Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle. Locational Analysis

José Moreno
Casiano Rodríguez
Natividad Jiménez

Heuristic cluster algorithm for multiple facility location-allocation problem

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle. Locational Analysis, tome 25, nº 1 (1991), p. 97-107.

<http://www.numdam.org/item?id=RO_1991__25_1_97_0>

Recherche opérationnelle/Operations Research
(vol. 25, n° 1, 1991, p. 97 à 107)

HEURISTIC CLUSTER ALGORITHM FOR MULTIPLE FACILITY LOCATION-ALLOCATION PROBLEM (*)

by José Moreno [1], Casiano Rodríguez [1], Natividad Jiménez [1]

Abstract. – *The multiple facility location-allocation problem consists of finding the optimal set of location points to establish the facility centers at them and the allocation of every demand point to a facility center. The problem can be solved by determining an optimal partition of the demand point set and solving the corresponding single facility location problems. We propose a general method based on Cluster Analysis for obtaining a heuristic partition and provide the specific algorithms for the standard models. We compare these procedures experimentally with other known heuristics on different sized randomly generated instances of the p-median problem. Its low memory requirements, its efficiency and the high degree of optimality attained mean that is a method which is particularly suited for using with personal computers.*

Keywords : Location; Facilities; Optimization.

Résumé. – *Le problème d'affectation-localisation multiple de services consiste à trouver l'ensemble optimal de points de localisation (pour y établir les centres de service) et l'affectation de chaque point de demande à un centre de service. Le problème peut être résolu en déterminant une partition optimale de l'ensemble des points de demande et en résolvant ensuite les problèmes correspondants de localisation à un seul service. Nous proposons une méthode générale basée sur l'Analyse des Données pour obtenir une partition heuristique et donnons des algorithmes spécifiques pour les modèles standards. Nous comparons expérimentalement ces procédures avec d'autres heuristiques connues sur des exemples engendrés aléatoirement pour le problème de la p-médiane. Le petit nombre de mémoires nécessaires, son efficacité et le haut degré d'optimalité atteint signifie que la méthode est particulièrement adaptée à un usage sur micro-ordinateur.*

1. INTRODUCTION

Consider a set of demand points where customers may require a service, and a set of location points where facility centers can be established to provide the service. The multiple facility location-allocation problem consists of finding the optimal location for the facility centers and the allocation of every demand point to a center which serves it [*see* Love *et al.* (1988)].

(*) Received April 1990.
(1) Departamento de Estadística e Investigación Operativa, Facultad de Matemáticas. Universidad de la Laguna, 38203 La Laguna, Tenerife, Spain.

Recherche opérationnelle/Operations Research, 0399-0559/91/01 97 11/$ 3.10

To formalize the problem, let S be a metric space that includes both demand and location points. Let $U \subset S$ be a finite set of demand points and let $L \subset S$ be the set of possible location points. Solving the problem involves findings an optimal pair consisting of a set of location points $X \subset L$ and an allocation function $a: U \rightarrow X$. For every demand point $u \in U$, $a(u) \in X$ denotes the center that serves $u \in U$. A function of the locations and allocations chosen that depends on the cost of serving the demand from the corresponding facility centers, must be minimized.

Let $C(x, u)$ be the function that evaluates the cost of serving the demand point $u \in U$ from a facility center located at point $x \in L$. If the size of the set X is known in advance to be equal to p then the problem is called the p-facility location-allocation problem. The corresponding single facility location problem arises when $p=1$.

Given a solution of the p-facility problem, the sets demand points allocated to each facility center constitute a partition of the demand set into p subsets. Given every subset of this partition, the optimal location for the facility center serving the demand points in it is the solution of the single facility location with respect to this subset. Therefore, in most of the models, the p-facility problem can be solved by choosing an optimal partition of size p and solving p single facility problems.

The typical optimization criteria are: to minimize the total cost of serving every demand points, and to minimize the worst cost of serving a demand point. The corresponding optimal locations are called medians and centers.

