J Optim Theory Appl
DOI 10.1007/s10957-010-9783-1

Siting and Sizing of Facilities under Probabilistic Demands

Luís M. Fernandes · Joaquim J. Júdice · Hanif D. Sherali · António P. Antunes

© Springer Science+Business Media, LLC 2011

Abstract In this paper a discrete location model for non-essential service facilities planning is described, which seeks the number, location, and size of facilities, that maximizes the total expected demand attracted by the facilities. It is assumed that the demand for service is sensitive to the distance from facilities and to their size. It is also assumed that facilities must satisfy a threshold level of demand (facilities are not economically viable below that level). A Mixed-Integer Nonlinear Programming (MINLP) model is proposed for this problem. A branch-and-bound algorithm is designed for solving this MINLP and its convergence to a global minimum is established. A finite procedure is also introduced to find a feasible solution for the MINLP that reduces the overall search in the binary tree generated by the branch-and-bound

Communicated by P.M. Pardalos.

This research is supported in part by the *National Science Foundation*, under Grant Number CMMI-0969169.

L.M. Fernandes (✉)
Instituto Politécnico de Tomar and Instituto de Telecomunicações, Tomar, Portugal
e-mail: lmerca@co.it.pt

J.J. Júdice
Departamento de Matemática, Universidade de Coimbra and Instituto de Telecomunicações, Coimbra, Portugal
e-mail: joaquim.judice@co.it.pt

H.D. Sherali
Grado Department of Industrial & Systems Engineering, Virginia Polytechnic Institute & State University, Blacksburg, VA, USA
e-mail: hanifs@vt.edu

A.P. Antunes
Departamento de Engenharia Civil, Universidade de Coimbra, Coimbra, Portugal
e-mail: antunes@dec.uc.pt

Published online: 07 January 2011

algorithm. Some numerical results using a GAMS/MINOS implementation of the algorithm are reported to illustrate its efficacy and efficiency in practice.

Keywords Facility location models · Mixed-integer nonlinear programming · Discrete optimization · Global optimization

1 Introduction

The study of problems involving the siting and sizing of facilities of various types (schools, post offices, etc.) has been the subject of intense research in the past forty years. A significant part of these efforts was directed towards the formulation and resolution of particular types of optimization models, generically known as discrete location (or location-allocation) models. These models are aimed at determining the best locations and capacities for a set of facilities, assuming that the demand for the services they provide arise at a given number of centers and that the facilities are to be located amongst a given set of sites. The reader is referred to [1] for a textbook presentation of location models, and recent surveys are available in [2–4] and [5]. Some particular discrete location models have a very strong presence in the optimization literature, including, for instance, the plant (or warehouse, or facility) location model [6] and the p-median model [7]. Both these and other basic location models, as well as most of their extensions, assume that demand is inelastic with travel cost (or travel distance, or travel time). Furthermore, they assume that demand can be assigned to the least-cost facility or is willing to patronize the closest facility. These assumptions may be plausible, when facilities such as schools or post offices are involved, but they are unlikely to hold for facilities such as shopping centers or cinema complexes. For the latter, it seems more reasonable to assume that demand is stochastic, as users would tend to, but not necessarily, patronize closer and larger facilities.

In this paper, we introduce a new model for siting and sizing facilities with probabilistic demand in the sense stated above. The model determines the locations and capacities of facilities, that maximize the (expected) demand attracted or covered by the facilities, assuming that the larger the facility is, the larger is the demand it attracts, and where the expected demand received by each facility exceeds a given threshold. The optimization model combines ingredients from elastic-demand location models (see e.g., [8]) and spatial-interaction location models (see e.g., [9]). It is different from the models described in [10] and [11], which also aim at determining optimum facility locations and capacities within a spatial-interaction framework, but ignore elastic-demand issues. The model addressed in [12] considers these types of issues (the location of a new facility is assumed to expand the market), but do not determine optimal facility capacities. To the best of our knowledge, the most similar model to the one dealt with in this paper is due to [13], but this does not consider a minimum demand threshold.

The proposed model is formulated as a Mixed-Integer Nonlinear Programming Problem (MINLP), which is quite difficult to tackle. A branch-and-bound algorithm is designed for finding a global minimum to this MINLP and is proven to converge

to such a solution. Furthermore, a finite procedure is developed for computing a first incumbent solution, which helps to reduce the overall effort of the algorithm. Computational experience, with a GAMS/MINOS implementation of the algorithm, is reported that reveals the efficacy of the algorithm to solve models having up to 20 centers. Moreover, the algorithm is shown to be efficient and compares favorably with the well-known MINLP commercial code BARON [14] for the solution of the same problems.

The remainder of this paper as organized as follows. The problem description and model formulation are presented in Sect. 2. Section 3 provides an illustrative example, and Sect. 4 develops the proposed branch-and-bound algorithm. A finite procedure for determining a good quality incumbent solution is discussed in Sect. 5. Computational experience is reported in Sect. 6, and, finally, Sect. 7 provides a summary and conclusions of the paper.

