2009 IEEE International Advance Computing Conference (IACC 2009)
Patiala, India, 6–7 March 2009

Heuristic Approaches for K-Center Problem

Rattan Rana, Deepak Garg
Thapar University

Abstract - The allocation of distribution centers or the facility center is an important issue for any company. The problem of facility location is faced by both new and existing companies and its solution is critical to a company's eventual success. This issue got the highest priority in last few years. It is equally important for both private as well as the public sector. The *k-center* problem is one of the basic problems in facility location. The aim is to locate a set of k facilities for a given set of demand points, such that for any demand point the nearest facility is as close as possible. Heuristics is a popular way to undertake such kind of typical problems. In this paper we present an intensive analysis of heuristic approach for *k-center* problem.

Keywords: Heuristic, Facility location, *k-center*, greedy, optimizing,

1. Introduction

A number of algorithms have been proposed for solving the location problems. This paper highlights the heuristic approach. The heuristic algorithm and its variants may be used to solve a large number of other location problems. Heuristic usually refers to a procedure that seeks an optimum solution but does not guarantee for an optimum solution. The heuristic approaches become popular technique in solving *k-center* problem. The *k-center* problem is nothing but the placement of multiple facilities (or centers) in such a way that the each center can cover maximum number of demand nodes and the distance of all centers should be as minimum as possible, it is one of the biggest issues for researchers. This problem is also known as *k-center* problem.

Formally, the *k-center* can be defined as Let $G = (V, E)$ be a complete undirected graph with edge costs satisfying the triangle inequality, and k be a positive integer not greater than $|V|$. For any set $S \subseteq V$, and vertex $v \in V$, we define $d(v, S)$ to be the length of a shortest edge from v to any vertex in S. The problem is to find such a set $S \subseteq V$, where $|S| \leq k$, which minimizes $\max_{v \in V} d(v, S)$. The vertex *k-center* problem is NP-hard. It can be mathematically represented as given below:

$$\underset{X \text{ on } G}{\text{Minimize}} \, f(X) = \max_{1 \leq i \leq n} h_i \, d(v_i, X)$$

In the category of heuristic algorithms, a number of algorithms have been proposed. Like pure greedy, improved heuristic, neighborhood search, interchange heuristic and meta-heuristic.

When we have to select a subset of objects, the greedy heuristic is used. Usually sequential approach applied in which individual site is accessed, which sited provides utmost impact on objective. This site is fixed and then next site with similar attributes is searched. Repeat the process until the required numbers of sites are identified. For this reason this approach is called as Greedy Heuristic, specifically it is known as Greedy-Add. Another version of Greedy Heuristic is known as Greedy-Drop, since it removes the site which has least impact on the objective during the process of site selection. We continue the removing process till the required number of facilities or sites remain.

Although the Greedy-Add and Greedy-Drop provide good (or feasible) solution for location model, but to provide consistent a new algorithms developed that starts from the results given by heuristic algorithms to improve the solution. These are known as Improved Heuristics

The Neighborhood search algorithm is one of the improvements heuristic. In this technique we start search from any feasible solution given by any of the greedy heuristics.

This Interchange heuristic approach is introduced by Teitz and Bart (1968). The problem with many search heuristics is that, instead of yielding the required optimal solution, they become stiff in local optima. Then researchers planned to apply the heuristics in more intelligent manner that is called as meta-heuristic. The basic idea behind this is to break out local optima and search other regions of the solution space.

A number of meta-heuristic approaches have been proposed in last few years. These include the two stage construction heuristic of Rosing & ReVelle (1997), the tabu search procedures of Mladenovic, Moreno & Moreno-Vega (1996), Voss (1996) and Rolland, Schilling & Current (1996); and the variable neighborhood search approaches of Hansen, Mladenovic & Perez-Brito (1998) and Hansen & Mladenovic (1998).

One of the popular meta-heuristic is Tabu search local heuristic originally proposed by Glover and Hansen (1986). Its crucial component is to prevent the search to cycle by forbidding some moves during a certain number of iterations. The Tabu Search heuristic involves defining what type of exchanges to restrict and nature of the aspiration criteria and short-term memory to utilize.

In the subsequent section a comparative analysis of all the above mentioned algorithms is provided. In fact, it will show different technical aspects and their implementation complexities.

