Yugoslav Journal of Operations Research
5 (1995), Number 2, 211-219

A DESCENT-ASCENT TECHNIQUE FOR SOLVING THE MULTI-SOURCE WEBER PROBLEM

Nenad MLADENOVIĆ
GERAD, HEC, University of Montreal,
Montreal, Quebec, Canada

Jack BRIMBERG
Department of Engineering Management, Royal Military College,
Kingston, Ontario, Canada

Abstract: The multi-source Weber problem requires locating m new facilities in continuous space in order to minimize a sum of transportation costs to n fixed points or customers with known demands. Several heuristic methods have been developed to solve this problem. Typically, these algorithms move in descent directions from a specified starting solution until a local minimum is reached. In this paper, we consider the original heuristic proposed by Cooper (1963, 1964, 1972), wich has no inherent neighbourhood structure. We show how a neighbourhood structure can be defined and a descent-ascent procedure employed to enhance the Cooper algorithm. Computational results are reported.

1. INTRODUCTION

The multi-source Weber problem, also referred to as the uncapacitated location-allocation problem, is one of the basic models in continuous location theory. The objective is to generate optimal sites in continuous space, notably R^2, for m new facilities in order to minimize a sum of transportation costs to a set of n fixed points or customers with known demands. Since capacity contraints are not imposed on the facilities, a customer will always be serviced by the facility which is nearest to it. The transportation cost associated with a customer is then given by the travel distance to its nearest facility multiplied by a weighting factor which is proportional to the customer's demand rate.

The model can be formulated as follows (e.g., see Love, Morris and Wesolowsky, 1988, (Chapter 7)):

$$\min_{W,X} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} d(a_i, x_j) \tag{1}$$

s.t.

$$\sum_{j=1}^{m} w_{ij} = w_i, \quad i = 1, \ldots, n$$

$$w_{ij} \geq 0, \quad \forall i, j$$

where:

- $a_i = (a_{i1}, a_{i2})$ is the known location of customer i, $i = 1,...,n$;
- $X = (x_1, \ldots, x_m)$ denotes the vector of location decision variables, with $x_j = (x_{j1}, x_{j2})$ being unknown location of facility j, $j = 1, \ldots, m$;
- w_j is the given total demand or flow required at customer i, $i = 1, \ldots, n$;
- $W = (w_{ij})$ denotes the vector of allocation decision variables, where w_{ij} gives the flow to costumer i from facility j, $i = 1,...,n$, $j = 1,...,m$;
- $d(x,y)$ is a function which measures the distance between any two points $x,y \in R^2$.

The objective function in (1) represents the total transportation costs in the system, while the constrain set ensures that the demands are satisfied at all the fixed points.

The main difficulty in solving (1) arises from the fact that the objective function is neither convex nor concave in all decision variables, and may contain several local optima. As an example, consider the research of Eilon et al. (1971) who used a descent algorithm by Cooper (1963, 1964, 1972) and 200 randomly-generated starting solution to obtain 61 local minima in a single problem with $m = 5$ and $n=50$. Furthemore, the worst solution deviated from the best one found by 40,9%!

Due to exponentialy - increasing computational times, global optimization methods for solving the location-allocation model are restricted to relatively small problem-sizes; e.g., see the branch-and-bound procedures of Kuenne and Soland (1972), and the set-reduction and p-median algorithm of Love and Morris (1975) for rectangular distances. Thus, heuristic procedures are required to obtain solutions to real problems where the customers are numbered in the hundreds or more. The first heuristic for solving (1) was proposed by Cooper (1963, 1964, 1972). Subsequently, Love and Juel (1982) developed five heuristics which are based on the observation that the dual formulation is a concave minimization problem. All these methods are descent-type algorithms, whereby an iteration is permitted to a nearby solution only if this solution is better than the current one. Thus, for each of these methods, the danger exists of being trapped at a local minimum.

