Simulated annealing metaheuristic for solving P-median problem

Article · January 2008

1 author:

Abdulrahman Alkhedhairi
King Saud University

Int. J. Contemp. Math. Sciences, Vol. 3, 2008, no. 28, 1357 - 1365

Simulated Annealing Metaheuristic for Solving P-Median Problem[1]

A. Al-khedhairi

Department of Statistics and OR, College of Science
King Saud University, P.O. Box: 2455
Riyadh: 11451, Saudi Arabia
akhediri@ksu.edu.sa

Abstract

We present an efficient metaheuristic based on Simulated Annealing for solving the p-Median problem. These ideas are tested on the well known OR-Lib set of problems with excellent results.

Keywords: p-Median Problem; location; Metaheuristic; Simulated Annealing (SA)

1 Introduction

The p-median problem, first introduced by Hakimi (1964), is to find p facility locations which will minimise the sum of weighted distances between demand points (customers) and their respective nearest facilities. Such a model would be useful in the cases where the service provided by the facilities is demanded on a regular, steady basis. So, locating supermarkets and administrative offices are examples of such a problem. The highlights in the development of the p-median problem are: (i) the development of the heuristic solution procedure by Tietz and Bart (1968), (ii) the formulation of the problem as an integer linear programme by ReVelle and Swain (1970) with the discovery that integer solutions result in most cases when the relaxed linear programming problem is solved, and (iii) the development of efficient procedures which almost always obtain and verify optimal solutions by solving the Lagrangian relaxation of the p-median problem by Cornuejols *et al.* (1977), Narula *et al.* (1977) and Marianov and Serra (2002).

The p-median problem is NP-hard (Kariv and Hakimi, 1979), optimal solutions to large problems are difficult to obtain; the problem size rapidly exceeds

[1]This project is supported by College of Science - Research centre project No (Stat/2006/13).

the capability of even modern computers and computer codes. Heuristics, particularly interchange heuristics, of one form or another, must then be resorted in order to estimate good solutions for these larger problems.

Let:

I = set of demand nodes indexed by i, $I = \{1, ..., N\}$.

J = set of candidate facility sites indexed by j, $J = \{1, ..., M\}$.

h_i= demand at node i.

d_{ij} = distance between demand node $i \in I$ and candidate site $j \in J$,

p = number of facilities to be located,

$$X_j = \begin{cases} 1 & \text{if we locate a facility at candidate site } j \in J. \\ 0 & \text{otherwise.} \end{cases}$$

$$Y_{ij} = \begin{cases} 1 & \text{if demand node } i \in I \text{ is assigned to a facility at candidate site } j \in J. \\ 0 & \text{otherwise.} \end{cases}$$

With these definitions and notations the p-median problem can be formulated as follows:

$$Minimise \qquad \sum_{i \in I} \sum_{j \in J} h_i \, d_{ij} \, Y_{ij} \tag{1}$$

subject to:

$$\sum_{j \in J} Y_{ij} = 1 \qquad \forall i \in I \tag{2}$$

$$Y_{ij} - X_j \leq 0 \qquad \forall i \in I, \; j \in J \tag{3}$$

$$\sum_{j \in J} X_j = p \tag{4}$$

$$X_j, \, Y_{ij} \in \{0, 1\} \qquad \forall i \in I, \; j \in J \tag{5}$$

The objective function (1) minimises the sum of the (weighted) distances between the demand nodes and its closest open facility. Constraint (2) ensures that each demand node is assigned to exactly one facility, while constraints (3) restrict demand nodes to be assigned to open facilities. Constraint (4) stipulates that p facilities are to be located. Finally, constraints (5) refer to integrality constraints.

The aim of this paper is to design an efficient simulated annealing metaheuristic which find the optimal (or near optimal) solution of the p-median problem.

Next section describes the basics of the simulated annealing metaheuristic. Section 3 represents the proposed simulated annealing metaheuristic to solve the p-median problem. The discussion and computational results of the proposed heuristic are given in section 4.