2. Literature Survey

In the category of heuristic algorithms, the first is greedy heuristic which is used to solve the *k-center* problem. The greedy heuristic selects centers one by one until the required numbers of centers are located. The centers are selected randomly. This heuristic is applied for *n*-times, if $n=|V|$. Each iteration is started from new vertex then a center is located by repeating this process. Now, we can select the best solution which highly impacts the objective. This is the simplest heuristic approach, but not much successful for large no of centers. It is also known as random, *1-center*, and plus version of greedy heuristic.

Another simplest heuristic is pure greedy technique. In which centers are located one by one and it reduces the objective function in each iteration as much as possible. The time complexity of this method is far less than the greedy heuristic.

Gonzalez [1] has implemented greedy heuristic and provide approximation factor 2. Jurij Mihelic and Borut Robic [4] has implemented this algorithm with random, *1-center* and plus version, they have reduced its time complexity to O(kn).

Hochbaum and Shmoys [2] have developed another parametric pruning technique to solve the *k-center* problem. They aimed to find a minimum dominating set in the pruned graph. But, M. R. Garey and D. S. Johnson [3] have proved that to compute the minimum dominating set is NP-hard. Then Jurij and Borut [4] have developed a new heuristic algorithm to solve the dominating set problems and provided better results.

Jurij Mihelic and Borut Robic [4] have also developed a new heuristic algorithm to solve the dominating set problem. In which the edge costs are stored in a non decreasing list which is used for getting the threshold value *r* and for solving the series of dominating set problems.

In the 1990 Wang and kam Hoi Cheng [5] has shown parallel time complexity of a heuristic algorithm for the *k-center* problem. They found that the results of greedy strategy are no greater than twice the optimal solution value.

N. Mladenovic, M. Labbe, and P. Hansen [6] have presented a basic Variable Neighborhood search and two Tabu search heuristics for the *p-center* problem without triangle inequality. Both proposed methods used the 1-interchange neighborhood structure. They have shown that how this neighborhood can be used even more efficiently than for solving the *p-median* problem.

Abhay K. Parekh [7] has made an analysis on simple greedy algorithm for finding small dominating sets in undirected graph of N nodes and M edges. In this, he found that $d_g \leq N+1-\sqrt{2M+1}$; here d_g is the cardinality of dominating set given by the algorithm

Tayyar Büyükbaşaran [12] has provided significant results with meta-heuristic and simulated annealing for vertex *p-center* problem. It produces better results than almost all other local search heuristics. The main drawback of Simulated Annealing algorithm is that it can not produce better results when the facility number is high. Making more iteration for problem instances with facility numbers higher than 10 or applying more complex move like interchange move of Mladenovic et al. (2003) can solve this problem.

3. Analysis of Heuristic Algorithms

As we go through the literature of *k-center* problem. Most of the researchers have used heuristic algorithms to solve the problem. This section presents the analytical results of heuristic algorithms used by different researchers to solve the *k-center* problem.

The most common method used to solve the *k-center* problem is to solve the series of set covering problem. If the number of facilities to be located is small then it works properly and provides good results. But the basic problem with this technique is that it doesn't work efficiently with large number of facilities.

In the literature survey we found that there is a great resemblance in dominating set problem and *k-center* problem, both lies in the category of NP-hard problems. Both of them intend to find a set of such vertices, which can control remaining nodes. Although, the *k-center* problem seeks more exact results than dominating set covering problem.

The analytical study shows [2, 13] that as algorithms reach to required number of dominating set value *k* it stops working. The worst case performance of this algorithms shows $O(|E|\log|E|)$ time complexity. If, we assume there is triangle inequality then the results shown by algorithms is *2-approximation* and author declares it as the best solution for *k-center* problem till now.

As we have already discussed about the heuristic algorithms, we also have different versions of greedy heuristic which are commonly used to solve the *k-center* problem. The most common is greedy (Gr) method. The selection of first center is based on maximum cost reduction. Similarly other centers are selected and added to the list of selected centers; the process is repeated until the required numbers of centers are located. The motive is to reduce the total cost in each iteration. There are some improved versions of greedy method like greedy first random (GrR), greedy 1-center (Gr1), and greedy plus (Gr+). The greedy first random selects first center randomly.

In the category of meta-heuristic approaches our next target of discussion is Variable Neighborhood Search (VNS). This has proved itself as a milestone for solving the combinatorial and optimization problems [8]. The major perceptions taken in this algorithm is to implement the local Binary Search algorithm with minor changes in basic structure of neighborhood techniques. Since, the neighborhood algorithms exhibit realistic property of proximity of local minima. So, VNS take its advantage in two ways; firstly, it searches in most attractive area so the search is not disturbed by any restricted moves. Secondly, it takes less iteration to search a solution than random search. Because it approach only the high quality local optima, where the possibility of getting solution is higher than random search with in same CPU time.

