European Journal of Operational Research 220 (2012) 320–327

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Some heuristic methods for solving *p*-median problems with a coverage constraint

Jesús Sáez-Aguado *, Paula Camelia Trandafir

Departamento de Estadística e Investigación Operativa Facultad de Ciencias, Universidad de Valladolid, c/Prado de la Magdalena s/n, 47005 Valladolid, Spain

ARTICLE INFO

Article history:
Received 25 October 2011
Accepted 7 February 2012
Available online 21 February 2012

Keywords:
Location
p-Median
Coverage constraint
Local search
GRASP
Lagrangean relaxation

ABSTRACT

The aim of this paper is to solve *p*-median problems with an additional coverage constraint. These problems arise in location applications, when the trade-off between distance and coverage is being calculated. Three kinds of heuristic algorithms are developed. First, local search procedures are designed both for constructing and improving feasible solutions. Second, a multistart GRASP heuristic is developed, based on the previous local search methods. Third, by employing Lagrangean relaxation methods, a very efficient Lagrangean heuristic algorithm is designed, which extends the well known algorithm of Handler and Zang, for constrained shortest path problems, to constrained *p*-median problems. Finally, a comparison of the computational efficiency of the developed methods is made between a variety of problems of different sizes.

1. Introduction

The *p*-median problem (PMP) is one of the most studied issues in combinatorial optimization, having many applications, among which cluster analysis [16,21], various location problems [10] and optimal diversity management [6] can be mentioned. For a general discussion of the problem, the reader is directed to [10,19].

Since Kariv and Hakimi [18] proved that the PMP is NP-hard, several heuristic algorithms have been found to solve the problem in an approximate way. Mladenovic et al. [20], Reese [24] and Alba and Domínguez [3] survey different approximations and metaheuristics designed for this problem, most of which are based on interchange or swap-based local search.

The interchange heuristic for the PMP, beginning with the well known Teitz and Bart algorithm [32], was improved with Whitaker's fast implementation [33] and, more recently, Resende and Werneck's [27,30]. Based on these implementations, Resende and Werneck [28,29] applied the GRASP (Greedy Randomized Adaptive Search Procedure) methodology to PMP, resulting in different algorithms which can be considered among the most efficient for the problem.

In this paper, the *p*-median problem with an additional coverage constraint (CONPMP) is considered. The constraint establishes that the total demand covered at a distance greater than some pre-established coverage distance should not exceed a previously chosen value. This constraint is related to the maximum covering location problem (MCLP), which is another important location problem (see [10,19]), whose objective is equivalent to minimizing the demand covered at a distance greater than the coverage distance. While the solution of the PMP is associated to maximum efficiency when designing a service, the solution of the CONPMP problem looks for maximum efficiency subject to a required minimum coverage.

CONPMP is a problem of interest in its own right, but it can also be used for exploring the trade-off between distance and coverage, by systematically varying the demand allowed to be covered at a distance greater than the coverage distance. This can be done, for example, by applying the ε-constraint method to the biobjective problem with extremum problems PMP and MCLP. This biobjective problem is considered in [10], which presents an elementary method, based on weighted sums, for finding a set of supported efficient solutions, and suggests that, with the ε-constraint method, all efficient solutions could be found. For an overview on available methods for solving multiobjective integer linear programming problems, the reader is directed to the excellent textbooks [9,11].

The main purpose of this paper is to provide methods for effectively solving the CONPMP subproblems. Although the aim is not to carry out a complete multiobjective analysis, there will be some discussion on the inherent trade-off between distance and coverage, which are the objectives of the PMP and MCLP problems, respectively. In a similar context, fixed charge facility location problems with coverage constrictions, and the corresponding trade-off between cost and coverage, have been considered in [23,34].

It is well known that adding a constraint to an optimization problem can make it much more difficult. This happens, for example, in the shortest-path [15], minimum spanning tree [2] or

* Corresponding author. Tel.: +34 983186417; fax: +34 983423013.
E-mail addresses: jsaez@eio.uva.es (J. Sáez-Aguado), camelia@eio.uva.es (P.C. Trandafir).

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2012.02.011

assignment problems [1]. In these cases, an additional constraint turns an easy problem into an NP-hard one. In our case, PMP is already NP-hard, so adding the CONPMP constraint does not modify this complexity, but the problem is computationally more difficult.

In this paper, different heuristic procedures for the CONPMP problem are developed. The first group of algorithms are modified versions of the local search and interchange algorithms used in the PMP. In this way, local search procedures are designed both for constructing and improving feasible solutions for CONPMP. These procedures are the basis for the design of a multistart GRASP heuristic for the CONPMP problem. Given that GRASP procedures are among the most efficient for solving PMP, it is hoped that this will also be the case for CONPMP.

The next heuristic procedure for CONPMP is based on Lagrangean relaxation. Lagrangean relaxation is a well-known methodology for solving large-scale combinatorial optimization problems. It is based on exploiting the inherent structure of each problem in order to obtain lower and upper bounds on the optimal value of the problem. For surveys on the theory and applications of Lagrangean relaxation, see for instance [14,22]. Daskin [10] is an excellent reference on the application of Lagrangean relaxation techniques to different location problems.

Lagrangean relaxation based heuristics have been proposed in the literature for numerous combinatorial optimization problems. Some of them are: [10,5] for the PMP, [7] for the set covering problem, [4] for the capacitated facility location problem, [6] for the diversity management problem, [21] for the cluster analysis problem, [8] for the train timetabling problem and [31] for the fixed charge transportation problem.

Lagrangean relaxation techniques have very often been applied to problems with an additional constraint. So, in [15] Handler and Zang develop specific procedures for constrained shortest-path problems, based on Lagrangean relaxation. Similar procedures have been developed, for instance, for constrained assignment problems [1] and constrained minimum spanning tree [2], to mention only a few. On the other hand, in [17], the Handler and Zang algorithm is extended to general resource constrained optimization problems, and the running time of the resulting algorithm is established.

In this paper, we apply Lagrangean relaxation to the CONPMP problem based on the resolution of subproblems of type PMP. Instead of using classical procedures, such as subgradient optimization [21,22] or bisection search for the case of a unique multiplier [13], the extension of the well-known algorithm of Handler and Zang [15] for the restricted shortest path problem to CONPMP has been found to be much more efficient here. The extension of the Handler and Zang algorithm to other structured problems with an additional constraint is termed the Lagrangean Relaxation Based Aggregated Cost (LARAC) algorithm in [17,35]. The solution found by the LARAC method can, in addition, be improved with the local search heuristic, finally resulting in a very efficient heuristic method that provides solutions near optimality.

