Journal of Industrial and Systems Engineering
Vol. 6, No. 1, pp48-57
Spring 2012

JISE

The P-Center Problem under Uncertainty

Majid Taghavi[1], Hassan Shavandi[2*]

[1,2]Dept. of Industrial Engineering, Sharif University of Technology, Tehran, Iran
[1]majid.t@gmail.com, [2]shavandi@sharif.edu

ABSTRACT

Facility location decisions play a prominent role in strategic planning of many firms, companies and governmental organizations. Since in many real-world facility location problems, the data are subject to uncertainty, in this paper, we consider the P-center problem under uncertainty of demands. Using Bertsimas and Sim approach, we develop a robust model of the problem as an integer programming model. Furthermore, we develop a tabu search algorithm for solving the problem. Finally we use design of experiments (DOE) to adjust the parameters of tabu search algorithm. The numerical results of algorithm are presented accordingly.

***Keywords*:** Facility Location, Robust Optimization, P-Center, Tabu Search.

1. INTRODUCTION

Facility location decisions play a prominent role in strategic planning of many firms, companies and governmental organizations. Deciding where to locate a new warehouse for a factory, where to place the fire stations, where to open a new branch of a bank and where to open a new store for chain storesare practical instances of facility location problems. These decisions are not only costly and irreversible (or at least very costly and time consuming to reverse), but they use up a lot of organization's resources also.

Facility location models can be classified according to their objectives, parameters, solutions and many other attributes. Jia *et al.* (2007), presented eight of the most common criteria that are used to classify the traditional facility location models. One of the most important criteria with which facility location models can be classified is the objective function. In the covering models, the objective is maximizing of covering demand nodes by a predefined number of facilities. If the objective function is to minimize the maximum distance between demand nodes and facilities, the problem belongs to the P-center problems.

Another criterion to categorize the facility location models is the state of input parameters. Sometimes the exact value of parameters is available and we formulate the problem under certainty,

*Corresponding Author

ISSN: 1735-8272, Copyright © 2012JISE . All rights reserved.

but it rarely happens in the real world. In contrast, Uncertainty assumption for problem parameters includes two main categories: Stochastic and robust models. In stochastic models, the parameters are not known exactly, but the information of their probability distribution is available. While in robust models, probabilities are not known, and uncertain parameters are specified either by discrete scenarios or by continuous ranges.

Dealing with facility location problems in real-world problems, it is clearly observable that scientists must tackle with parameters uncertainty. In fact, in the real-world application of facility location models, one cannot ignore the possibility that a small uncertainty in data can make the acquired optimal solution completely invalid. Because of this high importance, a vast literature could be found on the subject of optimization under uncertainty. Soyster (1973) developed a linear programming to deal with uncertainty that were feasible for all parameters that change in a specific interval. However, his method was creating over-conservatism solutions. To address this issue Ben-Tal and Nemirovski(1998), and El-Ghaoui(1997) considered an ellipsoid as the allowed space for parameter changes. They presented an efficient algorithm for convex optimization. However, a practical drawback was that it leaded to solving a Conic Quadratic problem which was not applicable directly to discrete optimization problems. Afterwards, Bertsimas and Sim (2003) presented a different approach. It had the ability of being applied directly to discrete optimization problems. Besides, they defined control parameter which controls the level of conservatism of the problem.

The p-center problem is one of the well-known NP-hard discrete location problems. There are a lot of applications of that in real-world problems. Some researchers worked on P-center problem under uncertainty. However, it is worthy to deal with it using Bertsimas and Sim approach as a new and easy-to-use method. In this paper, we apply the Bertsimas and Sim approach to the p-center problem under uncertainty to deal with the uncertainty of demands with having the ability to control the conservatism of the solutions.

2. PROBLEM FORMULATION

Let $G(N, L)$ be a graph with a set N of n nodes and a set L of m links. We define d_{ij} the shortest distance between nodes$i$and$j$. Also, we define w_i as the demand at any node$i \in N$. We assume that in case of several facilities availability on the network, the customer will choose the nearest one. The objective of the p-center problem is to locate p facilities on the network in such a way that the maximum weighted distance in the network is minimized.

