Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 179 (2007) 927–939
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

The *p*-median problem: A survey of metaheuristic approaches

Nenad Mladenović [a,*], Jack Brimberg [b], Pierre Hansen [c], José A. Moreno-Pérez [d]

[a] *School of Mathematics, University of Birmingham, United Kingdom and GERAD*
[b] *Department of Business Administration, Royal Military College of Canada, Kingston, Ont., Canada and GERAD*
[c] *GERAD and HEC Montreal, University of Montreal, Canada*
[d] *University of La Laguna, Tenerife, Spain*

Received 15 November 2004; accepted 15 May 2005
Available online 13 June 2006

Abstract

The *p*-median problem is one of the basic models in discrete location theory. As with most location problems, it is classified as NP-hard, and so, heuristic methods are usually used to solve it. Metaheuristics are frameworks for building heuristics. In this survey, we examine the *p*-median, with the aim of providing an overview on advances in solving it using recent procedures based on metaheuristic rules.

Keywords: Metaheuristics; Location; *p*-Median

1. Introduction

Let us consider a combinatorial or global optimization problem

$$\min\{f(x) \mid x \in X\}, \tag{1}$$

where $f(x)$ is the *objective function* to be minimized and X the set of *feasible solutions*. A solution $x^* \in X$ is *optimal* if

$$f(x^*) \leqslant f(x), \quad \forall x \in X. \tag{2}$$

An *exact algorithm* for problem (1), if one exists, finds an optimal solution x^*, together with the proof of its optimality, or shows that there is no feasible solution ($X = \emptyset$), or the problem is ill-defined (solution is unbounded). On the other hand, a *heuristic algorithm* for (1) finds quickly a solution x' that is "near" to being optimal. The metaheuristics are general strategies to design heuristic algorithms.

Location analysis is a field of Operational Research that includes a rich collection of mathematical models. Roughly speaking, a problem is classified to belong to the location field if some decision regarding the position of new facilities has to be made. In general, the objective or goal of the location problem is related with the distance between new facilities and other elements of the space where they have to be positioned. Location models may be divided into three groups: *continuous* ($X \subseteq R^q$),

* Corresponding author. Address: School of Mathematics, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom. Tel.: +44 121 414 6198; fax: +44 121 414 3389.

E-mail addresses: N.Mladenovic@bham.ac.uk, nenad.mladenovic@brunel.ac.uk (N. Mladenović), Jack.Brimberg@rmc.ca (J. Brimberg), Pierre.Hansen@gerad.ca (P. Hansen), jamoreno@ull.es (J.A. Moreno-Pérez).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2005.05.034

discrete (X is finite) and *network* models (X is a finite union of linear, continuous sets). Another possible classification is as a *median* (minisum) or *center* (minimax) problem, depending on the nature of the objective function considered. Location models are also deterministic or stochastic, linear or nonlinear, single or multi criteria, and so on. See several survey articles and books (Love et al., 1988; Brandeau and Chiu, 1989; Mirchandani and Francis, 1990; Drezner, 1995; Daskin, 1995; Drezner and Hamacher, 2002, etc.). Moreover, several special issues of journals have been devoted to locational analysis [e.g., more recently, *Annals of Operations Research*, vol. 111 (2002); *Computers and Operations Research*, vol. 29 (2002)]. Also, the main topic of two journals [*Location Theory* (1993–1997) and *Studies in Locational Analysis*] deals exclusively with location problems.

Numerous instances of location problems, arising in Operational Research and other fields, have proven too large for an exact solution to be found in reasonable time. It is well-known from complexity theory (Garey and Johnson, 1978; Papadimitriou, 1994; Yannakakis, 1997) that thousands of problems are *NP-hard*, that no algorithm with a number of steps polynomial in the size of the instance is known, and that finding one for any such problem would entail obtaining one for any and all of them. Moreover, in some cases where a problem admits a polynomial algorithm, the power of this polynomial may be so large that instances of realistic size cannot be solved in reasonable time in the worst case, and sometimes also in the average case or most of the time.

So one is often forced to resort to *heuristics*, that are capable of yielding quickly an approximate solution, or sometimes an optimal solution but without proof of its optimality. Some of these heuristics have a worst-case guarantee, i.e., the solution x_h obtained satisfies

$$\left|\frac{f(x_h)-f(x)}{f(x_h)}\right| \leqslant \varepsilon, \quad \forall x \in X, \tag{3}$$

for some ε, which is, however, rarely small. Moreover, this ε is usually much larger than the error observed in practice and may therefore be a bad guide in selecting a heuristic. In addition to avoiding excessive computing time, heuristics address another problem, that of local optima. A local optimum x_L of (1) has the property that

$$f(x_L) \leqslant f(x), \quad \forall x \in N(x_L) \cap X, \tag{4}$$

where $N(x_L)$ denotes a neighborhood of x_L. (Ways to define such a neighborhood will be discussed below.) If there are many local minima, the range of values they span may be large. Moreover, the globally optimum value $f(x^*)$ may differ substantially from the average value of a local minimum, or even from the best such value among many, obtained by some simple heuristic (a phenomenon called by Baum (1986), the central-limit catastrophe). There are, however, many ways to get out of local optima, or, more precisely, the valleys, which contain them (or set of solutions followed by the descent method under consideration towards the local solution).

In the last decade, general heuristic methods, usually called *metaheuristics*, have engendered a lot of success in OR practice. Metaheuristics provide a general framework to build heuristics for combinatorial and global optimization problems. They have been the subject of intensive research since Kirkpatrick et al. (1983) proposed *Simulated Annealing* as a general scheme for building heuristics able to escape the local optimum "trap". Several other metaheuristics were soon proposed. For a discussion of the best-known among them the reader is referred to the books edited by Reeves (1993) and Glover and Kochenberger (2003). Some of the many successful applications of metaheuristics are also mentioned there.

In this survey, we give an overview of heuristic methods with emphasis on recent results of metaheuristic approaches used to solve one of the basic discrete facility location problems, the p-median problem (PMP). It is classified as NP-hard (Kariv and Hakimi, 1969). Significant advances in the state-of-the-art may be attributed to these newer methods.

2. Formulation

Consider a set L of m facilities (or location points), a set U of n users (or customers or demand points), and a $n \times m$ matrix D with the distances traveled (or costs incurred) d_{ij} for satisfying the demand of the user located at i from the facility located at j, for all $j \in L$ and $i \in U$. The objective is to minimize the sum of these distances or transportation costs

$$(\min) \quad \sum_{i\in U} \min_{j\in J} d_{ij},$$

where $J \subseteq L$ and $|J| = p$. PMP can be defined as a purely mathematical problem: given an $n \times m$

matrix D, select p columns of D in order that the sum of minimum coefficients in each line within these columns be the smallest possible.