Clusters Analysis [*see* Hartigan (1975)] is related to the partition of a large set of items into highly dissimilar clusters made up of similar items. We propose to use hierarchical ascending algorithms to obtain efficiently a good partition of the set of demand points that provides a heuristic solution to any multiple facility location problem. The success of this heuristic depends on the algorithm used for the solving of the single facility location problem and on the appropriate selection of the way of evaluating the similarity.

2. FORMALIZATION OF THE PROBLEM

Three kinds of location models are selected: discrete, network and planar models. For each model, a space S with a distance function $d(.,.)$ is considered. A weight function $w(.)$ on the demand points is used to evaluate the cost of serving the demand point from a location point. Typical transportation costs are proportional to the distance travelled and the weight of a demand point represents a rate of demand. Thus, the cost of serving the demand at u from the center at x is the weighted distance:

$$C(x, u) = d(x, u) \,.\, w(u), \text{ for every } x \in L \text{ and } u \in U.$$

In a discrete location model, S is a finite set of points and the distance is usually the euclidean distance. If the location model is a network, then S is the set of points on it and the distance is the shortest path length. In planar models, S is the whole plane. Two different distance functions are considered: the euclidean and the rectilinear (Manhattan) distances.

In all models, the demand set U is an arbitrary finite set of points in S and the possible location points are all the points in S. The standard optimization criteria are the total cost criterion and the worst cost criterion. We focus our attention here on the first one.

The total cost of selecting the set of location points X and the allocation function $a: U \to X$ is evaluated by:

$$\text{TOT}(X, a) = \sum_{u \in U} C(a(u), u). \tag{1}$$

The p-median problem is the p-facility problem with minimum total cost criterion. In particular, the 1-median problem, or simply the median problem of U in L is to minimize the total cost function on L:

$$\text{TOT}(x, U) = \sum_{u \in U} C(x, u), \qquad x \in L. \tag{2}$$

3. THE PARTITION ASSOCIATED WITH A SOLUTION

If a point x is the location chosen for a single facility center, then the total cost of serving from it all points in U is evaluated by:

$$\text{TOT}(x, U) = \sum_{u \in U} C(x, u) = \sum_{u \in U} d(x, u) \,.\, w(u), \qquad x \in L. \tag{3}$$

The median problem of the demand set U in the location set L is to minimize the total cost function $\text{TOT}(x, U)$ on L. Let $C_1(L, U)$ denote this minimum value. A point $m \in L$ is a median of U in L if:

$$\text{TOT}(m, U) = C_1(L, U) = \underset{x \in L}{\text{MIN}}\, \text{TOT}(x, U) = \underset{x \in L}{\text{MIN}} \sum_{u \in U} d(x, u) \,.\, w(u). \tag{4}$$

Let X be a set of location points where facility centers are established. The optimal allocation is always achieved by allocating every demand point to the nearest facility center. Then the total cost of a set X of location points is the total cost of the optimal allocation of the demand points to the location points in X:

$$\mathrm{TOT}(X, U)=\sum_{u\in U} C(a(u), u)=\sum_{u\in U} \mathrm{MIN}_{x\in X}\, d(x, u).w(u). \tag{5}$$

Let $\mathrm{TOT}_p(L, U)$ be the optimal total cost of a set of p location points of L. A set $M\subset L$ with size p, $|M|=p$, is a p-median of U in L if:

$$\mathrm{TOT}(M, U)=\mathrm{TOT}_p(L, U)=\mathrm{MIN}\{\mathrm{TOT}(X, U): X\subset L, |X|=p\}. \tag{6}$$

Given the locations X of the facility centers and the optimal allocations of all demand points, let $U(x)$ denote the set of demand points allocated to the facility center x, for every $x\in X$. Then the total cost of the set of location points X is:

$$\mathrm{TOT}(X, U)=\sum_{x\in X}\sum_{u\in U(x)} d(x, u).w(u)=\sum_{x\in X}\mathrm{TOT}(x, U(x)). \tag{7}$$