In order to formulate the problem as an integer programming model, it is needed to define binary variables x_{ij} and x_j as follows:

$$x_j = \begin{cases} 1, & \text{if one of the } p \text{ facilities is located at node } j \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{if demand node } i \text{ is assigned to a facility located at node } j \\ 0, & \text{otherwise} \end{cases}$$

According to above explanations, the objective function of p-center problem is:

$$Minimize \left\{ \max_i \sum_{j=1}^{n} w_i . x_{ij} . d_{ij} \right\}$$

Now we assume that the demand of each node is subjected to data uncertainty. There is no information on their probability distribution, but they are allowed to change in an interval, i.e.,

$$w_i \in [w_i - \delta_i , w_i + \delta_i] \quad \forall i$$

The initial integer programming model of p-center problem under uncertainty can be written as follows.

$$Minimize\ T$$

$$s.t.$$

$$\sum_{j=1}^{n} w_i . x_{ij} . d_{ij} \leq T \qquad , \forall i \tag{1}$$

$$\sum_{j=1}^{n} x_j = p \tag{2}$$

$$x_{ij} - x_j \leq 0 \qquad , \forall i , j \tag{3}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad , \forall i \tag{4}$$

$$x_{ij} \in \{0,1\} \qquad , \forall i , j \tag{5}$$

$$x_j \in \{0,1\} \qquad , \forall j \tag{6}$$

Constraints (1) calculate the weighted distance for every node and bound it to T. Constraint (2) assures that p facilities must be located on the network. Constraints (3) do not let the problem to assign a demand node to a node in which no facility has been located. Constraints (4) make certain that each demand node is assigned to just one facility. Constraints (5) and (6) limit the decision variables to binary values. It is obvious from above model that there are $n^2 + n$ binary decision variables and $(n + 1)^2$ constraint.

Since there is no exact information about the value of demands, we should tackle with uncertainty using robust techniques. Among all of them, the Bertsimas and Sim (2003) approach seems to be more appropriate than others because:

- It provides a linear counterpart problem for the main problem. This makes the technique available for discrete network optimization problems.
- It allows a full control on the level on conservatism of the solution.
- It is computationally tractable both practically and theoretically.
- The number of added variables and constraints is less than other techniques.

According to Bertsimas and Sim approach, the counterpart problem which is the final model that should be solved will be:

$$Minimize\ T$$

$$s.t.$$

$$\sum_{j=1}^{n} w_i . x_{ij} . d_{ij} \ + \ z\Gamma + \sum_{j \in J_i} p_j \leq T \ , \quad \forall i \tag{7}$$

$$z + p_j \geq \delta_i . x_{ij} . d_{ij}, \quad \forall i \quad , \forall j \in N$$

$$\sum_{j=1}^{n} x_j = p \tag{8}$$

$$x_{ij} - x_j \leq 0, \forall i \quad , \forall j \in N$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2, \dots , n$$

$$x_{ij} \in \{0,1\}, \quad i, j = 1,2, \dots , n$$

$$x_j \in \{0,1\}, \quad j = 1,2, \dots , n$$

$$p_j \geq 0, \quad \forall j \in N$$

$$z \geq 0$$

Where, z and p_j are variables of dual problem and Γ is the parameter that adjusts the level of conservatism of the solution.

If $\Gamma = n$, the model will be over conservatism (equal to Soyster method). If $\Gamma = 0$, the model will be no longer a robust optimization problem. By changing Γ in the interval $[0, n]$, we can control the level of conservatism of the solution. According to Bertsimas and Sim, we assume that some of demand nodes face the uncertainty on demand. If the demand of up to Γnodes change, the solution will be feasible. Even if more than Γ of demand nodes change, the solution will remain feasible with a very high probability. We will present these probabilities in last section.

3. SOLUTION METHOD

Appropriate modeling is just the first challenge of solving a facility location problem. In fact, the main (and perhaps the most difficult) challenge is to find the optimal solution through a specific approach. Besides, the above problem is NP-hard because it is proved that the p-center problem is NP-hard. Therefore we should try to solve it through heuristic approaches. A vast range of different heuristic approaches could be found in the literature of facility location. Arostegui *et al.* (2006) compared the relative performance of Tabu Search, Simulated Annealing and Genetic Algorithms on various types of facility location problems under time-limited, solution-limited, and unrestricted conditions. The results indicated that tabu search shows very good performance in most cases. The performance of simulated annealing and genetic algorithm were more partial to problem type and the criterion used. Thus, in general it can be concluded that tabu search should be applied to solve the derived model.