The Simulated Annealing Metaheuristic

Simulated Annealing (SA) is a local optimization method for solving hard combinatorial optimization problems. Interest in this method started from the work of Kirkpatrick *et al.* (1983), and Cerny (1985) who showed how a model for simulating the annealing process of solids, as proposed by Metropolis et al. (1953), could be used for optimization problems. Since then, SA has been applied to many optimization problems in areas such as locational analysis, image processing, molecular physics and chemistry, and job shop scheduling Eglese (1990). The optimization problem which usually solved by SA can be described in simple words as follows: For each optimization problem, there is a set S of feasible solutions, each solution s having a cost function $f(s)$. The goal is to find a feasible solution of minimum cost function. In order to adapt a search algorithm to solve the above-described problem, one must additionally define a neighborhood. A neighborhood of a solution is a set of solutions that can be obtained by making a move in the current solution, a move being the change in value of one or more variables.

A local optimization algorithm like SA finds a solution in the following way: Starting with an initial solution s generated by other means, it repeatedly attempts to find a better solution by moving to a neighboring solution with lower cost function, until it reaches a solution, for which none of its neighbors have a lower cost function. Such a solution is called a local optimum.

Simulated annealing is motivated by the desire to avoid getting trapped in poor local optima, and hence, occasionally allows "uphill moves" to solutions of higher cost, under the guidance of a control parameter called the "temperature".

SA is essentially a type of local search algorithm. However, it differs from other local search algorithms in that it not only accepts solutions that improve the objective function, it also accepts inferior solutions with a certain probability in order to avoid being trapped in a local optimum.

The general SA algorithm starts with an initial solution, perhaps chosen at random. A neighborhood of the solution is generated by some suitable mechanism and the change in the objective function is calculated. If a reduction in the objective function is found, the current solution is replaced by the neighborhood solution, otherwise, the neighborhood solution is accepted with a certain probability. The probability of accepting an uphill move is normally set to $exp(-\Delta/T)$ where T is a control parameter which corresponds to temperature in the analogy with physical annealing, and Δ is the change in the objective function value. The acceptance function implies that small increases in the objective function are more likely to be accepted than large increases. When T is high, most moves will be accepted, but as T approaches zero most uphill moves will be rejected. So, in SA, the algorithm starts with a relatively high temperature, to avoid being prematurely trapped at a local optimum. The algorithm attempts a certain number of moves at each temperature while the temperature parameter drops gradually.

To implement SA for a particular combinatorial optimization problem, a number of generic choices must to be made:
(i) the initial temperature value,
(ii) a temperature function, T, to determine how tile temperature is to be changed as the algorithm proceeds,
(iii) the number of iterations, L, to be performed at each temperature, and
(iv) a stopping criterion to terminate the algorithm.

SA has a few attractive features. First of all, it is simple to implement. Secondly, it can be applied to a wide range of problems. Thirdly, it can provide high quality solutions to many problems. On the other hand, there are also disadvantages associated with SA. One of them is the large CPU time often required in obtaining the final solution. Another is the lack of memory: The basic SA algorithm does not have a mechanism to prevent the procedure from repeating a solution evaluated previously.

Implementation of Simulated Annealing to P-median problem

A basic SA heuristic for p-median location problem has been proposed in Murray and Church (1996). The SA heuristic proposed in Chiyoshi and Galvao(2000) combines elements of the vertex substitution method of Teitz and Bart with the general methodology of simulated annealing. The cooling schedule adopted incorporates the notion of temperature adjustments rather than just temperature reductions. Computational results are given for OR-Library test instances. Optimal solutions were found for 26 of the 40 problems tested. Recently, an SA heuristic that uses the 1-interchange neighborhood structure has been proposed in Levanova and Loresh (2004). Results of good quality are reported on Kochetov data sets,and on the first 20 (among 40) OR-Library test instances. For example, 17 out of the 20 OR-Library instances are solved exactly.

With the concept of SA, which discussed in the previous section, the following heuristic is proposed to solve the p-median location problem:
Step 1: Initialization.

1.1 Select an initial solution s from S;
1.2 Select an initial temperature T_0;
1.3 Set temperature change counter $t = 0$;

Step 2 : Improving the solution.
2.1 Set repetition counter $n = 0$;
2.2 Repeat the following operations until $n = L$;
2.2.1 Generate, randomly, a solution $s^{'}$ from the neighborhood $N(s)$;
2.2.2 Calculate $\Delta = f(s^{'}) - f(s)$;
2.2.3 If $\Delta \leq 0$ then set $s := s^{'}$
else select a random number, X, from$U(0, 1)$, if $X < exp(-\Delta/T)$ then set $s := s^{'}$;
2.2.4 Set $n = n + 1$;
2.3 Set $t = t + 1$ and $T = R - \left(\frac{R*t}{L}\right)$;
2.4 Stop if the stopping condition is met or Go to Step 2.

