# An Evaluation of K-Center Problem Solving Techniques, towards Optimality

Rattan Rana, Deepak Garg
Department of Computer Science & Engineering
Thapar University, Patiala (Punjab)
India- 147004
Email: rattanrana77@gmail.com, deep108@yahoo.com

## Abstract

Optimization of facility location problem is one of the prominent areas of research since last few decades. The vertex k-center problem represents a common occurring problem with in public and private sectors. Consistent efforts are going towards optimal solution of this NP-hard nature problem. The aim of this paper is to highlight the different aspects of algorithmic approach made in this area, so far. Relevant suggestions and modifications are also incorporated in this paper.

***Keywords*****:** vertex k-center, facility location, optimal, demand node, lower bound, upper bound.

## 1. Introduction

Facility location is a crucial type of problem having an extensive range of sub problems that have been scrutinized within various fields including operation research, computational geometry, data analysis, computational complexity and graph theory. This area of research has a long history and persistent activity accompanied by well off literature. The motive is to find the best location of facilities in a network graphs in realistic situations such as planting the ambulance services, fire stations, workstations and many more, entailed both the public and private sectors. Facility location problems proposed in operations research provide mathematical formulations of the common optimization aspects of these problems. One of the well recognized facility location problems is the vertex *k*-center problem. Where given a graph $G = \{V, E\}$ having a set of vertices $V = \{v_1, v_2, v_3,\dots\dots v_n\}$ and $E = \{e_1, e_2, e_3,\dots\dots e_m\}$ is a set of edges. The n and m are the cardinality of vertex and edge sets respectively. Let $G = \{V, E\}$ is graph with edge costs satisfying the triangle inequality, and $k$ be a positive integer greater than $|V|$. The objective is to find such a set $S \subseteq V$ and vertex $v \in V$ where $|S| \leq K$ which minimizes $\max_{v \in V} d(v,S)$. In simple words the objective of vertex *k*-center is to minimize the maximum distance between a center and a demand node. This problem is well formulated by Daskin [4] as given below:

***Inputs:***

$d_{ij}$= distance between demand node $i$ to candidate facility site $j$
$h_i$ = demand at node $i$
$P$ = number of facilities to locate

***Decision variables***

$$X_j = \begin{cases} 1 & \text{if we locate at candidate site j} \\ 0 & \text{if not} \end{cases}$$

$Y_{ij}$ = fraction of demand at node *i* that served by a facility at node *j*
$W$ = maximum distance between a demand node and the nearest facility
We can be formulate the vertex *k*-center problem as follow:

**Minimize** $W$ $(a)$
**Subject to:**

$$\sum_j Y_{ij} = 1 \quad \forall i \quad (b)$$

$$\sum_j X_j = P \quad (c)$$

$$Y_{ij} \le X_j \quad \forall i, j \quad (d) \qquad W \ge \sum_j d_{ij} Y_{ij} \quad \forall i \quad (e)$$

$$X_j = 0, 1 \quad \forall j \quad (f) \qquad Y_{ij} \ge 0 \quad \forall i, j \quad (g)$$

(*a*) is the objective function and minimizes the maximum distance between a demand node and the closest facility to the node. The constraint (*b*) state that all the demand node at *i* must be assigned at a facility at some node *j* for all nodes *i*. constraint (*c*) specify that *P* facilities be located. Constraint (*d*) state that the demand at node *i* can not be assigned to a facility at node *j* unless a facility is located at node *j*. constraint (*e*) state that the maximum distance between a demand node and the nearest facility to the node (*W*) must be greater than the distance between any demand node *i* and the facility to which it is assigned. Constraint (*f*) and (*g*) are the integrity and non negativity constraints. Hence the ultimate objective is to minimize the maximum distance between a center and a demand node.

Facility location problem is a well known problem of computer science and operation research. The vertex k-center problem is the part and parcel of facility location problem which is NP-hard in nature. The intensive efforts of researchers have turned the problem to NP-Complete. In other words the NP hard problem has been solved polynomially, although the work is going on towards optimal value. The subsequent section highlights the latest achievements available in literature excluding the earlier stages work.

## 2. Related Work

The vertex k-center problem has received considerable recent attention both in computer science and operation research literature. This section provides a brief review of extensive literature of vertex k-center problem. It will concentrate only on the latest accomplishments till date. Hochbaum and Shmoys (1985) have presented a 2-approximation algorithm for k-center problem with triangle inequality [8]. Using linear programming theory they have provided an interesting insight to problem and enable us to derive in $O(|E|log|E|)$ time a solution with value no more than twice the k-center problem.

