



A Maximal Client Coverage Algorithm for the p -Center Problem¹

Sittipong Dantrakul[†] and Chulin Likasiri^{†,‡,2}

[†]Department of Mathematics, Faculty of Science
Chiang Mai University, Chiang Mai 50200, Thailand
e-mail : sittiphong2002@hotmail.com (S. Dantrakul)
Chulin.l@cmu.ac.th (C. Likasiri)

[‡]Centre of Excellence in Mathematics
CHE, Si Ayutthaya Rd., Bangkok 10400, Thailand

Abstract : In this work, we propose a maximal client coverage algorithm for solving the p -center problem. The algorithm is created to locate p facilities and assign clients to them in order to minimize the maximum distance between clients and the facilities. We consider both uncapacitated and capacitated cases where demands of clients and capacities of facilities are taken into account. The simulations to test the proposed algorithm are also given and compared with method given by Albareda-Sambola et al. in 2010. Optimal solutions of the test problems are found using branch and bound algorithm to compare the optimality gaps of the proposed heuristics. The proposed heuristics solutions are found to be statistically faster than the reference algorithm at the significance level $\alpha = 0.01$ in both uncapacitated and capacitated cases.

Keywords : p -center problem; Facility location problem; Set covering problem; Binary integer programming; Greedy algorithm.

2010 Mathematics Subject Classification : 90C10.

¹This research was financially supported by the Centre of Excellence in Mathematics, CHE, Si Ayutthaya Rd., Bangkok 10400, Thailand.

²Corresponding author.

1 Introduction

Location problems consist of a set of clients and a set of potential sites where facilities can be located. The p -center problem (PC) is a class of location problems having a specific objective function to minimize the maximum distance between each client and the facility it is assigned to, providing that the number of opened facilities are not greater than p . This problem can be partitioned into the uncapacitated and capacitated cases. The uncapacitated case is a basic p -center problem that does not include the demands of clients and capacities of facilities. In the capacitated case, each client has a certain demand to meet and the facilities have capacity restrictions, i.e. the total demands of clients assigned to a facility cannot exceed the facility's capacity. The p -center problem has been studied together with p -median problem.

The p -center problem has been of interest since its first appearance in a publication in 1964 by Hakimi [1]. Problems on general connecting graphs have been proved to be NP-Hard ([2, 3]). An algorithm for the 1-center problem in $O(n)$ time was presented in [4]. Gonzalez [5] introduced a greedy heuristic for the p -center problem. Hochbaum and Shmoys [6] proposed a heuristic for p -center problem as follows. Initially, all distances in a graph are sorted in nondecreasing order. An edge with minimum distance was found and removed so that the number of connected graphs after removing all edges with higher distances is fewer than p . Hansen et al. [7] proposed the variable neighborhood search to solve p -median problem. The basic idea was to carry out a systematic change of neighborhood within a local search algorithm. The new solution obtained from partitioning the current solution set, randomly changing some variables elements in the partitions, and repartitioning until a better set of solutions is found. Then Mladenovic et al. [8] showed that heuristics for the p -median problem could be adapted for solving the p -center problem. They used the tabu search and variable neighborhood methods to solve the p -center problem and compared the result with method in [6]. Both tabu search and variable neighborhood methods significantly outperform the method in [6]. The variable neighborhood method is generally better than the tabu search method, which otherwise generally performs slightly better for smaller p . Pallottino et al. [9] used local search heuristics to solve the p -center problem in capacitated case.

In 1970, Minieka [10] first used a series of set covering problems which, later, Daskin [11] modified using the maximum cover version of this approach. Later, Ilhan and Pinar [12] applied this idea to a method, under which at each step, a cover distance was chosen and increased until all clients can be covered within this distance by using at most p facilities. A polynomial exact algorithm for capacitated p -center problem in tree networks is developed by Jaeger and Goldberg [13]. The method described in the article solves set-covering subproblems on trees with a polynomial algorithm. Bläser [14] designed an exact algorithm for the p -partial vertex cover problem and showed that this problem can be solved in a polynomial time for a certain p . In 2006, Ozsoy and Pinar [15] used [12] idea to solve the capacitated case by labeling the quantity of demand to clients and the capacity to

facility. Then, Albareda-Sambola et al. [16] improved their result by considering only clients in a given radius and improving the result using Lagrangian relaxation.

We develop a maximal client coverage algorithm to solve the p -center problem by combining set covering problem and greedy algorithm. The algorithms are given for both uncapacitated and capacitated cases.