2 Optimization Model

The optimization model to be introduced in this paper represents problems involving the siting and sizing of facilities under probabilistic demand and is predicated on the following assumptions:

1. Demand for the services offered by the facilities is concentrated at a finite set, say J, of centers.
2. The location of facilities is restricted to a set, say N, of candidate sites. At most one facility can be located at each site.
3. The *potential* demand (or number of potential users) from each center $j \in J$ is known and denoted by u_j. The *actual* demand from each center for the services offered at a facility located at site k, $k \in N$, increases with the size, say z_k, of the facility and decreases with the cost, say d_{jk}, for traveling to the facility (when a center j coincides with a facility location, we take $d_{jk} = \epsilon_0 > 0$, where ϵ_0 is a small positive quantity).
4. The size of a facility is proportional to the expected demand it attracts, and must exceed a given threshold, say $z_{\min}$.
5. The objective is to maximize the total expected demand attracted (covered) by the facilities.

For the sake of simplicity, we assume without loss of generality that $J \equiv N$, where a facility can be explicitly prohibited from being located at a non-viable site, and where the potential demand can be taken as zero at a non-center. Assumption 3 describes the spatial choice behavior of facility users. According to this, the probability, say p_{jk}, of a user (or the proportion of users) from demand center j patronizing a facility located at site k is given by

$$p_{jk} = \left(1 - \alpha d_{jk}^{\beta}\right) \frac{z_k d_{jk}^{-\gamma}}{\sum_{i \in N} z_i d_{ji}^{-\gamma}} \tag{1}$$

where $\alpha, \beta, \gamma > 0$ are calibration parameters, which can be estimated from real-world data using maximum likelihood methods [15] and guarantee that $p_{jk} \geq 0$ given the distance data (e.g., $\alpha \equiv D^{-\beta}$, where $D \equiv \max_{j,k}\{d_{jk}\}$).

In the formula (1) the first factor $(1 - \alpha d_{jk}^{\beta})$ is a demand decay term and the second ratio factor is a demand proportionality split term. These expressions are usually used in the literature [13] and fit empirical data extremely well. Together, they capture three essential stylized facts about the demand for the services provided by the facilities: (1) not all potential demand materialize into actual demand; (2) for equally sized facilities, the closer the facility is the larger is the demand for service; (3) for equally distant facilities, the larger the facility is the larger is the demand for service.

Assumption 4 asserts that in our model, and unlike the vast majority of the spatial interaction location models described in the literature, the attractiveness of a facility is endogenously determined, being related with demand according to:

$$z_k = \theta \sum_{j \in J} u_j p_{jk}, \quad \forall k \in N, \tag{2}$$

where θ is a parameter, that expresses the size per user (e.g., commercial floor space per person, number of cinema seats per person, etc). Note that (2) requires the size of the facility to accommodate the expected demand, where (1) implies that the probability of attracting demand is itself dependent on the size of the facility.

Given the assumptions stated above (recall that $J \equiv N$), the optimization model can be formulated as follows:

$$\textbf{P1}: \quad \text{Maximize} \quad \sum_{k \in N} z_k \tag{3}$$

$$\text{subject to} \quad z_k \leq U y_k, \quad \forall k \in N, \tag{4}$$

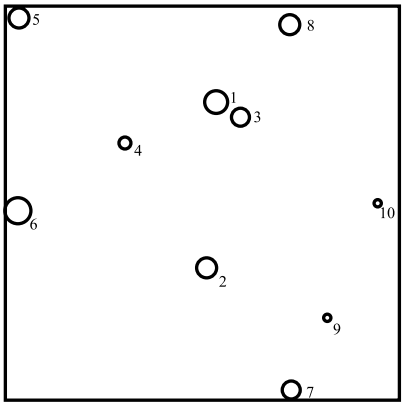
$$z_k \geq z_{\min} y_k, \quad \forall k \in N, \tag{5}$$

$$z_k = \theta \sum_{j \in N} u_j \left(1 - \alpha d_{jk}^{\beta}\right) \frac{z_k d_{jk}^{-\gamma}}{\sum_{i \in N} z_i d_{ji}^{-\gamma}}, \quad \forall k \in N, \tag{6}$$

$$y_k \in \{0, 1\} \text{ and } z_k \geq 0, \quad \forall k \in N, \tag{7}$$

where $U \equiv \sum_{j \in N} u_j$ is the total potential demand, and y_k is equal to 1, if a facility is located at site k and is equal to 0 otherwise, $\forall k \in N$.

The objective function (3) of this Mixed-Integer Nonlinear Programming Problem (MINLP) is concerned with the aggregate size of facilities. Thus the total expected demand attracted to the facilities is maximized (since the size of a facility is proportional to the expected number of users it attracts). Constraint (4) ensures that users visit center k only if there is a facility located at that center. Constraint (5) requires that the size of a facility located at center k exceeds the threshold level $z_{\min}$. Since the number of facilities to locate is not pre-defined and no facility costs are explicitly stated, (5) (along with (2)) plays an essential role in determining the number and location of facilities. Constraint (6) combines (1) and (2) and specifies that the size of a facility is proportional to the total number of users obtaining service from that facility. This constraint set is nonlinear and poses a principal challenge in solving the model. Finally, constraint (7) asserts logical restrictions on the decision variables.