3.1. Tabu Search Method

Glover (1989, 1990) introduced tabu search algorithm as a heuristic method. Afterwards, a large number of scientists applied it to their combinatorial optimization problems. In very complex problems, tabu search is able to find a near optimum solution or in some cases, the exact optimum solution.

Tabu search needs a feasible solution to launch the process of solving the problem. There are different methods to create a feasible solution. For example, one can choose first p demand nodes by their indexes $(x_1, x_2, \dots, x_p)$ to locate facilities. However, the point is that selecting a good feasible solution to start will affect not only the run time of the tabu search, but the quality (nearness to the optimal solution) of the solution also. Therefore it is better to choose a more wise approach. In this paper, the greedy method has been selected since it is found very common in creating a feasible solution in literature. This method sorts the demand nodes by their weights (w_i) and selects the first p nodes having the greater weights and locates the facilities. Then, the remained demand nodes will allocate to the nearest located facility. The value of z and p_j will be the minimum possible values which satisfy constraints (2) of counterpart problem.

3.1.1. Neighborhood structure

Another important stage in designing a tabu search method is the neighborhood structure. The neighborhood, according to definition, are solutions obtained by applying a single local transformation in the search space. Thus, the definition of a neighborhood could be as one of the following:

1- Changing the assignment of one demand node from one facility to another facility

$$x_{ij} = 1 \quad \leftrightarrows x_{ik} = 1$$

2- Changing the location of a facility and move all of the assignment to new location

$$x_j = 1 \ , \qquad x_{ij} = 1 \quad \leftrightarrows x_k = 1 \ , \qquad x_{ik} = 1$$

After each transformation, the values of z and p_j will be adjusted accordingly.

When the algorithm starts, it considers the given feasible solution by greedy algorithm. Then the method creates all possible neighborhoods and calculates the objective function for all neighborhoods. If there exists a neighborhood with better objective function value, the method will replace that solution with that of greedy algorithm. This will continue until no better neighborhood could be found.

3.1.2. Tabu List

Tabu list is an element which distinct the tabu search method from other methods such as local search. In fact, the algorithm needs something to prevent it from moving back to visited area when it is moving toward the optimal solution. In other words, tabu list is the short term memory of the algorithm which prevents cycling when moving away from local optima through non-improving moves. In defined problem, there are two possible way to insert a transformation to tabu list:

1- After each transformation which includes a change in the allocation of a demand node to a facility (x_{ik} instead of x_{ij}), if a better value for objective function found, allocating demand node i to facility located in j is a tabu move.
2- After each transformation which includes a change in the location of a facility (j instead of i), if a better value for objective function found, Locating a facility in demand node i again is a tabu move.

Designing a tabu search algorithm, the size of tabu list is a very important factor. The size of tabu list depends on the problem. However, it should neither be very short nor very long. In next section,

some experiences to adjust the tabu list size are presented.

3.1.3. Aspiration Criteria

After some iteration and considering all neighborhoods, algorithm will reach a state in that no better neighborhood could be found. In this case, we need to direct the algorithm into some unexplored areas of the search space. But beforehand, in order to be sure about any possible interesting neighborhood in current area of search space, we check the solutions currently in tabu list. If a better solution could be found, the algorithm will continue the process with that solution. Otherwise, we are almost sure that no interesting solution will be found in current area of search space. Therefore, the algorithm goes to other areas of search space by diversification.

3.1.4. Diversification

Diversification is another strength point of tabu search algorithm, since it survives the algorithm from getting stuck into local optimums. For that, it is needed to define a long term memory which counts the number that every component (facility or allocation) participates in forming a solution. For instance, it counts the iterations in that a facility has been located in demand node j. Then it must be calculated as a percentage of all solutions. Diversification is forming new solutions using components which their percentages are below a pre-specified percentage called "low bound of diversification". In this problem, if r demand nodes fall behind the low bound of diversification, the algorithm will replace them with r demand nodes which participated in forming solution the most. The low bound of diversification is an important parameter of algorithm which affects the quality and run time of the algorithm. Thus it is needed to adjust for the algorithm. Design of experience to adjust this parameter is presented in next section.

3.1.5. Stop Criteria

Similar to every other algorithm, this algorithm needs a criterion to terminate the program. Some different criteria could be found in the literature of tabu search such as after reaching a pre-specified iteration or value of objective function. However, since the diversification is the key part of the algorithm, the number of diversification is chosen as the termination criterion. This could be also an assurance that the algorithm will do the diversification sufficiently.