2 The Model Formulation

To formulate the mathematical model, we introduce the following notations.

$I = \{1, 2, \dots, n\}$ is a set of clients,

$J = \{1, 2, \dots, m\}$ is a set of potential facility sites,

f_j is a facility set up cost for facility $j \in J$ where f_j is a constant number,

d_{ij} is a transportation cost or distance from client i to facility j , we assume that $I \cup J$ is a node set of a complete graph, and d_{ij} is the minimum transportation cost or shortest path between facility j and client i ,

$$x_{ij} = \begin{cases} 1 & \text{if client } i \text{ is assigned to facility } j, \\ 0 & \text{otherwise,} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if facility } j \text{ is opened,} \\ 0 & \text{otherwise.} \end{cases}$$

The model can be formulated as follows:

$$\begin{aligned} \text{(UPC)} \quad & \min \quad z \\ & \text{s.t.} \quad \sum_{j \in J} d_{ij} x_{ij} \leq z \quad \forall i \in I \\ & \quad \sum_{j \in J} y_j \leq p \\ & \quad \sum_{i \in I} x_{ij} \leq n y_j \quad \forall i \in J \\ & \quad \sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \\ & \quad x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J \\ & \quad y_j \in \{0, 1\} \quad \forall j \in J \end{aligned}$$

where p is a number of facilities to be located, z , in the first constraints, are the maximum distance between a client and the facility it is assigned to. The second constraint is to ensure that at most p facilities can be opened. The third constraints are to guarantee that the client is assigned to the opened facility. The last constraints are to ensure that each client is assigned to some facility.

The model for p -center capacitated case (CPC) can be formulated as the UPC model with an extra constraint on the demands and capacities,

$$\sum_{i \in I} h_i x_{ij} \leq Q_j y_j,$$

where h_i is the demand of client i and Q_j is the capacity of facility j .

3 The Proposed Method for Solving p -Center Problem

We adapted the idea for solving p -center problem, from Ilhan and Pinar [15] which used a series of set covering problem and improved the algorithm created by Albareda-Sambola [16] as follows:

In order to solve the problem, let us denote $D_1 < D_2 < \dots < D_{max}$ be the sorted distinct entries of the distance d_{ij} . Obviously, the value of the optimal solution is one of the element in $D = \{D_1, D_2, \dots, D_{max}\}$.

Define radius δ for each facility $j \in J$ and for each client $i \in I$,

Y_j^δ denote the set of clients whose distance to facility j does not exceed the radius.

X_i^δ denote the set of facilities whose distance to client i does not exceed the radius. Obviously, if $X_i^\delta = \emptyset$. $\exists i$ cannot cover all clients by opened facility hence infeasible.

Let $D^* = \{D_i^* = \min d_{ij}, \forall i = 1, \dots, n\}$ is a set of minimum distance for each clients connect to facility, $D_{max}^* = \max\{D_i^*, \forall i\}$ is a minimum value which $X_i^\delta \neq \emptyset$.

D_u and D_l are upper bound and lower bound where u and l are the indices of the upper bound and lower bound in the set D .

3.1 Main algorithm

We carried out our idea for solving p -center problem as follows:

Step 1: Select initial lower and upper bounds, set the radius to the median of the lower and upper bounds.

Step 2: Solve the maximal client coverage problem with the given radius.

Step 3: If the solution of the maximal client coverage problem is possible to cover all clients with p facilities, set the upper bound to be the radius. Else, set the lower bound to be the radius.

Step 4: If the lower and upper bounds are close enough to each other, go to Step 5. Else, set the radius to be the median of the lower and upper bounds and go to Step 2.

Step 5: If the lower bound is the solution of PC, the solution is the lower bound. Else, the solution is the upper bound.

This method which can be done in $O(n^3 \log n)$ can be summarized in a block diagram shown in Figure 1.