The p-median problem and its extensions are useful to model many real world situations, such as the location of industrial plants, warehouses and public facilities (see, for example, Christofides, 1975, for a list of applications). PMP can also be interpreted in terms of cluster analysis; locations of users are then replaced by points in an m-dimensional space (see Hansen and Jaumard, 1997, for a survey of cluster analysis from a mathematical programming viewpoint). It may thus offer a powerful tool for data mining applications (Ng and Han, 1994).

Beside this combinatorial formulation, the PMP has also an integer programming one. Let us define two sets of decision variables: (i) $y_j = 1$, if a facility is opened in $j \in L$, and 0, otherwise; (ii) $x_{ij} = 1$, if customer i is served from a facility located in $j \in L$, and 0, otherwise. Then the integer programming formulation is as follows:

$$\min \quad \sum_i \sum_j d_{ij} x_{ij} \tag{5}$$

subject to

$$\sum_j x_{ij} = 1, \quad \forall i, \tag{6}$$

$$x_{ij} \leqslant y_j, \quad \forall i, j, \tag{7}$$

$$\sum_j y_j = p, \tag{8}$$

$$x_{ij}, y_j \in \{0, 1\}. \tag{9}$$

Constraints (6) express that the demand of each user must be met. Constraints (7) prevent any user from being supplied from a site with an unopened facility. The total number of open facilities is set to p by constraint (8).

3. Test problems

Most often test instances used in comparing heuristics for PMP are:

(i) *OR-Library instances.* There are 40 ORLIB problems from Beasley (1985, 1990), where the set of facility sites is identical to the set of users. The problem parameters range from instances with $n = 100$ nodes and $p = 5$, 10, 20 and 33 up to instances with $n = 900$ and $p = 5$, 10, 90. All these test problems are solved exactly (Beasley, 1985), which makes them suitable for computational comparisons. OR-Library is available at the webpage.[1]

(ii) *TSP-Lib instances.* The larger problem instances are usually taken from the travelling salesman library, Reinelt (1991). They are available at the TSP-Lib webpage.[2]

(iii) *Rolland et al. instances.* Rolland et al. (1996) tested their heuristics with non Euclidean instances with up to 500 nodes and potential facilities. Distances between nodes are random numbers from some interval. This set is available (from the authors or from us) upon request.

(iv) *Alberta, Galvão, Koerkel, Daskin and Pizzolato instances.* Five different sets of older instances were recently collected and used in Alp et al. (2003). They are available at Erkut's webpage.[3]

(v) *Resende and Werneck instances.* A new class of instances for PMP is introduced recently in Resende and Werneck (2004). These instances are generated in the same way as those in the Rolland et al. set above: each instance is a square matrix in which each entry (i,j) represents the cost of assigning user i to facility j. Instances with 100, 250, 500 and 1000 users were tested, each with values of p ranging from 10 to $n/2$. This set is available from the authors upon request.

(vi) *Kochetov instances.* This is a set of small size test instances. The number of potential facilities and users ranges from $n = 100$ to $n = 144$, with p around 10. This collection is classified into four groups: (a) instances on perfect codes (PCodes); (b) instances on chessboards (Chess); (c) instances on finite projective planes (FPP); (d) instances with large duality gap (Gap-A, Gap-B, Gap-C). They are down-loadable at the webpage.[4] At the same site the codes of several solution methods are also provided: (a) exact branch and bound; (b) simulated annealing; (c) probabilistic TS (described below as well); (d) genetic algorithm.

4. Classical heuristics

Heuristics for solving PMP may be divided into two groups: (I) Classical heuristics and (II) Metaheuristics. Methods in each group may be further

[1] http://mscmga.ms.ic.ac.uk/info.html.

[2] http://www.iwr.uni-heidelberg.de/groups/compt/software/TSPLIB95.

[3] http://www.bus.ualberta.ca/eerkut/testproblems.

[4] http://www.math.nsc.ru/AP/benchmarks/P-median/p-med_eng.html.

Table 1
Classification of p-median heuristics (the general types are: Constructive heuristics (CH), Local search (LS), Mathematical programming (MP) and MetaHeuristics (MH))

Type	Heuristic	References
CH	Greedy	Kuehn and Hamburger (1963), Whitaker (1983)
	Stingy	Feldman et al. (1966), Moreno-Pérez et al. (1991), Salhi and Atkinson (1995)
	Dual ascent	Galvão (1980, 1993), Erlenkotter (1978), Captivo (1991)
	Composite	Moreno-Pérez et al. (1991), Captivo (1991), Pizzolato (1994), Salhi (1997)
LS	Alternate	Maranzana (1964)
	Interchange	Teitz and Bart (1968), Whitaker (1983), Densham and Rushton (1992), Hansen and Mladenović (1997), Resende and Werneck (2003), Kochetov et al. (2005)
MP	Dynamic programming	Hribar and Daskin (1997)
	Lagrangian relaxation	Cornuejols et al. (1977), Mulvey and Crowder (1979), Galvão (1980), Beasley (1993), Daskin (1995), Senne and Lorena (2000), Barahona and Anbil (2000), Beltran et al. (2004)
	Aggregation	Hillsman and Rhoda (1978), Goodchild (1979), Erkut and Bozkaya (1999), Casillas (1987), Current and Schilling (1987), Hodgson and Neuman (1993), Hodgson and Salhi (1998), Bowerman et al. (1999), Francis et al. (2000, 2003)
MH	Tabu search	Mladenović et al. (1995, 1996), Voss (1996), Rolland et al. (1996), Salhi (2002), Kochetov (2001), Goncharov and Kochetov (2002)
	Variable neighborhood search	Hansen and Mladenović (1997), Hansen et al. (2001), García-López et al. (2002), Crainic et al. (2004)
	Genetic search	Hosage and Goodchild (1986), Dibbie and Densham (1993), Moreno-Pérez et al. (1994), Estivill-Castro (1999), Alp et al. (2003), Chaudhry et al. (2003)
	Simulated annealing	Murray and Church (1996), Chiyoshi and Galvão (2000), Levanova and Loresh (2004)
	Heuristic concentration	Rosing et al. (1998), Rosing and ReVelle (1997), Rosing et al. (1999)
	Scatter search	García-López et al. (2003)
	Ant colony	Levanova and Loresh (2004)
	Neural networks	Domínguez Merino and Muñoz Pérez (2002), Domínguez Merino et al. (2003)
	Decomposition	Dai and Cheung (1997), Taillard (2003)
	Hybrids	Resende and Werneck (2004)

classified according to their similarities. In Table 1 we give one possible classification of both groups of methods for the PMP.

Classical heuristics for the p-median problem often cited in the literature may be divided into three groups (see Table 1): Constructive (CH); Local search (LS); those based on Mathematical programming (MP) formulations. Within these groups we have: (i) *Greedy*, (ii) *Stingy*, (iii) *Dual ascent*, (iv) *Composite* (v) *Alternate*, (vi) *Interchange*, (vii) *Dynamic programming* (DP); (viii) *Lagrangian relaxation* (RL), and (ix) *Aggregation* (AG) heuristics. The first four are constructive heuristics, while the next two need a feasible initial solution. The last three may be classified in the Mathematical programming group.

