



A flexible model and efficient solution strategies for discrete location problems

Alfredo Marín^a, Stefan Nickel^{b,d,*}, Justo Puerto^c, Sebastian Velten^b

^a Universidad de Murcia, Departamento de Estadística e Investigación Operativa, 30100 Murcia, Spain

^b Saarland University, Chair of Operations Research and Logistics, Faculty of Law and Economics, P.O. Box 15 11 50, 66041 Saarbrücken, Germany

^c University of Seville, Faculty of Mathematics, 41012 Seville, Spain

^d Fraunhofer Institute for Industrial Mathematics (ITWM), Germany

ARTICLE INFO

Article history:

Received 5 April 2007

Received in revised form 27 February 2008

Accepted 9 March 2008

Available online 17 April 2008

Keywords:

Discrete location

Discrete ordered median problem

Valid inequalities

Variable fixing

ABSTRACT

Flexible discrete location problems are a generalization of most classical discrete locations problems like p -median or p -center problems. They can be modeled by using so-called ordered median functions. These functions multiply a weight to the cost of fulfilling the demand of a customer, which depends on the position of that cost relative to the costs of fulfilling the demand of other customers.

In this paper a covering type of model for the discrete ordered median problem is presented. For the solution of this model two sets of valid inequalities, which reduces the number of binary variables tremendously, and several variable fixing strategies are identified. Based on these concepts a specialized branch & cut procedure is proposed and extensive computational results are reported.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Location Analysis is one of the most active fields in Operations Research. Especially, discrete location problems have been widely studied due to their importance in practical applications and also due to their interesting structural properties (see for example [1–3] and references therein). Discrete location problems typically involve a finite set of sites at which facilities can be located, and a finite set of clients, whose demands for service or goods have to be fulfilled by these facilities. The simplest and well studied discrete location problems are the discrete p -Median Problem and the Uncapacitated (and Capacitated) Facility Location Problem (see [4,5]). Evidently many extensions of these basic location problems have been developed. The extensions range from capacity restrictions, over multi-echelon structures (see [6]), to time dynamic models and models with choice of facilities (see [7]). Recently several articles have been published addressing strategic supply chain decisions in the context of location problems (see [8] and references therein). This development led to a highly flexible and general framework of location models in terms of side constraints.

Another important aspect of a location model is the right choice of the objective function, and in most classical location models the objective function is the main differentiator. Therefore, a great variety of objective functions has been considered. The median objective is to minimize the sum of the costs of fulfilling all demand requests from clients. The center objective is to minimize the maximum cost of fulfilling the demand of a client, from amongst the sites chosen, over all clients. The centdian objective is a convex combination of median and center objectives; it aims to keep both the average cost behavior as well as the highest cost in balance. Despite the fact that all three objectives (and some more) are frequently encountered

* Corresponding author at: Saarland University, Chair of Operations Research and Logistics, Faculty of Law and Economics, P.O. Box 15 11 50, 66041 Saarbrücken, Germany. Tel.: +49 0 681 302 64738; fax: +49 0 681 302 64737.

E-mail addresses: amarin@um.es (A. Marín), s.nickel@orl.uni-saarland.de (S. Nickel), puerto@us.es (J. Puerto), s.velten@orl.uni-saarland.de (S. Velten).

in the literature (see for example [1,2]), not much has been done in the direction of a unified framework for handling all these objectives.

The increasing need for discrete location models in strategic supply chain planning, see for example [8], has made it necessary to develop new and flexible location models. To that end, [9] introduced a new type of objective function which generalizes the most popular objective functions mentioned above. This objective function applies a penalty to the cost of fulfilling the demand of a client which is dependent on the position of that cost relative to the costs of supplying other customers. For example, a different penalty might be applied if the cost of supplying the client was the 5th-most expensive cost rather than the 2nd-most expensive. It is even possible to neglect some customers by assigning a zero penalty. This adds a “sorting”-problem to the underlying facility location problem, making formulation and solution much more challenging.

For planar and network location problems the generalized model was studied in [10–12]. The research in this area even led to a recent monograph, see [13]. In [9], a formulation of the discrete case, later called the Discrete Ordered Median Problem, is discussed. Starting with a nonlinear formulation, several linearizations were developed. Structural results as well as a specially tailored branch and bound procedure are presented in [14]. Moreover, modeling and solution approaches for special cases in which the objective can be represented as a sum of k -centrum objectives are given in [15].

However, none of these approaches leads to satisfactory results concerning solution times of even medium size instances. Therefore, we provide a different formulation which is based in part on the idea of sorting radii already used in [16] (see also [17]) and, recently, by [18] for modeling the p -center problem. This new formulation will be the basis of our specialized branch & cut approach. In this context, we use a heuristic approach developed in [19], to get good upper bounds. This heuristic is based on Variable Neighborhood Search, first introduced in [23], and by employing this heuristic, high quality heuristic solutions for up to 900 possible locations can be obtained.

The rest of the paper is organized as follows. First we will recall the Discrete Ordered Median Problem formally and give a new mathematical programming formulation. In Section 3 we strengthen the initial formulation by using variable fixing and by adding valid inequalities. These results are put together in Section 4 to get a specialized Branch & Cut approach. In Section 5 we will show the efficiency of the described approach by reporting on extensive numerical experiments. The paper ends with some conclusions and an outlook to future research.

2. The Discrete Ordered Median Problem

In order to introduce the *Discrete Ordered Median Problem* (DOMP) formally, we define a set V of M discrete locations. These locations represent clients as well as potential plant locations.

Moreover, let $C = (c_{ij})$ ($i, j = 1, \dots, M$) be a non-negative $M \times M$ cost matrix, where c_{ij} denotes the cost of satisfying the total demand of client i from a plant at location j . Thereby, we assume that $c_{ii} = 0$ ($\forall i = 1, \dots, M$). This property of C is called *free self-service* (FSS) and will be needed for some of the subsequent results. For $i \neq j$ we assume $c_{ij} > 0$.

Let N with $1 \leq N \leq M - 1$ be the number of new plants which have to be located at the candidate sites. Then the costs for satisfying the demand of the respective clients, given a feasible solution $X \subset V$ with $|X| = N$, can be represented by the following vector

$$c(X) := (c_1(X), \dots, c_M(X)) \quad \text{with } c_i(X) = \min_{j \in X} \{c_{ij}\} \quad \forall i \in V.$$

However, due to the desired flexibility, $c(X)$ cannot directly be used to define the objective function of the DOMP. Instead, consider a permutation σ_X on $\{1, \dots, M\}$ for which the inequalities

$$c_{\sigma_X(1)}(X) \leq c_{\sigma_X(2)}(X) \leq \dots \leq c_{\sigma_X(M)}(X)$$

hold. Using this permutation we define the *sorted cost vector* $c_{\leq}(X)$ corresponding to a feasible solution X as follows:

$$c_{\leq}(X) := (c_{\sigma_X(1)}(X), \dots, c_{\sigma_X(M)}(X))$$

or for short

$$c_{\leq}(X) := (c_{(1)}(X), \dots, c_{(M)}(X)).$$

Furthermore, let $\lambda = (\lambda_1, \dots, \lambda_M)$ be an M -dimensional vector, with $\lambda_i \geq 0$ ($\forall i = 1, \dots, M$) representing a weight on the i -th lowest component of the cost vector $c(X)$. With respect to these assumptions the DOMP is defined as:

$$\min_{\substack{X \subset V \\ |X|=N}} f_{\lambda}(X), \tag{1}$$

with

$$f_{\lambda}(X) = \sum_{i=1}^M c_{(i)}(X) \cdot \lambda_i. \tag{2}$$

The function $f_{\lambda}(X)$ is called *ordered median function* (omf). An example illustrating the structure of the DOMP and the calculation of the ordered median function is given below.

Table 1
Modeling possibilities

λ	$f_\lambda(X)$	Meaning
$(1, \dots, 1)$	$\sum_{i=1}^M c_i(X)$	N -median
$(0, \dots, 0, 1)$	$\max_{1 \leq i \leq M} c_i(X)$	N -center
$(\alpha, \dots, \alpha, 1) \alpha \in [0, 1]$	$\alpha \cdot \sum_{i=1}^M c_i(X) + (1 - \alpha) \cdot \max_{1 \leq i \leq M} c_i(X)$	α -centdian
$(0, \dots, 0, \underbrace{1, \dots, 1}_k)$	$\sum_{i=M-k+1}^M c_{(i)}(X)$	k -centrum

Example 1. Let $V = \{1, \dots, 5\}$ and assume that $N = 2$ plants have to be located. Moreover, let the cost matrix C be as follows:

$$C = \begin{pmatrix} 0 & 6 & 5 & 4 & 8 \\ 4 & 0 & 8 & 5 & 7 \\ 6 & 2 & 0 & 8 & 5 \\ 6 & 5 & 4 & 0 & 1 \\ 5 & 5 & 2 & 6 & 0 \end{pmatrix}.$$

With $\lambda = (2, 0, 1, 1, 0)$, the unique optimal solution of this problem instance is $X = \{2, 5\}$. Therefore, the demands of locations 1 and 3 are satisfied by plant 2 whereas the demand of location 4 is satisfied by plant 5. Hence, $c_\leq(X) = (0, 0, 1, 2, 6)$ and

$$f_\lambda(X) = 0 \cdot 2 + 0 \cdot 0 + 1 \cdot 1 + 2 \cdot 1 + 6 \cdot 0 = 3.$$

Note that by using appropriate values for λ , nearly all classical discrete facility location problems can be modeled by the above definition. In addition, a wide range of new and interesting problems can be derived. Some of these modeling possibilities are given in Table 1. For a more extensive list the interested reader is referred to [13,20].

Since the DOMP contains the discrete N -median problem, which is \mathcal{NP} -hard (see [21]), as a special instance, the DOMP is \mathcal{NP} -hard, too. Moreover, due to the sorting process in the objective function, the above formulation of the DOMP is non-linear. But different formulations have also been proposed which are quadratic (see [9]) or even linear (see [9] and [14]). Nevertheless, none of these approaches provides satisfactory results concerning solution times for large problem instances. Therefore, we will develop a different formulation for the DOMP which is based in part on the idea of [18] for modeling the N -center problem.

In order to introduce this new formulation we first define G as the number of different non-zero elements of the cost matrix C . Hence, we can order different values of C in increasing sequence:

$$c_{(0)} := 0 < c_{(1)} < c_{(2)} < \dots < c_{(G)} := \max_{1 \leq i, j \leq M} \{c_{ij}\}.$$

Given a feasible solution (i.e. $X \subset V, |X| = N$) we can use this ordering to perform a sorting process of allocation costs. Thereby, we omit allocation costs of plant locations, because they are equal to zero anyway. Consequently, we only have to sort $M - N$ cost elements. This can be done by the following variables ($j = 1, \dots, M - N$ and $k = 1, \dots, G$):

$$x_{jk} := \begin{cases} 1 & \text{if the } j\text{-th smallest allocation cost is at least } c_{(k)}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

With respect to this definition the j -th smallest cost element is equal to $c_{(k)}$ if and only if $x_{jk} = 1$ and $x_{j, k+1} = 0$. Using these variables the ordered median function can be reformulated as follows:

Let X be feasible solution, i.e. $|X| = N$. Then, for $j = 1, \dots, N, C_{(j)}(X) = 0$ because of free self service. On the other hand, if $i \in \{N + 1, \dots, M\}$, there exists a $1 \leq k' \leq G$ so that $C_{(j)}(X) = c_{(k')}$. Therefore, by choosing values for the x_{jk} -variables according to their definitions, it holds:

$$x_{jk} = 1 \quad \forall k = 1, \dots, k' \quad \text{and} \quad x_{jk} = 0 \quad \forall k = k' + 1, \dots, G$$

Hence, by using a telescopic sum, we get

$$C_{(j)}(X) = \sum_{k=1}^G (c_{(k)} - c_{(k-1)}) \cdot x_{jk}$$

and the ordered median function can be written as:

$$f_\lambda = \sum_{j=1}^M \lambda_j \cdot C_{(j)}(X) = \sum_{j=1}^{M-N} \sum_{k=1}^G \lambda_{N+j} \cdot (c_{(k)} - c_{(k-1)}) \cdot x_{jk}. \tag{4}$$

The role of x_{jk} -variables in the sorting process is further demonstrated by the following example.