The heuristics of Love an Juel (1982) have a well-defined neighbourhood structure. These algorithms consider up to two exchanges of customer allocations

from one facility to another. In the dual formulation, this is equivalent to testing adjacent corner points in a radius up to two moves away from the current corner point. Recently, Brimberg and Mladenović (1994) demonstrated that Tabu search rules can be applied to the heuristics of Love and Juel using the same neighbourhood structure. (See Glover (1989, 1990) and Glover and Laguna (1993) for a general introduction to Tabu search). The main idea is to allow moves in ascent directions when further descent is no longer possible; i.e., a local minimum has been reached. In Brimberg and Mladenović, the iterations always proceed to the best neighbouring solution even if this solution is inferior to the current one. Thus, downward moves are made in steepest descent fashion while upward moves take mildest ascent directions. The motivation is to move in an opposite direction from the one in which the local minimum was entered. (For further discussion, see Hansen (1986) and Hansen and Jaumard (1990).) A Tabu list is also constructed to avoid cycling back to a previous local minimum.

Brimberg and Mladenović (1994) is a first application of Tabu search in a continuous setting. Furthemore, the combinatorial nature of the customer allocations has a complexity of order m^n $(n > m)$, while the standard problems in the literataure which use Tabu search treat lower order complexities such as $n!$ or n^m . The objective of this paper is to pursue further of application od descent-ascent procedures in a continuous setting. This time we examine the earlier algorithm by Cooper (1963, 1964, 1972), which has no inherent neighbourhood structure. We show that by first defining an appropriate neighbourhood structure, this type of algorithm can also be augmented by descent-ascent rules to provide a systematic approach for examining different regions of the solution space. Computational results are provided to compare our structured method with a multi-start version of Cooper 's algorithm.

2. COOPER'S ALGORITHM WITH NEIGHBOURHOOD SEARCH

We begin by reviewing the iterative procedure proposed by Cooper (1963, 1964, 1972) to solve the location-allocation problem. Since the iterations consist of alternately locating the facilities and then allocating demands to them, this method will be referred to as the *alternating* algorithm or ALT for short.

ALT Algorithm

Step 1: Choose initial locations for the m facilities. These locations can be specified at random, or using the judgement of the decision-maker. (Typically the distance function $d(x,y)$ is a norm, and these locations are restricted to the convex hull of the fixed points (Hansen, Perreur and Thisse (1980).) Allocate each customer to the nearest facility, arbitrarily breaking any ties.

Step 2: For the current set of allocations, optimaly locate the facilities by solving m independent single facility minisum problems (e.g., use a generalized Weiszfeld procedure when $d(x,y)$ is an l_p norm (Brimberg and Love, 1993)).

Step 3: Re-allocate each customer to its nearest facility for the new set of facility locations found in step 2. If the new allocations are identical to those in the preceding iteration, stop. (A local minimum has been found.) Otherwise, return to step 2.

The tendancy of ALT to be trapped in a local minimum can be illustrated by a simple numerical example. Consider three points in the plane, a_1=(0,0), a_2 = (0,1) and a_3=(2,0), with weights w_1=2, w_2=w_3=1, and Euclidean travel distances. Suppose that two facilities (F1, F2) are to be located to service these demands, and the initial sites chosen in step 1 of ALT are given by $x_1^0 = (1,0)$, $x_2^0 = (1,\frac{1}{2})$, where the superscript denotes the iteration number. Allocation customers to the nearest facility results in a_1 and a_3 belonging to F1 and a_2 to F2. In step 2, the facilities are re-located to $x_1^1 = (0,0)$ and $x_2^1 = (0,1)$. The new allocations in step 3 of ALT are found to be the same as in the previous iteration. Thus, the algorithm is terminated with an objective function value of $w_3 d(a_3, x_1) = 2$. The globaly optimal solution has $x_1 = (0,0)$, $x_2 = (2,0)$, and an objective function value of $w_2 d(a_2, x_1) = 1$. We see that ALT can get trapped in a local ninimum even for very small examples.