(i) *Greedy.* The Greedy heuristic (Kuehn and Hamburger, 1963) starts with an empty set of open facilities, and then the 1-median problem on L is solved and added to this set. Facilities are then added one by one until the number p is reached; each time the location which most reduces total cost is selected. An efficient implementation is given in Whitaker (1983).

(ii) *Stingy.* The Stingy heuristic (Feldman et al., 1966), also known as *Drop* or *Greedy-Drop*, starts with all m facilities opened, and then removes them one by one until the number of facilities has been reduced to p; each time the location which least increases total cost is selected. A modified implementation of the stingy heuristic is to start from a subset instead of the entire set of potential sites (Salhi and Atkinson, 1995).

(iii) *Dual ascent.* Another type of heuristic suggested in the literature is based on the relaxed dual of the integer programming formulation of PMP and uses the well-known *Dual ascent* heuristic DUALOC (Erlenkotter, 1978). Such heuristics for solving the p-median problem are proposed in Galvão (1980) and Captivo (1991).

(iv) *Composite heuristics.* Several hybrids of these heuristics have been suggested. For example, in the *GreedyG* heuristic (Captivo, 1991), in each step of *Greedy*, the *Alternate* procedure is run. A combination of *Alternate* and *Interchange* heuristics has been suggested in Pizzolato (1994). In Moreno-Pérez et al. (1991), a variant of *Stingy* (or *Greedy-Drop*) is compared with *Greedy* + *Alternate* and *Multistart Alternate*. In Salhi's (1997) perturbation heuristic, *Stingy* and *Greedy* are run one after another, each having a given number of steps. The search allows exploration of infeasible regions by oscillating around feasibility. The combination of *Greedy* and *Interchange*, where the *Greedy* solution is chosen as the initial one for *Interchange*, has been most often used for comparison with other newly proposed methods (see, for example, Voss, 1996; Hansen and Mladenović, 1997).

(v) *Alternate.* In the first iteration of Alternate (Maranzana, 1964), facilities are located at p points chosen in L, users assigned to the closest facility, and the 1-median problem solved for each facility's set of users. Then the procedure is iterated with these new locations of the facilities until no more changes in assignments occur. Since the iterations consist of alternately locating the facilities and then allocating users to them, this method will be referred to as the *alternating* heuristic. This heuristic may switch to an exhaustive exact method if all possible $\binom{m}{p}$ subsets of L are chosen as an initial solution. However, this is not usually the case since the complexity of the algorithm is then increased by an $O(m^p)$.

(vi) *Interchange.* The Interchange procedure (Teitz and Bart, 1968) is commonly used as a standard to compare with other methods. Here a certain pattern of p facilities is given initially; then, facilities are moved iteratively, one by one, to vacant sites with the objective of reducing total cost; this local search process is stopped when movement of any single facility fails to decrease the value of the objective function.

(vii) *Dynamic programming (DM).* A heuristic that uses a dynamic programming idea is suggested by Hribar and Daskin (1997). It may be viewed as reduced dynamic programming or as an extended greedy constructive method. Instead of considering only the best facility as in *Greedy*, the q best solutions are stored in each iteration (q is a parameter). The procedure stops when p facilities are reached, as in *Greedy*. This heuristic was tested using three small datasets of size $m = n = 49$, 55, and 88.

(viii) *Lagrangian heuristics (LH).* This heuristic procedure for solving PMP, originally proposed by Cornuejols et al. (1977), is based on the mathematical programming formulation (5)–(9). Different variants are suggested in Mulvey and Crowder (1979), Galvão (1980), and Beasley (1993). Usually, the constraint (6) is relaxed so that the Lagrangian problem becomes:

$$\max_u \min_{x,y} \left(\sum_i \sum_j d_{ij} x_{ij} + \sum_i u_i \left(1 - \sum_j x_{ij} \right) \right) = \max_u \min_{x,y} \left(\sum_i \sum_j (d_{ij} - u_i) x_{ij} + \sum_i u_i \right) \tag{10}$$

subject to (7)–(9). Note that the objective function (10) is minimized with respect to the original variables and is maximized with respect to the Lagrangian multipliers.

In Lagrangian heuristics the following steps are repeated iteratively after setting the initial values of the multipliers u_i: (1) solve the Lagrangian model, i.e., find the x_{ij} and y_i; (2) adjust the multipliers u_i. Thus, it may be seen as an "Alternate" type heuristic. The largest value of (10) (over all iterations) represents a lower bound of PMP. When the variables u_i are fixed, the resulting model (in step 1) is easy to solve (see, e.g., Daskin, 1995). The solution found may not be feasible, since the constraint (6) may be violated. However, feasibility is obtained by assigning the users to their closest open facility. The best of the feasible solutions found over all iterations would also give the best (lowest) upper bound. Therefore, Lagrangian heuristics provide both lower and upper bounds of the problem considered. The final most complex task is to modify the multipliers based on the solution just obtained. A common approach is by subgradient optimization. In Beasley (1993), at each subgradient iteration, Lagrangian solutions are made primal feasible and the reallocation improved by the classical Alternate heuristic. A faster variant, called the Lagrangian/surrogate heuristic has recently been proposed by Senne and Lorena (2000). We also refer the reader to the *Volume* subgradient approach introduced by Barahona and Anbil (2000). A semi-Lagrangian relaxation (SLR) method is suggested in Beltran et al. (2004). The idea is to get a better lower bound in the Lagrangian relaxation by treating the set of equality constraints in (6) and (8) twice: in the relaxation and in the set of constraints replacing relation "=" with "$\leqslant$". In theory SLR closes the integrality gap.

(ix) *Aggregation (AG).* In order to reduce the computational time, and sometimes keep the customer data confidential, a common practice is to reduce the number of demand points by demand data aggregation. The process of aggregation, however, results in the loss of locational information and induces errors to the problem solution. Thus, AG methods may be viewed as heuristic in nature.

There are three types of errors that occur when demand points are replaced by aggregation points: source A, B and C errors. Hillsman and Rhoda (1978) defined them for the first time. Their measurements assumed a uniform population density. Source A is defined separately and independently for each group of demand points. It arises due to the replacement of actual distances with their approximated values. Source B is a particular case of source A error that occurs when a facility point coincides with an aggregation point. Source C error occurs when demand points are wrongly allocated. In Plastria (2001), two types of *allocation* errors are considered in the continuous version of PMP, known as the multisource Weber problem.

Casillas (1987) is the first author to distinguish between cost and optimality errors. Current and Schilling (1987) present a method for removing source A and B aggregation errors if unaggregated data are available. Hodgson and Neuman (1993) concentrated on source C errors, outlining a complete enumeration method that can be used to eliminate this source, using GIS (geographical information systems). A faster variant of the method is suggested in Hodgson and Salhi (1998). They use a quadtree database structure to allocate groups of origins to destinations, basically aggregating when aggregation will not produce error, and disaggregating when it would.