Example 2. We consider the same problem instance as in Example 1. Then, since the sorted cost vector of matrix C is $(0, 1, 2, 4, 5, 6, 7, 8)$, the values of the x_{jk} -variables corresponding to the optimal solution $X = \{2, 5\}$ are given by:

$$\begin{aligned} x_{11} &= 1; & x_{12} &= x_{13} = x_{14} = x_{15} = x_{16} = x_{17} = 0 \text{ (location 4)} \\ x_{21} &= 1; & x_{22} &= 1; & x_{23} &= x_{24} = x_{25} = x_{26} = x_{27} = 0 \text{ (location 3)} \\ x_{31} &= 1; & x_{32} &= 1; & x_{33} &= 1; & x_{34} &= 1; & x_{35} &= 1; & x_{36} &= x_{37} = 0 \text{ (location 1)}. \end{aligned}$$

Furthermore, the objective function value can be calculated by

$$\begin{aligned} \sum_{j=1}^{M-N} \sum_{k=1}^{G_j} \lambda_{N+j} \cdot (c_{(k)} - c_{(k-1)}) \cdot x_{jk} &= \lambda_3 \cdot (x_{11} + x_{12} + 2 \cdot x_{13} + x_{14} + x_{15} + x_{16} + x_{17}) \\ &\quad + \lambda_4 \cdot (x_{21} + x_{22} + 2 \cdot x_{23} + x_{24} + x_{25} + x_{26} + x_{27}) \\ &\quad + \lambda_5 \cdot (x_{31} + x_{32} + 2 \cdot x_{33} + x_{34} + x_{35} + x_{36} + x_{37}) \\ &= 1 \cdot 1 + 1 \cdot (1 + 1) + 0 \cdot (1 + 1 + 2 + 1 + 1) = 3. \end{aligned}$$

Next, we are interested in variables reflecting the location decisions of a feasible solution X . In this regard we define for each row i ($i = 1, \dots, M$) of the cost matrix C , G_i as the number of different non-zero elements in this row. Thus, we obtain, as for the whole matrix C , the ordering

$$c_{(0)}^i := 0 < c_{(1)}^i < \dots < c_{(G_i)}^i := \max_{j=1, \dots, M} \{c_{ij}\}$$

of the cost elements of row i . Then we can define the following variables ($i = 1, \dots, M$ and $k = 1, \dots, G_i$):

$$z_{ik} := \begin{cases} 1 & \text{if the allocation cost for location } i \text{ is at least } c_{(k)}^i, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

In addition, for convenience we define $z_{iG_i+1} := 0$, for all $i = 1, \dots, M$. These definitions imply that, if $z_{ik} = 1$, no plants are established at locations j with $c_{ij} < c_{(k)}^i$. Moreover, notice that a plant is opened at location i if and only if $z_{ik} = 0$ for all $k = 1, \dots, G_i$ which is due to the fact that the allocation cost of a plant is equal to zero. An example illustrating the meaning of these variables is given below.

Example 3. We reconsider the data of Example 1 and recall that $X = \{2, 5\}$ is the optimal solution. Furthermore, consider the first row in matrix C (i.e. $(0 \ 6 \ 5 \ 4 \ 8)$). Then the variables z_{11}, z_{12}, z_{13} and z_{14} are associated with the sorted non-zero entries 4, 5, 6 and 8, and they assume, with respect to the optimal solution, the values

$$z_{11} = 1, \quad z_{12} = 1, \quad z_{13} = 1, \quad z_{14} = 0,$$

since location 1 is allocated to the plant at location 2 with a cost of 6. On the other hand, the variables z_{51}, z_{52} and z_{53} are all equal to 0, because there is a plant at location 5.

From the definition of z_{ik} it follows that a plant is opened at location i if and only if $z_{i1} = 0$, which is equivalent to the condition $1 - z_{i1} = 1$. Furthermore, one can observe that $z_{ik} - z_{ik+1} = 1$ if and only if the client at location i is allocated to an open plant at cost $c_{(k)}^i$. Hence, we can write the cost for allocating a client at location i to a plant, like in the objective function, as

$$\sum_{k=1}^{G_i} (c_{(k)}^i - c_{(k-1)}^i) \cdot z_{ik} = \sum_{k=1}^{G_i-1} c_{(k)}^i \cdot (z_{ik} - z_{ik+1}). \tag{6}$$

Therefore, the location and allocation decisions can be totally modeled by these variables whereas the following constraints ensure that only correct values will be assumed:

$$z_{ik} \geq 1 - \sum_{\substack{j=1, \dots, M \\ c_{ij} < c_{(k)}^i}} (1 - z_{j1}) \quad \forall i = 1, \dots, M, k = 2, \dots, G_i. \tag{7}$$

That these constraints indeed accomplish the desired task can be seen as follows. On the one hand, if no plant can be reached by customer i at a cost less than $c_{(k)}^i$ then the constraint for pair (i, k) becomes $z_{ik} \geq 1$. On the other hand, if a plant can be reached at a cost less than $c_{(k)}^i$ then it holds

$$\sum_{\substack{j=1, \dots, M \\ c_{ij} < c_{(k)}^i}} (1 - z_{j1}) \geq 1,$$

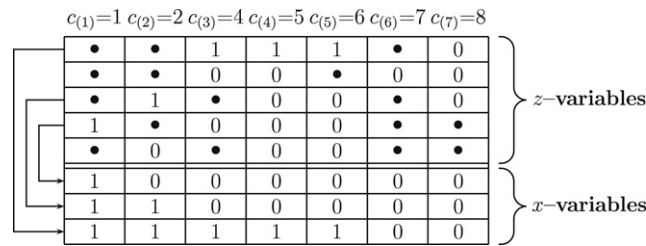


Fig. 1. Relationship between z- and x-variables.

and for (i, k) the constraint is redundant. Nonetheless, we still need to guarantee that exactly N plants will be opened among the M possibilities. This can be ensured by the constraint

$$\sum_{i=1}^M z_{i1} = M - N \tag{8}$$

since $z_{i1} = 0$ if and only if there is a plant at location i .

Up to now we have defined two kinds of variables where, for a feasible solution, the x_{jk} can be used to sort allocation costs and the z_{ik} are capable of modeling the location and allocation decisions. However, to obtain the new formulation of the DOMP we need to link these variables. This can be done by enforcing the number of locations with allocation costs at least $c_{(k)}$, once represented by x_{jk} - and once by z_{ik} -variables, to be equal:

$$\sum_{j=1}^{M-N} x_{jk} = \sum_{\substack{i=1, \dots, M \\ i_k \leq G_i}} z_{ik} \quad \forall k = 1, \dots, G \tag{9}$$

where

$$i_k := \begin{cases} \min\{s : c_{(s)}^i \geq c_{(k)}\} & \text{if } c_{(k)} \leq c_{(G_i)}^i, \\ G_i + 1 & \text{otherwise.} \end{cases} \tag{10}$$

Observe that these constraints ensure that if $z_{i i_k} = 1$ then, given the definition of x_{jk} -variables, there exists at least one j so that $x_{jk} = 1$ for all k with $c_{(k)} \leq c_{(i_k)}^i$. Because of this, it is guaranteed that x_{jk} -variables take the correct values, according to the location and allocation decision induced by z_{ik} -variables, even if cost elements are missing in the respective rows.

Therefore, values of x_{jk} -variables always represent sorted costs corresponding to the location and allocation decision of a solution. Furthermore, by using (4) as objective, the ordered median function is minimized.

The relationship between these variables is illustrated by the following example:

Example 4. Given again the problem data of Example 1, the graphic provided in Fig. 1 demonstrates how the z- and x-variables are linked to each another. Thereby, values for the optimal solution $X = \{2, 5\}$ have been used.

At last, to complete the new formulation of the DOMP, we need to impose the following group of sorting constraints on the x_{jk} -variables:

$$x_{jk} \geq x_{j-1k} \quad j = 2, \dots, M - N; k = 1, \dots, G. \tag{11}$$

Summarizing the constraints (7)–(9) and (11) and the objective function (4), the DOMP can be formulated as (DOMP)

$$\text{Min} \sum_{j=1}^{M-N} \sum_{k=1}^G \lambda_{N+j} \cdot (c_{(k)} - c_{(k-1)}) \cdot x_{jk} \tag{4}$$

$$\text{s.t. } z_{ik} \geq 1 - \sum_{\substack{j=1, \dots, M \\ c_{(j)} < c_{(k)}}} (1 - z_{j1}) \quad i = 1, \dots, M; k = 2, \dots, G_i \tag{7}$$

$$\sum_{i=1}^M z_{i1} = M - N \tag{8}$$

$$\sum_{j=1}^{M-N} x_{jk} = \sum_{\substack{i=1, \dots, M \\ i_k \leq G_i}} z_{ik} \quad k = 1, \dots, G \tag{9}$$

$$x_{jk} \geq x_{j-1k} \quad j = 2, \dots, M - N; k = 1, \dots, G \tag{11}$$

$$x_{jk} \in \{0, 1\} \quad j = 1, \dots, M - N; k = 1, \dots, G \tag{12}$$

$$z_{ik} \in \{0, 1\} \quad i = 1, \dots, M; k = 1, \dots, G_i. \tag{13}$$

Remark 1. Note that there may be feasible, and even optimal, solutions for the model (DOMP) which, by definition, are not feasible for the problem. For example, in the case of some λ -values being 0, a z_{ik} -variable may be equal to 1 even if $z_{ik'}$, $k' < k$, is already equal to 0. However, for each of these solutions a feasible solution for the problem can be derived easily by allocating locations according to location decisions, i.e. the values of the z_{i1} -variables.

Since the proposed formulation contains $O(M^3)$ binary variables and $O(M^3)$ constraints, fast solution times for large problem instances, using standard software-tools, are very unlikely (see Section 5). Therefore, we propose a specialized branch & cut procedure and several variable fixing strategies, exploiting the special structure of the presented model.

Remark 2. With respect to the definition of the x_{jk} - and z_{ik} -variables there are two more groups of sorting constraints which have to be satisfied. Namely,

- If the j -th smallest positive allocation cost is at least $c_{(k)}$, it is also at least $c_{(k-1)}$:

$$x_{jk-1} \geq x_{jk} \quad j = 1, \dots, M - N; \quad k = 2, \dots, G \tag{14}$$

- If the allocation cost of location i is at least $c_{(k)}^i$, it is also at least $c_{(k-1)}^i$:

$$z_{ik-1} \geq z_{ik} \quad i = 1, \dots, M; \quad k = 2, \dots, G_i. \tag{15}$$

If the model given by (DOMP) is optimized, these constraints are fulfilled automatically. However, they are needed for some of the variables fixing strategies detailed in the next section.

3. Variable fixing and valid inequalities

In this section we describe a number of variable fixing possibilities which are useful in the overall solution process. Moreover, a group of valid inequalities is introduced which leads to a specialized branch & cut procedure for solving the DOMP.