Where L determines the finishing condition of the search. When L iterations are performed without decreasing the cost function f , then the search finishes.

Several means of choosing the initial solution were considered such as the greedy ascent (ADD) and descent (DROP) algorithms or random choice. However, it deserve noting that, after performing many experiments, the best results are obtained by the AS algorithm using the initial solution provided by the greedy descent algorithm.

As literature shows, the two key issues for properly tuning this heuristic search method are the number of iterations L and the temperature decreasing rate R . Although they are not shown here due to space limitations, we obtained the best results for SA heuristic with $L = 3000$ iterations. Performing more iterations increased the required execution times and it did not provide better values of the cost function f. Regarding to the rate of temperature decreasing, it did not have an effect on the required execution time of the algorithm. However, we obtained the best $f(s)$ values with R =1.25.

Computational Results

The proposed SA metaheuristic was studied experimentally using instances taken from OR-Lib (Beasley, 1990) for p-median problems. Further, we coded our proposed SA metaheuristic in C and tested it on PC with 2 GB of RAM and 4.1 GHz processor. The programme of our proposed heuristic was run 50 times for each instance and record the best solution and its CPU time. The solutions were compared among themselves as well as with the optimal solutions of the instances. The time of one run for the SA algorithm for each instance was short. However, the solutions obtained by the SA algorithm, for all instances, were optimal or close to the optimal solution.

The results are summarised in Table 1 where *file no.* represents the file number of p-median data in OR-Lib, N refers to the number of vertices in the network (or number of customers), P is the optimal number of facilities needed to locate, *opt solution* represents the optimal solution of the instance, *Best Solution* refers to the best solution found by the proposed SA heuristic, and *dev* (in %) is the gap between the optimal solution and our best solution, *Time* represents the CPU time (in seconds) elapsed by the proposed heuristic to find the best solution. The deviation is calculated as:

$$dev = \frac{F_{best} - F_{opt}}{F_{opt}} \times 100$$

where F_{best} is the best solution found by the proposed heuristic algorithm, and F_{opt} refers to the optimal solution of the instance.

Table 1 shows that 33 instances out of 40 have been solved optimally and the best solution of 6 (1) instances is one unit (two units) greater than the optimal solution. The worst case is instance number 34 (Pmed34) where the best solution found by the heuristic is 13 whereas the optimal solution is 11 with a deviation of 18.18% from the optimal solution and running time of 108.7 s. Moreover, it deserves noting that CPU time needed to find the optimal solution increased with P for the same N (number of vertices in the network). Also, we can see that the proposed SA heuristic solved the largest instance (Pmed40) optimally in less than one minute.

References

[1] Beasley J. E., 1985, A note on solving large P-median problems, *European Journal of Operational Research*, 21, 270-273.

[2] Beasley J. E., 1990, OR Library: Distributing Test Problems by Electronic Mail, *Journal of the Operational Research Society,* 41(11), 1069-1072.

[3] Cerny V., 1985, A Thermodinamical Approach to the Traveling Salesman Problem: An Efficient Simulated Algorithm, *J. Optimiz. Theory Appl.*, 45, 41-55.

[4] Chiyoshi F., and Galvao, D., 2000. A statistical analysis of simulated annealing applied to the p -median problem. *Annals of Operations Research,* 96, 61 –74.

[5] Cornuejols G., Fisher, M. L., and Nemhauser, G. L. (1977). Location of bank accounts to optimize float: An analysis study of exact and approximate algorithms, *Management Sciences*, 23, 789-799.

[6] Eglese R. W., 1990, Simulated Annealing: A Tool for Operational Research, *European Journal of Operational Research*, 46, 271-282.

[7] Pirlot M.,1992, General local search heuristics in combinatorial optimisation: a tutorial, *JORBEL-Belgian Journal of Operations Research, Statistics and Computer Science*, 32, 7-67.

[8] Pirlot M., 1996, General local search methods, *European Journal of Operational Research*, 92, 493-511.