Erkut and Bozkaya (1999) consider three types of perturbations in the p-median model: cost error, optimality error, and location error. More formally, the functions f and g, and the solutions (x, y) and (x_a, y_a) are defined as follows:

- f the original (unaggregated) p-median objective function (5);
- g aggregated p-median objective function;
- (x, y) the optimal solution to the original (unaggregated) PMP;
- (x_a, y_a) the optimal solution to the aggregated PMP (5)–(9).

With this notation, the three types of errors are as follows:

$$\text{Cost error} = \frac{f(x_a, y_a) - g(x_a, y_a)}{f(x_a, y_a)},$$

$$\text{Optimality error} = \frac{f(x_a, y_a) - f(x, y)}{f(x, y)},$$

$$\text{Location error} = \|(x, y) - (x_a, y_a)\|.$$

A more theoretical approach on aggregation error bounds for a class of location models is given in Francis et al. (2000). A survey of methods developed in the last decade, based on viewing the aggregation as a second-order location problem, and using error bounds, to do aggregation in such a way as to keep the error small, can be found in Francis et al. (2003).

Modern computers are able to store very large data sets. Furthermore, the complexity of most PMP methods depends more on the number of facilities (m) than on the number of users (n). Therefore, it is reasonable to question whether heuristics based on aggregation can be competitive with other approaches. In other words, do the efforts to aggregate the input data for PMP pay-off? Are we dealing with a problem that is in fact, from a computational point of view, not a problem at all?

5. Implementation of interchange local search

The Interchange method is one of the most often used classical heuristics either alone or as a subroutine of other more complex methods or within metaheuristics. Therefore, it would seem that an efficient implementation is extremely important. The formula of benefit (or profit) w_{ij} in applying an interchange move is

$$w_{ij} = \sum_{u:c_1(u)\neq j} \max\{0, [d_1(u) - d(u,i)]\} - \sum_{u:c_1(u)=j} [\min\{d_2(u), d(u,i)\} - d_1(u)], \tag{11}$$

where u, i and j are the indices of a user, and the ingoing and outgoing facilities, respectively; $c_1(u)$ represents the index of the closest facility of user u; $d_1(u) = d(u, c_1(u))$ and $d_2(u)$ represent distances from u to the closest and second closest facilities, respectively. The first sum in (11) accounts for users whose closest facility is not j. The second sum refers to users assigned to j in the current solution; since they lose their closest facility, they will be reassigned either to the new facility i or to their second closest, whichever is more advantageous.

An important study has been done by Whitaker (1983), who described the so-called *fast interchange* heuristic. This method was not widely used (possibly because of an error in that paper) until Hansen and Mladenović (1997) applied it as a subroutine of a variable neighborhood search (VNS) heuristic.

Among other results reported is that *Add* and *Interchange* moves have similar complexity. Moreover, p times fewer operations are spent for one *fast interchange* move as compared to one *interchange* move of Teitz and Bart (1968). In fact, the following three efficient ingredients are incorporated in the interchange heuristic in Whitaker (1983): (i) *move evaluation*, where a best removal of a facility is found when the facility to be added is known; (ii) *updating* of the first and the second closest facilities of each user; (iii) restricted *first improvement* strategy, where each facility is considered to be added only once. In the implementation of Whitaker's interchange algorithm by Hansen and Mladenović (1997), only (i) and (ii) are used; i.e., instead of (iii), a best improvement strategy is applied. Hence, the restricted first improvement strategy is removed as well. Moreover, the complexity of steps (i) and (ii) is evaluated.

Recently, a new efficient implementation has been suggested by Resende and Werneck (2003). Its worst case complexity is the same (O(mn)), but it can be significantly faster in practice. The formula (11) is replaced with

$$w_{ij} = \sum_{u\in U} \max\{0, d_1(u) - d(u,i)\} - \sum_{u:c_1(u)=j} [d_2(u) - d_1(u)] + e_{ij}.$$

The first sum represents gains by inserting facility i, the second losses by dropping facility j, while the last term is from a matrix $E = [e_{ij}]$ called *extra*, which contains mostly values of zero, and whose updating makes this implementation efficient for large problem instances:

$$e_{ij} = \sum_{u:c_1(u)=j; d(u,i)<d_2(u)} [d_2(u) - \max\{d(u,i), d_1(u)\}].$$

Therefore, the extra memory required for the matrix E allows for significant accelerations. Several variants have been considered: full matrix (FM) and sparse matrix (SM) representation of E; with preprocessing, i.e., ranking distances from each user to all potential facilities (FMP and SMP), and so on. For example, the average speedups obtained by SMP on OR-Library and TSP-Lib test instances were by factors of 8.7 and 177.6, respectively, if the running times for preprocessing were not included. If they were included, then SMP was 1.8 and 20.3 times faster, respectively, than the fast interchange. As expected, the greatest gains were observed on Euclidean instances, since a significant number of the e_{ij} are equal to 0 in this case.

Another step forward in solving PMP by interchange local search has recently been suggested in Kochetov et al. (2005), where a new neighborhood structure, called LK (Lin–Kernigham), has been proposed. A depth parameter k that counts the number of interchange moves within one step of local search is introduced. The LK(k) neighborhood can be described by the following steps: (a) find two facilities i_{add} and i_{drop} such that the best solution in the 1-interchange neighborhood is obtained (b) exchange them to get a new solution; (c) repeat steps (a) and (b) k times such that a facility to be inserted has not previously been dropped in steps (a) and (b). The set LK(k) is thus defined as

$$\mathrm{LK}(k) = \left\{(i^t_{\text{add}}, i^t_{\text{drop}}), t = 1, \ldots, k\right\}.$$

The best solution from LK(k) is the local minimum with respect to the LK neighborhood structure. This local search has successfully been used within Lagrangian relaxation (LR), random rounding (after linear relaxation) (RR), and within ant colony optimization (ACO) (Dorigo and Di Caro, 1999).

6. Metaheuristics

We briefly describe here some of the metaheuristic methods developed for solving the PMP. They include: (i) Tabu search (TS), (ii) Variable neighborhood search (VNS), (iii) Genetic search, (iv) Scatter search, (v) Simulated annealing, (vi) Heuristic concentration, (vii) Ant colony optimization, (viii) Neural Networks, (ix) Decomposition heuristics, (x) Hybrid heuristics.