First of all, the following proposition states that we only need to enforce z_{i1} -variables ($i = 1, \dots, M$) to be binary:

Proposition 1. In (DOMP), $z_{ik} \in \{0, 1\}$ can be replaced by $0 \leq z_{ik} \leq 1$ for $i = 1, \dots, M$ and $k = 2, \dots, G_i$, since each optimal solution of the relaxed problem leads to an optimal solution of the original problem, too.

Proof. Let (\tilde{x}, \tilde{z}) be an optimal solution of (DOMP) with $z_{ik} \in \{0, 1\}$ replaced by $0 \leq z_{ik} \leq 1$ for $i = 1, \dots, M$ and $k = 2, \dots, G_i$. If all variables in (\tilde{x}, \tilde{z}) are binary, we are done. Otherwise, there are at least two pairs of indices (i', k') and (i'', k'') ($i', i'' \in \{1, \dots, M\}$ and $k', k'' \in \{2, \dots, G_i\}$) so that $0 < \tilde{z}_{i'k'}, \tilde{z}_{i''k''} < 1$. This is because of the Constraints (9) and all \tilde{x}_{jk} -variables being binary. Moreover, for these variables the Constraints (7) reduce $z_{i'k'} \geq 0$, since $z_{i1} \in \{0, 1\}$ ($i = 1, \dots, M$). Hence, fractional variables can be changed to 0 without violating these constraints. However, in order to maintain feasibility, some of the \tilde{x}_{jk} -variables have to be reduced to 0 as well. But obviously, this can be done without violating any constraints and without increasing the objective function value, because $\lambda_{N+j} \cdot (c_{(k)} - c_{(k-1)})$ ($j = 1, \dots, M - N$ and $k = 1, \dots, G$) is always greater than or equal to 0. Therefore, we obtain a new solution which is feasible, optimal and fulfills all binary constraints. \square

Furthermore, some of the x - and z -variables can be fixed to 0 or 1 in a preprocessing step. For some of these variable fixing strategies an upper bound on the optimal objective value is needed, which can be obtained, for example, by the Variable Neighborhood Search (VNS) presented in [13,19].

3.1. Fixing x_{jk} -variables to 1

Following the definition of the x_{jk} -variables, we first observe that $x_{j1} = 1$ for $j = 1, \dots, M - N$. This is true because if a location has to be allocated, the corresponding cost is at least $c_{(1)}$ (i.e. the smallest positive element in C). Now assume, for fixed j and k with $1 \leq j \leq M - N$ and $2 \leq k \leq G$, that $x_{jk} = 0$. Moreover, we define a set L as follows:

$$L := \left\{ i \in \{1, \dots, M\} : l_k^i = 1 \right\}.$$

Then, a solution with $x_{jk} = 0$ can only be feasible if it holds:

$$M - N - j \geq \sum_{j'=1}^{M-N} x_{j'k} = \sum_{\substack{i=1, \dots, M \\ l_k^i \leq G_i}} z_{ij_k} \geq |L| - N. \tag{16}$$

Thereby, the first inequality is due to the definition of the x_{jk} -variables:

$$x_{jk} = 0 \Rightarrow x_{j'k} = 0 \quad \forall j' \leq j.$$

Therefore, at most $M - N - j$ $x_{j'k}$ -variables can be equal to 1 if $x_{jk} = 0$. In addition, the second inequality follows from the fact that $|L|$ is the number of locations the smallest allocation costs of which are at least $c_{(k)}$. Hence, in a feasible solution, the

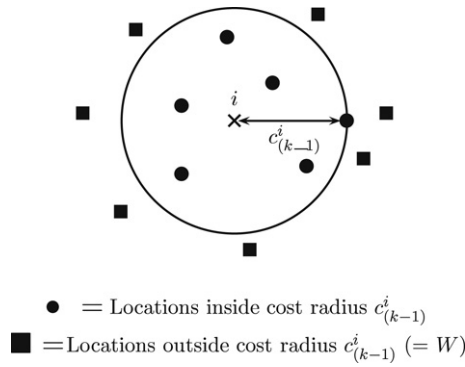


Fig. 2. Illustration of cost radius $c_{(k-1)}^i$ around location i .

number of locations which are allocated with costs at least $c_{(k)}$ has to be at least $|L| - N$, since at N locations a facility could be opened. At last, the equation in the middle follows from (9).

Consequently, if (16) does not hold (i.e. $M - j < |L|$), a solution with $x_{jk} = 0$ can never be feasible. Thus, in this case, x_{jk} has to be equal to 1.

Using this test, each x_{jk} -variable can be checked, whether it has to be equal to 1 or not. However, not all combinations of j and k need to be tested as indicated by the subsequent remark.

Remark 1. If x_{jk} has been fixed to 1 (applying the test of this subsection), then $x_{j'k'} = 1$ for $j' = j + 1, \dots, M - N$ and $k' = 1, \dots, k - 1$, too. This follows directly from the definition of the x_{jk} -variables.

3.2. Fixing x_{jk} -variables to 0

In order to derive a test for fixing some of the x_{jk} -variables to 0, we first assume that $x_{jk} = 1$ (for some fixed j ($1 \leq j \leq M - N$) and k ($1 \leq k \leq G$)). On the one hand, this means that the j -th smallest allocation cost is at least equal to $c_{(k)}$. Therefore, the $(j + 1)$ -th to $(M - N)$ -th smallest allocation costs are also at least $c_{(k)}$, and hence, $x_{j'k'} = 1$ for $j' = j + 1, \dots, M - N$. On the other hand we know that $x_{j'1} = 1$ for all $j' = 1, \dots, j - 1$ (see Section 3.1). Given this information, it is easy to see that the following expression provides a lower bound on the optimal objective value for the case where $x_{jk} = 1$:

$$\underbrace{c_{(1)} \cdot \left(\sum_{l=N+1}^{j-1} \lambda_l \right)}_{(*)} + \underbrace{c_{(k)} \cdot \left(\sum_{l=j}^{M-N} \lambda_l \right)}_{(**)}. \tag{17}$$

- (*) The 1-st to $(j - 1)$ -th smallest allocation costs are at least $c_{(1)}$.
- (**) The j -th to $(M - N)$ -th smallest allocation costs are at least $c_{(k)}$.

This lower bound can now be compared to the upper bound mentioned above or another heuristic. Then, if the lower bound for the case $x_{jk} = 1$ is greater than the upper bound, a solution with $x_{jk} = 1$ can never be optimal. Thus, x_{jk} can be fixed to 0 without losing any optimal solution. Moreover, by the following remark, this test needs not to be executed for all combinations of j and k .

Remark 2. If x_{jk} has been fixed to 0 (applying the test of this subsection), then $x_{j'k'} = 0$ for $j' = 1, \dots, j$ and $k' = k, \dots, G$, too. This follows directly from the definition of the x_{jk} -variables.

3.3. Fixing z_{ik} -variables to 0

As in the previous section, assume that $z_{ik} = 1$ for some fixed $1 \leq i \leq M$ and $1 \leq k \leq G_i$. Then we know, for an associated feasible solution, that no plant (including plant i) is opened within cost radius $c_{(k-1)}^i$ around location i . Otherwise, location i would be allocated to such a plant and z_{ik} would be equal to 0. Consequently, facilities can only be opened at locations lying outside this cost radius. This set of locations we denote by W (see Fig. 2).

If now, on the one hand, the cardinality of W is less than N , it is easy to observe that a solution with $z_{ik} = 1$ can never be feasible, since exactly N facilities have to be located. Moreover, if on the other hand the cardinality of W is greater than or equal to N , we can again derive a lower bound on the optimal objective value for the case $z_{ik} = 1$. For that purpose we define a cost vector $\tilde{c}(W) = (\tilde{c}_1(W), \dots, \tilde{c}_M(W))$ in the following way:

$$\tilde{c}_i(W) = \begin{cases} \min_{j \in W \setminus \{i\}} \{c_{ij}\} & \text{if } i' \in W \\ \min_{j \in W} \{c_{ij}\} & \text{otherwise.} \end{cases} \tag{18}$$

This cost vector contains, for each $i' \in V$, the smallest cost for allocating location i' to a possible plant in W (which is not i' itself). If we now change the N largest entries of this vector to 0 (i.e. we assume a plant is located there), $\tilde{c}(W)$ contains only the costs which have at least to be paid if $z_{ik} = 1$. Therefore, the scalar product $\langle \lambda, \tilde{c}_{\leq}(W) \rangle$, whereas $\tilde{c}_{\leq}(W)$ is equal to $\tilde{c}(W)$ with elements sorted in non-decreasing sequence, provides a lower bound for the case $z_{ik} = 1$. Hence, this lower bound can be compared to a given upper bound and in case it is greater z_{ik} has to be equal to 0 in any optimal solution. However, this test has again not to be performed for all i and j :

Remark 3. If z_{ik} has been fixed to 0 (applying the test of this subsection), then $z_{ik'} = 0$ for $k' = k, \dots, G_i$, too. This follows directly from the definition of the z_{ik} -variables.

3.4. Valid Inequalities

In designing the specialized branch & cut procedure for the new formulation of the DOMP, the following family of inequalities turned out to be very useful. The subsequent proposition proves that they are valid for the proposed formulation.

$$\sum_{j=1}^{|A|} x_{(M-N+1-j)k} \geq \sum_{\substack{i \in A \\ i'_k \leq G_i}} z_i i'_k \quad \forall A \subset V, |A| \leq M - N; k = 2, \dots, G. \tag{19}$$

Proposition 2. The family of inequalities (19) is valid for the formulation of the DOMP given in Section 2, i.e. they are fulfilled for all feasible solutions.

Proof. Let (\tilde{x}, \tilde{z}) be a feasible solution for the formulation of the DOMP given in Section 2. Moreover, let $A \subset V$ with $|A| \leq M - N$ and $k = 2, \dots, G$ be arbitrarily chosen. Then by (11) it holds:

$$\sum_{j=1}^{|A|} \tilde{x}_{(M-N+1-j)k} = \min \left\{ |A|, \sum_{j=1}^{M-N} \tilde{x}_{jk} \right\}.$$

On the other hand, we know by (9) and $0 \leq \tilde{z}_{ik} \leq 1$ that

$$\sum_{\substack{i \in A \\ i'_k \leq G_i}} \tilde{z}_i i'_k \leq \sum_{j=1}^{M-N} \tilde{x}_{jk} \quad \text{and} \quad \sum_{\substack{i \in A \\ i'_k \leq G_i}} \tilde{z}_i i'_k \leq |A|.$$

Thus,

$$\sum_{j=1}^{|A|} \tilde{x}_{(M-N+1-j)k} \geq \sum_{\substack{i \in A \\ i'_k \leq G_i}} \tilde{z}_i i'_k. \quad \square$$

Furthermore, apart from being valid for the proposed formulation, the inequalities (19) can be used to replace binary restrictions on x_{jk} -variables. This statement is proved by the following proposition.

Proposition 3. If a feasible solution satisfies $z_{ik} \in \{0, 1\}$ for all $i = 1, \dots, M$ and $k = 1, \dots, G_i$, then the family of Valid Inequalities (19) enforces x_{jk} to be binary for all $j = 1, \dots, M - N$ and $k = 1, \dots, G$ as well.