Therefore, if M is a p-median then each $m\in M$ must be a median with respect to the demand point set $U(m)$. So, the optimal solution of the p-median problem has total cost:

$$\mathrm{TOT}_p(L, U)=\mathrm{TOT}(M, U)=\sum_{m\in M}\mathrm{TOT}_1(L, U(m))=$$

$$=\mathrm{MIN}\left\{\sum_{i=1}^{p}\mathrm{TOT}_1(L, U_1):\{U_1, \ldots, U_p\}\in P_p(U)\right\} \tag{8}$$

where $P_p(U)$ is the set of partitions of U into p subsets.

Therefore, the problem can be solved by choosing the optimal partition of the set of demand points U into sets U_i $(i=1, \ldots, p)$, and solving the median problem of every demand set U_i in L. Let m_i be the median of U_i, then the p-median of U is $M=\{m_i;\ i=1, \ldots, p\}$ and every demand point $u\in U_i$ is allocated to the facility center at m_i.

4. HEURISTIC SEARCH FOR A PARTITION

Several greedy heuristics for the p-facility location-allocation problem can be applied to obtain a good partition of the demand set. The p facility centers are selected one by one. The usual greedy heuristic starts by locating the first facility center at the optimal location point for the single facility location problem with respect to the whole demand set. Given the locations of a set of facility centers, the new facility center is located at the point that minimizes the resulting cost [*see* Kuehn and Hamburger (1963)]. A partition is obtained by allocating every demand point to the nearest facility center.

Dyer and Frieze (1985) proposed a very simple greedy heuristic for searching for a partition of the set of demand points. First, take the demand point of largest weight and locate a facility center at this point. Given a set of facility centers, evaluate the cost of serving any demand point from these and locate a new facility center at the demand point with highest cost. A partition is also obtained by allocating every demand point to the nearest facility center.

A local search can improve a given partition for any p-facility location-allocation problem. The local search must find an allocation such that every facility center is the optimal solution of the single facility location problem of the demand points served by it. It involves two steps: (i) reallocating a demand point to another facility center, and (ii) finding a better location point for the facility center serving the demand points in a set of the partition. These steps are carried out, applying a suited strategy, until no improvement is obtained [*see* Hansen *et al.* (1983)].

Cooper (1964) proposed a local search that consists of solving the single facility location problem for each set of the partition. Then, find the demand points which are allocated to one facility center but are closer to another. Allocate each one of these demand points to its nearest facility center. Solve again the single facility location problems of the partition. Repeat these steps until there are no further demand points to be allocated to a different facility center. Only the problems with respect to the modified sets of the partition must be solved.

A good solution of a multiple facility location-allocation problem can be obtained applying clustering algorithms. To do this, the demand points are identified as the items and the sets of the demand points allocated to the same facility center as the clusters. The Algorithm uses a dissimilarity function in accordance with the location-allocation objective. Then, an efficient heuristic solution of the multiple facility location-allocation is found by solving the

p single facility location problems corresponding to each set of the final partition.

A hierarchical ascending clustering algorithm joins, in successive iterations, the two most similar clusters to form a new one. Usually, it starts with every item in a unitary cluster. However the algorithm could start with a lower initial number of clusters which is greater than the required number of centers. Any heuristic algorithm can be used to provide this initial partition.

An appropriate way to evaluate the dissimilarity between two clusters is to compute the increment of the cost function on the partition when these clusters are joined. The HACA heuristic (called HACA for Heuristic Algorithms from Cluster Analysis) consists of applying a hierarchical ascending classification algorithm that uses this dissimilarity function.

We propose to apply this heuristic to any multiple facility location-allocation problem by choosing: (*a*) a *greedy* heuristic to obtain the initial partition, (*b*) a heuristic function to *guess* the increment of the cost, and (*c*) a *local search* to improve the final partition.