(i) *Tabu search (TS).* Several tabu search (Glover, 1989, 1990) methods have been proposed for solving PMP (see also Glover and Laguna, 1997, for an introduction to tabu search). In Mladenović et al. (1995, 1996), the 1-interchange move is extended into a so-called *1-chain – substitution* move. Two tabu lists (TL) are used with given and random TL sizes. Another TS heuristic is suggested by Voss (1996), where a few variants of the so-called *reverse elimination* method are discussed. In Rolland et al. (1996), a 1-interchange move is divided into add and drop moves which do not necessarily follow each other and so feasibility is not necessarily maintained during the search; this approach, within TS, is known as *strategic oscillation* (see Glover and Laguna, 1993). The same restricted neighborhood structure is used in a more recent TS for solving PMP in Salhi (2002). After a drop move, the set of potential ingoing facilities is restricted to the K (a parameter) closest ones to the one just dropped. Moreover, the functional representation of the TL size and a flexible concept of the aspiration level are proposed. Although results reported do not improve significantly upon those obtained with a purely random TL size, this analysis gives possible directions in designing efficient TS heuristics. A simple probabilistic TS (PTS) is suggested by Kochetov (2001). Denote by $N(x)$ the 1-interchange neighborhood of any solution x (a set of open facilities). A restricted neighborhood $N_r(x) \subset N(x)$ (with a given probabilistic threshold $r < 1$) is obtained at random. The simple TS heuristic based on $N_r(x)$ does not use aspiration criteria, or intensification and diversification rules, but it allows the author to establish a connection with irreducible Markov chains and to develop asymptotic theoretical properties. For solving PMP by PTS, good results on Kochetov test instances (see above) are reported in Goncharov and Kochetov (2002).

(ii) *Variable neighborhood search* (VNS). There are several papers that use VNS (Mladenović and Hansen, 1997; Hansen and Mladenović, 2001a,b) for solving the PMP. In the first one (Hansen and Mladenović, 1997), a basic VNS is applied and extensive statistical analysis of various strategies performed. Neighborhood structures are defined by moving $1, 2, \ldots, k_{\max}$ facilities and correspond to sets of 0–1 vectors at Hamming distance $2, 4, \ldots, 2k_{\max}$ from x. In other words, if x_1 and x_2 denote two solutions (two sets of open facilities), the distance $\rho(x_1, x_2)$ between them is given as

$$\rho(x_1, x_2) = \frac{|x_1 \Delta x_2|}{2}, \tag{12}$$

where Δ is the symmetric difference operator. The descent heuristic used is 1–interchange, with the efficient fast interchange (FI) computational scheme described in Section 5. Results of a comparison of heuristics for OR-Library and some TSP-Lib problems are reported. In order to solve larger PMP instances, in Hansen et al. (2001), both reduced VNS and a decomposition variant of VNS (VNDS) are applied. Subproblems with increasing numbers of users (that are solved by VNS) are obtained by merging subsets of users (or market areas) associated with k ($k = 2, \ldots, p$) medians. Results on instances of 1400, 3038 and 5934 users from the TSP library show that VNDS improves notably upon VNS in less computing time, and gives much better results than FI, in the same time that FI takes for a single descent. Moreover, reduced VNS, which does not use a descent phase, gives results similar to those of FI in much less computing time.

Heuristics with Parallel VNS are found in García-López et al. (2002) and Crainic et al. (2004). The first of the three parallelization strategies analyzed in García-López et al. (2002) attempts to reduce computation time by parallelizing the local search in the sequential VNS. The second one implements an independent search strategy that runs an independent VNS procedure on each processor. The third one applies a synchronous cooperation mechanism through a classical master–slave approach. The *Cooperative* VNS parallelization proposed in Crainic et al. (2004) applies a cooperative multi-search method based on a central-memory mechanism.

(iii) *Genetic algorithm (GA)*. Several genetic search heuristics have been suggested. Hosage and Goodchild (1986) encoded a solution as a string of m binary digits (genes). In order to reach feasibility (p open facilities), the authors penalized the number of open facilities. The results reported are poor, even on small problems. In Dibbie and Densham (1993), each individual has exactly p genes, and each gene represents a facility index. This appears to be a better representation of the solution. The authors used conventional genetic operators: selection, cross-over and mutation. Reported results are similar to Interchange local search, but with considerably longer processing time. The size of the instances tested was $n = m = 150$ (user and facility sites coincide) and $p = 9$. Moreno-Pérez et al. (1994) designed a parallelized GA for the PMP. Each gene represents a facility index as well. Beside conventional GA operators, they used multiple population groups (colonies), which exchange candidate solutions with each other (via migrations).

Finally, in Alp et al. (2003), much better results are reported, but still not as good as those obtained by VNS, TS or hybrid approaches. It is even not clear if the suggested method belongs to the class of GA. The mutation operator is avoided, and the new members of the population are not generated in the usual way (i.e., by using selection and crossover operators). Two solutions are selected at random, and then the union of them taken, obtaining an infeasible solution with number of genes (facilities) larger than p. To reach feasibility, the Stingy or Greedy-Drop classical heuristic is applied. Better results would be obtained if the Interchange heuristic was applied after Stingy and the resulting method would then be similar to VNS. Results on OR-Library, Galvão, Alberta and Koerkel test instances are reported.

(iv) *Simulated annealing (SA)*. A basic SA heuristic for PMP has been proposed in Murray and Church (1996). The SA heuristic proposed in Chiyoshi and Galvão (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.

(v) *Heuristic concentration (HC)* method (Rosing and ReVelle, 1997) has two stages. In stage one, a set of solutions is obtained by repeating q times the Drop/Add heuristic, and then retaining the best m solutions found. The elements of desirable facility sites selected from the set of solutions (i.e., facilities that most often appeared in the m solutions) form a *concentration set*. Stage two of HC limits the set of potential facilities to this set and resolves the model. Such a restricted model can be solved heuristically or even exactly. An extension of HC, known as the *Gamma heuristic* (Rosing et al., 1999) includes a third stage as well. Testing is performed on 81 randomly generated instances with 100–300 nodes. The results in Rosing et al. (1998) compare successfully with the TS of Rolland et al. (1996).

(vi) *Scatter search (SS)* metaheuristic (Glover et al., 2000) is an evolutionary strategy based on a moderated set of good solutions (the *Reference Set*) that evolves mainly by combining its solutions to construct others exploiting the knowledge of the problem at hand. García-López et al. (2003) design a SS for the PMP by introducing a distance in the solution space. This VNS idea is used to control the diversification of the method. They consider the case where $U = L$. The distance between two solutions x_1 and x_2 is defined differently than in (12):

$$\eta(x_1, x_2) = \sum_{i \in x_1} \min_{j \in x_2} d_{ij} + \sum_{j \in x_2} \min_{i \in x_1} d_{ij}.$$

The reference set consists of k (a parameter) best solutions from the population and $r - k$ randomly chosen solutions following some diversification criteria (r denotes the reference set size). Solutions of a selected subset of the reference set are combined as follows: first, as in heuristic concentration, the set of facilities that appear in each solution of the subset is found; then to get the size p, new facilities are added iteratively according to predefined rules. The combined solutions are then improved by a local search based on interchanges. The resulting solution is incorporated in the reference set because it improves one of the k best solutions or because it improves the diversity of the set according to the distance between its solutions. Good results are reported on TSP-Lib instances. Three types of parallelization have been proposed in García-López et al. (2003) to achieve either an increase of efficiency or an increase of exploration. The procedures have been coded in C using OpenMP (1997) and compared in a shared memory machine with large instances.