Proof. Let (\tilde{x}, \tilde{z}) be feasible for (7)–(11), $z_{ik} \in \{0, 1\}$ for all $i = 1, \dots, M$ and $k = 1, \dots, G_i$. Take $k', 1 \leq k' \leq G$, arbitrary but fixed. Moreover, let

$$\sum_{\substack{i=1, \dots, M \\ i'_k \leq G_i}} z_i i'_k = q \in \mathbb{Z}_0^+.$$

Then, if $q = 0$, Constraints (9) imply $x_{jk} = 0$ for all $j = 1, \dots, M - N$. Otherwise, if $q = 1$, the Valid Inequalities (19) imply (with $|A| = 1$) that

$$\tilde{x}_{M-Nk'} \geq \max_{i=1, \dots, M} \{\tilde{z}_i i'_k\} = 1,$$

and thus $\tilde{x}_{M-Nk'} = 1$. Furthermore, if $q > 1$, again the Valid Inequalities (19) imply (with $|A| = 2$) that

$$\tilde{x}_{M-Nk'} + \tilde{x}_{M-N-1k'} \geq \max_{i, r=1, \dots, M} \{\tilde{z}_i i'_k + \tilde{z}_r r'_k\} = 2,$$

which yields $\tilde{x}_{M-N-1k'} = 1$. Repeating the same argument one can observe that $\tilde{x}_{M-N-2k'}$ to $\tilde{x}_{M-N-(q-1)k'}$ have also to be equal to 1. At last, by Constraints (9), $\tilde{x}_{N-M-qk'}$ to $\tilde{x}_{1k'}$ have to be equal to 0, which completes the proof. \square

Given Propositions 1 and 3, the DOMP can be formulated as a mixed-integer linear program with only M binary variables (z_{i1} for all $i = 1, \dots, M$). However, as the family of Valid Inequalities (19) is exponential in size, a solution, using standard software-tools, seems not to be appropriate. Therefore, the next section is dedicated to a specialized branch & cut procedure, which applies the valid inequalities dynamically whenever they are violated.

4. Specialized branch & cut

As already explained in the previous section, a solution for the presented model of DOMP can be found using a branch & cut procedure with only M binary variables, whereas Valid Inequalities (19) are added whenever they are needed. Thereby, the solution of a subproblem of the branching tree can be obtained as follows:

- (1) Solve the LP-relaxation.
- (2) Add, for each $2 \leq k \leq G$, all (or some) violated cuts of type (19) (if there are any). If some cuts have been added, goto Step 1.

Algorithm 1: Specialized Branch & Cut for the presented model of the DOMP

Input: Problem (DOMP) as given in Section 2.

Output: Optimal solution (x^*, z^*) and optimal objective value F .

```

1 Initialization and Preprocessing.
  Set  $S = \emptyset$  (set of pairs (subproblem, lower bound), which still have to be
  considered), determine the initial upper bound  $F$  using VNS, execute variable
  fixing following subsections 3.1-3.3, and initialize  $(x^*, z^*)$  accordingly.

2 repeat
  | Solve LP-relaxation of (DOMP).
  | forall  $k = 2, \dots, G$  do
  |   | Add violated cuts (20) according to the cutting strategy
  | end
  until No cuts have been added
  Set  $S = \{(DOMP, F^0)\}$  ( $F^0$  = lower bound of the LP-relaxation with all cuts
  added.)

3 while  $S \neq \emptyset$  do
  | Choose  $(P^i, F^i) \in S$  with  $F^i = \min_{(P^j, F^j) \in S} F^j$  and let  $(\tilde{x}, \tilde{z})$  be the optimal
  | solution of the LP-relaxation of  $P^i$ .
  | if  $\underline{F}^i > \bar{F}$  then
  |   | Set  $S = S \setminus \{(P^i, F^i)\}$ .
  | else if  $\underline{F}^i \leq \bar{F}$  and  $\tilde{z}_{i1} \in \{0, 1\}$  for all  $i = 1, \dots, M$  then
  |   | Set  $F = \underline{F}^i$ ,  $(x^*, z^*) = (\tilde{x}, \tilde{z})$  and  $S = S \setminus \{(P^i, F^i)\}$ .
  | else
  |   | Chose  $0 < \tilde{z}_{i1} < 1$  and define  $P_1^i$  and  $P_2^i$  by setting  $z_{i1} = 0$  and  $z_{i1} = 1$ ,
  |   | respectively, to  $P^i$ .
  |   | for  $P_1^i$  and  $P_2^i$  do
  |     | repeat
  |       | Solve LP-relaxation.
  |       | forall  $k = 2, \dots, G$  do
  |       |   | Add violated cuts (20) according to the cutting strategy
  |       | end
  |     | until No cuts have been added
  |   | end
  |   | if  $P_1^i$  ( $P_2^i$ ) is feasible then
  |     | Set  $S = S \cup \{(P_1^i, \underline{F}_1^i)\}$  ( $S = S \cup \{(P_2^i, \underline{F}_2^i)\}$ ).
  |   | end
  | end
  end

4 return  $(x^*, z^*)$  and  $\bar{F}$ .
```

However, because of the exponential size of (19), adding all violated cuts in each iteration proved to be inefficient. Therefore, only cuts of a specific type have been tested and, if violated, added.

Let $2 \leq k \leq G$ and \tilde{x} and \tilde{z} be the solution of the linear relaxation of a node in the branching tree. Moreover, let j_0 be the largest index so that $\tilde{x}_{j_0 k} = 0$ and let j_1 be the smallest index so that $\tilde{x}_{j_1 k} = 1$. In addition, let $\tilde{z}_{i1}^>$ be the vector of $\tilde{z}_{i1}^>$ variables with $i_k \leq G_i$ and sorted in non-increasing order. Then, for each $j_0 + 2 \leq j^* < j_1$ with $M - N - (j^* - 1)$ being less than or equal

Table 2
Tested modeling vectors

	λ		λ
T1	(1, ..., 1)	T6	(1, 0, 1, 0, ..., 1, 0, 1, 0)
T2	(0, ..., 0, 1)	T7	(..., 0, 1, 1, 0, 1, 1)
T3	(0, ..., 0, 1, ..., 1)	T8	(..., 0, 0, 1, 0, 0, 1)
T4	(0, ..., 0, 1, ..., 1, 0, ..., 0)	T9	(0.1, 0.2, ..., M/20, M/20, ..., 0.2, 0.1)
T5	(0, 1, 0, 1, ..., 0, 1, 0, 1)	T10	(M/20, ..., 0.1, 0.1, ..., M/20)

to the number of elements in \tilde{z}_k^{\geq} , the following cut has been tested:

$$\sum_{j=j^*}^{M-N} x_{jk} \geq \sum_{i=1}^{M-N-(j^*-1)} (z_k^{\geq})_i. \tag{20}$$

Note that these cuts are maximal in some sense, since if they are not violated, no valid inequality of type (19) with $|A| = M - N - (j^* - 1)$ is violated. Hence, by Proposition 3, we only need to test them in order to skip binary restrictions on x -variables. Thus, the specialized branch & cut procedure for solving the proposed model of the DOMP is given in Algorithm 1.

However, the number of cuts which have to be checked in order to omit binary restrictions on x -variables can even be further reduced. Observe that if for some $2 \leq k \leq G$ not all \tilde{x}_{jk} -variables are binary, $\tilde{x}_{(j_0+1)k}$ has to have a fractional value and, if \tilde{z}_k^{\geq} is binary (which will be the case at some point in the branching tree), cut (20) for $j^* = j_0 + 2$ is violated. Therefore, only these valid inequalities need to be tested and, furthermore, the test is only necessary if all z -variables are binary.

As can be seen from this discussion, we need, on the one hand, only some cuts of type (20) to replace binary constraints on the x -variables. This leads to relatively small linear programs in each iteration. On the other hand, adding all cuts of type (20) leads to better lower bounds in each node. On this account we tested three different cutting strategies (see Section 5.2):

- (1) Add each violated cut (20);
- (2) Add violated cuts (20) only if $j^* = j_0 + 2$;
- (3) Add violated cuts (20) only if $j^* = j_0 + 2$ and all z -variables are binary.

Note that the corresponding steps in Algorithm 1 are performed according to the cutting strategy.

5. Numerical results

5.1. Computer and test problems

The branch & cut procedure described in the previous section has been implemented using Visual C++ 7.0. Moreover, ILOG Concert Technology 2.0 and ILOG CPLEX 9.0 have been used for the implementation and solution of linear programs. All computational studies have been performed on a PC with a Pentium IV processor with 2.4 GHz and 512 MB of RAM.

In order to test the performance of the proposed solution method, problem instances with eight different λ -vectors have been used. These λ -vectors are given in Table 2 (for T1–T8 see also [13,19]), where Median (T1), Center (T2), k -Centra (T3) and $k_1 + k_2$ -Trimmed Mean (T4) are well known special cases. Furthermore, for each of these λ -vectors problem instances have been generated with $M = 30, 40, 50, 60, 70$ and 80 and $N = 3, 8$ and $M/3$. In this process, k (in T3) has been set to $\lceil M/3 \rceil$ and k_1 and k_2 (in T4) have been set to $\lceil M/10 \rceil$.

For each combination of M, N and λ five problem instances have been tested. Thereby, the cost elements of C have been chosen using a uniform distribution between 1 and 200. Thus, a total number of 900 problems have been used to test the performance of the proposed method (see Section 5.5).

In addition, the distance matrices of the OR-Lib instances pmed1–pmed5 (see [22]), extended by T1–T10, have been tested (see Section 5.6).

5.2. Configuration of the specialized branch & cut algorithm

Based on the discussion at the end of Section 4, we wanted to know which of the three possibilities for adding valid inequalities provides the best performance in solving the respective test problems. Therefore, each of the smallest problem instances (i.e. $M = 30$ and $M = 40$) has been calculated once for each possibility. The results of these computations, for T2 and T4 (for all other λ -values the results are similar to one of these two), are given in Tables 3–5 whereas the different possibilities for adding cuts are denoted as follows:

- (1) ALL: Add each violated cut (20);
- (2) ONE: Add violated cuts (20) only if $j^* = j_0 + 2$;
- (3) ONEBIN: Add violated cuts (20) only if $j^* = j_0 + 2$ and all z -variables are binary.

In each of these tables the first column contains the number of locations M , the second the respective λ -vectors (T2 or T4) and the third the number of plants N . The next three columns show the minimal, average (over all five instances) and maximal solution times (in seconds) for each problem type. Thereby, the first number in each column is the preprocessing

Table 3
Results for T2 and T4 using ONE

<i>M</i>	λ	<i>N</i>	Min.	Ave.	Max.	# N.	# C.
30	T2	3	0.14 / 4.22	0.16 / 8.48	0.19 / 22.66	9	1210
		8	0.47 / 0.73	0.64 / 2.99	0.91 / 9.81	60	1656
		<i>M</i> /3	0.63 / 1.59	1.00 / 3.69	1.66 / 8.77	112	2287
	T4	3	0.19 / 1.02	0.19 / 1.73	0.20 / 3.38	4	251
		8	0.63 / 0.81	0.72 / 1.01	0.86 / 1.39	4	158
		<i>M</i> /3	0.86 / 1.06	1.12 / 1.53	1.31 / 2.14	9	223
40	T2	3	0.31 / 15.94	0.37 / 99.70	0.45 / 263.64	27	3918
		8	1.02 / 1.84	1.29 / 12.79	2.27 / 40.28	73	2757
		<i>M</i> /3	2.80 / 4.78	3.60 / 8.99	5.06 / 19.39	292	3784
	T4	3	0.44 / 10.69	0.44 / 15.83	0.45 / 22.05	13	868
		8	1.16 / 1.45	1.63 / 3.99	2.59 / 9.63	12	383
		<i>M</i> /3	3.41 / 3.45	3.94 / 4.18	4.69 / 5.44	3	104