[9] Hakimi S. L., 1964, Optimum locations of switching centers and the absolute centers and medians of a graph, *Operations Research*, 12, 450-459.

[10] Hakimi S. L., 1965, Optimum distribution of switching centers in a communications network and some related graph theoretic problems, *Operations Research*, 13, 462-475.

[11] Johnson D., Aragon C. , Mcgeoch L., and Schevon C. , 1989, Optimization by Simulated Annealing: An Experimental Evaluation: Part I, Graph Partitioning, *Operations Research*, 37, 865-892.

[12] Kariv O., and Hakimi S., 1979, An algorithmic approach to network location problems. Part 2: The P-medians, *SIAM J. Appl. Math.*, 37, 539-560.

[13] Kirkpatrick S., Gelatt C., and Vecchi M., 1983, Optimization by Simulated Annealing, *Science*, 220, 67 l-680.

[14] Levanova T., and Loresh M.A., 2004, Algorithms of ant system and simulated annealing for the p -median problem. *Automation and Remote Control* , 65, 431 –438.

[15] Marianov V., and Serra D. , 2002, Location problems in the public sector, In Drezner, Z., and Hamacher, H. W. (eds.), *Facility Location: Applications and Theory*, Springer, Berlin, pp 119-150.

[16] Metropolis N., Rosenbluth A. W., Teller A. H., and Teller E., 1953, Equation of state calculation by fast computing methods, *Journal of Chemical Physics*, 21, 1087-1091.

[17] Minieka E., 1970, The m-center problem, *SIAM Review*, 12, 138-139.

[18] Murray A.T., and Church R. L., 1996. Applying simulated annealing to planning-location models. *Journal of Heuristics,* 2, 31 –53.

[19] Narula S. C., Ogbu U. I. and Samuelsson, H. M. (1977). An algorithm for the P-median problem, *Operations Research*, 25, 709-713

[20] ReVelle C., and Swain, R.W. (1970). Central facilities location, *Geographical analysis*, 2, 30-42.

[21] Teitz M.B., and Bart P., 1968, Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16, 955-961.

Received: May 19, 2008

File No.	N	P	Optimal Solution	SA heuristic		
				Best Solution	dev. (%)	Time (s)
Pmed1	100	5	127	127	0	0.03
Pmed2	100	10	98	98	0	0.10
Pmed3	100	10	93	93	0	0.32
Pmed4	100	20	74	74	0	0.38
Pmed5	100	33	48	49	2.08	0.16
Pmed6	200	5	84	84	0	0.41
Pmed7	200	10	64	64	0	0.43
Pmed8	200	20	55	56	1.82	1.21
Pmed9	200	40	37	37	0	1.32
Pmed10	200	67	20	21	5.00	0.66
Pmed11	300	5	59	59	0	0.36
Pmed12	300	10	51	51	0	1.74
Pmed13	300	30	36	36	0	5.52
Pmed14	300	60	26	27	3.85	6.89
Pmed15	300	100	18	18	0	12.09
Pmed16	400	5	47	47	0	3.64
Pmed17	400	10	39	39	0	4.27
Pmed18	400	40	28	28	0	10.25
Pmed19	400	80	18	18	0	11.75
Pmed20	400	133	13	13	0	16.38
Pmed21	500	5	40	40	0	3.46
Pmed22	500	10	38	38	0	8.29
Pmed23	500	50	22	22	0	25.76
Pmed24	500	100	15	15	0	29.44
Pmed25	500	167	11	12	9.09	26.84
Pmed26	600	5	38	38	0	1.05
Pmed27	600	10	32	32	0	9.16
Pmed28	600	60	18	18	0	30.48
Pmed29	600	120	13	13	0	41.84
Pmed30	600	200	9	9	0	49.77
Pmed31	700	5	30	30	0	8.81
Pmed32	700	10	29	29	0	29.14
Pmed33	700	70	15	15	0	93.94
Pmed34	700	140	11	13	18.18	108.7
Pmed35	800	5	30	30	0	105.69
Pmed36	800	10	27	27	0	118.55
Pmed37	800	80	15	16	6.67	172
Pmed38	900	5	29	29	0	90.14
Pmed39	900	10	23	23	0	137.03
Pmed40	900	90	13	13	0	248.49

Table 1: Comparison of OR-Lib instances for the proposed SA heuristc and the optimal solution.