(vii) *Ant colony optimization (ACO)* was first suggested in Dorigo et al. (1991) (see also Dorigo and Di Caro, 1999). The motivation for the method comes from nature. Ants deposit pheromone on the path while walking looking for food, and foragers follow the path with the stronger pheromone concentration with a higher probability. The shorter route between the nest and the food source will be used more often than the longer paths. So, all the ants will finally follow the shortest path since it receives the largest amount of pheromone. In mathematical terms, the idea of ACO is to use the statistical information obtained from previous iterations to guide the search into the more promising areas of the solution space. Usually the method contains several parameters, whose estimation and updating (as in SA) mostly influence the quality of the obtained solution.

In Levanova and Loresh (2004) and Kochetov et al. (2005), a randomized *stingy* or drop heuristic is used within ACO: initially a solution x is taken as the set of all potential facilities L; a facility j to be dropped is chosen at random (with probability r_j) from the restricted drop neighborhood set: $S_j(\lambda) = \{j \mid \Delta f_j \leqslant (1-\lambda)\min_\ell \Delta f_\ell + \lambda \max_\ell \Delta f_\ell\}$, for $\lambda \in (0,1)$ and $\Delta f_j = f(x) - f(x \backslash \{j\})$. The probability r_j is defined in a usual way, also introducing some more parameters. This basic variant of ACO was able to solve exactly only 8 of the first 20 OR-Library test instances. That is why the authors suggest two improvements when the cardinality of x reaches p; (i) the 1-interchange heuristic (Resende and Werneck, 2003) is applied; or (ii) local search with the LK neighborhood structure (Kochetov et al., 2005) is performed with x as an initial solution. A randomized drop routine followed by a 1-interchange or LK local search is repeated a given number of times, and the best overall solution is kept. Both improved versions were able to solve all 20 OR-Library instances.

(viii) *Neural networks (NN).* In Domínguez Merino and Muñoz Pérez (2002), a new integer formulation of the p-median problem allows the application of a two-layer neural network to solve it. In Domínguez Merino et al. (2003), a competitive recurrent neural network consisting of a single layer with $2np$ neurons is used to design three different algorithms. A competitive neural network is a two layer and fully connected network in which neurons compete to became active under certain conditions.

(ix) *Decomposition (DC).* In Dai and Cheung (1997), two decomposition heuristics aiming at problems of large scale are proposed. Firstly, a level-m optimum is defined. Starting from a local optimum, the first heuristic efficiently improves it to a level-2 optimum by applying an existing exact algorithm for solving the 2-median problem. The second heuristic further improves it to a level-3 optimum by applying a new exact algorithm for solving the 3-median problem. In Taillard (2003), three heuristics have been developed for solving large centroid clustering problems. Beside the p-median, this includes the multisource Weber problem and minimum sum-of-squares clustering. The first heuristic, named *candidate list strategy* (CLS), may be seen as a variant of VNS (in the first version of the paper appearing as a technical report in 1996, CLS was called VNS): the alternate heuristic is used as a local search procedure; a random perturbation, or shaking, of the current solution is done by choosing solutions from the restricted interchange neighborhood. The other two, called LOPT and DEC, use decomposition for solving large problem instances. An interesting idea of finding the partition of L, and thus the number of subproblems, by using dynamic programming is developed in the DEC procedure.

(x) *Hybrid heuristic (HH)* that combines elements of several "pure" metaheuristics is suggested in Resende and Werneck (2004). Like GRASP (Greedy Randomized Adaptive Search Procedure, Feo and Resende, 1995), their heuristic is a multi-start approach where each iteration consists of the construction of initial points by a randomized greedy step, followed by local search. As in TS and SS, their method borrows the idea of path-relinking (Laguna and Martíi, 1999). That is, a path between any two solutions from a set of good or *elite* solutions is found and local search performed starting from each solution on that path. Since the distance between two solutions (defined by the symmetric difference) is systematically changed by one before local search is performed, their path-relinking shares a similarity with VNS as well. Moreover, they augment path-relinking with the concept of multiple generations, a key feature of genetic algorithms. A large empirical analysis includes OR-Library, TSP-Lib, Galvão and Resende-Werneck (see above) sets of instances. Compared with other methods, their procedure often provides better results both in terms of running time and solution quality.

7. Conclusions

Table 1 presents an overview on the development of heuristics for solving the p-median problem (PMP). We should ask a basic question given the nature of this survey: Has the advent of metaheuristics advanced the state-of-the-art significantly? Based on a large body of empirical evidence, the answer should be a resounding Yes! While the earlier methods of constructive heuristics and local searches have been successful on relatively small instances of PMP, the empirical results show that solution quality may deteriorate rapidly with problem size. The use of metaheuristics has led to substantial improvements in solution quality on large scale instances within reasonably short computing time. Using nomenclature from tabu search, the success may be attributed to the ability of these metaheuristic-based methods to "intensify" the search in promising regions of the solution space, and then "diversify" the search in a systematic way when needed.

Some brief conclusions on the use of metaheuristics are as follows: (i) The neighborhood structure used in descent plays the most important role for the efficiency and effectiveness of any metaheuristic for PMP. The interchange neighborhood appears to be a better choice than the alternate, or drop/add. The variable depth neighborhood structure LK(k) (Kochetov et al., 2005) seems to be a better choice than the 1-interchange. (ii) The implementation of 1-interchange local search is the second very important issue. The implementation of Whitaker (1983) is better than that suggested by Teitz and Bart (1968), but not better than that proposed by Hansen and Mladenović (1997). This one in turn is outperformed by the implementation of Resende and Werneck (2003). Therefore, it is not easy to conclude what metaheuristic approach dominates others.

References

Alp, O., Erkut, E., Drezner, D., 2003. An efficient genetic algorithm for the *p*-median problem. Annals of Operations Research 122, 21–42.

Barahona, F., Anbil, R., 2000. The volume algorithm: producing primal solutions with a subgradient algorithm. Mathematical Programming 87, 385–399.

Baum, E.B., 1986. Toward practical "neural" computation for combinatorial optimization problems. In: Denker, J. (Ed.), Neural Networks for Computing. American Institute of Physics.

Beasley, J.E., 1985. A note on solving large *p*-median problems. European Journal of Operational Research 21, 270–273.

Beasley, J.E., 1990. OR-Library: Distributing test problems by electronic mail. Journal of Operational Research Society 41, 1069–1072.

Beasley, J.E., 1993. Lagrangian heuristics for location problems. European Journal of Operational Research 65, 383–399.