Table 4
Results for T2 and T4 using ONEBIN

<i>M</i>	λ	<i>N</i>	Min.	Ave.	Max.	# N.	# C.
30	T2	3	0.13 / 1.14	0.16 / 1.73	0.19 / 3.09	10	280
		8	0.41 / 0.69	0.57 / 1.83	0.81 / 5.39	61	1323
		<i>M</i> /3	0.61 / 1.61	0.95 / 3.44	1.56 / 8.95	171	2944
	T4	3	0.17 / 0.97	0.18 / 1.30	0.19 / 2.00	3	203
		8	0.61 / 0.77	0.68 / 0.95	0.80 / 1.22	6	134
		<i>M</i> /3	0.80 / 0.97	1.08 / 1.28	1.30 / 1.50	6	142
40	T2	3	0.30 / 3.72	0.35 / 11.95	0.44 / 25.17	28	720
		8	0.95 / 1.41	1.23 / 4.32	2.16 / 9.83	84	1552
		<i>M</i> /3	2.70 / 4.20	3.50 / 11.06	4.89 / 28.58	578	6039
	T4	3	0.42 / 7.77	0.43 / 14.25	0.44 / 18.61	18	1201
		8	1.09 / 1.39	1.60 / 4.15	2.69 / 10.67	29	636
		<i>M</i> /3	3.34 / 3.38	3.84 / 4.04	4.55 / 5.09	4	88

Table 5
Results for T2 and T4 using ALL

<i>M</i>	λ	<i>N</i>	Min.	Ave.	Max.	# N.	# C.
30	T2	3	0.13 / 25.95	0.15 / 119.59	0.17 / 342.61	7	9532
		8	0.42 / 0.77	0.59 / 3.81	0.84 / 13.80	24	3044
		<i>M</i> /3	0.61 / 1.47	0.94 / 2.62	1.56 / 5.38	26	2090
	T4	3	0.17 / 2.45	0.18 / 6.38	0.19 / 17.00	3	1939
		8	0.61 / 0.81	0.68 / 1.06	0.81 / 1.47	4	432
		<i>M</i> /3	0.81 / 0.97	1.07 / 1.40	1.27 / 1.92	6	370
40	T2	3	0.30 / 1014.73	0.35 / 2800.31	0.45 / 4585.02	17	34790
		8	0.97 / 2.84	1.25 / 54.62	2.20 / 221.05	27	7287
		<i>M</i> /3	2.78 / 3.44	3.46 / 5.42	4.83 / 8.16	34	2147
	T4	3	0.42 / 36.86	0.43 / 122.47	0.45 / 244.50	11	7685
		8	1.08 / 1.50	1.54 / 9.17	2.45 / 33.44	13	1915
		<i>M</i> /3	3.27 / 3.30	3.82 / 3.99	4.55 / 5.03	2	222

time and the second number (in bold face) the total solution time. The last two columns represent the average number of nodes in the branching tree (# N.) and the average number of cuts which have been applied (# C.).

As can be observed, for T2 and $N = 3-8$ the possibility ONEBIN provides the best performance. Thereby, it is better than ONE (e.g. 11.95–99.70 for $M = 40$ and $N = 3$) and much better than ALL (e.g. 11.95–2800.31 for $M = 40$ and $N = 3$). Moreover, it can be seen that for these problem instances adding as few cuts as possible leads to much better solution times (e.g. 720 (ONEBIN) to 3918 (ONE) to 34790 (ALL) for $M = 40$ and $N = 3$). More or less the same effect (i.e. ONEBIN has the best performance) can be observed for T4 and $N = 3-8$ even though not so distinctive. Because of these results we decided to use the possibility ONEBIN for the solution of the test problems with $N = 3$ and $N = 8$. On the other hand, for $N = M/3$, possibility ALL seems to have the best performance. However, the differences for $N = M/3$ between ALL, ONE and ONEBIN are rather small. Therefore, we compared the solution times for ALL and ONEBIN for $M = 50$ and $N = M/3$, too (see Table 6). Note that ONE has not been tested since the performance for $M/3$ is very similar to the one of ONEBIN. Table 6 has the same structure as the Tables 3–5 and its first part contains the results for ONEBIN whereas the second part shows the results for ALL. From these results it is easy to see that, at least for T2, solution times for ALL are much better than those for ONEBIN. Furthermore, the maximal solution time for T2 with ONEBIN is 3600 s which is the time limit that has been applied. Hence, some instances cannot be solved within the time limit using ONEBIN and so we decided to use the possibility ALL for test problems with $N = M/3$.

Apart from adding a different number of cuts, the special case T2 (with $N = 3-8$) is sensitive with respect to the initial upper bound. In this regard, running the VNS of [19] three times to get the initial upper bound (with different starting

Table 6
Results for T2 and T4 ($M = 50$) using ONEBIN resp. ALL

M	λ	N	Min.	Ave.	Max.	# N.	# C.
50	T2	$M/3$	5.91 / 8.75	8.05 / 1467.37	12.27 / 3600	6207	58 298
	T4	$M/3$	10.02 / 10.80	12.62 / 18.64	16.13 / 29.97	141	1 467
50	T2	$M/3$	6.02 / 9.31	8.11 / 28.29	12.33 / 42.95	109	8 740
	T4	$M/3$	10.05 / 10.34	12.61 / 16.17	16.08 / 29.25	36	1 477

Table 7
Results for CPLEX and CPLEX + Var. Fix compared to B.&C.

M	λ	N	CPLEX	CPLEX + Var. Fix.	Spec. B. & C.
30	T2	3	3600/ 3600 /3600	17.25/ 896.57 /3600	1.14/ 1.73 /3.09
		8	21.88/ 64.04 /116.78	0.80/ 1.44 /2.14	0.69/ 1.83 /5.39
		$M/3$	1.30/ 12.50 /23.86	0.73/ 1.19 / 1.66	1.47/ 2.62 /5.38
30	T4	3	61.69/ 1565.19 /3600	8.48/ 15.96 /36.44	0.97/ 1.30 /2.00
		8	1.14/ 15.29 /36.44	0.81/ 1.20 / 2.13	0.77/ 0.95 /1.22
		$M/3$	0.48/ 2.58 /6.52	0.88/ 1.44 / 1.95	0.97/ 1.40 /1.92
40	T2	3	3600/ 3600 /3600	3600/ 3600 /3600	3.72/ 11.95 /25.17
		8	120.64/ 1270.98 /3600	3.72/ 720.73 /3600	1.41/ 4.32 / 9.83
		$M/3$	6.72/ 13.76 /21.30	2.97/ 3.80 /5.20	3.44/ 5.42 / 8.16
40	T4	3	3600/ 3600 /3600	772.17/ 2204.36 /3600	7.77/ 14.25 /18.61
		8	2.52/ 203.74 /722.73	1.78/ 10.06 /20.06	1.39/ 4.15 /10.67
		$M/3$	0.64/ 2.77 /10.20	3.36/ 3.93 / 4.73	3.30/ 3.99 /5.03

Note that again a time limit of 3600 s has been used.

solutions), instead of once, sometimes reduced the total solution time by a factor of ten or more. Therefore, for T2 and $N = 3-8$, this possibility has been implemented. For all other λ -vectors and for $N = M/3$ this strategy does not lead to better solution times.

Summarizing the results of this section we conclude that for problems with a small number of plants (i.e. N is small compared to M) it is better to use as few cuts as possible to keep the linear programs in each iteration as small as possible. For these types of problems the lower bound improvement seems not to be a crucial advantage. By contrast, if N gets larger the improvement of the lower bound becomes more and more important and is sometimes even necessary to obtain an optimal solution in reasonable time.

5.3. Strength of the variable fixing and the valid inequalities

Given the results of the previous section it is also interesting to see whether the specialized branch & cut procedure, in its best configuration, provides better solution times than just using CPLEX for the solution of the basic problem or using CPLEX after applying the variable fixing strategies of Sections 3.1–3.3. Therefore, these solution methods were implemented as well and the test problems with $M = 30$ and $M = 40$ were used to compare them with the specialized branch & cut. The outcomes of these calculations are given in Table 7. In this table the first three columns are the same as in the Tables 3–6. The fourth column shows the solution times when using just CPLEX, the fifth column when using CPLEX with variable fixing and the last column when using the branch & cut procedure. In addition, the first number in each column represents the minimal, the second the average (over all five instances) and the third the maximal solution time. Note that again only the results for T2 and T4 are given, but they are, as above, representative for more or less every λ -value which has been considered.

Analyzing the entries of Table 7 one can observe that for $N = 3$ and $N = 8$ the solution times when using CPLEX are always worse than those when using CPLEX after applying variable fixing. This is due to the fact that the problems are much smaller after variable fixing which is substantiated by Table 8 where the average number of fixed variables is given. Moreover, considering the values for $N = 3$, it is easy to see that CPLEX without variable fixing was usually not able to find a solution within the time limit. However, for $N = 3$ and $N = 8$, specialized branch & cut provides solution times that are even better than those of CPLEX plus variable fixing. Thereby, the improvement is sometimes only marginal for small problems (e.g. $M = 30$, $N = 8$ and T4) but for larger problems it is tremendous (e.g. $M = 40$, $N = 3$ and T2). One reason for this improvement is the fact that the Valid Inequalities (19) sometimes close the duality gap considerably. This can be seen in Table 9 where reduction of the root node gap, after adding violated cuts of type (20), is reported. Nonetheless, the main reasons for the improvement, especially for small values of N , are Propositions 1 and 3 because they allow to reduce the number of binary variables to M .

For $N = M/3$, on the other hand, it is not clear which solution method is the most appropriate. As can be observed from Table 7 for each alternative there is at least one example where it shows the best performance. In addition, for example for $M = 40$ and T4, CPLEX provides the smallest average solution time but it needs 10.20 s for one instance which is twice as long as the maximal solution time when using CPLEX with variable fixing or the branch & cut. The reason for CPLEX being much better in solving problems with larger N values seems to be cuts that are added in the root node using the default

Table 8

Average number of fixed variables

M	λ	N	# x -Var. fixed to 1	# z -Var. fixed to 0	# x -Var. fixed to 0
30	T2	3	167	440	3180
		8	99	525	3190
		$M/3$	80	499	2916
	T4	3	167	114	3156
		8	99	249	2914
		$M/3$	80	262	2635
40	T2	3	164	735	4136
		8	110	988	4953
		$M/3$	67	865	3738
	T4	3	164	111	4139
		8	110	363	4548
		$M/3$	67	525	3450

Table 9

Root node gap improvement

M	λ	N	R.-N. gap (in %)	R.-N. gap with cuts (in %)	Improvement (in %)
30	T2	3	56	36	20
		8	45	20	25
		$M/3$	39	19	20
30	T4	3	10	6	4
		8	6	3	3
		$M/3$	5	1	4
40	T2	3	59	39	20
		8	51	22	29
		$M/3$	41	19	22
40	T4	3	18	14	4
		8	8	5	3
		$M/3$	3	0	3

Table 10

Comparing the solution times with those obtainable by previously published procedures

	λ	T1	T2	T3	T4	T5	T6	T7	T8
$N = 8$	[13] and [20]	167.19	13.54	86.44	56.37	160.75	131.82	170.57	151.38
	Spec. B. & C.	0.82	2.17	5.99	0.96	1.78	2.64	1.52	3.20
$N = 10$	[13] and [20]	303.11	6.39	117.02	154.46	242.00	253.11	269.65	188.97
	Spec. B. & C.	0.72	2.60	1.81	1.31	1.13	2.28	1.23	2.09
$N = 15$	[13] and [20]	274.74	4.81	125.19	649.54	209.68	327.44	179.30	101.19
	Spec. B. & C.	0.78	1.95	0.96	9.47	1.06	3.84	0.88	1.03

configuration of the solver. These cuts improve the initial lower bound, which is for larger N values sometimes even worse than for smaller ones, considerably, and hence the total solution time decreases. However, for small values of N these cuts are not very effective and the specialized branch & cut shows a much better performance.