3.2. Adjusting the Parameters of Tabu Search Method using Design of Experiments (DOE)

As it is mentioned in advance, it is needed to tune the algorithm parameters in order to work more efficiently. This includes both adjusting the algorithm parameters -including tabu list size (factor A), lower bound of diversification (factor B), and the number of diversification (factor C) - and considering their effects on each other. In design of experiments, the 2^k method has been used. It determines 2 levels (low and high) for each factor and considers the effects of them on each other. The levels are shown in table 1.

Table 1 Low and high level for each factor.

	Factor		
	A	B	C
Low	$\frac{n}{4}$	5%	$\frac{n}{5}$
High	n	75%	n

For 3 factors and 2 levels, we need at least 8 experiments. We have done 32 experiments in order to make our results more precise. All experiments have been done under below circumstances:

$$n = 20 \ , \ p = 5 \ , \ \Gamma = 10$$

Besides, weights of demand nodes, distance of nodes and the noise of each weight have been produced by a program randomly and are fixed for all 32 experiments. The optimum value of objective function (acquired by exact algorithm which explores all possible solutions) is 1765.89. The results for 32 experiments are shown in table 2.

Table 2 Results for 32 experiments

Factors			**Objective function values**			
A	B	C	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Low	Low	Low	2200.72	2058.34	1836.75	2058.34
Low	Low	High	1869.36	1895.25	1904.86	1955.03
Low	High	Low	2116.52	1869.36	2177.93	2173.67
Low	High	High	1889.97	1988.77	1765.89	1836.22
High	Low	Low	2200.72	2116.52	2116.52	2200.72
High	Low	High	1978.63	1765.89	1869.36	1901.82
High	High	Low	2141.52	1871.37	2200.72	2177.93
High	High	High	2058.34	1819.64	1871.37	1765.89

Some calculations have been done on data of table 2 related to 2^k method. The results of them including the variance analysis are shown in table 3.

Table 3 Variance analysis of experiments

	Sum of Squares	**Degree of freedom**	**Mean of Squares**	**F_0**	**F-distribution**
A	6611.925013	1	6611.925013	0.509750571	0.495546705
B	1296.93245	1	1296.93245	0.099987834	0.759937038
C	357299.8578	1	357299.8578	27.54626018	0.000775289
AB	2504.19645	1	2504.19645	0.193062621	0.672006633
AC	11582.42	1	11582.42	0.892954049	0.372339356
BC	223.1328125	1	223.1328125	0.017202566	0.89888826
ABC	10133.89661	1	10133.89661	0.781279216	0.402531475
Error	311301.6625	24	12970.9026		
Total	700954.024	31			

If $\mathrm{F-distribution} \leq \mathrm{F}_0$, it may be concluded that the factor or the interaction between factors are important to our algorithm. In this case, it is obvious that factor C is very important and factor A and interaction between factors A and C (AC) is of less important. Now, in order to set the regression line:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_3 + \hat{\beta}_2 x_1 x_3 + \hat{\beta}_3 x_1 x_2 x_3$$
$$= 1989.18 - 105.66 x_3 - 19.025 x_1 x_3 + 17.78 x_1 x_2 x_3$$

In which $x_1$and x_2 and x_3 are coded variables[2]. The results for $\hat{y}$ are shown in table 4.

Table 4 $\hat{y}$ results in each combination of factors

A	B	C	$\hat{y}$
-1	-1	-1	2058.035
-1	-1	1	1920.325
-1	1	-1	2093.595
-1	1	1	1884.765
1	-1	-1	2131.645
1	-1	1	1846.715
1	1	-1	2096.085
1	1	1	1882.275

As we expected in advance, the best case is related to high level of factor C. The next stage of importance is related to high level of AC.As a result, the tabu list size must be set equal to n and the number of diversification equal to n too. There is no specific result for the low bound of diversification. So, we adjust it according to some different experiments as 50 percent.

4. EXPERIMENTAL RESULTS

According to above adjustments, the results of some experiments are presented in table 5. In some cases, the comparison between the result of tabu search and that of exact algorithm are presented. The weight of each demand node has been produced randomly between 10 and 100. Besides, all experiments have done on a Sony computer which has a 1.66 GHz CPU and 1 GB RAM. The used software for implementation of algorithm is Wolfram Mathematica 6.0[3].

