International Journal of Civil Engineering

Construction Management Engineering

Hybrid harmony search for conditional p-median problems

A. Kaveh[1,*], **H. Nasr Esfahani**[2]

Received: August 2010, Accepted: April 2011

Abstract

In this paper the conditional location problem is discussed. Conditional location problems have a wide range of applications in location science. A new meta-heuristic algorithm for solving conditional p-median problems is proposed and results are compared to those of the previous studies. This algorithm produces much better results than the previous formulations.

Keywords: Conditional location problem, P-median problem, Harmony search algorithm, Meta-heuristics.

1. Introduction

Conditional location problem consists of finding p new facilities to serve a set of demand points, with q facilities already being available. When $q = 0$, the problem becomes unconditional. In the conditional p-median problems, once the new p locations are located, a demand can be served either by one of the existing facility or by one of the new facilities which ever is the nearest facility to the demand [1].

Every application to the p-median problems becomes a conditional model when already there exist some facilities in the area under study. As an example, if one wants to locate p warehouses in an area, it is an unconditional p-median problem. However, when q warehouses already exist in the area and we need to add p new warehouses, it becomes a conditional p-median problem.

The first studies on conditional location problems are due to [2,3]. In those references, the studied conditional problem was a conditional 1-center problem ($p=1$, and $q\geq 1$). The conditional location problem with $p\geq 1$ new facilities has been studied in two papers by Chen [4,5]. The former is about conditional p-median and p-center problem and the latter is on conditional p-center problems. Drezner solves the conditional p-center problem by an algorithm that requires the solution of $O(\log n)$ unconditional p-center problems (n being the number of demand nodes) [6]. A heuristic for the conditional p-median problem requiring solution of several unconditional p-median problems is discussed in [7]. The method proposed by Drezner is applicable to both planar and network configurations. Berman and Simchi-Levi suggested to solve the conditional p-median and p-center problems on a network by an algorithm that requires one time solution of an unconditional $(p + 1)$ median or $(p + 1)$-center problem [8]. Chen and Handler presented a relaxation algorithm for solution of the conditional p-center problem [5]. Chen and Chen presented an algorithm [9] for solving the conditional p-center problem, which relies on Drezner's observations [1], as well as a relaxation-based algorithm developed by Chen and Chen [10]. This new relaxation algorithm is applicable to both the continuous Euclidean and the discrete conditional p-center problems.

Berman and Drezner suggested a simple algorithm which solves both the conditional p-median and p-center problems on a network [1]. The algorithm requires one-time solution of an unconditional p-median or p-center problem using a shortest distance matrix. The conditional p-median problems were solved by CPLEX program providing superior results compared to those obtained by the method presented by Berman and Simchi-Levi.

In this article the conditional location problem is studied. Conditional location problems have a wide range of applications in location science. A new meta-heuristic algorithm for the solution of the conditional p-median problems is proposed and results are compared to those of the previous studies. This algorithm produce much better results than the previous formulations.

2. Formulation of the problem

Let $G = (N, L)$ be a network with N being the set of nodes, $|N|=n$ and L being the set of links. Consider a non-negative number w_i that represents the demand weight at node $i\in N$. Let

* *Corresponding Author: alikaveh@iust.ac.ir*
1 Professor, Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran
2 Graduate Student, School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran

d_{xy} be the shortest distance between any two nodes $x, y \in G$. Suppose that there is a set Q ($|Q| = q$) of existing facilities. Let $X=(X_1,...,X_p)$ and $Y=(Y_1,...,Y_q)$ be vectors of size q and p, respectively, where Y_i is the location of the existing facility i and X_j is the location of new facility j. The conditional p-median location problem can now be expressed as minimizing

$$f(x) = \sum_{i=1}^{n} w_i \min\{d(X,i), d(Y,i)\} \quad (1)$$

Where $d(X,i)$ (or $d(Y,i)$) is the shortest distance from the closest facility in X (or Y) to node i [1].

3. The algorithm of Berman and Simchi-Levi

The idea is to produce a new potential location representing all the existing facilities. If a demand point is utilizing the services of an existing facility, it will use the services of the closest existing facility. Thus, the distance between a demand point and the new location is the minimum distance among all the existing facilities. In order to force the formation of a facility at the new location, a new demand point is considered with a distance of zero from the new potential location and a large distance from all the other potential locations. The new distance matrix, denoted by D', is constructed by adding a new location a_0 (a new column) to D which represents the Q existing locations and a new demand point v_0 with an arbitrary positive weight. For each regular demand point (node) i, we have $d(i,a_0) = min_{k \in Q} \{d_{ik}\}$ and $d(v_0, a_0) = 0$. For each regular potential location node j, $d(v_0, j) = M$, where M is a large number. Again the nodes in Q and in the potential locations Q are removed. The program is tackled by solving the unconditional $p + 1$ median (or center) problem using D' [8].