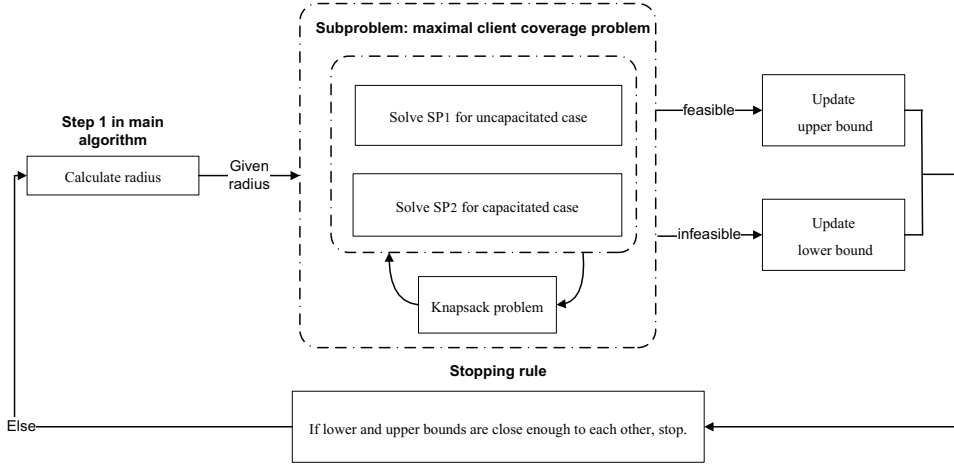


Figure 1. The process of the proposed algorithm for the p -center problem

Now, we show step by step of main algorithm as the follows:

Step 1: Select an initial $D_l = D_{max}^*$ and $D_u = D_{max}$ and set $\varepsilon = \lceil (u + l)/2 \rceil$, $\delta = D_\varepsilon$.

Step 2: Solve SP1 using radius= δ .

Step 3: If the solution of SP1 $< n$, set $D_l = \delta$, $l = \varepsilon$.
Else, D_u is set to be δ and $u = \varepsilon$.

Step 4: If $u - l > 1$, set $\varepsilon = \lceil (u + l)/2 \rceil$, $\delta = D_\varepsilon$ and go to Step 2.
Else, go to Step 5.

Step 5: Solve SP1 by using radius D_l .
If the solution of SP1 $< n$, the solution of PC= D_u .
Else, D_l is the solution of PC.

3.2 Subproblem

A maximal client coverage problem to use in the method of the uncapacitated p -center problem can be formulated as follows:

$$\begin{aligned}
 \text{SP1} \quad & \max \sum_{i \in X_i^\delta} \sum_{j \in Y_j^\delta} x_{ij} \\
 \text{s.t.} \quad & \sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \\
 & \sum_{i \in X_j^\delta} x_{ij} \leq ny_j \quad \forall j \in J \\
 & \sum_{j \in J} y_j \leq p.
 \end{aligned}$$

The SP1 is a model for finding the maximum number of clients assigned to opened facilities in the radius δ . Recall n is the number of clients, if the solution of SP1 $< n$, then PC is infeasible.

For the capacitated p -center problem, the model can be formulated by using the maximal client coverage problem with capacitated restriction, where the second constraints in SP1 are replaced by

$$\sum_{i \in X_j^\delta} h_i x_{ij} \leq Q_j y_j \quad \forall j \in J$$

called the modified problem, SP2.

Our idea for solving a maximal client coverage problem was carried out as follows:

Step 1: Find clients who connect to only one facility in the given coverage radius, then open that facility.

Step 2: Assign clients who connect exclusively to the opened facility in Step 1 by using knapsack problem, delete all clients who have been assigned a facility.

Step 3: If the number of opened facilities is fewer than p , go to Step 4. Else, stop.

Step 4: If there are clients who connect to only one facility in the given coverage radius go to Step 1. Else, choose one unopened facility having maximum number of connecting clients to be opened and go to Step 2.

The knapsack problem used in Step 2 can be formulated as follows:

$$\begin{aligned} \max \quad & \sum_{i \in I} h_i x_{ij} \\ \text{s.t.} \quad & \sum_{i \in I} h_i x_{ij} \leq Q_j. \end{aligned}$$

For uncapacitated case, we assume $h_i = 1$ and $Q_j =$ a number of clients. Our method for solving the knapsack problem was carried out as follows:

Step 1: Construct a candidate set whose distances to the facility are less than the given radius and whose demands are less than the facility's capacities.

Step 2: Assign clients in candidate set who have a maximum demand to the facility.

Step 3: Update capacity of the facility and update the candidate set.

Step 4: If the candidate set is empty, stop. Else, go to Step 2.

4 Computational Results and Discussions

The experiments are run on Dell Inspiron 1440 Intel Core 2 Duo T6500 2.10 GHz and 2 GB of RAM with MATLAB 7.9.0. In each case, 100 random data sets were tested using the algorithms of both uncapacitated and capacitated cases. Tables 1 and 2 show the results in the uncapacitated case. Minimum, maximum and average CPU times of the simulations are compared in Table 1 between the

proposed algorithm and the algorithm in [16]. The percentage gap between the solutions found and the optimal solution using branch and bound algorithm is illustrated in Table 2. The CPU times and the optimality gaps of the capacitated case are shown in Table 3 and Table 4.