According to table 5, it is obvious that the implemented algorithm works properly for $n \leq 20$. For greater number of nodes, because of run time of exact algorithm, it was practically impossible to find the optimum value. However it can be guessed that for very large number of n, there is possibly a gap between the result obtained from tabu search and optimal value. But, since there is no access to optimal value, the result of tabu search would be applicable in real-world problems.

As we mentioned in advance, the role of Γ in this problem is to control the degree of conservatism of the solutions. It is assumed that up to Γ number of demand nodes are allowed to have fluctuation in their weights. Bertsimas and Sim proved that even if more than Γ changes, the solution will remain feasible with a very high probability. We analyzed the role of Γ in a problem with 10 demand nodes and 2 facilities. The data were fixed for 10 iteration with different possible values for Γ. The results are shown in table 6.

---

[2] Coded Variables' formulation is :

$$x = \frac{x - \frac{x_{high}+x_{low}}{2}}{\frac{x_{high}-x_{low}}{2}}$$

[3]http://www.wolfram.com/products/mathematica/index.html

Table 5 Numerical results

n	p	Γ	$RunTime$	TS^1	$OptimalValue^2$
3	1	2	0.06	1570.71	1570.71
4	1	3	0.14	3257.06	3257.06
5	3	4	0.64	1715.54	1715.54
6	2	4.5	0.9	1380.17	1380.17
7	2	3	2.1	1646	1646
8	2	7.5	3.79	2213.13	2213.13
9	2	7	4.89	2671.96	2671.96
10	2	5	4.34	2334.99	2334.99
11	3	7	8.93	1242.52	1242.52
12	7	8	60.93	978.2	978.2
12	8	8	189.04	596.34	596.34
15	3	10	29.15	3158	3158
20	5	13	378.75	1840.63	1840.63
30	5	20	271.29	3908.43	---
40	6	30	732.92	4180.82	---
80	4	60	7138.47	1024.4	---
100	10	70			---

1- The result of Tabu search algorithm
2- The result of exact algorithm (Optimal value)

Table 6 Different results for one specific problem with changes in Γ

Γ	**Value**	**Probability of violation**	**Increase in value of objective function (%)**
0	2738.59	0.62	0
1	2738.59	0.5	0
2	2821.77	0.37	3.037
3	2849.38	0.27	4.046
4	2849.38	0.17	4.046
5	2850.23	0.11	4.077
6	2902.99	0.054	6.003
7	2902.99	0.03	6.003
8	2902.99	0.01	6.003
9	2902.99	0.005	6.003
10	2982.28	0.00099	8.898

In table 6, the probabilities of violation for constraints have been presented too. Clearly, the more the method moves toward more conservatism solutions, the more the probability move toward zero.

5. CONCLUSION

To sum up, we used the Bertsimas and Sim approach of dealing with uncertainty to model the p-center problem under uncertainty of demands. The derived model is a mixed integer programming model and we developed a tabu search method to solve the problem. The parameters of the tabu

search method were adjusted using DOE method. The performance of developed tabu search method is very efficient and we see the most solutions of sample problems are optimal. A potential future research area is analyzing this problem considering an uncertain number of facilities may be constructed in future. Therefore considering this assumption and modeling it by robust approaches can be an interesting problem.

REFRENCES

[1] Arostegui M., Kadipasaoglu N., Khumawala M. (2006), An empirical comparison of Tabu Search, Simulated Annealing, and Genetic Algorithms for facilities location problems; *International Journal of Production Economics*103; 742-754.

[2] Ben-Tal A., Nemirovski A. (1998), Robust convex optimization, *Math. Operations Research* 23; 769-805.

[3] Bertsimas D., Sim M. (2003), Robust discrete optimization and networkflows;*Mathematical Programming Series B*; 98, 49–71.

[4] El-Ghaoui, Lebret H. (1997), Robust solutions to least-square problems to uncertain data matrices;*SIAM Journal on Matrix Analysis and Applications* 18; 1035-1064.

[5] Glover F. (1989), Tabu Search Part I;*ORSA Journal on Computing* 1; 190-206.

[6] Glover, F (1990), Tabu Search Part II;*ORSA Journal on Computing* 2; 4-32.

[7] Jia H. Fernando, Dessouki G. (2007), A modeling framework for facility location of medical services for large-scale emergencies;*IIE Transactions* 39; 41–55.

[8] Soyster A.L. (1973), Convex programming with set-inclusive constraints and applications to inexact linear programming; *Operations Research* 21; 1154-1157.