This weakness in Cooper's algorithm is well-known (e.g., see Eilon et. al. (1971)). To improve the chances of finding the best solution, or at least a reasonably good one, a multi-start version of ALT can be employed. All we do is repeat the algorithm for several randomly-generated starting locations of the facilities. The aim is to explore different regions of the solution space by using different starting solutions, and hence, to obtain a number of local minima throughout the solution space. The best one of these will be retained as the final solution. Thus, the multistart version, which will be reffered to as MALT for brevity, combines the elements of random search with Cooper's algorithm.

An important feature of Cooper's algorithm is that there is no defined neighbourhood structure. In proceeding from one iteration of locating facilities and allocating demands to the next, the total number of exchanges of customer allocations from one facility to another will be variable, taking on in general any value from (0,1,.....,n-m). In addition to this variable neighbourhood size, the algorithm does not pursue a local search, but rather, moves directly to one point in the neighbourhood.

Some procedure is needed to force the algorithm to investigate different regions of the solution space. The random nature of the multi-start version provides a means of achieving this. Eventually, if we find a sufficient number of local minima from random starting points, the global solution should be among them. However, depending on the shape of the objective function, this can take a very, very long time. A more systematic approach is desired, and to this end we introduce a new algorithm which combines Cooper's local search with predefined neighbourhood structure. This will be given by the set of all possible points around the current local minimum obtained from a specified number of customer exchanges from one facility to another. The number of customer-to-facility exchanges is identified by parameter k_1 which can be set experimentally by the analyst. The neighbourhood constructed in this fashion is a generalization of the neighbourhoods defined in the H-heuristic of Love an Juel (1982), wich use one or two exchanges.

A specified number of points in the neighbouhood are chosen for further investigation. This number, denoted by the parameter k_2 , is again set by the analyst. The neighbourhood points may be obtained in a random manner, the aim being to mo-

ve in different directions from the current solution. A local minimum is then generated from each of these neighbourhood points using Cooper's algorithm, and the best one is retained as the new current solution, even if it is inferior to the predecessor. Thus, ascent directions are permitted. The next iteration proceeds in the same fashion from the new current solution. Since only a few points (k_2) in the neighbourhood are chosen randomly at each iteration, the likelihood of cycling is almost zero. Thus a Tabu list containing moves which are not allowed in the next iteration is not required. Algorithm is terminated after a specified number of iterations (k_3) or by some other stopping criterion.

The parameter k_1 must be set with great care. It should be large enough to ensure that each iteration moves to a new region in the solution space (i.e., new local minimum). On the other hand, if k_1 is too large, the systematic movement through the solution space is replaced by a random movement similar to the multi-start version (MALT). Our Cooper algorithm augmented by neighbourhood search rules is summarized below. This procedure will be called H+ALT to indicate that a neighbourhood structure similar to the one in the H-heuristic has been added to ALT.

H+ALT Algorithm

Step 1: Run ALT to obtain a local minimum. Label this solution as the current solution (CS). Set counter=1.

Step 2: Choose k_2 points in the neighbourhood of CS. (This neighbourhood consists of all possible points obtained by k_1 customer-to-facility exchanges). For each of these k_2 points, repeat steps 2 and 3 of ALT until a local minimum is reached. Of the k_2 local minima thus obtained, retain the one with the lowest value of the objective function, and label it as the new current solution (CS).

Step 3: Augment counter by one. If counter $< k_3$, return to step 2; otherwise, stop. The final solution is the best from all the local optima obtained.

3. COMPUTATIONAL RESULTS

Two sets of experiments were conducted, the first one using the Ruspini data [17] for 75 fixed points in the plane, while the second set considered test problems randomly generated in a 100 x 100 square. In all cases, the weights w_i were given a value of unity, and Eucledean distances were used; that is,

$$d(x, y) = ((x_1 - y_1)^2 + (x_2 - y_2)^2)^{\frac{1}{2}}, \quad \forall x, y \in R^2,$$

where $x=(x_1,x_2)$, $y=(y_1, y_2)$.