Center	Potential demand
1	72
2	70
3	58
4	34
5	76
6	99
7	63
8	70
9	19
10	16
Total	577

Fig. 1 Example input data

3 An Illustrative Example

This section illustrates the type of results that can be obtained through the application of the model. Consider a region of 100 × 100 length units and having 10 population centers as depicted in Fig. 1, where the larger the circle in the figure, the larger is the potential demand at that center. There is one center (Center 6) that is significantly larger than all the others and there exist three noticeably small centers (Centers 4, 9, and 10). The total potential demand of the region is 577. Each population center is assumed to be a candidate site for locating a facility. The model parameters are given by $z_{\min} = 100$ and $\theta = 1$, with calibration parameters $\alpha = D^{-\beta}$, where $D = \max_{j,k \in N}\{d_{jk}\}$, $\beta = 1$, and $\gamma = 1$. The optimal solution consists of locating facilities at three centers: 1, 2, and 6 having sizes 199, 117, and 123, respectively (see Table 1). For every center where a facility is located ($d_{jk} = \epsilon_0 > 0$) almost all the potential demand is attracted to that facility. The users at centers without a coincident facility split their demand between the available facilities according to their proximity to facilities and the size of the facilities. The facilities attract a total expected demand of 439, which corresponds to 76% of the potential demand.

Next, we discuss the interpretation of the optimal solution as displayed in Fig. 2. Center 4 is at a central location in relation to the facilities. Since the distances between this center and each of the facilities are very similar, the size of the facility plays the leading role in demand attraction. Indeed, the largest facility receives 41% of the potential demand at Center 4 while the other two facilities attract only 15% and 18%, respectively. Now, consider the example of Center 7, which is located peripherally with respect to the facilities. The closest facility is the smallest among the three. However, because the other two facilities are much further away, Facility 2 attracts 29% of the potential demand at Center 7. The effect of the size of the facility is also noticeable, although Facilities 1 and 6 are almost equally distant from Center 7, Facility 1 receives 14% of the demand while Facility 6 attracts only 5%. For Center 8, the relative location of the center is similar to that of Center 7. However, in this case, the closest facility is the largest one (Facility 1). Therefore, the demand attracted by

Table 1 Optimum solution statistics

Center	Attracted demand							
	Facility 1		Facility 2		Facility 6		Total	
	Users	%	Users	%	Users	%	Users	%
1	69	96%	1	1%	0	–	70	97%
2	2	3%	65	93%	1	1%	68	97%
3	42	72%	6	10%	2	3%	50	86%
4	14	41%	5	15%	6	18%	25	74%
5	19	25%	5	7%	14	18%	38	50%
6	1	1%	1	1%	94	95%	96	97%
7	9	14%	18	29%	3	5%	30	48%
8	34	49%	6	9%	2	3%	42	60%
9	4	21%	7	37%	1	5%	12	63%
10	5	31%	3	19%	0	–	8	50%
Total	199	34%	117	20%	123	21%	439	76%

Facility 1 from Center 8 is much larger than that from Center 7. Indeed, Facility 1 attracts more demand from Center 8 than the three facilities do from Center 7. Finally, we illustrate the case of Center 10. Facilities 1 and 2 are equally distant from this center while Facility 6 is relatively further away. As a consequence, only Facilities 1 and 2 attract demand from Center 10, with Facility 1 enjoying a larger demand share (31% versus 19%) because it is the largest facility.

4 A Branch-and-Bound Algorithm

In this section we design a branch-and-bound algorithm for finding a global minimum to Problem **P1**. Since Constraint (5) requires that the size of any open facility located at center k must exceed the threshold level $z_{\min}$, then

$$\sum_{j \in N} y_j \leq V, \tag{8}$$

where

$$V \equiv \lfloor U/z_{\min} \rfloor \tag{9}$$

and where $\lfloor (.) \rfloor$ represents the largest integer smaller than or equal to $(.)$. Since for each j, k,