N. Mladenovic et al. (2003) has performed an experimental analysis on different heuristic algorithms. In this study diverse versions of greedy, Alternate and Interchange heuristic algorithms have been compared such as, Binary Search (BS), Interchange (I), Greedy (Gr), Greedy Plus (GrP), Alternate (A), Greedy Interchange (Gr+I), Greedy Plus Interchange (GrP+I) and Alternate Interchange (A+I) produced their performance results in average percentage deviation as shown below in table 1.

Algorithms	Percentage Deviations
BS	48.5
I	62.4
Gr	119.7
Gr+I	90.1
GrP	81.9
GrP+I	67.0
A	94.0
A+I	62.9

Table 1: Average percentage deviation

This study illustrates the computational efficiency of above shown algorithms with different values of *n* and *p* (where *n* is the total number of vertices and *p* is the number of facilities to be identified). We found that

the overall results of Binary Search are much better than other algorithms. If we talk about only the Local Search and Binary Search techniques then the basic difference found in them is that the Local search shows fast and good results for smaller number of *n* and *p*. But, the Binary Search performs better for the large number of *n* and *p*. This performance deviation is shown in chart 1.

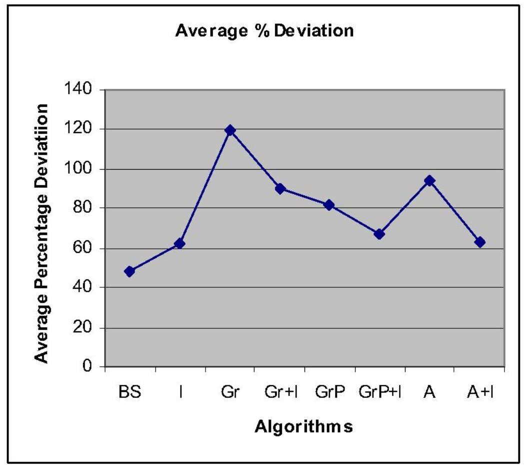

Chart 1: Average percentage of heuristic Algorithms

In another study of heuristic algorithms Gonzalez, Dyer and Frieze have made a comparative analysis on different algorithms with heuristics. They have shown approximation factor 2 using greedy heuristic for the *k-center* problem. They have shown average and deviation ratio of different algorithms in their results. Here we are presenting it in the form of chart as shown in Chart 2..

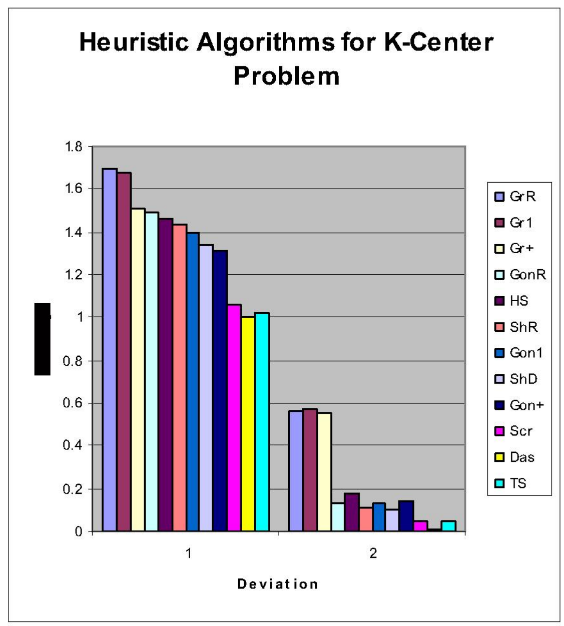

Chart 2: Analysis of heuristic Algorithms

The quality of greedy heuristic results is seems much better than the results generated randomly. But, it has been observed that there is a disadvantage of greedy heuristic that it can generate only a limited number of different solutions. Moreover, their results in early stages of process strongly constrain the available possibilities at later stages, often causing very poor moves in the final phases of the solution. As far as the complexity of algorithm is concern, that could not be reduced at any level of heuristic techniques.

The quality of greedy heuristic results is seems much better than the results generated randomly. But, it has been observed that there is a disadvantage of greedy heuristic that it can generate only a limited number of different solutions. Moreover, their results in early stages of process strongly constrain the available possibilities at later stages, often causing very poor moves in the final phases of the solution. As far as the complexity of algorithm is concern, that could not be reduced at any level of heuristic techniques.