4. The algorithm of Drezner

Drezner solved the conditional problems by defining a modified shortest distance matrix D' as

$$D'_{ij} = \min\{d_{ij}, \min_{k \in Q}\{d_{ik}\}\} \quad \forall_{i,j \in N} \quad (2)$$

It should be noted that D' is not symmetric even when D is symmetric. The unconditional p-median or p-center problem using the appropriate D' solves the conditional p-median or p-center problem. This is true since if the shortest distance from node i to the new p facilities is larger than $\min\{_{k \in Q}\{d_{ik}\}\}$ then the shortest distance to the existing q facilities is utilized [1].

5. Hybrid harmony search algorithm

In this paper we use a new meta-heuristic algorithm which is inspired by music, harmony search algorithm [11]. Kaveh and Nasr combined the modified harmony search algorithm with the greedy heuristic and obtained a robust algorithm called Greedy Harmony Search (GHS) [12]. It is discussed there that with a constant Harmony Memory Considering Rate (HMCR) there is a problem with diversification if the number of nodes in Harmony Memory (HM) is a small fraction of total nodes. That means if we choose HMCR around 0.95 to 0.99, the search space will totally be limited to the nodes of the HM matrix and the algorithm gets trapped in local optimum or at least the convergence rate decreases. On the other hand, if we choose the HMCR parameter around 0.4 to 0.7, this will be a good strategy at the beginning because the diversification of the search remains high at this stage and the algorithm searches the entire search space. However, as the algorithm gets through, the diversification should be decreased and the low HMCR makes the algorithm not to converge fast and damages the intensification. In order to conquer this problem, variable HMCR is introduced (similarly, Mahdavi et al., defined variable Pitch Adjusting Rate (PAR) and Bandwidth (bw) [13]). To combine harmony search with the greedy heuristic, we generate several random numbers in interval [0 1] for a pre-defined number of times (t) and compare those random numbers with variable HMCR (VarHMCR, will be defined in the next section) t times. In this way, we select several arrays of the column of the HM and several random nodes of the graph (depending on the VarHMCR). Finally we select the best of them according to their fitness values. Here we use this algorithm to solve the conditional p-median problem. Figure 1 illustrates how the greedy heuristic finds the best value from the first column for the first variable.

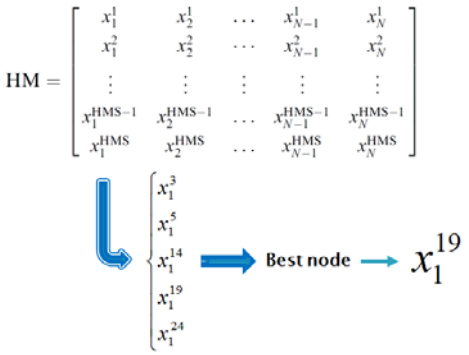

Fig. 1. Greedy heuristic finds the best node from the five randomly selected nodes

5.1. Definition of the problem and the algorithm parameters

The problem is similar to the unconditional p-median problem. As mentioned before, the purpose is to locate some facilities on some nodes of the graph in the way that the sum of distances between the customers and their nearest facilities becomes minimum. This is the discrete p-median problem and the minimum distance between each two nodes is calculated by the Dijkstra's algorithm. This algorithm was first suggested by Edsger Dijkstra [14] in 1959 and is a graph search algorithm that finds the shortest path between each two nodes. The objective function for p-median problem is:

$$f(x) = \sum_{i=1}^{m} w_i d(x, P_i) \quad (3)$$

$$d(x, P_i) = \min(d(x_j, P_i)) \quad : \quad x_j \in X \quad (4)$$

Where P_i is the node i of the graph and X is the set of selected medians and x represents the array of the set X. Formula 3 finds the nearest facility or service point *"x"* among the set of the medians x_j (set X) to the demand point p_i with Dijkstra's algorithm. Once the demand point specifies its nearest facility, a connection will be produced through the shortest path we found by Dijkstra's algorithm. Formula 4 calculates the sum of distances to their nearest service point multiplied by w_i (demand weight at node i) for all demand points (the number of demand points is m and the number of medians is p).