Thereafter, J. Plesnik (1987) has generalized the results of Hockbaum and Shmoys, a polynomial algorithm with a worst case error ratio of 2 is described for the p-center problem in connected graphs with edge lengths and vertex weights [5]. A slight modification of this algorithm provides ratio 2 also for the absolute p-center problem. Both these heuristics are the best possible in the sense that any smaller ratio would imply P=NP. Plesnik has set the objective to minimize the weighted eccentricity and gave a polynomial heuristic having ratio 2 also for the p-center problem. In the subsequent publication Plesnik (1987) has proved that the ratio can be arbitrarily large in case of multi-centers. J. Plesnik continued his interest and efforts with k-center problem. He has given two faster heuristics for the absolute p-center problem in graphs [6] (1988).

Daskin (1995) has given an improved algorithm to find the optimal solution of the vertex p-center problem. He has taken the lower and upper bound on the value of p-center objective function to solve the set covering problem [4]. Daskin presented a new approach to solve the k-center problem to optimality in July, 2000. The major change was the replacement of set covering problem with maximal covering problem and provided better results [2].

Ilhan and Pinar (2001) developed an interesting 2-phase approach to solve the vertex p-center problem optimally for a specific covering radius [3]. Al-Khedhairi et al. (2005) have made some modification to the Daskin and Al-Khedhairi's algorithms to solve the vertex p-center problem [1]. In the 2006 Al-Khedhairi [7] has given a new enhancement to the Daskin's algorithm, he has shown how to speed up the process of shrinking the gap between lower and upper bounds in order to solve the vertex p-center problem. The next section of this paper presents the technical aspects of all mentioned algorithms with their pros and cons.

## 3. Discussion

This section provides the detailed discussion over the results provided by contemporary authors using different approaches.

Ilhan and Pinar [3] have designed an interesting two phase algorithm to solve the vertex p-center problem optimally. This algorithm has two phases LP and IP respectively.

**Preliminaries:** R is the specific covering radius and refers the problem as IP:

$$\sum_{j \in J} b_{ij} w_j \geq 1 \qquad \forall i \in I$$

$$\sum_{j \in J} w_j \leq P$$

$$w_j \in \{0,1\} \qquad \forall j \in J$$

Where $I$ is set of demand nodes and $J$ is set of candidate facility sites. The $b_{ij}$ can be calculated as given below

$$b_{ij} = \begin{cases} 1 & \text{If } d_{ij} \leq R,\ \forall i \in I,\ \forall j \in J \\ 0 & \text{otherwise,} \end{cases}$$

As stated above the algorithm has two phases the first phase is the LP relaxation for the problem is solved for given $R$. The second phase is IP feasibility problem which is almost similar to first phase except that the integrity constraints are relaxed and replaced by

$$0 \leq w_j \leq 1 \quad \forall j \in J$$

The basic reason for the success of this algorithm is that if first phase is not feasible then need not to go for the second phase for given $R$. If the first phase is feasible for $R$ then the smallest value of $R$ (i.e. $R_0$) is treated as lower bound ($R_0$) for starting the covering radius as $R_0$ for second phase. If the solution is infeasible for $R_0$ then it is increased until the solution is reached. When it reaches to solution this value of $R$ represents the optimal solution for vertex p-center problem. The full algorithm is shown in fig 1. In 2005 Al-khedhairi has suggested some modifications in Ilhan's algorithm [1].

***Modifications suggested*:**

Al-khedhairi has suggested first change in step 5 of phase 1, that

**If the LP relaxation is not feasible then set R=U, else keep the old value of R from step 2.**

This minor change may reduce the number of iterations in a great extent. An important point has been observed by the Al-khedhairi that is in step 6 of phase 2. It shows that the value of radius $R$ produced by step 2 is used for IP formulation but if this value of $R$ does not exist in distance matrix then there is no use of any iteration made with this value of $R$.