For the Ruspini data, the number of changes of customer allocations was set $k_1 = 30$. We found that this large parameter setting was required in order to provide a

sufficiently large neighbourhood to leave some very deep local minima traps. The number of solutions generated at random around a current local minimum was set at k_2=4. The H+ALT algorithm was run with MALT for different values of m. These results are summarized in **Table 1**, where f_{opt} denotes the lowest value of the objective function found by either algorithm. The same stopping criterion was used for both of them; namely, 1000 calls to a Weiszfeld subroutine for solving the single facility minisum location (or Weber) problem. Thus, both algorithms had comparable computation times as shown in **Table 1**. We see that as the number of facilities to be located increases, the quality of the solution obtained by H+ALT becomes much better than MALT. For example, when m=32, a 42% improvement is obtained.

The random test problems varied in size from n=10 to 100 in multiples of 10. For each n, a different test problem was run for m=2,3,.....,[n/2], for a total of [n/2] - 1 trials. The computational results are summarized in **Table 2**. The score in column 3 indicates how many trials H+ALT gave a better result than MALT, and how many MALT was superior. For example, a score 9 : 5 means that H+ALT was better than MALT in 9 of the trials, while MALT was better than H+ALT in 5 of them. The maximum deviations in **Table 2** indicate the relative improvement of H+ALT over MALT. The '+' column gives the maximum % improvement of H+ALT, while '–' column gives the analogous result for MALT. In all cases the following formula is used:

$$deviation = \frac{f_{opt1} - f_{opt2}}{f_{opt1}},$$

where f_{opt1} and f_{opt2} are the values of the objective function for the best solutions obtained by MALT and H+ALT respectively.

Table 1: Results for Ruspini data, n = 75 (Computation times are for 1000 calls to a Weiszfeld algorithm)

Objective function f_{opt}			CPU Time (sec)		Improvement
m	H+ALT	MALT	H+ALT	MALT	%
2	2385.548	2385.548	4.723	4.763	0.000
3	1609.278	1609.278	3.274	4.269	0.000
4	854.620	854.620	2.672	3.974	0.000
5	772.391	772.391	3.123	4.003	0.000
6	708.014	708.406	3.591	4.133	0.055
7	641.876	661.469	3.866	4.259	2.962
8	591.057	595.779	4.270	4.398	0.793
9	544.512	561.776	4.674	4.579	3.073
10	504.604	552.990	5.184	4.800	8.750
11	476.901	518.918	5.402	5.031	8.097
12	442.898	540.159	5.805	5.276	18.006
13	421.888	510.957	6.238	5.530	17.432
14	402.364	503.669	6.378	5.691	20.113
15	391.549	464.031	6.607	5.994	15.620
16	371.487	464.463	7.102	6.167	20.018

Objective function f_{opt}			CPU Time (sec)		Improvement
m	H+ALT	MALT	H+ALT	MALT	%
17	356.321	461.782	7.301	6.378	22.838
18	347.258	449.114	7.550	6.627	22.679
19	333.155	426.391	7.684	6.861	21.866
20	319.996	443.338	7.944	7.036	27.821
21	307.795	438.304	8.096	7.264	29.776
22	308.957	401.540	8.360	7.552	23.057
23	299.046	430.822	8.288	7.686	30.587
24	280.311	427.454	8.449	7.924	34.423
25	280.073	384.715	8.542	8.169	27.200
26	273.151	405.434	8.721	8.261	32.627
27	249.972	383.874	8.708	8.575	34.882
28	251.206	403.712	9.064	8.835	37.776
29	257.929	377.949	9.142	8.990	31.756
30	242.001	383.088	9.141	9.139	36.829
31	223.904	390.618	9.252	9.404	42.680
32	217.057	375.123	9.187	9.523	42.137