Table 1. Average CPU times of 100 random data sets in uncapacitated case.

size		Time spent using the proposed algorithm (s)			Time spent using algorithm 3 given in [16] (s)		
m,n	p	min t	max t	avg t	min t	max t	avg t
50	5	0.0000	0.0312	0.0087	0.0312	0.0624	0.0229
50	10	0.0156	0.0468	0.0123	0.0468	0.0624	0.0270
50	15	0.0156	0.0468	0.0148	0.0468	0.0780	0.0309
50	20	0.0156	0.0468	0.0144	0.0468	0.0780	0.0326
100	10	0.0624	0.0936	0.0373	0.1716	0.2184	0.1005
100	20	0.0936	0.1560	0.0526	0.2184	0.2652	0.1203
100	30	0.1092	0.1560	0.0640	0.2496	0.3276	0.1388
100	50	0.1092	0.1560	0.0658	0.2652	0.3120	0.1437
500	50	4.8828	5.4288	5.1730	8.8873	10.0729	9.6751
500	100	5.5536	6.0840	5.9046	10.7593	11.2897	11.0042
500	150	6.6300	7.1448	6.9015	11.5549	12.7405	12.4347
500	200	7.4880	8.2369	7.9561	13.1929	14.3473	13.8824

Table 2. The percentage gap between the solutions found and the optimal solution in uncapacitated case.

size		The percentage gap using the proposed algorithm (%)			The percentage gap using algorithm 3 given in [16] (%)		
m,n	p	min %	max %	avg %	min %	max %	avg %
50	5	0.0000	3.0339	0.1389	0.0000	4.1125	0.1593
50	10	0.0000	4.1942	0.0827	0.0000	4.1942	0.1526
50	15	0.0000	4.3970	0.0562	0.0000	4.3970	0.0067
50	20	0.0000	4.0071	0.9283	0.0000	4.0071	0.3970
100	10	0.0000	9.1812	3.0878	0.0000	12.1491	4.1354
100	20	0.0000	11.9220	3.0679	0.0000	11.9220	2.9341
100	30	0.0000	11.2499	3.0199	0.0000	15.0339	3.0014
100	50	0.0000	15.7710	5.2499	0.0000	14.3823	6.0095
500	50	0.0000	17.0468	7.2184	0.0000	16.1159	7.5501
500	100	0.0000	15.5630	9.2319	0.0000	15.6411	9.3462
500	150	0.0000	16.1031	10.2265	0.0000	17.9902	10.7621
500	200	0.0000	20.1152	10.9789	0.0000	20.5501	11.0052

Table 3. Average CPU times of 100 random data sets in capacitated case.

size		Time spent using the proposed algorithm (s)			Time spent using algorithm 3 given in [16] (s)		
m,n	p	min t	max t	avg t	min t	max t	avg t
50	5	7.9280	8.9420	8.5848	9.8201	11.5829	10.8357
50	10	10.5801	11.9997	11.3257	12.2661	13.2957	12.7372
50	15	11.9545	13.7017	13.1089	13.3737	20.4718	15.4735
50	20	13.6705	14.8093	14.2960	15.6810	20.6090	17.5951
100	10	100.6463	110.5853	107.2575	143.3517	151.0540	148.3112
100	20	104.7094	117.2238	106.8407	148.8308	155.5308	153.5285
100	30	108.7547	120.7513	118.2543	153.5853	161.7792	160.1656
100	50	111.2760	124.2551	120.8143	164.5497	178.9340	167.6020
400	40	338.6797	349.5060	345.2435	440.9172	469.1299	465.2630
400	80	351.6551	370.6991	363.9293	455.2858	478.5688	474.6541
400	120	362.1626	379.8909	371.3500	471.7572	493.4694	489.6892
400	150	380.1190	399.9593	387.1966	500.7537	529.0119	522.7482

Table 4. The percentage gap between the solutions found and the optimal solution in capacitated case.