**Phase-1 (LP relaxation)**

**Step 1:** Set $L=\min\{ d_{ij}: \forall i \in I, \forall j \in J\}$

$U= \max\{ d_{ij}: \forall i \in I, \forall j \in J\}$

**Step 2:** Calculate $R= \left[ \dfrac{(U+L)}{2} \right]$

If $d_{ij} \leq R$, then set $b_{ij}=1$, else set $b_{ij}=0$

**Step 3:** Solve the LP relaxation of (IP)

If LP relaxation is infeasible then

Set $L=R$, else set $U=R$

**Step 4:** Calculate ($U$-$L$), If $(U\text{-}L) \leq 1$ Then go to step 5, else go to step 2

**Step 5:** If LP relaxation is infeasible, Then set $R=U$, else set $R=L$, Phase-2 (IP feasibility problem)

**Step 6:** If $d_{ij} \leq R$ , then set $b_{ij}=1$, else set $b_{ij}=0$

**Step 7:** Solve the IP problem, If the IP Problem is feasible then stop, else set $R= \min\{d_{ij}: d_{ij}<R, \forall i \in I, \forall j \in J\}$ and go to step 6.

Fig 1:The Original algorithm of Ilhan and Pinar

Therefore it must be checked in the distance matrix whether the value of $R$ exists in distance matrix or not before performing step 6. If it exists then it can be continue as before otherwise reset the value of $R$ to next minimum distance greater than current value of $R$ and continue. The replacement of step 6 suggested by Al-Khedhairi are as below:

**Step 6.1** Check if $R$ exists in the distance matrix or not:
If $R$ exists then go to step 6.2, else set $R=\text{Min}\ \{d_{ij}{:}d_{ij}<R,\ \forall i \in \text{I},\ \forall j \in \text{J}\}$ and go to step 6.2.
**Step 6.2** If $d_{ij} \leq R$, then set $b_{ij}=1$, else set $b_{ij}=0$

Second suggested modification is very effective to improve the efficiency of algorithm; it also reduces the number of iterations. It is known as jump-based update

The basic idea behind this method is to reduce the iterations; it works same as the basic algorithm works the difference is only that if problem is infeasible for $R$ then we select the second next minimum distance instead of next minimum distance $R'$. It can be presented as:

$$R=R''$$

The functionality of idea can be shown as given below in figure 2.

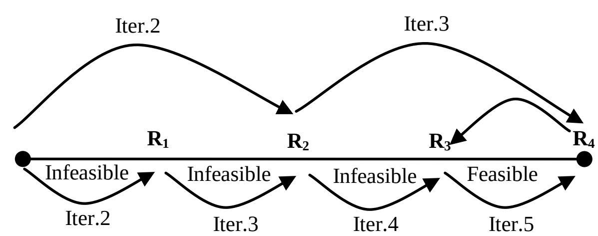


Fig 2: Illustrative example of jump-based approach

***Description*:**

Fig. 2 shows that $R$ is the minimum radius selected as lower bound and algorithm tries to solve the IP problem. If it is not feasible then goes to next minimum radius $R_1$ and continues the process until solution is reached. On the contrary the arrows shown above the line are working differently it shows that in case of infeasible then it does not select to next minimum radius rather it jump to second next minimum radius. It checks whether the solution is feasible, if yes; then before concluding this point as optimal solution, an additional check is also performed at go one step back for feasibility if it's also feasible for this radius then it is selected as lower bound to get the optimal solution otherwise the next one is the lower bound. It is summarized that this modification enable the algorithm to perform in less number of iterations but the results shows that in few cases it is false, sometimes it also takes one more iteration to get optimal solution.

M. S. Daskin has developed an algorithm to solve the *p-center* problem (1995). It is nothing but the improved version of Minieka's (1970) algorithm. The basic steps of this algorithm are given in fig 3. Fig 3 shows that initially select the lower and upper bound. Then solve the set covering problem using the average of lower and upper bounds as coverage distance. Suppose $k$ be the number of facilities required to cover all nodes. If $k \leq P$, reset set the value of upper bound to $D$ else reset the lower bound to $D$+1. If the lower and upper bounds are equal then the lower bound is optimal solution to $p$-center problem and stop. Otherwise solve the SCP with new coverage distance $D$ and continue the process.

In the continuation of this M.S. Daskin (2000) has given a new approach to solve the vertex p-center problem [2]. This algorithm is somewhat different from previous one. This approach uses a maximal covering problem as a subroutine instead of set covering problem.