5. THE HACA ALGORITHM FOR THE p-FACILITY PROBLEM

The procedure starts taking k_0 initial clusters, where k_0 is chosen between p and $|U|$, and successively decreasing the number of clusters until value p is reached. At any iteration of the algorithm there are k clusters; each cluster i has an associated set of demand points U_i and a facility center x_i to serve them with cost $c_i = C(x_i, U_i)$. The p sets of demand points in each cluster constitute a partition of the set U.

The clusters i and j, with minimum dissimilarity between them, are joined in a new cluster with demand set $U_i \cup U_j$ and its facility center x_{ij} is chosen to be the median of $U_i \cup U_j$. Clusters i and j are substituted by the new one with the corresponding cost. After $k_0 - p$ iterations, there are p clusters with the corresponding demand sets U_i, centers x_i and costs c_i. These p facility centers constitute a heuristic solution to the location-allocation problem.

The HACA algorithm uses four subroutines:

– SING (U) returns the median of the demand points in set U.

– COST (x, U) returns the cost of serving the demand points in set U from a center located at x.

– DISS (i, j) returns the dissimilarity between clusters i and j.

– MIN(D) returns indices i and j (with $i<j$) of the minimum entry of matrix D.

HACA Algorithm

1. Initialization.
   1.1. Apply the DYER-FRIEZE greedy heuristic.
   1.2. Obtain the sets of initial partition U_i, $i=1, \ldots, K_0$.
   1.3. Do: $x_i \leftarrow \text{SING}(U_i)$, for $i=1, \ldots, k_0$.
   1.4. Do: $c_i \leftarrow \text{COST}(x_i, U_i)$, for $i=1, \ldots, k_0$.
   1.5. Do: $D_{ij} \leftarrow \text{DISS}(i, j)$, for $i, j=1, \ldots, k_0$.
2. Iterations. For k going from k_0 down to $p+1$ do:
   2.1. Do: $(i, j) \leftarrow \text{MIN}(D)$.
   2.2. Do: $U_1 \leftarrow U_i \cup U_j$ and $U_j \leftarrow U_k$.
   2.3. Do: $x_i \leftarrow \text{SING}(U_i)$ and $x_j \leftarrow x_k$.
   2.4. Do: $c_i \leftarrow \text{COST}(x_i, \text{U}_i)$ and $c_j \leftarrow c_k$.
   2.5. Do: $D_{is} \leftarrow \text{DISS}(i, s)$ and $D_{js} \leftarrow D_{ks}$, $s=1, \ldots, k-1$.
3. Termination.
   3.1. Apply the COOPER local Search.

The p-median problem in a network can be solved by obtaining the discrete p-median in the vertex set [*see* Hakimi (1964)]. Thus, there are two kinds of model for the p-median problem: (*a*) the discrete model which includes the p-median problem in the network, and (*b*) the continuous model which is the p-median problem on the plane.

It is possible to apply different procedures, heuristic or exact, for the four subroutines. We have selected the following:

SING is performed by applying an exact algorithm:

(*a*) Discrete model: An exhaustive search.

(*b*) Continuous model: The Weiszfeld algorithm.

COST is computed by the corresponding formula, although the following should be noted:

(*a*) Discrete model: The distances are stored in a matrix.

(*b*) Continuous model: The distances are computed when required.

DISS is computed in each case by the following *ad hoc* heuristic:

(*a*) Discrete model: $\text{DISS}(i, j)=\text{COST}(x_i, U_j)+\text{COST}(x_j, U_i)-c_i-c_j$.

(*b*) Continuous model: let x be the weighted average of the demand points in $U_i \cup U_j$. Then $\text{DISS}(i, j)=\text{COST}(x, U_i \cup U_j)-c_i-c_j$.