The paper is structured as follows. Section 2 states the problem and establishes the notation used in the rest of the article. Section 3 develops the basic procedures of local search for the CONPMP problem, which lead to the multistart GRASP algorithm in Section 4. Section 5 develops a Lagrangean relaxation which is the basis of the LARAC algorithm. Section 6 presents the computational results obtained when comparing the different approximations on two whole batteries of problems. Section 7 gives the final conclusions.

2. Statement and notation

CONPMP may be stated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} h_i d_{ij} y_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} y_{ij} = 1 \quad i = 1,\ldots,m \tag{2}$$

$$y_{ij} \leqslant x_j \quad i = 1,\ldots,m,\ j = 1,\ldots,n \tag{3}$$

$$\sum_{j=1}^{n} x_j = p \tag{4}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} q_{ij} y_{ij} \leqslant \varepsilon \tag{5}$$

$$x_j \in \{0,\ 1\} \quad j = 1,\ldots,n \tag{6}$$

$$y_{ij} \in \{0,\ 1\} \quad i = 1,\ldots,m,\ j = 1,\ldots,n \tag{7}$$

In this statement, m is the number of points of demand of a service, n the number of points of possible establishment of service, p the number of service points which will be opened, h_i the demand at point i, d_{ij} the distance between the demand point i and service point j; $x_j \in \{0, 1\}$ is the location variable, defined for $j = 1, \ldots, n$, with $x_j = 1$ if a facility is located at point j, and $y_{ij} \in \{0, 1\}$ is the assignment variable, defined for $i = 1, \ldots, m$, $j = 1, \ldots, n$, with $y_{ij} = 1$ if all demand at point i is assigned to a facility at point j.

The objective function (1) expresses the sum total of distances traversed by all clients of the system. The constraints (2) indicate that all demand points must be serviced or assigned to a unique service point. The constraints (3) guarantee that if a service point is not open, then it cannot service any demand point, that is, $x_j = 0 \Rightarrow y_{ij} = 0, i = 1,\ldots,m$. The constraints (4) mean that the total number of open installations must be the pre-established number p. Finally, (6) and (7) state that all variables take only the values 0 and 1.

The coefficients q_{ij} in the constraints (5) are defined as follows, based on a coverage distance DC established by the user:

$$q_{ij} = \begin{cases} h_i & \text{if} \quad d_{ij} > DC \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

So, the left hand side of (5) is the total demand covered at a distance greater than DC, which must not exceed ε, also specified by the user.

If we denote by $\mathcal{F}$ the feasible set given by the constraints (2)–(4) and (6) and (7), by $f_1(y) = \sum_{i=1}^{m}\sum_{j=1}^{n} h_i d_{ij} y_{ij}$, and by $f_2(y) = \sum_{i=1}^{m}\sum_{j=1}^{n} q_{ij} y_{ij}$, the problem can be stated in a more compact form as:

$$\text{(P)} \quad \text{Minimize} \quad f_1(y) \tag{9}$$
$$\text{subject to} \quad (x, y) \in \mathcal{F}$$
$$f_2(y) \leqslant \varepsilon \tag{10}$$

It should be noted that $\mathcal{F}$ is the feasible set of a PMP problem without constraints. To avoid trivial cases, in all that follows, it will be supposed that $\mathcal{F}$ is nonempty. It will also be assumed that exact or approximate algorithms are available to solve PMP problems with any linear objective.

Regarding the feasibility of problem (P), for $k = 1, 2$, consider first the two PMP problems:

$$(P_k) \quad \text{Minimize} \quad f_k(y)$$
$$\text{subject to} \quad (x, y) \in \mathcal{F} \tag{11}$$

Let $(x^k, y^k) \in \mathcal{F}$ be an optimal solution of (P_k). Defining $\varepsilon_1 = f_2(y^1)$ and $\varepsilon_2 = f_2(y^2)$, the following is immediately verified:

Proposition 1. *If the feasible set for PMP $\mathcal{F}$ is nonempty, then*

1. *Problem (P) is feasible if and only if $\varepsilon_2 \leqslant \varepsilon$.*
2. *If $\varepsilon \geqslant \varepsilon_1$, then (x^1, y^1) fulfills the constraint $f_2(y) \leqslant \varepsilon$, and is therefore an optimum for (P).*

Thus, the only case of interest for (P) is when $\varepsilon_2 \leqslant \varepsilon \leqslant \varepsilon_1$. These two values are obtained by solving the two p-median problems (P_k), for $k = 1, 2$.

3. Local search heuristics for the CONPMP problem

In this section, two heuristic procedures are developed based on local search. These procedures are modifications of the usual interchange search for PMP [32,33,27,30], in order to take into account the additional constraint (5). In every case, the neighborhood structure is the following: given $(x, y) \in \mathcal{F}, N(x, y)$ consists of those (x', y') such that x' is obtained from x via a simple interchange and y' by assigning the closest j with $x'_j = 1$ to each demand point i.

3.1. Heuristic for finding a feasible solution

This subsection looks at how to construct feasible solutions for problem (9) and (10). It is assumed that heuristic algorithms to solve any PMP with a linear objective are available and, in this way, the solutions verifying $(x, y) \in \mathcal{F}$ will be generated. Although it is possible to design simple greedy heuristics that consider the constraint $f_2(y) \leqslant \varepsilon$, analogous to those developed in [23] for uncapacitated fixed charge facility location, we have observed that, for p-median problems, these heuristics frequently end without finding a solution that verify the constraint (10). The following algorithm is essentially a local search algorithm for PMP with the objective of minimizing $f_2(y)$, which ends with the first solution that satisfies $f_2(y) \leqslant \varepsilon$.

Algorithm 1. *H*1 Algorithm

Input: Any solution $(x, y) \in \mathcal{F}$.
Output: A solution $(x^*, y^*) \in \mathcal{F}$ verifying $f_2(y^*) \leqslant \varepsilon$.

General Step.
1. If $f_2(y) \leqslant \varepsilon$, let $(x^*, y^*) = (x, y)$ and stop.
2. Examine $N(x, y)$. If there is no $(x', y') \in N(x, y)$ such that $f_2(y') < f_2(y)$, stop. A local optimum has been reached but no feasible solutions have been found.
3. Choose $(x', y') \in N(x, y)$ such that $f_2(y') < f_2(y)$, redefine $(x, y) = (x', y')$ and go back to Step 1.

End of *H*1

3.2. Improvement heuristic

Now we shall start from a feasible solution, that is, one satisfying $(x, y) \in \mathcal{F}$ and $f_2(y) \leqslant \varepsilon$, with the aim of improving the objective $f_1(y)$.