$$\frac{z_k d_{jk}^{-\gamma}}{\sum_{i \in N} z_i d_{ji}^{-\gamma}} \leq 1,$$

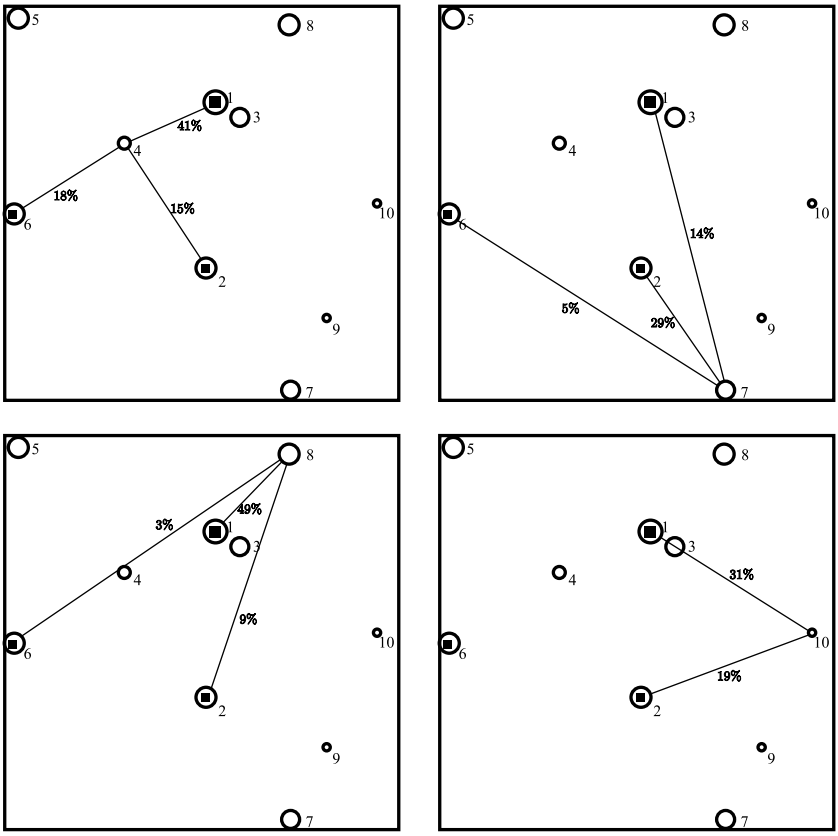


Fig. 2 Illustration of user spatial choice behavior

it follows from (6) that

$$z_k = \theta \sum_{j\in N} u_j\left(1-\alpha d_{jk}^{\beta}\right)\frac{z_k d_{jk}^{-\gamma}}{\sum_{i\in N} z_i d_{ji}^{-\gamma}} \le \theta \sum_{j\in N} u_j\left(1-\alpha d_{jk}^{\beta}\right), \quad \forall k \in N. \tag{10}$$

Defining

$$U_k \equiv \sum_{j\in N} \theta u_j\left(1-\alpha d_{jk}^{\beta}\right), \quad \forall k \in N, \tag{11}$$

we can improve the upper bounds in (4) to assert:

$$z_{\min} y_k \le z_k \le U_k y_k, \quad \forall k \in N. \tag{12}$$

Also, for convenience in notation, denote

$$a_{ji} \equiv d_{ji}^{-\gamma}, \quad \forall i, j \in N, \quad \text{and} \quad b_{jk} \equiv \theta u_j\left(1-\alpha d_{jk}^{\beta}\right) d_{jk}^{-\gamma}, \quad \forall j, k \in N. \tag{13}$$

Then Problem **P1** can be written in the following equivalent form, where we have introduced certain auxiliary decision variables ϕ_j and ψ_j, $\forall j \in N$, for the sake of structural convenience:

$$\textbf{P2}: \quad \text{Maximize} \quad \sum_{k\in N} z_k \tag{14}$$

$$\text{subject to} \quad z_{\min} y_k \le z_k \le U_k y_k, \quad \forall k \in N, \tag{15}$$

$$z_k \left[1 - \sum_{j\in N} b_{jk}\phi_j \right] = 0, \quad \forall k \in N, \tag{16}$$

$$\psi_j = \sum_{i\in N} a_{ji} z_i, \quad \forall j \in N, \tag{17}$$

$$\phi_j \psi_j = 1, \quad \forall j \in N, \tag{18}$$

$$y_k \in \{0, 1\} \text{ and } z_k \ge 0, \quad \forall k \in N, \tag{19}$$

$$\sum_{j\in N} y_j \le V. \tag{20}$$

In order to reformulate **P2** into a more amenable form, we establish the following results.

Proposition 4.1 *Let* $P \equiv \max_{k\in N}\{U_k\} = U_{k^*}$. *Then* $(z, y, \phi, \psi) \equiv (\hat{z}, \hat{y}, \hat{\phi}, \hat{\psi})$ *with* $\hat{z}_{k^*} = P$, $\hat{y}_{k^*} = 1$, $\hat{z}_i = \hat{y}_i = 0$, $\forall i \ne k^*$, *and* $\hat{\psi}_j = a_{jk^*} z_{k^*} = 1/\hat{\phi}_j$, $\forall j \in N$, *is a feasible solution to Problem* ***P2*** *with an objective value equal to* P.

Proof The feasibility of the given solution to (15) and (17)–(20), as well as to (16) for $k \ne k^*$ is obvious. Moreover, in (16) for $k = k^*$, we have using (11) and (13) that,

$$\sum_{j\in N} b_{jk^*}\hat{\phi}_j = \frac{1}{P}\sum_{j\in N}\left[\frac{b_{jk^*}}{a_{jk^*}}\right] = \frac{1}{P}\sum_{j\in N}\theta u_j(1 - \alpha d_{jk^*}^{\beta}) = 1.$$

Hence, (16) is also satisfied for $k = k^*$. Moreover, the objective value equals $\hat{z}_{k^*} = P$. □

Proposition 4.2 *Let*

$$l_j^{\phi} \equiv \frac{1}{\sum_{i\in N} a_{ji} U_i}, \quad \forall j \in N, \quad \textit{and} \quad u_j^{\phi} \equiv \frac{1}{P \min_{i\in N}\{a_{ji}\}}, \quad \forall j \in N. \tag{21}$$

Then $l_j^{\phi} \le \phi_j \le u_j^{\phi}$, $\forall j \in N$, *in any optimal solution to Problem* ***P2***.