The parameters are defined variable as

$$VarHMCR = MinHMCR + (MaxHMCR - MinHMCR) \times \frac{j}{Max\text{Imp}} \quad (5)$$

$$VarPAR = MinPAR + (MaxPAR - MinPAR) \times \frac{j}{Max\text{Imp}} \quad (6)$$

$$Varbw = Maxbw + (Minbw - Maxbw) \times \frac{j}{Max\text{Imp}} \quad (7)$$

MaxImp denotes the number of iterations, and j is the number of current iteration. In Figure 2, only neighbors that are in the pre-defined coverage distance (bandwidth) are considered. Here, the maxbw is taken as 9 and the minbw is 5. Thus at the beginning of the algorithm the Varbw is 9 and the considered nodes are 2, 6, 8, 9, 15 and 21. At the end of the algorithm nodes with numbers 2, 8 and 15 can only be substituted by node number 1. If the number of the nodes is high and the number of medians we want to locate is low, MaxHMCR, MinHMCR, MaxPAR and MinPAR can be defined around 0.85, 0.7, 0.4 and 0.2 respectively, to control the intensification and diversification essentially. Maxbw and Minbw is defined considering the distances between the nodes. Therefore, we define MaxHMCR, MinHMCR, MaxPAR, MinPAR, Maxbw and Minbw first.

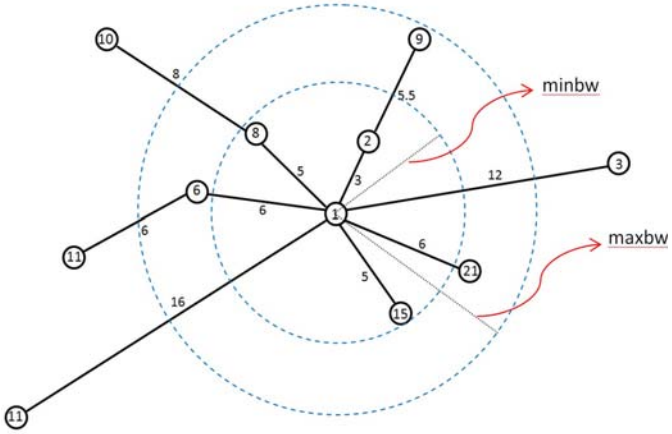

Fig. 2. A sample node and the coverage distance

5.2. Initialization of the harmony memory

In conditional p-median problems we want to locate p facilities on the network in which q facilities haves already been located. That means we need a harmony memory that have the same arrays (located facilities) in all rows. If we suppose $N=\{n_1,...,n_n\}$ is the set of nodes of the graph ($|N|=n$), $Q=\{a_1,...,a_q\}$ is the set of nodes in which facilities have already been located on ($|Q|=q$) and $M=N-Q=\{n_1, n_2,...,n_{m=n-q}\}$ is the set of free nodes of graph in which new facilities can be established on ($|M|=n-q$). The solutions in this problem are p-combinations of the set of M with n - q nodes (free nodes) because the order of locations is not relevant and is denoted by $C(M,p)$. Consequently to build the *HM*, we only need to produce a random p-combination of the nodes of the graph *HMS* times. For conditional p-median problems the *HM* will have the following general form:

$$HM = \begin{bmatrix} a_1^1 & a_2^1 & \cdots\cdots & \cdots & a_q^1 & C^1(M,p) \\ a_1^2 & a_2^2 & \cdots\cdots & \cdots & a_q^2 & C^2(M,p) \\ & & & \vdots & & \\ a_1^{HMS} & a_2^{HMS} & \cdots & a_q^{HMS} & C^{HMS}(M,p) \end{bmatrix} \quad (8)$$

5.3. Improvising a new harmony

Similar to HM, for improvising a new harmony (hnew), we should first mention the facilities that have already been located. Thus the hnew has the following form:

$$hnew = \{a_1, a_2, \ldots, a_q, b_1, b_2, \ldots, b_p\} \quad (9)$$

Again $Q=\{a_1,...,a_q\}$ is the set of located facilities. The algorithm tries to find p medians for hnew after a_q (for all hnews $\{a_1, a_2,...,a_q\}$comes first). Thus to improvise a new harmony, it is enough to improvise a harmony with p arrays out of set M, so the length of hnew is $p+q$. The medians after a_q are determined based on the following three rules:

1. Memory consideration (considering VarHMCR),
2. Pitch adjustment (considering VarPAR), and
3. Random selection (considering VarHMCR).