Algorithm 2. *H*2 Algorithm

Input: A solution $(x, y) \in \mathcal{F}$ with $f_2(y) \leqslant \varepsilon$.
Output: A solution $(x^*, y^*) \in \mathcal{F}$ verifying $f_2(y^*) \leqslant \varepsilon$

General Step.
1. Examine the neighborhoods $(x', y') \in N(x, y)$, which satisfy $f_2(y') \leqslant \varepsilon$. If there is no $(x', y') \in N(x, y)$ which simultaneously satisfies $f_1(y') < f_1(y)$ and $f_2(y') \leqslant \varepsilon$, let $(x^*, y^*) = (x, y)$ and stop, since a local optimum has been reached.
2. Take $(x', y') \in N(x, y)$ such that $f_1(y') < f_1(y)$ and $f_2(y') \leqslant \varepsilon$, let $(x, y) = (x', y')$ and go back to Step 1.

End of *H*2

As with any local search algorithm, both *H*1 and *H*2 can be implemented using either a best improving or a first improving strategy.

4. Multistart GRASP heuristic

The *H*1 (construction) and *H*2 (improvement) heuristics of the previous section can be included in various forms in a multistart algorithm which considers two important aspects of any metaheuristic: diversification and intensification. In this paper, the GRASP methodology has been chosen for CONPMP because it is one of the most efficient metaheuristics for the PMP problem. For comprehensive information on GRASP methodology see [25,26].

A GRASP is a multistart process in which each iteration consists of two main blocks: construction of a feasible solution, and improvement by local search. Diversification is obtained, in the construction phase, with a randomized greedy algorithm, while intensification is obtained, in the improvement phase, with a local search algorithm. The different iterations are independent, and the best solution found is saved and returned as the result.

In the construction phase, a feasible solution is iteratively constructed, one element at a time. To this end, at each iteration, let the set of candidate elements be set by all elements that can be incorporated to the partial solution under construction without destroying feasibility. Then a restricted candidate list (RCL) is considered, formed by the elements that can be incorporated to the partial solution under construction and have the best incremental cost. The element to be incorporated is randomly selected from those in the RCL. The RCL can be chosen with different criteria, but one of the more important is the cardinality-based criteria. In this case, the cardinality of RCL is a priori fixed to some chosen value r, and the RCL is made up of those elements having the r best costs. See [25,26] for many other ways of forming the RCL.

For CONPMP, we have found that a randomized greedy algorithm frequently ends with no feasible solution, due to the simultaneous presence of the constraints (4) and (10). This is significantly different for the fixed charge facility problem with a coverage constraint considered in [23], for which it is possible to open enough facilities for constraint (10) to be fulfilled. In our case, the number of open facilities is fixed to p, and it is not so easy to design a greedy algorithm that takes into account the constraint (10).

The first phase of a GRASP iteration, which usually corresponds to a randomized greedy algorithm for CONPMP, has been replaced by two steps: a randomized greedy algorithm for the PMP (P_1), with objective $f_1(y)$, followed by an *H*1 algorithm to restore feasibility. It is expected that algorithm *H*1 will restore feasibility without deteriorating the objective $f_1(y)$ excessively. The second phase is the improvement heuristic *H*2, with the aim of improving the objective $f_1(y)$. The complete algorithm, with K starts, is the following (*it* is the iteration index):

Algorithm 3. *H*3**GRASP Algorithm**

Input: K.
Output: A solution $(x^*, y^*) \in \mathcal{F}$ verifying $f_2(y^*) \leqslant \varepsilon$.

Initial Step. Set $f_1^* = \infty, it = 0$.
General Step. While $it \leqslant K$, **do**:
1. **Iteration.** Set $it = it + 1$.
2. **Greedy randomized construction**. Apply a greedy randomized construction method to the PMP problem (P_1) with objective $f_1(y)$, and let $(x^1, y^1) \in \mathcal{F}$ be the obtained solution.
3. **Feasibility.** Beginning with $(x^1, y^1) \in \mathcal{F}$, apply algorithm *H*1 to find a feasible solution. If one is found, let $(x^2, y^2) \in \mathcal{F}$, with $f_2(y^2) \leqslant \varepsilon$. If not, exit the iteration and go back to step 1.
4. **Local search.** Beginning with (x^2, y^2), apply algorithm *H*2, and let (x^3, y^3) be the obtained solution.
5. **Update.** If $f_1(y^3) < f_1^*$, set $f_1^* = f_1(y^3), (x^*, y^*) = (x^3, y^3)$.

end do.
End of *H*3

5. A Lagrangean relaxation based heuristic

5.1. Lagrangean relaxation

Lagrangean relaxation is one of the most commonly used techniques for large scale combinatorial optimization problems. For general information on this technique, the reader is remitted to the references given in Section 1. It is based on exploiting the inherent structure of each problem to obtain lower and upper bounds on the optimal value of the problem. For CONPMP, it is clear that the constraint (10) is what makes the problem more difficult, so it is a candidate to be dualized.

Dualizing the constraint (10) in the formulation (9) and (10), with a multiplier $\lambda \geqslant 0$, the following Lagrangean function is obtained:

$$L(\lambda) = \min_{(x,y)\in\mathcal{F}} f_1(y) + \lambda f_2(y) - \lambda\varepsilon.$$

After eliminating the constant term $-\lambda\varepsilon$, for any $\lambda \geqslant 0$ the subproblem that has to be solved is

$$P(\lambda) \quad \min_{(x,y)\in\mathcal{F}} \{f_1(y) + \lambda f_2(y)\} \qquad (12)$$

Note that $P(\lambda)$ is a p-median problem with an aggregated cost objective. The corresponding dual problem is:

$$(D_L) \max_{\lambda\geqslant 0} L(\lambda)$$

5.2. LARAC algorithm

In general, the maximization of $L(\lambda)$ is achieved by a subgradient algorithm [14,21,22]. When a single constraint is dualized, as in our case, simpler procedures could be used, based on the properties of $L(\lambda)$, such as a bisection search (see [13]). Based on the Handler and Zang algorithm [15] for the restricted shortest path problem, Jüttner [17] and Xiao et al. [35] have extended its use to other problems of combinatorial optimization with an additional constraint, and called the resulting algorithm LARAC (Lagrangean Relaxation Based Aggregated Cost), because of the form of the objective function of subproblem $P(\lambda)$ in (12). In this section, the LARAC method is applied to CONPMP.

The following properties and observations are derived from the well-known theory of Lagrangean Relaxation (see [14,21,22,17]), and give the motivation and main algorithmic ideas. The following notation will be used: $v(\bullet)$ is the optimal value of problem $(\bullet)$, and $z = v(P)$.