Proof By (15), (17) and (18), we have $\phi_j = [\frac{1}{\sum_{i\in N} a_{ji} z_i}] \ge l_j^{\phi}$, $\forall j \in N$. Moreover, since $\sum_{k\in N} z_k \ge P$ for any optimal solution to Problem **P2**, by Proposition 4.1,

we get

$$\phi_j \le \frac{1}{\min\{\sum_{i\in N} a_{ji} z_i : \sum_{i\in N} z_i \ge P, z \ge 0\}} = \frac{1}{P \min_{i\in N}\{a_{ji}\}} = u_j^{\phi}, \quad \forall j \in N. \qquad \square$$

Based on Proposition 4.2, and using (15) and (19), we can linearize Constraint (16) as follows: Define

$$L_k^{\phi} \equiv \min\left\{ \sum_{j\in N} b_{jk}\phi_j : l_j^{\phi} \le \phi_j \le u_j^{\phi}, \forall j \in N \right\}, \quad \forall k \in N, \tag{22}$$

$$U_k^{\phi} \equiv \max\left\{ \sum_{j\in N} b_{jk}\phi_j : l_j^{\phi} \le \phi_j \le u_j^{\phi}, \forall j \in N \right\}, \quad \forall k \in N. \tag{23}$$

Then we can rewrite (16) as

$$y_k + (1 - y_k)L_k^{\phi} \le \sum_{j\in N} b_{jk}\phi_j \le y_k + (1 - y_k)U_k^{\phi}, \quad \forall k \in N. \tag{24}$$

In fact, whenever $y_k = 0$, we have that $z_k = 0$ by (15), and then (16) holds trivially and (24) is redundant by (22)–(23). On the other hand, when $y_k = 1$, we have that $z_k > 0$ by (15), and in this case, (24) implies that $\sum_{j\in N} b_{jk}\phi_j = 1$, so that (16) again holds true.

Using the foregoing constructs, we can equivalently write Problem **P2** as the following problem $\mathbf{P}(\Omega)$, predicated on the bounding hyperrectangle Ω, where the bounds on the ψ-variables are derived directly from those on the ϕ-variables, based on (29).

$$\mathbf{P}(\Omega): \quad \text{Maximize} \quad \sum_{k\in N} z_k \tag{25}$$

$$\text{subject to} \quad z_{\min} y_k \le z_k \le U_k y_k, \quad \forall k \in N, \tag{26}$$

$$y_k + (1 - y_k)L_k^{\phi} \le \sum_{j\in N} b_{jk}\phi_j \le y_k + (1 - y_k)U_k^{\phi}, \quad \forall k \in N, \tag{27}$$

$$\psi_j = \sum_{i\in N} a_{ji} z_i, \quad \forall j \in N, \tag{28}$$

$$\phi_j \psi_j = 1, \quad \forall j \in N, \tag{29}$$

$$(y, \phi) \in \Omega \equiv \left\{ (y, \phi) : y \in \{0, 1\}^n, l_j^{\phi} \le \phi_j \le u_j^{\phi}, \forall j \in N \right\}, \tag{30}$$

$$\left(1/u_j^{\phi}\right) \le \psi_j \le \left(1/l_j^{\phi}\right), \quad \forall j \in N,\ z \ge 0, \tag{31}$$

$$\sum_{j \in N} y_j \le V. \tag{32}$$

We now design a branch-and-bound algorithm for finding a global minimum to Problem $\mathbf{P}(\Omega)$ given by (25)–(32). This procedure is based on partitioning Ω by branching on the binary y-variables as well as by splitting the intervals for the ϕ-variables as detailed below. Any node problem in this methodology, indexed by r, say, is predicated on the set Ω^r (in lieu of Ω) in (25)–(32), which is defined as follows:

$$\begin{aligned}\Omega^r \equiv \big\{ (y,\phi)&: y_k \equiv 0, \forall k \in J_0^r;\ y_k \equiv 1, \forall k \in J_1^r; \\ & y_k \in \{0,1\}, \forall k \in N^r \equiv N - \left(J_0^r \cup J_1^r\right); l_j^r \le \phi_j \le u_j^r, \forall j \in N\big\},\end{aligned} \tag{33}$$

where J_0^r and J_1^r are the index sets for the y-variables that are currently fixed at zero and one, respectively, with the remaining y-variables being free, and where $l_j^r \ge l_j^{\phi}$ and $u_j^r \le u_j^{\phi}, \forall j \in N$. At the initial node, or *root node*, we have $r = 1$ with $\Omega^1 \equiv \Omega$. For any other node r, the corresponding node problem is denoted by $\mathbf{P}(\Omega^r)$, and is given as follows:

$\mathbf{P}(\Omega^r)$: Maximize $\displaystyle\sum_{k \in N} z_k$ (34)

subject to $z_k = 0, \forall k \in J_0^r$, and $z_{\min} \le z_k \le U_k, \quad \forall k \in J_1^r$, (35)

$$z_{\min} y_k \le z_k \le U_k y_k, \quad \forall k \in N^r, \tag{36}$$

$$\sum_{j \in N} b_{jk}\phi_j = 1, \quad \forall k \in J_1^r, \tag{37}$$

$$y_k + (1 - y_k)L_k^r \le \sum_{j \in N} b_{jk}\phi_j \le y_k + (1 - y_k)U_k^r, \quad \forall k \in N^r, \tag{38}$$

$$\psi_j = \sum_{i \in N} a_{ji} z_i, \quad \forall j \in N, \tag{39}$$

$$\phi_j \psi_j = 1, \quad \forall j \in N, \tag{40}$$

$$(y,\phi) \in \Omega^r, \tag{41}$$

$$\left(1/u_j^r\right) \le \psi_j \le \left(1/l_j^r\right), \quad \forall j \in N,\ z \ge 0, \tag{42}$$

$$\sum_{j \in N} y_j \le V, \tag{43}$$

where L_k^r and $U_k^r, \forall k \in N^r$ are derived below based on Ω^r. Toward this end, for any node r, we first tighten Ω^r as possible based on the valid restrictions

$$v^* \le \sum_{k \in N} z_k \le v_r, \tag{44}$$

where ν^* is the current best-known objective value, and ν_r is a known (initial) upper bound on the objective value of $\mathbf{P}(\Omega^r)$. To begin with, for $r = 1$, we have by Proposition 4.1 that

$$\nu^* = P \quad \text{and} \quad \nu_r = U \equiv \sum_{j \in N} u_j. \tag{45}$$

In general, we can take ν_r as the upper bound computed by solving the upper-bounding linear programming relaxation $LP(.)$ defined in the sequel for the *parent node* of the current node r, and ν^* is, as always, the current overall incumbent solution value. Accordingly, based on the definitions of ψ_j in (39) and Ω^r in (33), and using (35), (36) and (40), we compute

$$\hat{l}_j^r = \frac{1}{\max\{\sum_{i \in N} a_{ji} z_i : \sum_{k \in N} z_k \le \nu_r, (35), 0 \le z_k \le U_k, \forall k \in N^r\}}, \tag{46}$$

$$\hat{u}_j^r = \frac{1}{\min\{\sum_{i \in N} a_{ji} z_i : \sum_{k \in N} z_k \ge \nu^*, (35), 0 \le z_k \le U_k, \forall k \in N^r\}} \tag{47}$$

and then we replace the bounds on the ϕ-variables within Ω^r by

$$l_j^r \leftarrow \max\{l_j^r, \hat{l}_j^r\}, \quad \forall j \in N, \quad \text{and} \quad u_j^r \leftarrow \min\{u_j^r, \hat{u}_j^r\}, \quad \forall j \in N. \tag{48}$$

In case $l_j^r > u_j^r$ for any $j \in N$, i.e., the resulting $\Omega^r = \emptyset$, we *fathom* node r. Otherwise, similar to (22)–(23), we compute

$$L_k^r \equiv \min\left\{\sum_{j \in N} b_{jk} \phi_j : l_j^r \le \phi_j \le u_j^r, \forall j \in N\right\}, \quad \forall k \in N, \tag{49}$$

$$U_k^r \equiv \max\left\{\sum_{j \in N} b_{jk} \phi_j : l_j^r \le \phi_j \le u_j^r, \forall j \in N\right\}, \quad \forall k \in N. \tag{50}$$

Using the values given by (46)–(48) and (49)–(50) in (34)–(43), produces the *pre-processed node problem* $\mathbf{P}(\Omega^r)$.

Next, we compute an upper-bounding linear programming relaxation for $\mathbf{P}(\Omega^r)$, denoted by $\mathbf{LP}(\Omega^r)$, by constructing a polyhedral outer-approximation for the hyperbolic constraints (40) and relaxing the binary restrictions on the free y-variables as explained next.

Consider the relationship $\phi_j \psi_j = 1$ in (40) for any $j \in N$ over the interval $l_j^r \le \phi_j \le u_j^r$. Figure 3 depicts this relevant segment of the hyperbolic curve. In lieu of this nonlinear equation, we use a polyhedral outer-approximation to this curve segment by examining the concave envelope chord (see Fig. 3) along with some three tangents, constructed at the end-points l_j^r and u_j^r, and at the geometric mean $\phi = \sqrt{l_j^r u_j^r}$, the last of which yields a tangent that is parallel to the chord (see Fig. 3). This polyhedral approximation is defined as follows:

$$\phi_j + (l_j^r u_j^r)\psi_j \le l_j^r + u_j^r, \quad \forall j \in N, \tag{51}$$