MIN is executed by comparing all the entries. It could be improved by using one of the efficient data structures designed to preserve the arrangement of a set of numbers (*i. e.* Balanced Binary Trees or Heaps). In this case, insertions and deletions must be performed repeatedly. However this is not crucial for the efficiency of this procedure.

We set $k_0 = 2p$ since our computational experiences show that taking k_0 greater than $2p$ does not give significantly better solutions.

6. THE EFFICIENCY OF HACA

The parameters that determine the size of the problems are n, the number of demand points, and p, the number of facility centers.

To initiate the HACA procedure, the greedy heuristic of Dyer and Frieze is performed to provide the initial partition. After HACA provides a partition into p sets, the local search of Cooper is performed to improve the solution.

TABLE I

Continuous p-*median with Manhattan Distance.*

p=5	GREEDY			RANDOM				HACA		
n	COST	TT	TS	MIN	COST	AT	ST	COST	TT	TS
100	490	10.88	10.38	578	606	2.19	1.82	468	4.78	1.8
200	1094	2.97	2.09	1205	1359	6.87	6.22	1079	10.05	2.7
300	1637	5.60	4.31	1932	2046	17.46	16.49	1509	16.48	5.5
400	2323	11.65	9.84	2366	2861	17.68	16.23	2341	21.32	6.8
500	2965	17.36	15.00	2945	3337	43.45	41.65	2973	33.02	14.7
600	3318	9.78	7.05	3931	4153	101.70	99.44	2999	34.23	12.3
700	3896	18.68	15.43	4668	4861	108.04	105.36	3546	40.05	14.5
800	4460	25.93	22.24	5015	5251	72.40	69.50	4626	112.14	82.8
900	5008	44.07	39.87	5439	6712	135.91	133.44	5085	247.03	213.2
1000	5957	90.55	85.80	6682	7075	112.16	108.77	5388	458.68	421.5

p=10	GREEDY			RANDOM				HACA		
n	COST	TT	TS	MIN	COST	AT	ST	COST	TT	TS
100	292	2.58	1.65	397	419	2.43	1.80	285	9.23	1.26
200	645	3.41	1.60	906	1009	10.18	9.06	632	20.60	4.45
300	1060	5.44	2.80	1388	1563	10.86	9.14	1062	28.57	4.29
400	1600	7.64	4.24	1961	2140	30.38	28.11	1402	37.97	6.21
500	2046	21.43	16.34	2412	2530	31.46	28.38	1818	50.27	10.66
600	2262	13.74	8.60	2808	2977	34.41	30.53	2340	76.15	28.08
700	2686	17.64	11.68	3149	3513	38.92	34.63	2601	96.48	40.27
800	3113	50.77	44.42	3495	4036	50.80	45.65	3003	81.26	17.75
900	3502	52.31	44.93	4348	5314	110.08	105.62	3344	97.64	52.31
1000	3824	93.35	85.60	4488	5062	107.77	101.68	4091	146.98	65.38

TABLE II

Continuous p-median with Euclidean Distance.

p=5	GREEDY			RANDOM				HACA		
n	COST	TT	TS	MIN	COST	AT	ST	COST	TT	TS
100	709.5	80	71	704.8	725.5	78	72.1	707.8	66	55
200	*1532.3*	*148*	*137*	*1528.3*	*1553.0*	*322*	*303.8*	*1546.9*	*97*	*80*
300	2497.4	405	381	2449.3	2456.7	566	541.5	2449.3	436	397
400	3299.8	876	835	3301.0	3308.8	1274	1230.0	3368.6	1587	1502
500	4185.9	912	876	4174.4	4174.9	1976	1917.4	4215.7	2077	1980
600	5057.6	1700	1644	4993.9	5020.9	1808	1628.8	5048.1	1728	1570
700	5829.8	1828	1772	5829.8	5891.6	2706	2649.7	5829.8	2978	2855