5.4. Updating the harmony memory

If the fitness function value of $hnew=\{a_1, a_2,...,a_q,b_1, b_2,...,b_p\}$ is better than the worst fitness function value of harmonies in the HM, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

5.5. Checking the stopping criterion

If the stopping criterion (maximum number of improvisations or unchanging the best objective function value for a predefined number of iterations) is satisfied, computation is terminated. Otherwise, Steps 3 and 4 are repeated.

6. Computational results

We applied our method to the OR-Library tests proposed by Beasley [15]. Similar to Drezner`s article we assume that the first ten nodes of the given network are existing facilities

($q=10$) and p new facilities need to be located. With this assumption the *HM* is calculated as follows:

$$HM = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & P^1(M,p) \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & P^2(M,p) \\ & & & & & & \vdots & & & & \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & P^{10}(M,p) \end{bmatrix} \quad (10)$$

For test numbers 35 to 40, Berman and Drezner could not solve the problem due to the lack of memory. We solved them but there is no guarantee that the obtained values are the optimal ones. Although the best results obtained by the greedy harmony search for tests with $n \geqq 800$ and $p \leqq 10$ (test numbers 35, 36, 38, 39) are very likely to be the optimal values. Table 1 shows the computational results. We have taken the CPU time for drezner`s algorithm and Berman and Simchi`s algorithm from Drezner`s paper (in seconds). From this table, it is obvious that when the number of medians is not more than a hundred, harmony search is much better to use and when the number of medians increases over one hundred, other two algorithms are better to employ. Best result in each test is made bold in the table.

7. Concluding Remarks

In this paper a brief description is provided for the algorithms by Berman, Drezner and Berman and Simchi, and

Table 1. Computational results

Test No.	N	P	Optimum	Berman , Simchi`s algorithm value	CPU time for Berman Simchi`s algorithm	Drezner`s algorithm value	CPU time for drezner`s algorithm	GHS value	Error % of GHS	CPU time for GHS
1	**100**	**5**	4369	4369	1.12	4369	1.64	4369	0	**0.000**
2	**100**	**10**	3087	3087	1.20	3087	1.73	3087	0	**0.015**
3	**100**	**10**	3417	3417	1.51	3417	2.31	3417	0	**0.031**
4	**100**	**20**	2392	2392	0.87	2392	1.28	2392	0	**0.047**
5	**100**	**33**	916	916	0.90	916	1.26	916	0	**0.218**
6	**200**	**5**	6153	6153	71.18	6153	62.96	6153	0	**0.015**
7	**200**	**10**	4778	4778	5.42	4778	5.31	4778	0	**0.078**
8	**200**	**20**	3877	3877	4.50	3877	4.46	3877	0	**0.250**
9	**200**	**40**	2370	2370	4.26	2370	4.25	2370	0	**1.015**
10	**200**	**67**	1086	1086	4.02	1086	3.91	1086	0	**3.525**
11	**300**	**5**	6682	6682	51.05	6682	33.05	6682	0	**0.031**
12	**300**	**10**	5661	5661	82.70	5661	90.71	5661	0	**0.110**
13	**300**	**30**	3967	3967	12.75	3967	12.35	3967	0	**1.718**
14	**300**	**60**	2716	2716	14.61	2716	15.24	2716	0	**9.594**
15	**300**	**100**	1569	1569	**11.84**	1569	11.86	1569	0	31.750
16	**400**	**5**	7443	7443	1221.44	7443	1094.01	7443	0	**0.094**
17	**400**	**10**	6136	6136	742.42	6136	577.35	6136	0	**0.281**
18	**400**	**40**	4449	4449	41.44	4449	42.31	4449	0	**2.891**
19	**400**	**80**	2668	2668	**27.22**	2668	28.13	2668	0	29.156
20	**400**	**133**	1648	1648	**26.74**	1648	27.39	1648	0	85.954
21	**500**	**5**	7693	7693	60.06	7693	57.56	7693	0	**0.093**
22	**500**	**10**	7662	7662	1113.70	7662	645.17	7662	0	**0.390**
23	**500**	**50**	4331	4331	58.50	4331	57.50	4331	0	**11.438**
24	**500**	**100**	2799	2799	**50.70**	2799	52.44	2799	0	51.875
25	**500**	**167**	1726	1726	48.95	1726	**48.05**	1726	0	416.243
26	**600**	**5**	8792	8792	5458.86	8792	3674.21	8792	0	**0.110**
27	**600**	**10**	7632	7632	*	7632	*	7632	0	**0.468**
28	**600**	**60**	4316	4316	102.58	4316	99.22	4316	0	**29.437**
29	**600**	**120**	2913	2913	82.49	2913	**80.55**	2914	0.03	501.875
30	**600**	**200**	1896	1896	**74.47**	1896	77.35	1897	0.05	610.328
31	**700**	**5**	9151	9151	*	9151	2449.34	9151	0	**0.156**
32	**700**	**10**	8637	10641	*	10641	*	8637	0	**0.500**
33	**700**	**70**	4460	4460	154.18	4460	157.55	4460	0	**133.140**
34	**700**	**140**	2904	2904	*	2904	*	2904	0	**186.062**
35	**800**	**5**	9374***	**	-	**	-	9374	0	**0.155**
36	**800**	**10**	9155***	**	-	**	-	9155	0	**0.609**
37	**800**	**80**	4879***	**	-	**	-	4879	0	**112.355**
38	**900**	**5**	10117***	**	-	**	-	10117	0	**0.375**
39	**900**	**10**	8799***	**	-	**	-	8799	0	**0.812**
40	**900**	**90**	5000***	**	-	**	-	5000	0	**77.481**