5.4. Comparison with previous results

To see whether the approach introduced in this work is better suited to solve the DOMP than the already existing solution procedures or not, Table 10 contains the best average solution times (in seconds) provided by [13,20] for problems with $M = 30$ and $N = 8$, $N = 10$ and $N = 15$. Optimal solution times for larger problems are not reported in these works. In addition, the average (total) solution times (in seconds) for solving problems of the same size using the new approach, are displayed in Table 10 as well. Thereby, the same computer as for the outcomes given in [13,20] and has been employed.

As can be observed from this table, the average solution times which can be reached by using the specialized branch & cut procedure, are nearly always at least one order of magnitude lower than those reported in the literature. Only for λ -vectors of type 2 (i.e. T2), the improvement is less significant.

Hence, the new approach is much better suited to solve the DOMP to optimality than the existing ones.

5.5. Complete results

The complete results for all 900 test instances can be found in the Tables 12–17 in the Appendix. These tables have the same structure as those of Section 5.2 but with columns added displaying the average gap between the initial heuristic and the final objective value, the maximum of these values and the average root node gap of the respective problem types (M , N and λ -vector). Moreover, for each problem type, test instances have been computed using the respective best configuration (see analysis in Section 5.2).

Table 11
Results for pmed1–pmed5

	M	N	λ	Solution time	# N.	# C.
pmed1	100	5	$T1$	9.16/ 23.80	2	156
			$T2$	10.50/ 52.19	10	934
			$T4$	9.17/ 282.41	31	8 451
			$T9$	8.86/ 319.36	33	3 009
pmed2	100	10	$T1$	20.64/ 24.34	1	107
			$T2$	13.91/ 3600	–	–
			$T4$	17.67/ 93.42	10	1 637
			$T9$	17.77/ 70.52	10	310
pmed3	100	10	$T1$	20.47/ 30.42	5	612
			$T2$	30.56/ 3600	–	–
			$T4$	19.03/ 252.70	39	3 807
			$T9$	20.56/ 417.33	103	4 216
pmed4	100	20	$T1$	104.28/ 160.67	1	120
			$T2$	66.06/ 3600	–	–
			$T4$	96.64/ 3600	–	–
			$T9$	104.53/ 1100.8	712	22 842
pmed5	100	33	$T1$	277.41/ 278.03	1	614
			$T2$	227.17/ 3600	–	–
			$T4$	249.47/ 3600	–	–
			$T9$	290.78/ 3600	–	–

As can be observed from these tables, these results are very promising. For nearly all λ -vectors problems can be solved with M up to 80 in reasonable time (depending on the number of plants N). Only for some λ -vectors (i.e. $T3$ and $T10$ and partly $T2$) and N values (i.e. $N = 3$ and $N = 8$) the outcomes are not really satisfying. For these λ -vectors and $M = 70$ and $M = 80$ an optimal solution cannot be found within the time limit of 3600 s (missing values in Tables 16 and 17). For smaller M values a solution within the time limit is possible but solution times are quite high compared to those of other λ -vectors. A reason for this may be the relatively large root node gaps which cannot be closed by the algorithm (in reasonable time) even if many cuts are added.

Considering the respective solution times in more detail, one can observe that there is sometimes a large difference between minimal and maximal solution times. But this is not a surprising fact since the size of the model depends on the number of different values in the distance matrix. Furthermore, it becomes clear that, in principle but not always, solution times decrease for increasing values of N which is again due to the definition of the model. In addition, root node gaps are decreasing for increasing values of N as well. Therefore, the hardest problems are those with $N = 3$ (this can also be seen in Table 7 for the solution with CPLEX). At last, the number of branch & bound nodes is increasing from $N = 3$ to $N = 8$ but decreasing from $N = 8$ to $N = M/3$. For the number of applied cuts, on the other hand, such a behavior cannot be detected.

The average gaps between the initial heuristic and the final objective value show that the heuristic indeed provides very good approximations. Nonetheless, sometimes the gap is relatively high, especially for $T2$. For problems with $N = 3$ and $N = 8$, a heuristic solution, using the approach of [13,19], can be found very quickly. On the other hand, for $N = M/3$ the heuristic solution time forms the main part of the total solution time. Thereby, a different heuristic which is not so sensitive with respect to the number of facilities to be located, may lead to even faster solution times.

Summarizing the numerical results, we conclude that the new model and the proposed solution procedure are well suited for solving the DOMP for a large number of different λ -vectors. Thereby, problem instances can be solved in a reasonable time (on average within less than 10 minutes) which are (in terms of the value of M) more than twice as large as those that can be solved by existing optimal solution approaches (see [13,20]).

5.6. Results for pmed1–pmed5

In addition to the problem instances that were generated for testing the performance of the new approach we used distance matrices of “standard test problems” pmed1–pmed5 (see [22]) in order to see whether the proposed solution method works for these (larger) problems as well. The results of these computations, for some selected λ -values, are given in Table 11 whereas the first four columns show the problem instances, the fifth the preprocessing and total solution time and the last two the number of nodes in the branching tree and the number of applied cuts. Note that for pmed1–pmed4 (pmed5) the possibility ONEBIN (ALL) has been used and that again a time limit of 3600 s has been applied.

Analyzing Table 11 one can observe that, on the one hand, for $T1$ all problems can be solved relatively fast. Furthermore, the solution times for pmed2 and pmed3 are a good deal lower than those for $T1$, $M = 80$ and $N = 8$ which may be due to the fact that the distance matrices of these problems are not symmetric. On the other hand, for $T4$ and $T9$, a solution in reasonable time can only be found for pmed1–pmed3. Moreover, if $T2$ is chosen as λ -vector only pmed1 can be solved within the time limit.

Therefore, it seems that with the provided solution method we are, in principle, able to solve even these problems. However, the size of these test problems already meet our upper size limit and so further research is necessary in this area.

6. Conclusions and further research topics

In this article we have introduced a new formulation for the Discrete Ordered Median Problem, based on two different sets of binary variables, z and x . The first set is used to measure the distance between each point and its closest opened plant, whereas the second set is used to sort all these distances in increasing order, which is necessary to define the ordered objective function. Making use of this new formulation, a preprocessing phase and a subset of valid inequalities, we are able to solve the largest instances for the DOMP approached to date.

Although our results imply a significant advance in the resolution of flexible discrete location problems, the new formulation can only be considered as a starting point inside a new research line. Many possible improvements of the method here developed come to our minds in a natural way. For instance, it is clear from the shape of our formulation that most of the variables in the right-hand side of our variables vectors z and x will take value zero in an optimal solution, since the customers are assigned to the closest opened plant and the right side of the matrix is associated to large distances. Therefore, either a large part of the formulation can be ignored from the beginning or a column generation method can be devised. We can also take advantage of the peculiarities of the coefficient vector λ used to define the objective function. In particular, the formulation can be reduced taking into account the null entries of this vector. More valid inequalities can also help to reduce the duality gap of the instances.

Lastly, the current formulation and the presented solution procedure is only valid for non-negative λ entries. If negative λ -values occur, several changes have to be made, as, for example, the minimization of the objective does not lead automatically to the correct values of z - and x -variables. This topic and those mentioned above are the subject of further research activities.

Acknowledgements

Alfredo Marín gratefully acknowledges support provided by *Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (I+D+I)*, projects HA2005-0056 and MTM2006-14961-C05-04, RDEF funds and *Fundación Séneca*, project 02911/PI/05. Stefan Nickel and Sebastian Velten thank the German Academic Exchange Service (DAAD) (grant number: D/05/5D/05/59) for partial support. Justo Puerto thanks for partial support through grant numbers P06-FQM-01366 and MTM2007-67433-C02-01.

Appendix

See Tables 12–17.

Table 12
Complete results ($M = 30$)

M	λ	N	Min. time	Ave. time	Max. time	Ave. H.gap	Max. H.gap	# N.	# C.	R.-N. gap
30	T1	3	0.16 / 0.52	0.18 / 1.01	0.19 / 1.64	0.04	0.20	9	472	0.11
		8	0.55 / 0.61	0.66 / 0.79	0.75 / 1.19	0.00	0.02	21	478	0.08
		M/3	0.75 / 0.77	0.92 / 0.96	1.09 / 1.11	0.00	0.02	4	516	0.04
	T2	3	0.13 / 1.14	0.16 / 1.73	0.19 / 3.09	0.05	0.16	12	374	0.59
		8	0.41 / 0.69	0.57 / 1.83	0.81 / 5.39	0.01	0.04	65	1418	0.47
		M/3	0.61 / 1.47	0.94 / 2.62	1.56 / 5.38	0.03	0.11	26	2194	0.40
	T3	3	0.19 / 3.95	0.19 / 25.08	0.22 / 44.53	0.01	0.07	112	12904	0.55
		8	0.50 / 1.00	0.59 / 3.69	0.69 / 8.83	0.02	0.08	116	3126	0.36
		M/3	0.73 / 1.00	1.02 / 2.39	1.27 / 6.61	0.00	0.00	16	1835	0.26
	T4	3	0.17 / 0.97	0.18 / 1.30	0.19 / 2.00	0.02	0.09	25	2783	0.22
		8	0.61 / 0.77	0.68 / 0.95	0.80 / 1.22	0.00	0.00	29	759	0.13
		M/3	0.81 / 0.97	1.07 / 1.40	1.27 / 1.92	0.00	0.00	9	737	0.11
T5	3	0.19 / 0.89	0.20 / 2.47	0.22 / 4.33	0.00	0.00	14	983	0.15	
	8	0.58 / 0.98	0.80 / 1.56	1.00 / 2.86	0.00	0.00	34	593	0.12	
	M/3	0.80 / 1.08	0.98 / 1.62	1.09 / 2.89	0.01	0.02	27	727	0.09	
T6	3	0.17 / 1.05	0.18 / 2.76	0.19 / 4.11	0.03	0.17	14	571	0.14	
	8	0.55 / 1.09	0.68 / 1.88	0.86 / 4.09	0.00	0.02	56	619	0.11	
	M/3	0.72 / 1.03	0.89 / 2.11	1.20 / 2.92	0.02	0.10	53	939	0.08	
T7	3	0.17 / 1.06	0.20 / 2.25	0.22 / 3.55	0.00	0.00	11	464	0.14	
	8	0.58 / 1.03	0.64 / 1.39	0.72 / 1.97	0.00	0.02	39	635	0.12	
	M/3	0.75 / 0.95	0.98 / 1.54	1.06 / 2.75	0.00	0.00	29	692	0.08	
T8	3	0.19 / 1.83	0.20 / 3.97	0.22 / 8.97	0.00	0.00	18	1006	0.18	
	8	0.48 / 1.11	0.57 / 2.86	0.67 / 3.59	0.01	0.03	96	1751	0.17	
	M/3	0.77 / 1.67	0.96 / 2.97	1.27 / 6.22	0.01	0.03	77	1981	0.13	
T9	3	0.16 / 0.84	0.18 / 1.07	0.20 / 1.53	0.00	0.00	6	243	0.11	
	8	0.56 / 0.63	0.70 / 0.82	0.91 / 1.00	0.00	0.00	21	364	0.06	
	M/3	0.78 / 0.83	1.08 / 1.12	1.67 / 1.69	0.00	0.00	16	404	0.03	
T10	3	0.17 / 3.69	0.19 / 8.27	0.20 / 15.02	0.00	0.00	32	1918	0.27	
	8	0.53 / 1.11	0.57 / 3.49	0.59 / 5.69	0.02	0.10	75	2282	0.22	
	M/3	0.75 / 1.11	0.89 / 3.97	1.05 / 14.20	0.00	0.00	16	2196	0.17	