1. For all $\lambda \geqslant 0$, $L(\lambda) \leqslant z$ (lower bound). In particular, if λ^* maximizes $L(\lambda)$, then $L(\lambda^*) = v(D_L) \leqslant z$.
2. $L(\lambda)$ is concave and piecewise linear.
3. A given value $\lambda \geqslant 0$ maximizes $L(\lambda)$ if and only if there are two solutions $(x^A, y^A) \in \mathcal{F}$ and $(x^B, y^B) \in \mathcal{F}$, both being optimal solutions to $P(\lambda)$ for which $f_2(y^A) \geqslant \varepsilon$ and $f_2(y^B) \leqslant \varepsilon$. These solutions can coincide, in which case $f_2(y^A) = f_2(y^B) = \varepsilon$. A proof of this property can be found in [17].
4. It is interesting to consider the biobjective problem min{$f_1(y)$,$f_2(y)$} subject to $(x, y) \in \mathcal{F}$. Observe that, via Lagrangean Relaxation, we will only obtain supported efficient solutions for the biobjective problem because, for any $\lambda > 0$, the subproblem $P(\lambda)$ is a special case of the method of weights, with $\lambda_1 = 1$, $\lambda_2 = \lambda$ (see [11] for the basic elements of multiobjective combinatorial optimization).
5. If the feasible solutions $(x, y) \in \mathcal{F}$ are represented in two-dimensional space by ($f_1(y)$,$f_2(y)$), a typical case can be seen in Fig. 1. The efficient supported solutions are A_1, A_2, A_4, and A_6, while the optimal solution of the problem corresponds to the unsupported point A_3. In this case, with Lagrangean relaxation, the best solution that can be obtained is A_4.
6. The main idea of the LARAC method can be illustrated using Fig. 2. At each iteration, there are two supported efficient solutions, which, in the objective space, correspond to a point $A = (a_1, a_2)$ satisfying $a_2 > \varepsilon$, and a point $B = (b_1, b_2)$ satisfying $b_2 \leqslant \varepsilon$, and are therefore associated to a feasible solution of (P). Let λ be the absolute value of the slope of the line connecting A and B. Minimizing $P(\lambda)$ will result in a point C, and one of the three following cases: (a) C is below the line $f_2 = \varepsilon$, as point C_1 is in Fig. 2. In this case, C corresponds to a new feasible solution that improves the present solution B, and so B is replaced by C. (b) C is above the line $f_2 = \varepsilon$, as point C_2 is in Fig. 2. Then, A is replaced by C. (c) C is equal to A or B. By the third observation, the maximum of $L(\lambda)$ has been reached and the algorithm stops.

Algorithm 4. LARAC algorithm

Output: A feasible solution $(x^*, y^*) \in \mathcal{F}$ verifying $f_2(y^*) \leqslant \varepsilon$ and $f_1^* = f_1(y^*)$.

Initial Step.

1.1 Solve problem P_1, which is the p-median problem with objective $f_1(y)$, and let $(\bar{x}, \bar{y})$ be an optimal solution. If $f_2(\bar{y}) \leqslant \varepsilon$, then $(\bar{x}, \bar{y})$ is an optimal solution of problem (P), so stop. Otherwise, set $A = (f_1(\bar{y}), f_2(\bar{y}))$.

2.2 Solve problem P_2, which is the p-median problem with objective $f_2(y)$, and let $(\bar{x}, \bar{y})$ be an optimal solution. If $f_2(\bar{y}) > \varepsilon$, then problem (P) is not feasible, so stop. Otherwise, take $B = (f_1(\bar{y}), f_2(\bar{y}))$ and $(x^*, y^*) = (\bar{x}, \bar{y})$.

General Step.

Given $A = (a_1, a_2)$, $B = (b_1, b_2)$ with $a_2 > \varepsilon$, $b_2 \leqslant \varepsilon$, $a_1 < b_1$,

2.1 Let $\lambda = \frac{b_1 - a_1}{a_2 - b_2}$.
Solve the p-median problem $P(\lambda)$, and let $(\bar{x}, \bar{y})$ be an optimal solution, with corresponding point $C = (f_1(\bar{y}), f_2(\bar{y}))$.

2.2 If C is equal to A or B, stop, since the maximum value of $L(\lambda)$ has been reached. Otherwise,

2.3 If $C \neq B$ and $f_2(\bar{y}) \leqslant \varepsilon$, then necessarily $f_1(\bar{y}) = c_1 < a_1$. Set $B = f_1(\bar{y}), f_2(\bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$ and go to (2.1).

2.4 If $C \neq A$ and $f_2(y) > \varepsilon$, then necessarily $c_2 = f_2(\bar{y}) < a_2$. Set $A = f_1(\bar{y}), f_2(\bar{y})$ and go to (2.1).

End of LARAC.

If $P(\lambda)$ has been solved optimally, the final bound on the optimal value z is valid (see Fig. 1):

$$LB \leqslant z \leqslant UB$$

where $LB = L(\lambda) - \lambda\varepsilon$ and $UB = f_1^*$. Alternatively, the lower bound can be computed as the intersection of the line joining A and B with the line $f_2 = \varepsilon$, giving the value:

$$LB = \frac{f_2^A \cdot f_1^B - f_2^B - f_1^A}{f_2^A - f_2^B} - \frac{f_1^B - f_1^A}{f_2^A - f_2^B}\varepsilon.$$

On the other hand, if $P(\lambda)$ has been solved by heuristic methods, the lower bound may not be correct, although the upper bound always is.

Additionally, the solution obtained by the LARAC method may be improved by the local search algorithm $H2$.

6. Computational results

In this section, the computational results obtained by applying the procedures in previous sections to a series of CONPMP

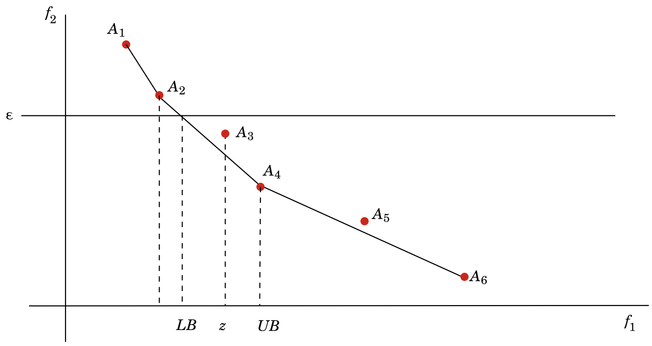

Fig. 1. Representation on the biobjective space.

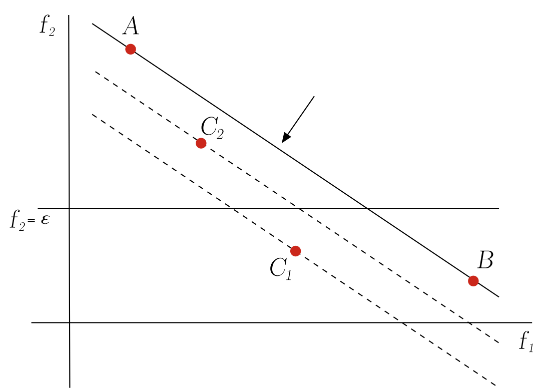

Fig. 2. Main iteration of the LARAC algorithm.

problems are presented. The aim of these experiments is twofold. On the one hand, the effectiveness of the GRASP and LARAC procedures has to be assessed and compared, in terms of solution quality as well as the required computation time. On the other hand, the analysis of the trade-off between distance and coverage, obtained when varying the value ε, can illustrate the main applicability of model CONPMP in the context of location decisions.