Beltran, C., Tadonki, C., Vial, J.-Ph., 2004. Solving the *p*-median problem with a semi-Lagrangian relaxation. Logilab Report, HEC, University of Geneva, Switzerland.

Bowerman, J., Calamai, P.H., Hall, G.B., 1999. The demand partitioning method for reducing aggregation errors in *p*-median problems. Computers & Operations Research 26, 1097–1111.

Brandeau, M.L., Chiu, S.S., 1989. An overview of representative problems in location research. Management Science 35 (6), 645–674.

Captivo, E.M., 1991. Fast primal and dual heuristics for the *p*-median location problem. European Journal of Operational Research 52, 65–74.

Casillas, P., 1987. Data aggregation and the *p*-median problem in continuous space. In: Ghosh, A., Rushton, G. (Eds.), Spatial Analysis and Location Allocation Models. Van Nostrand Reinhold Publishers.

Chaudhry, S.S., He, S., Chaudhry, P.E., 2003. Solving a class of facility location problems using genetic algorithm. Expert Systems 20, 86–91.

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

Christofides, N., 1975. Graph Theory: An Algorithmic Approach. Academic Press, New York.

Cornuejols, G., Fisher, M.L., Nemhauser, G.L., 1977. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. Management Science 23, 789–810.

Crainic, T., Gendreau, M., Hansen, P., Mladenović, N., 2004. Cooperative parallel variable neighborhood search for the *p*-median. Journal of Heuristics 10, 293–314.

Current, J., Schilling, D., 1987. Elimination of source A and B errors in *p*-median location problems. Geographical Analysis 19, 95–110.

Dai, Z., Cheung, T.-Y., 1997. A new heuristic approach for the *p*-median problem. Journal of the Operational Research Society 48 (9), 950–960.

Daskin, M., 1995. Network and Discrete Location. Wiley, New York.

Densham, P.J., Rushton, G., 1992. A more efficient heuristic for solving large *p*-median problems. Papers in Regional Science 71 (3), 307–329.

Dibbie, C., Densham, P.J., 1993. Generating intersecting alternatives in GIS and SDSS using genetic algorithms. In: GIS/LIS Symposium, Lincoln.

Domínguez Merino, E., Muñoz Pérez, J., 2002. An efficient neural network for the *p*-median problem. Lecture Notes in Artificial Intelligence 2527, 460–469.

Domínguez Merino, E., Muñoz Pérez, J., Jerez Aragonés, J., 2003. Neural Network Algorithms for the *p*-median problem. In: ESANN'2003 Proceedings – European Symposium on Artificial Neural Networks Bruges (Belgium), pp. 385–391.

Dorigo, M., Di Caro, G., 1999. The ant colony optimization meta-heuristic. In: Corne, G., Dorigo, M., Glover, F. (Eds.), New Ideas in Optimization. McGraw-Hill, New York, pp. 11–32.

Dorigo, M., Maniezzo, V., Colorny, A., 1991. Ant system: An autocatalytic optimizing process. Technical report TR-91-016, Politecnici di Milano.

Drezner, Z. (Ed.), 1995. Facility Location. A Survey of Applications and Methods. Springer, New York.

Drezner, Z., Hamacher, H. (Eds.), 2002. Facility Location: Applications and Theory. Springer, Berlin.

Erkut, E., Bozkaya, B., 1999. Analysis of aggregation errors for the *p*-median problem. Computers & Operations Research 26, 1075–1096.

Erlenkotter, D., 1978. A dual-based procedure for uncapacitated facility location. Operations Research 26, 992–1009.

Estivill-Castro, V., Torres-Velazquez R., 1999. Hybrid genetic algorithm for solving the *p*-median problem. In: Yao, X. et al. (Eds.), Seal 1998, LNCS 1585, New York, pp. 18–25.

Feldman, E., Lehrer, F.A., Ray, T.L., 1966. Warehouse locations under continuous economies of scale. Management Science 12, 670–684.

Feo, T., Resende, M., 1995. Greedy randomized adaptive search. JOURNAL Global Optimization 6, 109–133.

Francis, R.L., Lowe, T.J., Tamir, A., 2000. Aggregation error bounds for a class of location models. Operations Research 48, 294–307.

Francis, R.L., Lowe, T.J., Tamir, A., 2003. Worst-case incremental analysis for a class of *p*-facility location problems. Networks 39, 139–143.

Galvão, R.D., 1980. A dual-bounded algorithm for the *p*-Median problem. Operations Research 28, 1112–1121.

Galvão, R.D., 1993. Use of lagrangean relaxation in the solution of uncapacitated facility location problems. Location Science 1, 57–70.

García-López, F., Melián Batista, B., Moreno Pérez, J.A., Moreno Vega, J.M., 2002. The parallel variable neighborhood search for the *p*-median problem. Journal of Heuristics 8, 375–388.

García-López, F., Melián Batista, B., Moreno Pérez, J.A., Moreno Vega, J.M., 2003. Parallelization of the scatter search