Table 13
Complete results ($M = 40$)

M	λ	N	Min. time	Ave. time	Max. time	Ave. H.gap	Max. H.gap	# N.	# C.	R.-N. gap
40	T1	3	0.41 / 4.84	0.44 / 7.75	0.48 / 10.94	0.01	0.03	28	1435	0.21
		8	1.34 / 1.67	1.83 / 2.19	2.77 / 3.31	0.01	0.04	48	1059	0.09
		M/3	3.38 / 3.50	4.37 / 4.41	5.61 / 5.64	0.00	0.00	4	445	0.04
	T2	3	0.30 / 3.72	0.35 / 11.95	0.44 / 25.17	0.14	0.32	34	1007	0.64
		8	0.95 / 1.41	1.23 / 4.32	2.16 / 9.83	0.08	0.27	93	1764	0.53
		M/3	2.78 / 3.44	3.46 / 5.42	4.83 / 8.16	0.13	0.30	35	2236	0.42
	T3	3	0.42 / 89.23	0.44 / 131.00	0.47 / 163.30	0.00	0.00	234	22156	0.61
		8	1.09 / 9.42	1.42 / 34.98	2.13 / 68.11	0.02	0.05	564	14690	0.42
		M/3	3.58 / 4.03	4.37 / 11.01	5.69 / 30.13	0.00	0.02	44	4034	0.27
	T4	3	0.42 / 7.77	0.43 / 14.25	0.44 / 18.61	0.01	0.05	65	5632	0.31
		8	1.09 / 1.39	1.60 / 4.15	2.69 / 10.67	0.00	0.00	142	3574	0.17
		M/3	3.27 / 3.30	3.82 / 3.99	4.55 / 5.03	0.00	0.00	11	1029	0.09
	T5	3	0.42 / 7.75	0.45 / 15.87	0.48 / 19.47	0.00	0.02	37	1784	0.24
		8	1.30 / 1.95	1.55 / 2.89	1.97 / 3.78	0.00	0.00	52	955	0.11
		M/3	3.52 / 3.78	3.99 / 4.88	4.67 / 5.70	0.01	0.02	39	925	0.08
	T6	3	0.41 / 6.47	0.46 / 14.61	0.50 / 21.16	0.00	0.02	28	957	0.21
		8	1.11 / 4.23	1.28 / 6.18	1.45 / 10.03	0.01	0.06	83	901	0.11
		M/3	3.59 / 4.05	3.80 / 4.65	4.03 / 5.55	0.01	0.06	49	681	0.07
	T7	3	0.41 / 6.83	0.44 / 13.91	0.45 / 20.91	0.00	0.00	27	710	0.22
		8	1.06 / 2.30	1.45 / 3.28	2.02 / 4.88	0.00	0.00	44	475	0.11
		M/3	3.59 / 3.80	4.65 / 5.00	5.97 / 6.17	0.00	0.00	23	433	0.06
	T8	3	0.44 / 12.91	0.45 / 25.60	0.47 / 36.05	0.00	0.00	47	2219	0.26
		8	1.25 / 5.23	1.73 / 9.20	2.09 / 13.72	0.00	0.02	144	2030	0.15
		M/3	3.41 / 4.47	5.74 / 7.54	8.98 / 11.27	0.01	0.02	76	1497	0.10
	T9	3	0.41 / 11.22	0.43 / 15.78	0.48 / 20.92	0.01	0.07	27	854	0.22
		8	1.02 / 1.31	1.40 / 2.86	1.97 / 4.61	0.00	0.00	46	438	0.09
		M/3	3.66 / 3.72	4.15 / 4.19	4.92 / 4.95	0.00	0.00	16	301	0.03
	T10	3	0.41 / 41.61	0.44 / 67.59	0.48 / 112.25	0.00	0.01	92	6652	0.32
		8	1.16 / 8.41	1.73 / 16.60	3.45 / 40.03	0.00	0.00	214	5158	0.24
		M/3	3.52 / 4.20	4.90 / 7.32	6.95 / 11.27	0.00	0.00	14	1835	0.17

Table 14
Complete results ($M = 50$)

M	λ	N	Min. time	Ave. time	Max. time	Ave. H.gap	Max. H.gap	# N.	# C.	R.-N. gap
50	T1	3	0.88 / 6.92	0.92 / 23.42	1.00 / 42.88	0.00	0.00	38	1870	0.19
		8	2.06 / 4.23	3.43 / 6.46	5.17 / 11.97	0.00	0.00	342	8169	0.14
		M/3	10.27 / 10.28	11.96 / 12.00	14.34 / 14.38	0.00	0.00	5	531	0.04
	T2	3	0.89 / 10.22	1.11 / 24.57	1.23 / 36.98	0.03	0.10	27	232	0.65
		8	6.31 / 8.64	6.76 / 13.35	7.63 / 22.05	0.17	0.27	105	1238	0.56
		M/3	6.02 / 9.31	8.11 / 28.29	12.33 / 42.95	0.32	0.67	110	8846	0.43
	T3	3	0.86 / 375.52	0.94 / 543.26	1.06 / 911.48	0.05	0.21	426	54103	0.63
		8	1.98 / 242.58	2.67 / 1717.30	3.63 / 3600	0.02	0.09	5890	187349	0.51
		M/3	10.41 / 11.41	11.66 / 15.99	14.31 / 19.34	0.00	0.00	45	4073	0.26
	T4	3	0.86 / 12.55	0.92 / 35.76	1.02 / 64.11	0.00	0.00	109	12134	0.29
		8	2.09 / 5.27	2.23 / 12.47	2.48 / 24.48	0.00	0.00	1235	38768	0.21
		M/3	10.05 / 10.34	12.61 / 16.17	16.08 / 29.25	0.00	0.00	45	2291	0.10
	T5	3	0.86 / 10.00	0.90 / 36.42	0.95 / 67.39	0.01	0.06	47	3119	0.22
		8	2.05 / 7.41	3.15 / 11.54	4.45 / 17.20	0.00	0.00	309	8166	0.16
		M/3	9.94 / 9.97	13.24 / 13.79	16.95 / 17.53	0.00	0.02	30	881	0.06
	T6	3	0.86 / 15.11	0.93 / 35.66	1.00 / 75.59	0.02	0.08	32	1190	0.19
		8	2.16 / 9.17	2.78 / 13.75	3.95 / 18.61	0.00	0.00	120	2079	0.14
		M/3	10.52 / 11.00	13.21 / 15.46	17.47 / 22.42	0.00	0.00	78	1048	0.05
	T7	3	0.88 / 13.55	0.90 / 33.83	0.94 / 56.17	0.01	0.06	30	836	0.19
		8	2.11 / 6.23	2.79 / 11.09	3.73 / 20.13	0.01	0.03	90	852	0.15
		M/3	10.64 / 10.80	14.22 / 15.14	23.28 / 25.78	0.00	0.00	44	734	0.05
	T8	3	0.84 / 19.38	0.92 / 55.63	1.06 / 96.66	0.01	0.07	46	1653	0.22
		8	1.86 / 12.14	2.81 / 25.90	3.94 / 35.58	0.01	0.02	234	2837	0.18
		M/3	10.00 / 11.42	12.63 / 16.19	17.52 / 20.91	0.01	0.04	113	1889	0.08
	T9	3	0.84 / 30.44	0.93 / 48.38	1.02 / 88.16	0.00	0.01	37	919	0.22
		8	2.14 / 4.95	2.34 / 10.06	2.58 / 17.09	0.00	0.00	82	689	0.11
		M/3	10.05 / 10.11	12.89 / 13.03	17.39 / 17.44	0.00	0.00	25	396	0.02
	T10	3	0.86 / 50.28	0.92 / 196.93	1.00 / 399.92	0.00	0.00	128	8938	0.31
		8	2.05 / 72.84	2.54 / 271.21	3.36 / 607.36	0.00	0.05	1626	40688	0.28
		M/3	11.47 / 11.92	13.33 / 18.40	16.20 / 26.61	0.01	0.03	19	2311	0.16

Table 15
Complete results ($M = 60$)

M	λ	N	Min. time	Ave. time	Max. time	Ave. H.gap	Max. H.gap	# N.	# C.	R.-N. gap
60	T1	3	1.64 / 57.22	1.68 / 106.38	1.73 / 216.77	0.01	0.03	112	9 109	0.26
		8	3.42 / 11.22	4.73 / 23.43	6.13 / 52.14	0.01	0.02	1055	26 202	0.18
		M/3	25.42 / 25.45	29.98 / 30.07	34.44 / 34.55	0.00	0.00	11	1 331	0.04
	T2	3	1.73 / 51.95	1.94 / 87.48	2.08 / 113.44	0.09	0.15	63	173	0.68
		8	9.22 / 11.64	11.22 / 31.72	15.44 / 86.56	0.10	0.18	180	1 565	0.60
		M/3	12.80 / 31.75	18.80 / 139.72	23.78 / 475.27	0.28	0.67	640	34 474	0.43
	T3	3	1.53 / 1483.02	1.69 / 2078.92	1.78 / 2774.53	0.01	0.03	907	129 980	0.68
		8	4.59 / 2492.08	5.63 / 3379.75	7.61 / 3600	0.00	0.00	9194	217 361	0.57
		M/3	28.94 / 40.19	39.76 / 71.35	55.31 / 144.33	0.00	0.01	231	16 461	0.28
	T4	3	1.66 / 39.89	1.72 / 78.68	1.84 / 149.28	0.00	0.02	218	27 240	0.34
		8	3.73 / 21.81	4.54 / 47.74	5.95 / 69.00	0.00	0.02	1942	45 557	0.25
		M/3	23.84 / 24.00	27.84 / 44.28	38.27 / 76.95	0.00	0.00	183	7 908	0.10
	T5	3	1.64 / 87.09	1.68 / 176.49	1.70 / 353.95	0.01	0.03	105	6 772	0.28
		8	3.56 / 22.58	4.74 / 42.48	5.30 / 88.52	0.00	0.01	522	9 537	0.19
		M/3	25.41 / 26.64	33.90 / 34.97	45.61 / 46.53	0.00	0.00	74	2 151	0.06
	T6	3	1.63 / 73.86	1.70 / 121.03	1.80 / 213.56	0.01	0.03	63	2 009	0.25
		8	3.69 / 19.14	4.58 / 60.48	7.05 / 128.80	0.01	0.06	255	2 731	0.17
		M/3	24.03 / 25.92	29.48 / 31.61	36.81 / 37.28	0.00	0.02	71	1 108	0.04
	T7	3	1.58 / 86.83	1.67 / 166.65	1.72 / 342.38	0.01	0.04	69	1 535	0.26
		8	3.39 / 16.73	4.22 / 47.02	5.11 / 107.98	0.04	0.13	209	1 050	0.17
		M/3	25.31 / 25.47	30.17 / 31.03	37.70 / 38.48	0.00	0.01	37	669	0.04
	T8	3	1.64 / 106.52	1.70 / 243.07	1.78 / 538.84	0.01	0.05	110	4 999	0.29
		8	3.20 / 24.03	4.49 / 93.29	5.77 / 200.84	0.01	0.02	458	4 400	0.20
		M/3	21.11 / 32.75	33.07 / 42.69	53.31 / 65.77	0.01	0.04	241	3 822	0.06
	T9	3	1.66 / 52.69	1.71 / 154.15	1.80 / 313.33	0.00	0.01	60	1 455	0.26
		8	3.69 / 19.14	4.73 / 54.03	6.86 / 113.72	0.00	0.00	189	1 195	0.15
		M/3	24.94 / 25.00	34.29 / 34.67	49.33 / 50.36	0.00	0.00	53	786	0.02
	T10	3	1.59 / 750.88	1.78 / 1214.11	2.14 / 2018.92	0.01	0.07	406	44 862	0.37
		8	3.73 / 304.63	5.77 / 1999.35	7.89 / 3600	0.00	0.02	5022	130 801	0.33
		M/3	26.91 / 35.08	33.90 / 57.88	40.05 / 104.86	0.00	0.01	47	6 215	0.17