size		The percentage gap using the proposed algorithm (%)			The percentage gap using algorithm 3 given in [16] (%)		
m,n	p	min %	max %	avg %	min %	max %	avg %
50	5	0.0000	19.2319	5.5630	0.0000	24.2942	7.3462
50	10	0.0000	14.4505	3.1818	0.0000	15.9619	3.5132
50	15	0.0000	18.0838	2.2638	0.0000	17.0046	3.4018
50	20	0.0000	17.2290	4.1455	0.0000	17.7749	4.0760
100	10	0.0000	23.9133	7.1361	0.0000	20.8173	5.2399
100	20	0.0000	25.1524	5.8693	0.0000	27.8687	7.1233
100	30	0.0000	26.8258	4.5797	0.0000	26.0844	5.1839
100	50	0.0000	19.5383	7.5499	0.0000	23.3998	7.2400
400	40	0.0000	35.9961	10.1450	0.0000	36.2599	10.4173
400	80	0.0000	40.0782	12.8530	0.0000	39.8001	13.0497
400	120	0.0000	37.4427	10.6221	0.0000	38.4314	10.9027
400	150	0.0000	43.1067	10.3510	0.0000	44.9106	9.9448

A modified algorithm is proposed in this paper to solve the p -center problem. We introduce the maximal client coverage problem with fixed radius to find the lower and upper bounds of p -center problem and use knapsack problem to find the solution to the maximal client coverage problem. The average-case results indicate that the proposed algorithm is statistically faster than algorithm 3 given in [16] with the significance level $\alpha = 0.01$ in both uncapacitated and capacitated cases. Branch and bound algorithm is used to find the optimal solution of each test which was compared to the solution of the proposed and reference algorithms. The average percentage errors of proposed algorithm are found to be similar at all significance levels for both cases.

Acknowledgements : This research has received the full financial support by the Centre of Excellence in Mathematics, CHE, Si Ayutthaya Rd., Bangkok 10400, Thailand. Sittipong Dantrakul has got the student scholarship from the Human Resources Development in Science Project (Science Achievement Scholarship of Thailand SAST) and supported by the Graduate School of Chiang Mai University, Thailand. The authors thank Wiriya Sungkhaniyom for her proofreading help.

References

- [1] S.L. Hakimi, Optimum locations of switching centers and the absolute centers and medians of a graph, *Operations Research* 12 (1964) 450–459.
- [2] O. Kariv, S.L. Hakimi, An algorithmic approach to network location problems part I: the p -centers, *SIAM Journal of Applied Mathematics* 37 (1979) 513–538.
- [3] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Bell Laboratories, Murray Hill, Freeman Co., N.J., 1978.
- [4] S. Olariu, A simple linear-time algorithm for computing the center of an interval graph, *International Journal of Computer Mathematics* 24 (1990) 121–128.
- [5] T. Gonzalez, Clustering to minimize the maximum intercluster distance, *Theoretical Computer Science* 38 (1985) 293–306.
- [6] S.D. Hochbaum, B.D. Shmoys, A best possible heuristic for the k -center problem, *Mathematics of Operations Research* 10 (1985) 180–184.
- [7] P. Hansen, N. Mladenovic, D. Perez-Brito, Variable neighborhood decomposition search, *Journal of Heuristics* 7 (2001) 335–350.
- [8] N. Mladenovic, M. Labbe, P. Hansen, Solving the p -center problem with tabu search and variable neighborhood search, *Networks* 42 (2003) 48–64.
- [9] S. Pallottino, M.P. Scappara, M.G. Scutellà, Large scale local search heuristics for the capacitated vertex p -center problem. *Networks* 43 (4) (2004) 241–255.
- [10] E. Minieka, The m -center problem, *SIAM Rev.* 12 (1970) 138–139.
- [11] M.S. Daskin, A new approach to solving the vertex p -center problem to optimality: Algorithm and computational results, *Communications of the Operations Research Society of Japan* 45 (9) (2000) 428–436.
- [12] T. Ilhan, M.C. Pinar, An efficient exact algorithm for the vertex p -center problem, <http://www.optimization-online.org/DBHTML/2001/09/376.html>, 2001.
- [13] M. Jaeger, J. Goldberg, A polynomial algorithm for the equal capacity p -center problem on trees, *Transportation Science* 28 (1994) 167–175.

- [14] M. Bläser, Computing small partial coverings, *Inf. Proc. Letters* 85 (2003) 327–331.
- [15] F.A. Ozsoy, M.C. Pinar, An exact algorithm for the capacitated vertex p -center problem, *Computers and Operations Research* 33 (2006) 1420–1436.
- [16] M. Albareda-Sambola, J.A. Diaz, E. Fernandez, Lagrangean duals and exact solution to the capacitated p -center problem, *European Journal of Operational Research* 201 (2010) 71–81.

(Received 27 January 2012)

(Accepted 19 March 2012)