In addition, the lower dimensional problems have been solved up to optimality with XPRESS-Optimizer version 20.00.11 [12]. The solver in XPRESS has one of the most efficient implementations of the Branch and Cut algorithm, and includes the most recent developments in integer programming, such as different cutting planes, heuristics and preprocessing methods, as well as sophisticated rules for guiding the enumeration tree. The availability of the optimal values enables the exact deviation of the values obtained by GRASP and LARAC methods to be obtained.

All the algorithms necessary to run the XPRESS, GRASP and LARAC methods have been programmed in C, compiled with Microsoft Visual C++.NET version 7.1 and run on a PC Intel (R) Core (TM)2 CPU 6300 @ 1.85 GHz with 1.98 GB of RAM.

In order to compare the different methods, two data sets have been used. The first one contains 60 CONPMP problems generated in the following manner: in all cases $m = n$ is taken with values 300, 500, 800 and 1000, respectively; integer demands are generated uniformly between 10 and 100, and Euclidean real distances are found from coordinates generated uniformly between 0 and 1000, without rounding. For each dimension, 5 problems have been randomly generated, thus obtaining 20 PMP problems. For each of these test problems, the following 3 values of ε are taken: $\varepsilon_2 + \beta(\varepsilon_1 - \varepsilon_2)$, with β = 0.2,0.5 and 0.8, respectively, finally obtaining 60 CONPMP problems. For every problem, p = 15 and DC = 12 have been taken.

The second data set consists of 32 real data extracted from the data set of the 3100 counties in the contiguous US, which can be obtained at the following web address: http://www.census.gov/geo/www/gazetteer/places2k.html. The 250, 500, 750 and 1000 most populous counties have been extracted and the distances calculated, from the latitude and longitude, assuming a spherical earth projected onto a plane, without rounding. For each dimension, the values p = 10 and p = 20 have been taken, and DC = 130 in all cases. Finally, for each of the eight cases, the following 4 values of ε have been taken: $\varepsilon_2 + \beta(\varepsilon_1 - \varepsilon_2)$, with β = 0.2, 0.4, 0.6, and 0.8, respectively, finally obtaining 32 CONPMP problems.

Before presenting the results, some details should be given on how each algorithm was run.

1. XPRESS was run with all default options, except for a time limit of 6 h, and this was done only for those problems of size 500 or lower, since for those of greater size the efficiency of a general purpose solver is drastically reduced. In fact, after 6 h, no solution is found for dimensions 750 or greater.
2. GRASP. The multistart algorithm $H3$ has been run with K = 20 iterations. This choice gives a good balance between the final time and the quality of the obtained solutions. In the greedy construction step, a restricted list of candidates based on cardinal, with a value of r = 80, has been used. From previous experimental tests, it was observed that this parameter setting is the most suitable for a large number of the studied problems. Increasing or decreasing this value does not, in general, much improve the quality of the obtained solution. In the local search algorithms, $H1$ and $H2$, the best improving rule has been chosen.
3. LARAC. The most important issue in this algorithm is the choice of method for solving the PMP aggregated cost subproblems. To do so, a GRASP method has been used, following the Resende and Werneck implementation [27,28,30] with K = 20 iterations and a restricted list of candidates based on the cardinal, with a value of r = 80, as in the previous case.

Tables 2–5 in Appendix A contain a summary of the results obtained for the two data sets. The first column contains the name of the problem, which refers to m and n. For the second data set, the second column shows the number of medians p, and the remaining columns have the same interpretation for both datasets. Instead of thinking in terms of the original objectives $f_1(y)$ and $f_2(y)$, it is easier and more meaningful to think in terms of average distance and percentage of total demand covered at a distance greater than DC. So, the column labeled ε shows the chosen value of the maximum percentage of the total demand covered at a distance greater than DC. For problems of size 500 or lower, the column labeled z shows the optimal value of the average distance obtained with XPRESS, whereas z^G shows the average distance obtained with GRASP and z^L with LARAC. The column labeled t indicates the time required by XPRESS, in seconds, and analogously t^G and t^L the time for the GRASP and LARAC methods. For all the problems for which the optimum value of the average distance z has been calculated, the said value is presented in boldface, and the percentages of the deviation obtained by GRASP and LARAC are $d^G = 100\frac{z^G - z}{z}$ and $d^L = 100\frac{z^L - z}{z}$, respectively.

Overall, the results displayed in Tables 2–5 show that both GRASP and LARAC are very efficient methods for solving the CONPMP, although it must be emphasized that LARAC is more efficient than GRASP, due to the quality of the solutions it produces and the execution time. Moreover, the behavior of GRASP and LARAC is fairly consistent, with a very low variability over the whole set of instances, both for the generated as well as the real cases. Next, we analyze these questions in some detail.

For those problems having dimension 500 or lower, for which XPRESS is able to obtain the optimal solutions, the average

Table 1
ε-Constraint method, pm250, $p = 20$.

Point	Av. dist.	Coverage	Point	Av. dist.	Coverage
1	82.9310	19.4739	19	85.8534	15.2977
2	83.0255	18.7309	20	86.1398	15.1566
3	83.2803	18.7223	21	85.9863	14.9672
4	83.7955	18.6573	22	86.3390	14.7794
5	83.8009	18.4940	23	86.6458	14.7328
6	83.8490	18.3954	24	87.4950	14.2661
7	83.7145	18.3940	25	87.8286	14.0900
8	83.8755	17.9143	26	87.8476	14.0784
9	83.8958	17.7698	27	88.1813	13.9022
10	83.9439	17.6712	28	88.4734	13.7611
11	83.9705	17.1901	39	89.3141	13.7401
12	84.1034	16.8596	30	89.6477	13.5640
13	84.4560	16.6719	31	89.6667	13.5524
14	85.0631	16.6252	32	90.0004	13.3762
15	85.2400	16.1732	33	90.4831	13.3415
16	85.7585	16.0219	34	93.3770	13.2908
17	85.7788	15.8774	35	96.4907	13.2862
18	85.8269	15.7788	36	93.8397	13.2561

percentages of deviation obtained by GRASP are 0.55, 0.15, 0.15, and 0.72 for problems s300, s500, pm250, and pm500 respectively; while those obtained by LARAC are 0.09, 0.05, 0.09, and 0.11, for the same problems. The total averages over the 46 cases are 0.39 and 0.08 for GRASP and LARAC, respectively, which shows that the solutions obtained by both methods are of very high quality, mainly those of LARAC. In fact, LARAC reaches the optimum in 23 out of the 48 cases, while GRASP reaches it in 11 out of the 48 cases.