Table 16
Complete results ($M = 70$)

M	λ	N	Min. time	Ave. time	Max. time	Ave. H.gap	Max. H.gap	# N.	# C.	R.-N. gap
70	T1	3	2.61 / 113.25	2.84 / 198.89	3.25 / 397.53	0.00	0.02	27	111	0.18
		8	6.77 / 20.03	7.39 / 39.64	8.36 / 66.44	0.00	0.02	49	34	0.11
		M/3	57.61 / 57.64	87.96 / 88.10	162.80 / 163.02	0.00	0.00	12	983	0.04
	T2	3	2.70 / 156.08	2.89 / 544.15	3.14 / 1179.95	0.08	0.13	288	1 116	0.70
		8	13.98 / 93.39	15.98 / 372.69	18.67 / 1147.52	0.18	0.25	1647	12 742	0.62
		M/3	24.34 / 35.48	26.99 / 63.37	31.91 / 137.13	0.83	1.17	96	8 736	0.39
	T3	3	- / -	- / -	- / -	-	-	-	-	-
		8	- / -	- / -	- / -	-	-	-	-	-
		24	57.81 / 64.08	71.64 / 93.12	105.66 / 143.19	0.00	0.00	107	8 285	0.24
	T4	3	2.67 / 115.38	2.89 / 147.47	3.23 / 173.98	0.01	0.03	118	2 205	0.36
		8	5.42 / 35.91	6.08 / 118.48	7.84 / 235.00	0.01	0.05	1282	16 876	0.27
		M/3	48.75 / 55.77	59.65 / 64.03	68.77 / 74.61	0.00	0.00	48	2 773	0.07
	T5	3	2.69 / 157.72	2.93 / 290.69	3.28 / 555.11	0.00	0.02	71	1 329	0.27
		8	6.36 / 40.61	9.10 / 91.36	11.83 / 183.20	0.00	0.00	379	3 854	0.19
		M/3	55.05 / 57.55	61.35 / 62.93	70.25 / 71.08	0.01	0.02	62	1 421	0.05
	T6	3	2.67 / 149.09	2.86 / 250.62	3.11 / 493.36	0.00	0.00	55	1 010	0.24
		8	5.89 / 59.16	6.80 / 115.19	7.64 / 183.14	0.01	0.03	229	1 394	0.17
		M/3	55.33 / 57.52	60.44 / 68.80	75.95 / 83.45	0.00	0.02	202	2 209	0.04
	T7	3	2.69 / 185.73	2.89 / 302.94	3.11 / 618.64	0.00	0.00	58	1 057	0.25
		8	6.06 / 33.50	7.76 / 88.23	11.11 / 169.59	0.01	0.02	187	747	0.17
		M/3	56.78 / 57.69	68.07 / 71.40	92.02 / 92.48	0.00	0.01	114	1 322	0.04
	T8	3	2.69 / 211.36	2.93 / 389.43	3.20 / 767.75	0.00	0.00	79	1 764	0.27
		8	5.27 / 42.34	7.99 / 162.98	9.92 / 362.38	0.01	0.06	340	2 695	0.19
		M/3	56.83 / 57.27	65.77 / 68.28	77.05 / 79.83	0.00	0.02	94	1 281	0.05
	T9	3	2.67 / 274.95	2.82 / 360.12	3.02 / 420.27	0.00	0.01	71	1 298	0.28
		8	5.44 / 27.73	6.59 / 65.58	7.72 / 123.80	132	0.00	0.00	678	0.14
		M/3	59.70 / 59.77	78.43 / 78.53	114.63 / 114.81	0.00	0.01	19	257	0.01
	T10	3	- / -	- / -	- / -	-	-	-	-	-
		8	- / -	- / -	- / -	-	-	-	-	-
		M/3	58.72 / 62.39	73.14 / 86.93	86.44 / 118.45	0.00	0.00	39	4 118	0.15

Table 17
Complete results ($M = 80$)

M	λ	N	Min. time	Ave. time	Max. time	Ave. H.gap	Max. H.gap	# N.	# C.	R.-N. gap
80	T1	3	4.25 / 226.06	4.36 / 449.46	4.69 / 673.59	0.01	0.03	36	167	0.18
		8	8.73 / 77.61	12.36 / 178.58	17.73 / 248.89	0.00	0.01	98	59	0.14
		M/3	113.38 / 113.42	138.93 / 139.05	192.14 / 192.20	0.00	0.00	46	5 168	0.04
	T2	3	3.17 / 493.98	3.42 / 950.00	3.75 / 1300.17	0.08	0.23	345	1268	0.73
		8	6.44 / 498.95	8.36 / 1752.73	12.88 / 3600	0.15	0.23	4287	26 109	0.67
		M/3	48.11 / 106.63	66.24 / 206.01	81.55 / 534.58	0.62	0.83	167	18 941	0.44
	T3	3	- / -	- / -	- / -	-	-	-	-	-
		8	- / -	- / -	- / -	-	-	-	-	-
		M/3	113.00 / 143.00	133.76 / 243.82	163.30 / 424.98	0.00	0.00	230	21 730	0.26
	T4	3	4.14 / 119.72	4.26 / 217.31	4.58 / 370.92	0.02	0.09	106	1 410	0.34
		8	7.91 / 148.89	9.32 / 365.37	11.03 / 769.67	0.00	0.00	1160	10 065	0.28
		M/3	102.25 / 107.81	126.35 / 132.66	147.61 / 151.61	0.00	0.02	71	5 570	0.08
	T5	3	4.19 / 361.53	4.33 / 633.52	4.72 / 908.78	0.01	0.03	80	1 223	0.27
		8	7.61 / 140.47	9.89 / 378.96	11.94 / 509.33	0.01	0.03	571	3 280	0.22
		M/3	102.17 / 103.17	155.99 / 157.27	203.83 / 205.52	0.00	0.01	46	1 695	0.04
	T6	3	4.22 / 216.55	4.36 / 423.55	4.70 / 605.80	0.00	0.00	66	919	0.24
		8	8.92 / 145.06	10.92 / 342.31	16.84 / 557.34	0.01	0.03	323	1 278	0.19
		M/3	105.33 / 115.39	123.68 / 131.93	138.92 / 154.73	0.01	0.01	112	1 420	0.03
	T7	3	4.20 / 283.09	4.35 / 587.86	4.70 / 896.86	0.01	0.02	68	876	0.24
		8	8.56 / 163.73	11.96 / 320.00	19.11 / 395.48	0.00	0.02	292	936	0.20
		M/3	106.42 / 111.02	125.99 / 130.18	169.11 / 169.41	0.00	0.01	94	1 580	0.03
	T8	3	4.17 / 436.36	4.23 / 855.20	4.31 / 1265.75	0.02	0.02	98	2 350	0.27
		8	8.78 / 285.47	11.25 / 670.93	14.78 / 1007.44	0.00	0.01	663	4 666	0.23
		M/3	94.16 / 101.80	122.59 / 126.52	157.27 / 158.17	0.00	0.02	100	1 862	0.04
	T9	3	4.19 / 438.31	4.28 / 621.11	4.58 / 933.38	0.01	0.07	69	1 207	0.26
		8	7.69 / 221.36	9.06 / 500.14	13.20 / 1222.34	0.00	0.00	391	1 801	0.18
		M/3	120.64 / 120.95	159.73 / 159.92	228.17 / 228.47	0.00	0.01	21	372	0.01
	T10	3	- / -	- / -	- / -	-	-	-	-	-
		8	- / -	- / -	- / -	-	-	-	-	-
		M/3	152.84 / 212.98	184.10 / 351.64	257.14 / 590.39	0.00	0.01	220	25 319	0.16

References

- [1] M. Daskin, Network and Discrete Location: Models, Algorithms, and Applications, Wiley, 1995.
- [2] Z. Drezner, H. Hamacher, Facility Location: Applications and Theory, Springer, Berlin Heidelberg, New York, 2002.
- [3] P. Mirchandani, R. Francis, Discrete Location Theory, Wiley, New York, NY, 1990.
- [4] L. Cánovas, M. Landete, A. Marín, On the facets of the simple plant location packing polytope, Discrete Applied Mathematics 124 (2002) 27–53.
- [5] K. Aardal, Capacitated facility location: Separation algorithms and computational experience, Mathematical Programming 81 (1998) 149–175.
- [6] A. Marín, Lower bounds for the two-stage uncapacitated facility location problem, European Journal of Operational Research 179 (2007) 1126–1142.
- [7] L. Cánovas, S. García, M. Labbé, A. Marín, A strengthened formulation for the simple plant location problem with order, Operations Research Letters 35 (2) (2007) 141–150.
- [8] M. Melo, S. Nickel, F. Saldanha da Gama, Dynamic multi-commodity capacitated facility location: A mathematical modeling framework for strategic supply chain planning, Computers and Operations Research 33 (2006) 181–208.
- [9] S. Nickel, Discrete Ordered Weber Problems, in: Operations Research Proceedings 2000, Springer Verlag, 2001, pp. 71–76.
- [10] R. Francis, T. Lowe, A. Tamir, Aggregation error bounds for a class of location models, Operations Research 48 (2000) 294–307.
- [11] S. Nickel, J. Puerto, A unified approach to network location problems, Networks 34 (1999) 283–290.
- [12] A. Rodríguez-Chía, S. Nickel, J. Puerto, F. Fernández, A flexible approach to location problems, Mathematical Methods of Operations Research 51 (2000) 69–89.
- [13] S. Nickel, J. Puerto, Facility Location - A Unified Approach, Springer Verlag, 2005.
- [14] N. Boland, P. Domínguez-Marín, S. Nickel, J. Puerto, Exact procedures for solving the discrete ordered median problem, Computers and Operations Research 33 (2005) 3270–3300.
- [15] W. Ogryczak, A. Tamir, Minimizing the sum of the k largest functions in linear time, Information Processing Letters 85 (3).
- [16] G. Cornuéjols, G. Nemhauser, L. Wolsey, A canonical representation of simple plant location problems and its applications, SIAM Journal on Algebraic and Discrete Methods 1 (3) (1980) 261–272.
- [17] A. Kolen, Solving covering problems and the uncapacitated plant location problem on trees, European Journal of Operational Research 12 (1983) 266–278.
- [18] S. Elloumi, M. Labbé, Y. Pochet, A new formulation and resolution method for the p -center problem, INFORMS Journal on Computing 16 (1) (2004) 84–94.
- [19] P. Domínguez-Marín, S. Nickel, N. Mladenović, P. Hansen, Heuristic procedures for solving the discrete ordered median problem, Annals of Operations Research 136 (2005) 145–173.
- [20] P. Domínguez-Marín, The discrete ordered median problem: Models and solution methods, Ph.D. Thesis, University of Kaiserslautern, 2003.
- [21] O. Kariv, S. Hakimi, An algorithmic approach to network location problems II: The p -medians, SIAM Journal on Applied Mathematics 37 (1979) 539–560.
- [22] J. Beasley, Or-library: Distributing test problems by electronic mail, Journal of the Operational Research Society 41 (1990) 1069–1072, available at <http://people.brunel.ac.uk/~mastjb/jeb/orlib/files>.
- [23] N. Mladenović, P. Hansen, Variable neighborhood search, Computers and Operations Research 24 (1997) 1097–1100.