As we have observed in different studies of heuristic algorithms, it has been realized that the computational performance of heuristics should be improved. In this context Wang and kam Hoi Cheng [5] has performed and experiment, in which they have compared the performance of heuristic algorithms on uniprocessor and multiprocessor systems. They found that the time complexity on uniprocessor system is $O(n^3)$ while on the multiprocessor it is $O(n \log^2 n)$, n is the total number of vertices. This experiment shows that despite of applying unlimited parallelism the algorithm has higher than polylogrithmic time complexity

Finally, it has been proved that the greedy heuristic couldn't provide optimal solution for *k-center* problem even on implemented with multiprocessor system.

4. Result Discussion

In the previous section we have discussed all the pros and cons of different heuristic algorithms as well as their working trends. Now it's essential to discuss different aspect of all heuristic algorithms. During the whole discussion, we found that the behavior of all algorithms vary with different problems. As Jurij Mihelic and Borut Robic [8] have shown some practical aspects of these techniques. Now, an important questions crop up, if all of these algorithms are performed on a particular set of vertices, then how these algorithms perform?

When they have tried these algorithms with a particular problem, undoubtedly the performance shown by all algorithms is different. The pure greedy algorithm shows worst performance, while the greedy plus algorithm provides slightly better results. The fundamental problem of pure greedy algorithm is that it is extremely dependent on the number of facilities to be located i.e. parameter *k*. If the number of facility is small then it shows better results. But, in case of large value of *k* performance is not so good.

The results of Gonzalez algorithms are also significant. Gonzalez plus version shows much better results. Although, the results of Gonzalez's algorithm is approximately 32% above the optimal. Similarly, HS, ShR behaves and returns same results. While the Scr algorithms of Jurij Mihelic and Borut Robic shows much better results than all other algorithms. Its results are on an average 6% above the optimal solution.

The computational performance of pure greedy algorithm is relatively better than others, but it also constraints the number of facilities. In other words with increasing number of facilities its performance goes down. While the greedy plus version runs much slower because it tries all the vertices for *1-center*.

5. Conclusion and Future work

During the detailed study of heuristic algorithms we have observed that the almost heuristic algorithms are providing good results in particular cases. As far as performance of these algorithms is concern, it's not bad, but it couldn't provide an optimum solution for *k-center* problem. The basic problem with each algorithm is that as the number of centers increases the performance of algorithm decreases proportionally. Even on implementing with multiprocessor system it couldn't improve its performance. Hence it has been proved that the heuristic algorithms proposed till now can provide good results with small number of centers.

So it is the open problem for researchers to develop a heuristic algorithm which can perform efficiently with any number of centers.

REFERENCES

[1] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science,* 38:293–306, 1985.

[2] Dorit S. Hochbaum and David B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research,* 10:180–184, 1985.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., San Francisco, 1979.

[4] Jurij Mihelic and Borut Robic. *Approximation algorithms for the k-center problem*: an experimental evaluation.

[5] Qingzhou Wang and Kam Hoi Cheng. Parallel time complexity of a heuristic algorithm for the k-center problem with usage weights. IEEE, 1990.

[6] N. Mladenovic, M. Labbe, and P. Hansen. *Solving the p-center problem with tabu search and variable neighborhood search.* Networks, Vol. 42(1), 48-64 2003.

[7] Abhay K. Parekh. *Analysis of a Greedy Heuristic For finding Small Dominating Sets in Graphs.* M.I.T., Cambridge

[8] Jurij Mihelic and Borut Robic. *Solving the k-center problem efficiently with a dominating set algorithm. Journal of Computing and Information Technology*-CIT 13, 2005, 3, pp 225-233.

[9] Doron Chen and Reuven Chen. *Revising relaxation-based optimal algorithms for the solution of the continuous and discrete p-center problems.*

[10] Zvi Drezner and H. W. Hamacher. *Facility Location Applications and Theory, 2004.*

[11] Mark S. Daskin. *Network and Discrete Location Models, Algorithms, and Applications.*

[12] Tayyar Büyükba□aran. *Solving the P-Center Problem With Simulated Annealing.* Middle East Technical University, Industrial Engg. Deptt., Ankara 2004

[13] J. Martinich, A vertex-closing approach to the *p*-Center problem, Naval Res. Log 35 (1988), 185–201.