For those problems having dimensions greater than 500, the deviations cannot be calculated, but LARAC gives a strictly better solution than GRASP in 35 out of the 48 cases, while the opposite happens in 7 cases, and in another 4 cases both methods are equal.

Regarding the total computation time, the first thing to note in Tables 1–4 is the clear difference between the times of XPRESS on the one hand and those of GRASP and LARAC on the other. This is to be expected when comparing a general purpose solver with a heuristic method, which is specifically designed for the task. Comparing GRASP and LARAC, significant differences are observed between the two datasets. In all cases, LARAC is faster than GRASP, but while for the first dataset, it is on average 2.3 times faster, for the second one, the ratio increases to 16.2. We have not found an explanation for this fact, except the differences in the data, mainly the scale of the demands.

It can be concluded that both GRASP and LARAC methods are efficient for solving CONPMP, although it has been somewhat surprising that LARAC is a little more efficient than GRASP, in both the quality of the solutions and for the computation time.

Tables 1–4 also show, although in a rather coarse way, the tradeoff between mean distance and percentage of coverage at distance greater than *DC*. Given that, in generating the CONPMP problems, only 3 values of ε were chosen for the first set and 4 for the second, the trade-off can only be visualized approximately. Even so, it can be observed how, in every problem, the average distance decreases as the value of the maximum percentage ε increases.

In order to obtain more complete information on the trade-off, it should be necessary to solve the problem (9) for more values of

Table 2
Computational performance, dataset 1, sizes 300 and 500. Boldface values are the optimum.

Problem	ε	XPRESS		GRASP			LARAC		
		z	t	z^G	t^G	d^G	z^L	t^L	d^L
s300_1	22.0256	**8.9323**	56	9.0380	2.75	1.18	**8.9323**	4.44	0
s300_1	23.6032	**8.7951**	34	**8.7951**	21.88	0	**8.7951**	2.52	0
s300_1	25.3111	**8.7627**	48	**8.7627**	32.47	0	8.7689	3.53	0.07
s300_2	20.6154	**8.9469**	57	8.9588	4.27	0.13	**8.9469**	3.94	0
s300_2	23.6522	**8.7545**	68	8.7912	26.75	0.42	**8.7545**	3.33	0
s300_2	24.0630	**8.7515**	84	8.7584	28.80	0.08	**8.7515**	2.50	0
s300_3	19.8059	**8.7967**	75	8.8536	6.92	0.65	8.8536	4.39	0.27
s300_3	21.6649	**8.6581**	19	8.6727	20.63	0.17	8.6640	9.20	0.07
s300_3	23.4026	**8.6074**	27	8.6186	31.48	0.13	8.6146	5.44	0.08
s300_4	21.8812	**9.0526**	45	9.2761	4.80	2.47	**9.0526**	10.81	0
s300_4	23.1715	**8.8219**	17	8.9112	15.61	1.01	8.8555	3.86	0.38
s300_4	24.4618	**8.7449**	77	8.7658	25.72	0.24	8.7590	3.38	0.16
s300_5	22.4227	**8.9115**	19	9.0052	6.30	1.05	**8.9115**	3.92	0
s300_5	24.4996	**8.8426**	20	8.8876	21.70	0.51	8.8679	3.09	0.29
s300_5	26.5567	**8.8273**	47	8.8449	30.09	0.20	8.8330	5.58	0.07
Average			46.2		18.68	0.55		4.66	0.09
s500_1	25.6196	**9.3343**	145	9.3395	39.25	0.06	**9.3343**	46.52	0
s500_1	26.7839	**9.2756**	146	9.2787	74.52	0.03	9.2787	14.03	0.03
s500_1	27.9560	**9.2636**	61	**9.2636**	85.55	0	**9.2636**	23.88	0
s500_2	24.5478	**9.2914**	292	9.2928	74.09	0.02	9.2944	16.86	0.03
s500_2	26.4250	**9.2142**	130	9.2407	103.94	0.29	9.2244	21.83	0.11
s500_2	28.2029	**9.1852**	91	**9.1852**	123.59	0	**9.1852**	20.16	0
s500_3	23.2652	**9.3909**	379	9.4181	15.78	0.28	9.3942	12.72	0.03
s500_3	25.76622	**9.3205**	418	9.3289	86.72	0.30	9.3289	16.84	0.09
s500_3	26.2049	**9.3130**	420	9.3272	65.13	0.15	**9.3130**	28.27	0
s500_4	22.4291	**9.2434**	245	9.2904	16.78	0.50	9.2448	46.39	0.02
s500_4	24.3661	**9.1618**	268	9.1729	77.63	0.12	9.1819	16.55	0.21
s500_4	26.3946	**9.1213**	85	**9.1213**	74.83	0	**9.1213**	13.20	0
s500_5	24.0793	**9.3236**	167	9.3526	32.06	0.31	**9.3236**	17.63	0
s500_5	26.2146	**9.2547**	288	9.2626	60.75	0.08	9.2761	9.20	0.23
s500_5	28.1745	**9.2079**	86	9.2218	86.63	0.15	**9.2079**	19.70	0
Average			214.73		67.81	0.15		21.59	0.05

Table 3
Computational performance, dataset 1, sizes 800 and 1000.

Problem	ε	GRASP		LARAC	
		z^G	t^G	z^L	t^L
s800_1	26.1298	9.5908	82.25	9.5892	78.22
s800_1	27.0324	9.5538	267.28	9.5241	45.81
s800_1	28.9135	9.4989	175.72	9.5061	43.20
s800_2	26.8938	9.5299	157.19	9.5080	62.44
s800_2	28.1132	9.4862	447.91	9.4463	49.33
s800_2	29.5640	9.4128	270.72	9.4128	61.77
s800_3	25.9320	9.6300	34.36	9.4708	38.77
s800_3	26.6025	9.4765	59.72	9.4319	64.91
s800_3	27.3099	9.4251	138.17	9.4251	36.98
s800_4	25.4831	9.6993	79.72	9.5671	37.31
s800_4	27.3945	9.5360	354.56	9.4773	54.78
s800_4	29.0262	9.4666	242.36	9.4434	94.17
s800_5	25.6175	9.5348	162.31	9.5152	39.11
s800_5	27.1858	9.4473	409.55	9.4504	65.11
s800_5	29.2019	9.4379	263.47	9.4323	37.08
Average			209.68		53.92
s1000_1	28.2056	9.7784	70.91	9.6809	241.41
s1000_1	29.4082	9.6348	306.22	9.6348	356.70
s1000_1	30.8361	9.6845	417.31	9.6213	105.75
s1000_2	26.4047	9.7778	32.94	9.6503	195.69
s1000_2	27.8202	9.6123	228.22	9.6060	46.06
s1000_2	29.4901	9.5746	437.09	9.5544	60.69
s1000_3	26.4553	9.6566	55.80	9.6324	167.17
s1000_3	28.1821	9.5486	277.42	9.5403	75.97
s1000_3	30.5749	9.5090	507.58	9.5073	81.61
s1000_4	27.0251	9.5889	96.13	9.5669	116.09
s1000_4	28.6249	9.5248	281.13	9.5158	363.91
s1000_4	30.2099	9.5094	388.67	9.4978	117.36
s1000_5	26.6755	9.5559	93.34	9.5424	143.34
s1000_5	28.1603	9.4971	235.88	9.4964	87.75
s1000_5	29.6899	9.4665	426.25	9.4666	117.09
Average			256.99		151.77

Table 5
Computational performance, dataset 2, sizes 750 and 1000.