p=10	GREEDY			RANDOM				HACA		
n	COST	TT	TS	MIN	COST	AT	ST	COST	TT	TS
100	421.2	27	19	423.2	430.7	48	39	408.9	40	23
200	999.2	86	66	1002.6	1022.6	219	185	973.2	109	71
300	1698.1	381	321	1668.6	1675.2	407	354	1648.3	373	292
400	2273.7	420	357	2289.6	2340.2	794	708	2269.0	549	441
500	2874.6	482	418	2866.0	2906.0	1513	1368	2871.1	984	835
600	3480.4	1490	1336					3464.6	747	619
700	4146.4	2061	1871	4067.9	4112.7	2188	2020	4102.0	857	717
800	4812.3	2026	1854	4750.5	4825.3	2612	2451	4764.7	1989	1762
900	5394.1	3899	3605	5336.1	5371.8	3964	3718	5343.7	3175	2868

In the HACA procedure the number of times that every subroutine is executed are: SING subroutine is executed $2p$ times, DISS and COST subroutines are both used $\theta(p^2)$ times and MIN subroutine is executed p times.

The greatest number of times is for DISS and COST subroutines. COST subroutine takes $\theta(n)$ time. To evaluate the dissimilarity between two clusters, a single facility problem must be solved. However, the use of an exact procedure involves spending a lot of time. This, of course, takes $\Omega(n)$ time, *i. e.* the time is greater than any linear function of n.

Therefore we decided to look for an *ad-hoc* way to guess the dissimilarity also in $O(n)$ time. The procedures described above to perform the subroutine DISS are $\theta(n)$ in time. Then the total time taken by subroutines DISS and COST is $\theta(p^2 n)$.

The time needed to execute SING is $\Omega(n)$. Therefore, the greatest computational time in HACA is taken up by this subroutine. Thus any research aimed at improving the efficiency of the procedure must be concentrated on the algorithm used to find the optimal single location of a demand set.

Another major question on the efficiency of these procedures is the size of the memory used. If the matrix with the distance between the demand points

TABLE III

The discrete p-*median euclidean problem.*

p=5	GREEDY			RANDOM				HACA		
n	COST	TT	TS	MIN	COST	AT	ST	COST	TT	TS
100	726.8	3.9	2.9	708.8	751.0	4.7	3.7	712.4	6.1	3.5
200	1554.4	15.9	13.7	1551.6	1581.4	12.4	10.8	1538.8	17.5	12.3
300	2504.8	35.1	31.5	2508.6	2539.2	29.9	27.4	2467.1	46.4	37.3
400	3315.7	60.1	55.2	3315.6	3375.6	83.8	77.9	3337.5	69.8	58.3
500	4196.0	93.8	87.5	4210.9	4251.9	90.4	85.0	4194.1	148.3	129.8
600	5106.0	129.0	121.0	5017.4	5071.1	210.8	199.5	5106.0	150.0	131.7
700	5864.9	170.6	161.0	5860.2	5942.5	242.5	230.4	5860.2	287.0	257.4
800	6803.7	450.0	142.3	6776.2	6845.4	339.5	324.1	6779.8	375.7	337.1
900	7781.2	281.3	268.1	7638.5	7690.6	359.5	344.5	7614.8	600.4	558.6
1000	8476.3	345.9	331.0	8438.6	8549.4	493.9	474.2	8430.9	532.6	486.4

p=10	GREEDY			RANDOM				HACA		
n	COST	TT	TS	MIN	COST	AT	ST	COST	TT	TS
100	423.9	2.8	1.2	411.1	446.8	3.3	1.7	425.5	5.5	1.4
200	1011.5	9.4	5.8	1025.0	1065.6	12.3	8.3	1034.6	15.4	6.8
300	1743.5	18.3	12.8	1764.7	1781.1	27.7	21.0	1703.9	30.5	15.8
400	2283.5	31.1	27.7	2306.5	2386.8	40.2	31.7	2263.0	58.2	34.8
500	2897.3	55.2	44.1	2959.2	3065.7	73.0	60.6	2906.0	68.0	43.0
600	3545.2	64.5	53.0	3509.4	3628.2	98.4	83.5	3484.4	120.2	79.3
700	4216.4	118.0	99.7	4152.4	4244.6	183.6	157.9	4141.5	141.2	97.0
800	4825.9	192.7	165.3	4846.5	4908.3	216.4	188.7	4840.3	179.4	128.1
900	5489.7	131.7	113.5	5414.5	5544.0	254.1	224.0	5387.9	241.5	180.1
1000	5985.1	251.6	220.0	6076.4	6092.1	306.5	273.2	6085.5	350.8	263.0