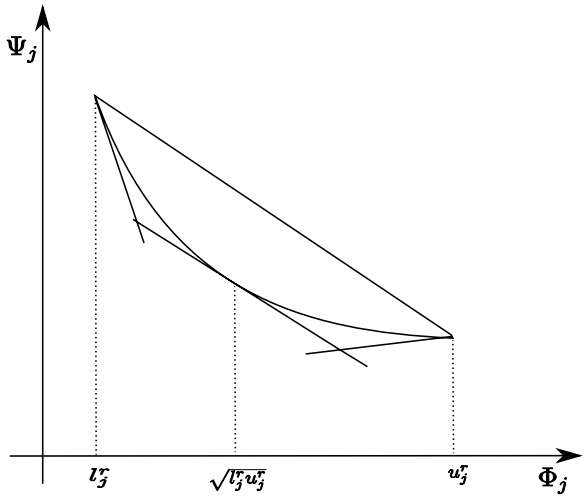


Fig. 3 Polyhedral outer-approximation

$$\phi_j + (l_j^r)^2 \psi_j \geq 2l_j^r, \quad \forall j \in N, \tag{52}$$

$$\phi_j + (l_j^r u_j^r)\psi_j \geq 2\sqrt{l_j^r u_j^r}, \quad \forall j \in N, \tag{53}$$

$$\phi_j + (u_j^r)^2 \psi_j \geq 2u_j^r, \quad \forall j \in N. \tag{54}$$

Then the linear programming relaxation is given by

$$\begin{aligned} \mathbf{LP}(\Omega^r): \quad \text{Maximize} \Big\{ \sum_{k \in N} z_k : (35, 36, 37, 38, 39, \\ 42, 43), (51, 52, 53, 54), (y, \phi) \in \bar{\Omega}^r \Big\} \end{aligned} \tag{55}$$

where $\bar{\Omega}^r$ is the continuous relaxation of Ω^r defined by

$$\begin{aligned} \bar{\Omega}^r \equiv \big\{ (y, \phi) &: y_k = 0, \forall k \in J_0^r; \; y_k = 1, \forall k \in J_1^r; \\ & 0 \leq y_k \leq 1, \forall k \in N^r; \; l_j^r \leq \phi_j \leq u_j^r, \forall j \in N \big\}. \end{aligned} \tag{56}$$

Remark 1 Note that the solution to LP(Ω^r) gives a tighter upper bound for Problem P(Ω^r) than ν_r, which could be further used in (46) to possibly tighten the bounds on the ϕ-variables, and hence the subsequent relaxation LP(.). Although this looping is possible (which could be continued so long as the bounds improve by at least some percentage tolerance), we do not perform this for the sake of algorithmic simplicity.

Let $\nu[.]$ denote the optimal objective function value of any problem [.]. If $\nu[LP(\Omega^r)] \leq \nu^*(1+\epsilon)$, for some tolerance $\epsilon \geq 0$, then we fathom node r. Furthermore, in case the optimal solution obtained for LP(Ω^r) satisfies the relaxed constraints (40) and the binary restrictions on the y-variables in Problem P(Ω^r), then

this solution is also optimal to the latter problem, and so, we update the incumbent solution if necessary and fathom node r. Otherwise, node r is declared *active*, in that $\nu[LP(\Omega^r)] > \nu^*(1+\epsilon)$, and then we partition this node problem using the following Branching Rule.

Branching Rule

Given: Active node r, with optimal solution $(z^r, y^r, \phi^r, \psi^r)$ to Problem LP(Ω^r). Let R denote the current total number of nodes generated.
Partitioning of Ω^r:

Step 1 If y^r is binary, go to Step 2. Otherwise, find

$$k^* \in \arg\min_{k \in N} |y_k^r - 0.5|$$

and partition Ω^r into two subsets Ω^{R+1} and Ω^{R+2} based on restricting y_{k^*} equal to 0 and 1, respectively, within each subset. Exit this rule.

Step 2 Find

$$j^* \in \arg\max_{j \in N} |\phi_k^r \psi_k^r - 1.0|$$

and partition Ω^r into two subsets Ω^{R+1} and Ω^{R+2} based on bounding ϕ_{j^*} within the intervals $[l_{j^*}^r, \phi_{j^*}^r]$ and $[\phi_{j^*}^r, u_{j^*}^r]$ respectively. Exit this rule.

The steps of the proposed branch-and-bound algorithm are presented next. Let

k: the index for the current upper bounding problem under analysis;
ν^*: the best known lower-bound;
L: the queue of indices of active subproblems;
R: the number of nodes generated besides the root node;
$LP(\Omega^r)$: the upper bounding problem for node r;
ϵ: the specified optimality tolerance ($\epsilon \geq 0$);
$\nu(.)$: the optimal value of Problem (.).

Branch-and-Bound Algorithm (B&B)

Step 0 (Initialization): Let $R = 1, k = 0, L = \{1\}$, and $\epsilon \geq 0$. Let $\nu^* = P$ be the value obtained by Proposition 4.1 or computed via the procedure described in Section 5.
Step 1 (Choice of node): If $L = \emptyset$ then stop; otherwise find k

$$k \in \arg\max_{r \in L} \{\nu[LP(\Omega^r)]\}.$$

Set $L \leftarrow L - \{k\}$.
Step 2 (Branching rule): Partition Ω^k into two subnodes Ω^{R+1} and Ω^{R+2} based on the Branching Rule.
Step 3 (Solve, Update, and Queue): Set $i = 1$.