Problem	p	ε	GRASP		LARAC	
			z^G	t^G	z^L	t^L
pm750	10	40.2299	197.4405	80.23	188.7478	21.27
pm750	10	41.963	183.0916	101.41	180.0362	26.55
pm750	10	43.6962	176.7785	118.64	177.1955	27.11
pm750	10	54.4293	175.4104	141.61	175.4104	28.53
pm750	20	22.1873	106.4569	921.00	105.4022	43.00
pm750	20	23.5294	104.4218	1206.64	104.2248	84.70
pm750	20	24.8716	103.232	1498.77	102.756	45.34
pm750	20	26.2138	102.1501	1413.77	101.5164	53.36
Average				685.25		41.23
pm1000	10	41.449	202.399	228.52	204.5442	40.16
pm1000	10	43.2836	188.2565	271.78	190.1617	39.56
pm1000	10	45.1181	183.9809	292.20	183.1855	49.64
pm1000	10	46.9526	181.4904	313.06	181.6898	43.38
pm1000	20	23.6156	109.4082	1846.66	107.9114	153.22
pm1000	20	24.7519	106.2316	2348.75	106.1406	100.63
pm1000	20	25.8882	105.247	2905.80	104.8802	114.94
pm1000	20	27.0245	104.5349	3068.34	104.0866	98.25
Average				1409.38		79.97

parameter ε. Better yet, a method of multi-objective integer programming should be used that guarantees obtaining the entirety of the efficient boundary, or a good approximation of it. The form of problem (9) directly suggests applying the ε-constraint method (see [9,11]). Two matters must be resolved in order to apply this method: how to modify the value of ε, and how to solve the resulting subproblems, which, in this case, are CONPMP. Regarding the first matter, if the actual solution produces objective values $\overline{f_1}$ and $\overline{f_2}$, it suffices to take $\varepsilon = \overline{f_2} - 1$ to obtain the next solution. As regards solving the CONPMP problems, the previous computational results suggest that the LARAC method is the most efficient.

By way of illustration, in Fig. 3 and Table 1, the result of applying the ε-constraint method to problem pm250, with $p = 20$ can be seen. The CONPMP subproblems have been solved using LARAC and the H2 local search method. The complete procedure obtained 36 solutions, and took 94.73 s. Given that the subproblems have been solved heuristically, it cannot be guaranteed that all the solutions obtained will be non-dominated. In fact, out of the 36 solutions, those numbered 20 and 35 are dominated and should be

Table 4
Computational performance, dataset 2, sizes 250 and 500. Boldface values are the optimum.

Problem	p	ε	XPRESS		GRASP			LARAC		
			z	t	z^G	t^G	d^G	z^L	t^L	d^L
pm250	10	32.72	**157.07**	106.00	157.38	4.89	0.19	**157.07**	1.83	0
pm250	10	35.13	**152.55**	99.95	152.94	5.88	0.25	152.94	2.33	0.25
pm250	10	37.54	**150.77**	110.48	**150.77**	7.13	0	**150.77**	2.00	0
pm250	10	39.95	**149.80**	124.92	**149.80**	7.16	0	**149.80**	1.95	0
pm250	20	14.90	**86.27**	33.47	86.33	24.86	0.07	**86.27**	5.27	0
pm250	20	16.55	**84.95**	48.69	85.463	32.50	0.60	85.24	3.09	0.34
pm250	20	18.20	**83.73**	71.58	83.86	39.22	0.15	83.87	2.95	0.16
pm250	20	19.85	**82.93**	61.47	**82.93**	43.75	0	**82.93**	3.39	0
Average				82.06		20.67	0.15		2.97	0.09
pm500	10	37.09	**186.87**	3553.23	**186.87**	23.05	0	187.19	6.59	0.16
pm500	10	38.74	**174.91**	718.20	175.55	26.06	0.36	175.44	8.03	0.30
pm500	10	40.39	**172.06**	562.38	173.42	31.86	0.79	**172.06**	5.28	0
pm500	10	42.04	**170.67**	1261.66	**170.67**	31.89	0	**170.67**	5.56	0
pm500	20	18.82	**99.85**	278.47	102.49	93.33	2.64	**99.85**	15.84	0
pm500	20	20.31	**96.67**	367.22	**96.67**	153.17	0	**96.67**	15.47	0
pm500	20	21.80	**94.78**	362.25	95.78	213.01	1.05	**94.78**	20.73	0
pm500	20	23.28	**93.91**	432.9	94.77	235.13	0.90	94.33	23.19	0.44
Average				942.03		100.94	0.72		12.58	0.11

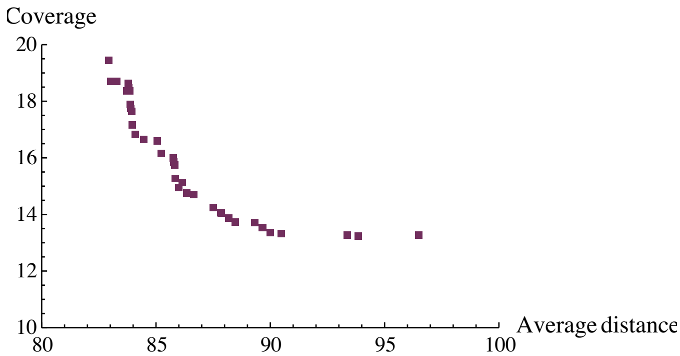

Fig. 3. Efficient frontier for problem pm250, $p = 20$.

eliminated. In spite of this, the tradeoff between both objectives can be observed, either in Table 1 or in the corresponding Fig. 3. This information could be valuable for a decision-maker faced with a location problem for which the objectives of distance and coverage are important.

7. Summary and conclusions

In this paper, different heuristic procedures are developed for the p-median problem with an additional coverage constraint (CONPMP). Local search algorithms for CONPMP, for constructing and improving feasible solutions, are obtained as modified versions of the local interchange search algorithms for the PMP. Based on these local search procedures, a GRASP multistart heuristic for the CONPMP problem is developed.