Table 2: Results for random test problems

n	Total number of trials	Score	Maximum +	Deviation (%) −
10	4	0:0	0	0
20	9	2:2	1.41	8.24
30	14	7:1	8.78	6.62
40	19	9:5	14.04	4.44
50	24	13:5	12.27	3.98
60	29	16:6	20.15	8.65
70	34	21:7	21.86	3.20
80	39	25:10	20.50	16.93
90	44	28:12	24.87	12.12
100	49	32:15	28.02	20.05

We conclude from the results in **Table 2** that H+ALT significantly out-performed MALT in the set of random test problems which were run. It is interesting to note that H+ALT gave consistently better results for larger values of m. For example, when $n = 50$, H+ALT was always superior for $m \geq 14$; when $n = 100$, H+ALT was always superior for $m \geq 24$. It appears from these observations that the structured approach of H+ALT out-performs the random approach of MALT when the number of local minima is large.

4. DISCUSSION AND CONCLUSION

A new algorithm (H+ALT) is presented which combines the features of Cooper's descent algorithm (1963, 1964, 1972) with the neighbourhood structure used in the H-heuristics of Love and Juel (1982). In addition, neighbourhood search rules are incorporated which permit moves in ascent derections. The multi-start version of Cooper's algorithm (MALT) and the H-heuristic can in fact be viewed as special cases of our new algorithm, where the parameters defining the neighbourhood and the number of points investigated in this neighbourhood are set to specific values. Computational experiments comparing H+ALT with MALT confirm that the structured approach used by H+ALT to search for new local minima can produce substantially better results than the random approach of MALT, particulary when the objective function contains a large number of local minima.

Future research includes studies related to fine-tuning the parameters of the neighbourhood search. A comparison of other location-allocation heuristic with our new algorithm should be made.

REFERENCES

[1] Brimberg, J., and R.F. Love, "Global convergence of a generalized iterative procedure for the minisum location problem with l_p distances", *Operations Research* 41, (1993) pp. 1153-1163.

[2] Brimberg, J., and N. Mladenović, "Application of Tabu search in the multi source Weber problem". *GERAD report,* (1994) *G-94-39,* Montreal, Canada.

[3] Cooper, L., "Location-allocation problems", *Operations Research 11,* (1963) *pp.* 331-343.

[4] Cooper, L., "Heuristic methods for location-allocation problems", *SIAM Rewiew 6,* (1964) pp. 37-53.

[5] Cooper, L., "The transportation - location problem", *Operations research 20,* (1972) pp. 94-108.

[6] Eilon, S., C.D.T Watson - Gandy, and N.Christofides, *Distribution Management*. New York: Hafner, (1971)

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

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

[9] Glover, F. and M.Laguna, "Tabu Search, in. C. Reeves ed. ", *Modern heuristic techniques for combinatorial problems* (Chapter 3), Oxford, Blackwell, (1993)

[10] Hansen, P., "The steepest ascent mildest descent heuristic for combinatorial programming", Presented at the Congress on numerical methods in combinatorial optimization, Capri, Italy, (1986)

[11] Hansen, P., B. Jaumard, "Algorithms for the maximum satisfiability problem". *Computing 44,* (1990) pp. 279-303.

[12] Hansen, P., J. Perreur, and J.F.Thisse, "Location theory, dominance and convexity: some further results". *Operations Research 28,* (1980) pp. 1241-1250.

[13] Kuenne, R.E., and R.M.Soland, "Exact and approximate solution to the multisource Weber problem". *Mathematical Programming 3,* (1972) pp. 193-209.

[14] Love, R.F., and H.Juel, "Properties and solution methods for large location-allocation problems". *Journal of the Operational Research Society 33,* (1982) pp. 443-452.

[15] Love, R.F., and J.G. Morris, "A computational procedure for the exact solution of location-allocation problems with rectangular distances". *Naval Research Logistic Quaterly 22,* (1975) pp. 441-453.

[16] Love, R.F., J.G.Morris, and G.O.Wesolowsky, *Facilities Location: Models and Methods*. New York, North Holland, (1988)

[17] Ruspini, E.H., "Numerical methods for fuzzy clustering", *Information Sciences*, 2, (1970) pp. 319-350.