* Berman and Drezner could not find optimal within two hours.
** Berman and Drezner could not solve the problem due to the lack of memory.
*** Since these tests solved only with GHS, there is no guarantee to get optimal value although it is very likely.

then a new harmony search algorithm is presented for solving the conditional p-median problem. Here we used the combinations of the harmony search algorithm and the greedy heuristic for solving a discrete problem. The discrete space and the conditional problem require the use of modified parameters and harmony memory for the algorithm. Our computational results for the harmony search are compared with the other two algorithms and it is shown that the presented algorithm is quite fast and more efficient. From the computational results table, it can be seen that as the number of medians inreases, the execution time for the harmony search algorithm increases, and the previous algorithms run slightly faster for a large number of medians. However, in general the proposed algorithm outperforms the previous exact algorithms, especially when the number of medians is not high.

Acknowledgement: The first author is grateful to the Iran National Science Foundation for the support.

References

[1] Beramn O, Drezner Z. A new formulation for the conditional p-median and p-center problems, Operations Research Letters, 2008, Vol. 36, pp. 481-483.
[2] Handler YG, Mirchandani PB. Location on Networks Theory and Algorithms, The MIT Press, Cambridge, MA, 1979.
[3] Lin CC. A note about the new emergency facility insertion in an undirected connected graph, in Sixth Annual Pittsburgh Conference on Modelling Simulation, Pittsburgh, Penn, 1975.
[4] Chen R. Conditional minisum and minimax location-allocation problems in euclidean space, Transportation Science, 1990, Vol. 22, pp. 158-160.
[5] Chen R, Handler GY. The conditional p-center in the plane, Naval Research Logistics, 1993, Vol. 40, pp. 117-127.
[6] Drezner Z. On the conditional p-center problem, Transportation Science, 1989, Vol. 23, pp. 51-53.
[7] Drezner Z. On the conditional p-median problem, Computers and Operations Research, 1995, Vol. 22, pp. 525-530.
[8] Beramn O, Simchi-Levi D. The conditional location problem on networks, Transportation Science, 1990, Vol. 24, pp. 77-78.
[9] Chen D, Chen R. A relaxation-based algorithm for solving the conditional p-center problem, Operations Research Letters, 2010, Vol. 38, pp. 215-217.
[10] Chen D, Chen R. New relaxation-based algorithms for the optimal solution of the continuous and discrete p-center problems, Computers and Operations Research, 2009, Vol. 36, pp. 1646-1655.
[11] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search, Simulation, 2001, Vol. 76, pp. 60-68.
[12] Kaveh A, Nasr H. Solving conditional and unconditional p-center problem with greedy harmony search and its applications, Scientia Iranica A , 2011, Vol. 18, pp. 867-877.
[13] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems. Applied Mathematiacl Computation, 2007, Vol. 188, pp. 1567-1579.
[14] Dijkstra EW. A note on two problems in connexion with graphs, Numerische Mathematik, 1959, Vol. 1, pp. 269-271.
[15] Beasley JE. A note on solving large p-median problems. European Journal of Operational Research, 1985, Vol. 21, pp. 270-273.