***Step 0: Set L=0 and U= max ($d_{ij}$)***

***Step 1: Calculate D=*** $\left[\dfrac{(L+U)}{2}\right]$

***Step 2: Solve set covering problem for the coverage distance D, and let k be the number of facilities found.***

i. ***If SCP is feasible (i.e. $k \leq P$) then set U=D***

ii. ***Else (i.e. SCP is infeasible, k>P) set L=D+1***

***Step 3: If L=U, then the optimal solution is L, and stop. Else go to Step 1.***

Fig 3: Original Algorithm of Daskin (1995)

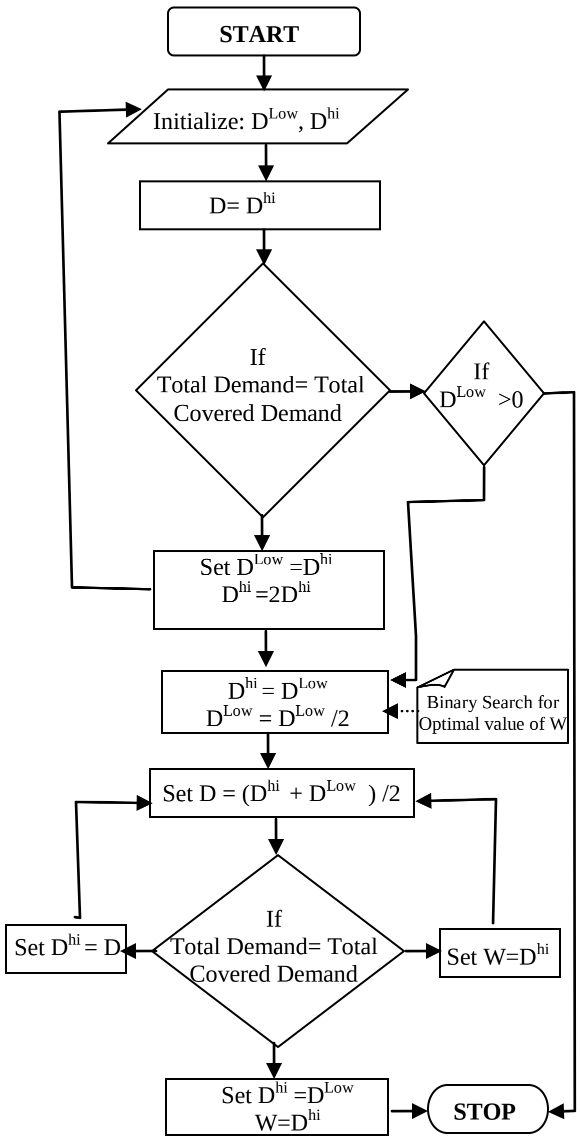


Fig 4: Daskin's Algorithm (2000) for vertex p-center problem

Because the set covering model fails to distinguish between large demand nodes and small demand nodes, it provides only the set of nodes without any demand consideration. The main steps of this algorithm are shown in fig 4.

***Description*:**

The flow chart shown in fig 4 describing the working of Daskin's algorithm. The variables $D^{hi}$ and $D^{low}$ are used to retain the value of upper and lower bound respectively over the value of $W$. First of all the Lower and upper bounds are initialized with some input values. Then it has been observed that if the all demand can not be covered in a distance $D^{hi}$, then reset the value of $D^{Low}$ as $D^{hi}$ and the value of $D^{hi}$ is doubled.
Thereafter, in the next step it is checked whether the demand is trial covered in distance $D^{Low}$ and $D^{Low}$ >0 then $D^{hi}$ is set to $D^{Low}$ and $D^{hi}$ is halved. Then binary search is performed to find the optimal value of $W$. $D$ is the average of $D^{Low}$and $D^{hi}$ that is called as trial coverage distance. If demand covered by distance $D$ is equals to total demand then new upper bound becomes the trial coverage distance, otherwise new lower bound becomes the trial coverage distance plus 1. The algorithm stops when the new lower and upper bounds are equal.

***Observations based on Daskin's algorithm (1995)*:**

The search procedure opts in this algorithm is like a simple binary search which result in large number of iterations. Al-Khedhairi in 2005 has suggested some modifications in this algorithm. Al-Khedhairi suggests that the number of iterations can be reduced either introducing tighter initial lower and upper bounds or by devising a more powerful binary search.

***Modification 1:***

Al-Khedhairi has suggested a new way to select the initial lower and upper bound that reduces the number of iterations as well as make it easy to find the exact solution [1]. Applying tightening of initial bounds:
**Lower bound:** The lower bound should be selected as $P^{th}$ minimum value of the distance in distance matrix including ties and excluding diagonal values. It guarantees that $R^* \geq L$ (where $R^*$ is the optimal covering radius).
**Upper bound:** The initial upper bound can be found by selecting the maximum distance value from each row and then select minimum from this set of maximum distance values. It guarantees the $R^* \leq U$.

