to assign each candidate student to the median (exam location) that is the nearest one to its home. The problem is that, since each median has a fixed capacity, some candidate students will have to be assigned to the second (or third, fourth ...) nearest median to their homes. Suppose there is an assignment conflict -e.g. there is just one position in one median, and that median is the nearest one for two candidate students.

In this case the student-assignment procedure prefers to assign to that median the student that would be most intolerant if she was assigned to its second nearest median. A student is "intolerant" to the extent of the difference between two distances, namely the distance between her home and her nearest median and the distance between her home and her second nearest median. Once the student-assignment procedure is complete, the fitness of an individual can be computed by formula (2.1).

• *Selection:* The first step consists in selecting individuals for reproduction. This selection is done randomly with a probability depending on the relative fitness of the individuals so that best ones are often chosen for reproduction than poor ones. Typically we can distinguish two types of selection scheme, proportionate selection and ordinal-based selection. Proportionate-based selection picks out individuals based upon their fitness values relative to the fitness of the other individuals in the Population. Ordinal-based selection schemes select individuals not upon their raw fitness, but upon their rank within the population.

• *Crossover:* crossover is the process of taking two parent solutions and producing from them a child. After the selection (reproduction) process, the population is enriched with better individuals. Reproduction makes clones of good strings but does not create new ones. Crossover operator is applied to the mating pool with the hope that it creates a better offspring.

Crossover is a recombination operator that proceeds in three steps:

i). The reproduction operator selects at random a pair of two individual strings for the mating.

ii). A cross site is selected at random along the string length.

iii). Finally, the position values are swapped between the two strings following the cross site.

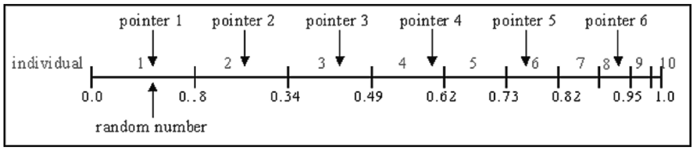

Fig. 1. Stochastic universal sampling

As a preprocessing step for the possible application of crossover, Cap-p-Med-GA computes two exchange vectors, one for each parent, as follows. For each gene of parent 1, Cap-p-Med-GA checks whether the allele (facility index) of that 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 indexes in parent 2's genotype. The same procedure is performed for each facility index in the genotype of parent 2. For instance, let the two parents be the facility-index vectors [1, 2, 3, 4, 5] and [2, 5, 9, 10, 12]. Their respective exchange vectors are: *vp1* = [1, 3, 4] and *vp2* = [9, 10, 12].

Once the facility indexes that can be exchanged have been identified, the crossover operator can be applied, as follows:

• *Evaluation:* Then the fitness of the new chromosomes is evaluated.

• *Replacement:* During the last step, individuals from the old population are killed and replaced by the new ones.

The algorithms stopped when the population converges toward the optimal solution.

The basic genetic algorithm is as follows:

• *[start] Genetic random population of n chromosomes (suitable solutions for the problem)*
• *[Fitness] Evaluate the fitness f(x) of each chromosome x in the population*
• *[New population] Create a new population by repeating following steps until the New population is complete*
  - *[selection] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to get selected)*
  - *[crossover] With a crossover probability, cross over the parents to form new offspring (children), If no crossover was performed, offspring is the exact copy of parents*
  - *[Mutation] With a mutation probability, mutate new offspring at each locus (position in chromosome)*
  - *[Accepting] Place new offspring in the new population*
• *[Replace] Use new generated population for a further sum of the algorithm*
• *[Test] If the end condition is satisfied, stop, and return the best solution in current population*
• *[Loop] Go to step 2 for fitness evaluation*

V. COMPUTATIONAL RESULTS

As mentioned earlier, the problem being solved consists of selecting 26 medians out of 43 facilities. Therefore, there are $^{26}C_{43}$ = 421,171,648,758 (roughly 421 billion) candidate solutions (Table 1).

Table 1: Computational Results

	Using GA
No. of eval. solutions	24,300
Run time	01:43:21
Avg distance	2.40Km
Total distance	47,313Km
Percent near facility	79%

VI. CONCLUSIONS AND FUTURE WORK

We have proposed a GA for the capacitated p-median problem, and have applied it to a real-world problem with a quite large search space, containing roughly 421 billion (4, 21 x 1011) candidate solutions. Our GA uses an individual representation and genetic operators developed specifically for the p-median problem. Capacitated p-median problem (CPMP) is an important deviation 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.

REFERENCES

[1]. BOOKER, L. B., "Improving Search in Genetic Algorithms", In: Genetic Algorithms and Simulated Annealing (Edited by L. Davis), 61-73, Morgan Kauffmann, Los Altos, CA, 1987.
[2]. CORREA, Elon S., "Algoritmos Geneticos e Busca Tabu Aplicados ao Problema das P-Medianas", (In Portuguese) M.Sc. Mestrado em Metodos Numericos em Engenharia. Universidade Federal do Parana (UFPR). Curitiba, Brazil 2000.
[3]. DIBBLE, C., DENSHAM, P. J., "Generating Interesting Alternatives in GIS and SDSS Using Genetic Algorithms", *GIS/LIS* symposium, University of Nebraska, Lincoln, 1993.
[4]. ERKUT, Erhan; BOZKAYA, Burçin; ZHANG, Jianjun., "An Effective Genetic Algorithm for the p-median Problem", (In press) 2001.
[5]. P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting stock problem," *Operations research*, vol. 9, pp. 849-859, Nov-Dec. 1961.
[6]. R. E. Kuenne and R. M. Soland, "Exact and approximate solutions to the multisource weber problem", *Mathematical Programming*, vol. 3, pp. 193-209, Dec. 1972.
[7]. T. Murray and R. L. Church, "Applying simulatedannealing to location-planning models," *Journal of Heuristics*, vol. 2, pp. 31-53, June 1996.
[8]. J. Brimberg and N. Mladenovic, "Solving the continuouslocation-allocation problem with tabu search," *Studies in Locational Annals*, vol. 8, pp. 23-32, Jan. 1996.
[9]. X. Li and A. G. Yeh, "Integration of genetic algorithms andgis for optimal location search," *International Journal of Geographical Information Science*, vol. 19, pp. 581-601, May.2005.
[10]. D. E. Goldberg, "*Genetic algorithms in search optimization and machine learning"*, Menlo Park, CA: Addison-Wesley, 1989.