On the other hand, starting from a Lagrangean relaxation of the CONPMP, which is based on the solution of subproblems of type PMP, the LARAC algorithm is designed as an extension of the well-known Handler and Zang algorithm for the restricted shortest path problem. In addition, the local search improvement heuristic is applied to the solution obtained by the LARAC algorithm, resulting in a very efficient heuristic, which, in general, yields better solutions than the GRASP algorithm.

In order to compare the GRASP and LARAC methods, two data sets have been used, the first randomly generated and the second extracted from the data set of the 3100 counties in the contiguous US, with a total number of 92 CONPMP problems. In addition, the lower dimensional problems have been solved up to optimality with XPRESS-Optimizer. After extensive computational experiments with the 92 problems, it can be concluded that both the GRASP and LARAC methods are efficient for solving CONPMP, although LARAC has turned out to be somewhat more efficient than GRASP, both because of the quality of the solutions as well as for the computation time.

Acknowledgement

The authors thank the anonymous referees for their useful suggestions and comments that improved the paper, and FICO for providing the XPRESS Optimizer application, which has been used to obtain the computational results. The second author is supported by the Consejería de Educación de la Junta de Castilla y León (Spain) under project VA056A09 and by the Ministerio de Ciencia e Innovación under project ECO2008-02358.

Appendix A

Tables 2–5.

References

[1] V. Aggarwal, A Lagrangean-relaxation method for the constrained assignment problem, Computers and Operation Research 12 (1985) 97–106.
[2] V. Aggarwal, Y.P. Aneja, K.P.K. Nair, Minimal spanning tree subject to a side constraint, Computers and Operations Research 9 (1982) 287–296.
[3] E. Alba, E. Domínguez, Comparative analysis of modern optimization tools for the p-median problem, Statistics and Computing 16 (2006) 251–260.
[4] P. Avella, M. Boccia, A. Sforza, I. Vasilev, An effective heuristic for large-scale capacitated facility location problems, Journal of Heuristics 15 (2009) 597–615.
[5] P. Avella, A. Sassano, I. Vasilev, Computational study of large-scale p-median problems, Mathematical Programming 109 (1) (2007) 89–114.
[6] O. Briant, D. Naddef, The optimal diversity management problem, Operations Research 52 (2004) 515–526.
[7] A. Caprara, M. Fischetti, P. Toth, A heuristic method for the set covering problem, Operations Research 47 (5) (1999) 730–743.
[8] A. Caprara, M. Monaci, P. Toth, P.L. Guida, A Lagrangian heuristic algorithm for a real-world train timetabling problem, Discrete Applied Mathematics 154 (2006) 738–753.
[9] J.L. Cohon, Multiobjective Programming and Planning, Academic Press, 1978.
[10] M.S. Daskin, Network and Discrete Location, Models, Algorithms and Applications, John Wiley and Sons, 1995.
[11] M. Ehrgott, Multicriteria Optimization, Springer Verlag, 2005.
[12] FICO Team XPRESS Optimization, XPRESS-MP Optimizer Reference Manual, Fair Isaac Corporation, Leam House, 64 Trinity Street, Leamington Spa, Warwickshire CV32 5YN, UK, 2010.
[13] B.L. Fox, D.M. Landi, Searching for the multiplier in one-constraint optimization problems, Operations Research 18 (1970) 253–262.
[14] M. Guignard, Lagrangean relaxation, TOP 11 (2003) 151–228.
[15] G. Handler, I. Zang, A dual algorithm for the constrained path problem, Networks 10 (1980) 293–310.
[16] P. Hansen, B. Jaumard, Cluster analysis and mathematical programming, Mathematical Programming 79 (1997) 191–215.
[17] A. Jüttner, On resource constrained optimization problems, in: 4th Japanese–Hungarian Symposium on Discrete Mathematics and Its Applications, 2005, pp. 1–10.
[18] O. Kariv, L. Hakimi, An algorithmic approach to network location problems. Part II: the p-medians, SIAM Journal of Applied Mathematics 37 (1979) 539–560.
[19] M. Labbe, D. Peeters, J.F. Thisse, Location on networks, in: M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemheuser (Eds.), Network Routing, Handbooks in Operations Research and Management Science, vol. 8, North-Holland, 1995, pp. 551–617. Chapter 7.
[20] N. Mladenovic, J. Brimberg, P. Hansen, J.A. Moreno-Pérez, The p-median problem: a survey of metaheuristic approaches, European Journal of Operational Research 179 (2007) 927–939.
[21] J.M. Mulvey, H.P. Crowder, Cluster analysis: an application of Lagrangean relaxation, Management Science 25 (1979) 329–340.
[22] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, Wiley and Sons, 1988.
[23] L.K. Nozick, The fixed charge facility location problem with coverage restrictions, Transportation Research Part E 37 (2001) 281–296.
[24] J. Reese, Solution methods for the p-median problem: an annotated bibliography, Networks 48 (2006) 125–142.
[25] M.G.C. Resende, J.L. González Velarde, GRASP: greedy randomized adaptive search procedures. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial 19 (2003) 61–76.
[26] M.G.C. Resende, C. Ribeiro, Greedy randomized adaptive search procedures, in: F. Glover, G.A. Kochenberger (Eds.), Handbook of Metaheuristics, Kluwer Academic Publisher, 2003, pp. 219–249.
[27] M.G.C. Resende, R.F. Werneck, On the implementation of a swap-based local search procedure for the p-median problem, in: Richard E. Ladner (Ed.), Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments (ALENEX'2003), SIAM, Philadelphia, 2003, pp. 119–127.
[28] M.G.C. Resende, R.F. Werneck, A hybrid heuristc for the p-median problem, Journal of Heuristics 10 (2004) 59–88.
[29] M. Resende, R.F. Werneck, A hybrid multistart heuristic for the uncapacitated facility location problem, European Journal of Operational Research 174 (2006) 54–68.
[30] M.G.C. Resende, R.F. Werneck, A fast swap-based local search procedure for location problems, Annals of Operations Research 150 (2007) 205–230.
[31] J. Sáez, Fixed charge transportation problems: a new heuristic approach based on Lagrangean relaxation and the solving of core problems, Annals of Operations Research 172 (2009) 45–69.
[32] M.B. Teitz, P. Bart, Heuristic methods for estimating the generalized vertex median of a weighted graph, Operations Research 16 (1968) 955–961.
[33] R. Whitaker, A fast algorithm for the greedy interchange of large-scale clustering and median location problems, INFOR 21 (1983) 95–108.
[34] J.G. Villegas, F. Palacios, A.L. Medaglia, Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example, Annals of Operations Research 147 (2006) 109–141.
[35] Y. Xiao, K. Thulasiraman, G. Xue, A. Jüttner, The constrained shortest path problem: algorithmic approaches and an algebraic study with generalization, AKCE International Journal of Graphs and Combinatorics 2 (2005) 63–86.