***Modification 2*:**

It has been observed that the binary search used in original algorithm is implemented to speed up the process of minimizing the gap between lower and upper bound is a slow technique. Another method is suggested that is called as golden section method. It is basically used to find the minimiser of one dimensional function within an interval [$a$, $b$]. The golden search method requires two interior points $x_1$, $x_2$ to be used, where

$$x_1 = a + \left(\frac{1}{\gamma}\right)^2 (b-a)$$

$$x_2 = a + \left(\frac{1}{\gamma}\right)(b-a)$$

$\gamma$ is the golden ratio which equal to 0.618. The proposed modification uses $x_1$ and $x_2$ as converge distance of SCP with $a$, $b$ replaced by the initial $L$ and $U$, respectively. The value of $x_1$ and $x_2$ are rounded down.

Now checking the feasibility at coverage distance $x_1$, if it is feasible then optimal solution lies between $a$ and $x_{1;}$ Otherwise feasibility will be checked at $x_2$, if it found feasible then optimal solution lies between $x_1$ and $x_2$. If optimal solution lies between $a$ and $x_1$ then problem will be solved using $x_1$ until it get infeasible solution at $x_1$. Each time check $x_1$ or $x_2$ and record the number of facilities found to cover all nodes. So it is easy to notice that the optimal solution of p-center problem is $U$, if $U\text{-}L \geq 2$ means there is no client at distance greater than $L$ and less than $U$. If there is no distance value between $U$ and $L$ then it is not required to check $x_1$ or $x_2$ and it is enough to use this distance value as the coverage distance of SCP and check it. If feasible then this is the optimal solution of the problem. If infeasible then the upper bound is optimal value of solution.

***Observations based on Daskin's algorithm (2000)*:**

One of the important improvements done by Daskin in comparison of previous algorithm is that the set covering problem is replaced by the maximal covering problem is solved as a subroutine. The basic idea behind this change is only to provide exact number of facilities because set covering problem provides excess number of facilities than required sometimes. Secondly, it has improved the execution speed of algorithm comparatively. The main problem with this algorithm is to find the exact lower and upper bounds. It can be concluded that the performance of algorithm depends upon the accuracy of lower and upper bounds.

## 4. Conclusion and Future Work

Various approaches have been studied during this evaluation process. Different points of attraction have come across. The conclusion of this discussion is that all the algorithms has given their wonderful results but the problem of finding exact initial lower and upper bounds is still there in all. Because the performance of algorithm is directly depends upon the number of iterations performed by it and until or unless the number of iterations are not reduced the performance can never be improved. Even though, the jump based approach improved the performance, but it is applicable only for problems where no of lower bound are greater than 5. As far as the future work is concerned, so improve the method which can provide the tighter upper and lower bounds. It can only help to improve the performance of algorithm as well as possibility of optimal solution.

## References

[1] Al-khedhairi A., Salhi S., "Enhancement to Two Exact Algorithms for Solving the Vertex P-center Problem", JMMA 2(2), pp 129-147, 2005.

[2] Daskin M., 2000, "A New Approach to Solve the Vertex P-center Problem to optimality: algorithm and computational results". Communications of the Operations Research Society of Japan, 45(9), 428-436.
[3] Ilhan T., and Pinar M.C., 2001, "An Efficient Exact Algorithm for the Vertex p-center Problem", http://www.optimization-online.org/DB_HTML/2001/09/376.html
[4] Daskin M., 1995, "Network and Discrete Locations: Models, algorithms, and applications", John Wiley & Sons, New York. Pp 154-191.
[5] Plesnik J., "A Heuristic for The p-center Problem in Graphs". Discrete Applied Mathematics 17 (1987), pp 263-268, North Holland.
[6] Plesnik J., "Two Heuristics for The Absolute p-Center Problem in Graphs". Math. Slovaca 38, 1988, No. 3, 227-233.
[7] Al-khedhari A., "An Enhancement of Daskin's Algorithm for Solving p-center Problem", Journal of Approximation Theory and Applications, 2(2), pp 121-131, 2006.
[8] Hochbaum, D.S. Shmoys, D.B.: "A Best Heuristic for the k-center Problem". Math. Operations Research 10, 1985, pp 180-184.