for the p-median problem. Parallel Computing 29 (5), 575–589.
Garey, M.R., Johnson, D.S., 1978. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New-York.
Glover, F., 1989. Tabu Search – Part I. ORSA Journal on Computing 1, 190–206.
Glover, F., 1990. Tabu Search – Part II. ORSA Journal on Computing 2, 4–32.
Glover, F., Kochenberger, G. (Eds.), 2003. Handbook of Metaheuristics. Kluwer Academic Publisher, Dordrecht.
Glover, F., Laguna, M., 1993. Tabu Search. In: Reeves, C. (Ed.), Modern heuristic techniques for combinatorial problems. Blackwell, Oxford (Chapter 3).
Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic Publishers, Norwell, MA.
Glover, F., Laguna, M., Marty, R., 2000. Fundamentals of scatter search and path relinking. Control and Cybernetics 39 (3), 653–684.
Goncharov, E., Kochetov, Y., 2002. Probabilistic tabu search for the unconstrained discrete optimization problems. Discrete Analysis and Operations Research 9 (2), 13–30 (in Russian).
Goodchild, M.F., 1979. The aggregation problem in location-allocation. Geographical Analysis 11, 240–255.
Hansen, P., Jaumard, B., 1997. Cluster analysis and mathematical programming. Mathematical Programming 79, 191–215.
Hansen, P., Mladenović, N., 1997. Variable neighborhood search for the p-median. Location Science 5, 207–226.
Hansen, P., Mladenović, N., 2001a. Variable neighborhood search: Principles and applications. European Journal of Operational Research 130, 449–467.
Hansen, P., Mladenović, N., 2001b. Developments of variable neighborhood search. In: Ribeiro, C., Hansen, P. (Eds.), Essays and surveys in metaheuristics. Kluwer Academic Publishers, Boston/Dordrecht/London, pp. 415–440.
Hansen, P., Mladenović, N., Pérez-Brito, D., 2001. Variable neighborhood decomposition search. Journal of Heuristics 7 (4), 335–350.
Hillsman, E.L., Rhoda, R., 1978. Errors in measuring distances from populations to service centers. Annals of Regional Science Association 12, 74–88.
Hodgson, M.J., Neuman, S., 1993. A GIS approach to eliminating source C aggregation error in p-median models. Location Science 1, 155–170.
Hodgson, M.J., Salhi, S., 1998. Using a quadtree structure to eliminate aggregation error in point to point allocation. Presented at IFORS, Montreal.
Hosage, C.M., Goodchild, M.F., 1986. Discrete space location-allocation solutions from genetic algorithms. Annals of Operations Research 6, 35–46.
Hribar, M., Daskin, M., 1997. A dynamic programming heuristic for the p-median problem. European Journal of Operational Research 101, 499–508.
Kariv, O., Hakimi, S.L., 1969. An algorithmic approach to network location problems; part 2. The p-medians. SIAM Journal on Applied Mathematics 37, 539–560.
Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680.
Kochetov, Y., 2001. Probabilistic local search algorithms for the discrete optimization problems. Discrete Mathematics and Applications, Moscow, MSU, 84–117 (in Russian).
Kochetov, Y., Alekseeva, E., Levanova, T., Loresh, N., 2005. Large neighborhood search for the p-median problem. Yugoslav Journal of Operations Research 15 (1), 53–63.
Kuehn, A.A., Hamburger, M.J., 1963. A heuristic program for locating warehouses. Management Science 9 (4), 643–666.
Laguna, M., Martíi, R., 1999. GRASP and path relinking for 2-layer straight line crossing minimization. INFORMS Journal on Computing 11, 44–52.
Levanova, T., Loresh, M.A., 2004. Algorithms of ant system and simulated annealing for the p-median problem. Automation and Remote Control 65, 431–438.
Love, R.F., Morris, J.G., Wesolowsky, G.O., 1988. Facilities Location: Models and Methods. North Holland, New York.
Maranzana, F.E., 1964. On the location of supply points to minimize transportation costs. Operations Research Quarterly 12, 138–139.
Mirchandani, P., Francis, R. (Eds.), 1990. Discrete location theory. Wiley-Interscience.
Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Computers & Operations Research 24, 1097–1100.
Mladenović, N., Moreno-Pérez, J.A., Moreno-Vega, J.M., 1995. Tabu search in solving p-facility location-allocation problems. Les Cahiers du GERAD, G-95-38, Montreal.
Mladenović, N., Moreno-Pérez, J.A., Moreno-Vega, J.M., 1996. A chain-interchange heuristic method. Yugoslav Journal of Operations Research 6, 41–54.
Moreno-Pérez, J.A., Rodríguez, C., Jimenez, N., 1991. Heuristic cluster algorithm for multiple facility location-allocation problem. RAIRO – Recherche Operationnelle/Operations Research 25, 97–107.
Moreno-Pérez, J.A., García-Roda, J.L., Moreno-Vega, J.M., 1994. A parallel genetic algorithm for the discrete p-median problem. Studies in Location Analysis 7, 131–141.
Mulvey, J.M., Crowder, H.P., 1979. Cluster analysis: An application of Lagrangian relaxation. Management Science 25, 329–340.
Murray, A.T., Church, R.L., 1996. Applying simulated annealing to planning-location models. Journal of Heuristics 2, 31–53.
Ng, B.T., Han, J., 1994. Efficient and effective clustering methods for spatial data mining. In: Bocca, J. et al. (Eds.), 20th International Conference on Very Large Data Bases. Morgan, Kaufmann, pp. 144–155.
OpenMP, October 1997. OpenMP: A proposed Industry Standard API for Shared Memory Programming. White Paper.
Papadimitriou, C., 1994. Computational Complexity. Addison-Wesley, Reading, MA.
Pizzolato, N.D., 1994. A heuristic for large-size p-median location problems with application to school location. Annals of Operations Research 50, 473–485.
Plastria, F., 2001. On the choice of aggregation points for continuous p-median problems: A case for the gravity center. TOP 9, 217–242.
Reeves, C. (Ed.), 1993. Modern Heuristic Techniques for Combinatorial Problems. Blackwell, Oxford.
Reinelt, G., 1991. TSPLIB – A traveling salesman problem library. ORSA Journal on Computing 3, 376–384.
Resende, M., Werneck, R.F., 2003. On the implementation of a swap-based local search procedure for the p-median problem. In: Ladner, Richard E. (Ed.), Proceedings of the 5th Workshop on Algorithm Engineering and Experiments. SIAM, Philadelphia, pp. 119–127.

Resende, M., Werneck, R.F., 2004. A hybrid heuristic for the p-median problem. Journal of Heuristics 10 (1), 59–88.

Rolland, E., Schilling, D.A., Current, J.R., 1996. An efficient tabu search procedure for the p-median problem. European Journal of Operational Research 96, 329–342.

Rosing, K.E., ReVelle, C.S., 1997. Heuristic concentration: Two stage solution construction. European Journal of Operational Research 97, 75–86.

Rosing, K.E., ReVelle, C.S., Rolland, E., Schilling, D.A., Current, J.R., 1998. Heuristic concentration and tabu search: A head to head comparison. European Journal of Operational Research 104, 93–99.

Rosing, K.E., ReVelle, C.S., Schilling, D.A., 1999. A gamma heuristic for the p-median problem. European Journal of Operational Research 117, 522–532.

Salhi, S., 1997. A perturbation heuristic for a class of location problems. Journal of Operational Research Society 48, 1233–1240.

Salhi, S., 2002. Defining tabu list size and aspiration criterion within tabu search methods. Computers and Operations Research 29, 67–86.

Salhi, S., Atkinson, R.A., 1995. Subdrop: A modified drop heuristic for location problems. Location Science 3, 267–273.

Senne, L.F.E., Lorena, A.N.L., 2000. Lagrangian/surrogate heuristics for p-median problems. In: Laguna, M., Gonzales-Velarde, J.L. (Eds.), Computing tools for modelling, optimization and simulation: Interfaces in computer science and operations research. Kluwer Academic Publisher, Dordrecht, pp. 115–130.

Taillard, É.D., 2003. Heuristic methods for large centroid clustering problems. Journal of Heuristics 9 (1), 51–73.

Teitz, M.B., Bart, P., 1968. Heuristic methods for estimating the generalized vertex median of a weighted graph. Operations Research 16, 955–961.

Voss, S., 1996. A reverse elimination approach for the p-median problem. Studies in Locational Analysis 8, 49–58.

Whitaker, R., 1983. A fast algorithm for the greedy interchange for large-scale clustering and median location problems. INFOR 21, 95–108.

Yannakakis, M., 1997. Computational complexity. In: Aarts, J.K., Lenstra, J.K. (Eds.), Local Search in Combinatorial Optimization. Wiley, Chichester, pp. 19–56.