is stored then the size of the memory used by HACA is $\theta(n^2)$. But if the distances are computed when required, HACA only needs $\theta(p^2)$ memory for the clusters and $\theta(n)$ memory for the demand points.

7. COMPUTATIONAL RESULTS

For all the instances of the problems used, n demand points were independently and uniformly generated using a random number generator within the square [0,10] × [0,10]. Their weights were also randomly generated in the interval [0,10]. The HACA heuristic procedure was compared with two other heuristic procedures called: the Greedy heuristic and the Random heuristic. The Greedy heuristic consists of the procedure proposed by Dyer and Frieze followed by the local search proposed by Cooper. The Random heuristic is a typical combination of the Montecarlo Method and a Local Search. This procedure consists of randomly generating a partition and improving it by carrying out the local search of Cooper.

Programs were coded in TURBO-PASCAL 5.0 and run on a personal computer based on the 80286 processor. The lack of memory (640 k) of MsDos compatible machines compelled us to consider suitable data structures to obtain efficiency in time and memory. Thus we were able to apply the HACA procedure to problem instances as large in size as $n = 1{,}000$ and it did not take an excessive amount of time to obtain a solution.

The run times and optimal values for the HACA, GREEDY and RANDOM heuristic algorithms applied to the instances are given in the tables. Run times shown are in seconds. The data in tables I and II are for the continuous problems, the first one using euclidean distance and the second one using rectangular distance. Tables III is for the discrete problem on the plane.

The columns of the tables contain the following data: Total Time employed by the heuristic (TT), Time taken to solve Single median problems (TS), and the cost provided by the heuristic (COST). The run time and cost shown for the RANDOM heuristic are average values in 5 repetitions. Moreover, the minimum cost reached is added in colum MIN.

REFERENCES

1. L. Cooper, Heuristic Methods for location-allocation problems, *S.I.A.M. Review*, 1964, *6*, pp. 37-52.
2. M. E. Dyer and A. M. Frieze, A Simple Heuristic for the *p*-Centre Problem, *Operations Research Letters*, 1985, *3*, N. 3, pp. 285-288.
3. S. L. Hakimi, Optimal Location of Switching Centres and the Absolute Centres and Medians of a Graph, *Oper. Res.*, 1964, *12*, pp. 450-459.
4. J. A. Hartigan, Clustering Algorithms, John Wiley, New York, 1975.
5. P. Hansen, D. Peeters and J. F. Thisse, Public Facility Location Models: A Selective Survey, in Thisse and Zoller, 1983.
6. A. P. Hurter and J. S. Martinich, Facility Location and the Theory of Production, Kluwer Acad. Pub., N.Y., 1989.
7. A. A. Kuehn and M. J. Hamburger, A Heuristic Program for Locating Warehouses, *Management Science*, 1963, *9*, pp. 643-666.
8. R. F. Love, J. G. Morris and G. O. Wesolowsky, Facilities Location. Models and Methods, North-Holland, Amsterdam, 1988.
9. J. Moreno, C. Rodríguez, N. Jimenez and R. Ramos, Computational Studies on Heuristics for the *p*-Center Problem. III Meeting of the E.W.G. on Location Analysis, 1988, Sevilla, Spain.
10. J. F. Thisse and H. G. Zoller eds., Location Analysis of Public Facilities, North-Holland, 1983.