(i) Solve Problem $LP(\Omega^{R+i})$.

(ii) If $\nu[LP(\Omega^{R+i})] > \nu^*(1+\epsilon)$ then set $L \leftarrow L \cup \{R+i\}$ and go to (iii); otherwise go to (iv).

(iii) If the optimal solution to $LP(\Omega^{R+i})$ is feasible to $P(\Omega^{R+i})$, update ν^* according to

$$\nu^* \leftarrow \max\{\nu^*, \nu[LP(\Omega^{R+i})]\}.$$

If ν^* changes remove all indices $t \in L$ for which $\nu[LP(\Omega^t)] \leq \nu^*(1+\epsilon)$.

(iv) If $i = 2$ then set $R \leftarrow R+2$ and go to Step 1; otherwise, let $i = 2$ and go to (i).

The following result establishes the convergence of the algorithm.

Proposition 4.3 *Consider Algorithm B&B run with $\epsilon = 0$. Then either this algorithm terminates finitely with the incumbent solution being optimal to Problem $P(\Omega)$, or else, an infinite branch-and-bound tree is generated such that along any infinite branch of the tree, any accumulation point of the sequence of solutions generated for the corresponding relaxations $LP(.)$ solves $P(\Omega^r)$.*

Proof The case of finite convergence is obvious by the validity of the upper and lower bounding strategies. Hence, suppose that an infinite branch-and-bound tree is generated, and consider any infinite branch. Define each step of selecting a node to partition (following the greatest upper bound rule) as a *stage*, and index the stages consecutively as $s = 1, 2, \ldots$. In particular, denote the sequence of nested Ω-sets along the identified infinite branch as $\{\Omega^{r(s)}\}$ for $s \in S$, and let $\xi^{r(s)} \equiv (z^{r(s)}, y^{r(s)}, \phi^{r(s)}, \psi^{r(s)})$ be the optimal solution obtained for Problem LP($\Omega^{r(s)}$), $\forall s \in S$, with $[l^{r(s)}, u^{r(s)}]$ defining the vector of bounds on the ϕ-variables within the corresponding set $\Omega^{r(s)}$, $\forall s \in S$. By taking any convergent subsequence if necessary (by the boundedness of the sequences generated), assume without loss of generality that

$$\left\{\xi^{r(s)}, l^{r(s)}, u^{r(s)}\right\}_S \rightarrow \left(\xi^*, l^*, u^*\right). \tag{57}$$

We need to prove that ξ^* solves Problem P(Ω).

First, by the greatest upper-bound node selection rule, we have that

$$\nu[P(\Omega)] \leq \nu[LP(\Omega^{r(s)})] = \sum_{k \in N} z_k^{r(s)}, \quad \forall s \in S. \tag{58}$$

Next, observe that over the infinite sequence of partitions $\{\Omega^{r(s)}\}_S$, we could have branched on the y-variables only finitely often, and that by the preference given to branching first on a fractional y-variable if necessary in the Branching Rule, we must have $y^{r(s)}$ binary for $s \in S$ large enough, so that by (57), we get that

$$y^* \quad \text{is binary-valued.} \tag{59}$$

Furthermore, there exists some ϕ_p, $p \in N$, which is branched on infinitely often for $s \in S_1 \subseteq S$, say, by splitting its interval at the corresponding solution value $\phi_p^{r(s)}$ for such $s \in S_1$. As a result of this process, in the limit, we must have $\phi_p^* = l_p^*$ or $\phi_p^* = u_p^*$. In either case, by (51)–(54), we therefore obtain

$$\phi_p^* \psi_p^* = 1. \tag{60}$$

But Step 2 of the Branching Rule asserts that

$$\left|\phi_p^{r(s)}\psi_p^{r(s)} - 1\right| \geq \left|\phi_j^{r(s)}\psi_j^{r(s)} - 1\right| \geq 0, \quad \forall j \in N, \text{ for each } s \in S_1. \tag{61}$$

Taking limits in (61) as $s \to \infty$ for $s \in S_1$, and using (57) and (60), we get

$$\phi_j^*\psi_j^* = 1, \quad \forall j \in N. \tag{62}$$

Therefore, by (59) and (62), we get that ξ^* is feasible to $P(\Omega^*)$, and hence to $P(\Omega)$ because $\Omega^* \subseteq \Omega$. Consequently,

$$\sum_{k\in N} z_k^* \leq \nu[P(\Omega)]. \tag{63}$$

Finally, taking limits in (58) as $s \to \infty$, $s \in S$, we get

$$\nu[P(\Omega)] \leq \sum_{k\in N} z_k^*,$$

which in concert with (63) asserts that $\nu[P(\Omega)] = \sum_{k\in N} z_k^*$. Since ξ^* is feasible to $P(\Omega)$ from above, we have that ξ^* solves $P(\Omega)$. □

5 A Finite Procedure for Computing Lower-Bounds

The efficiency of the branch-and-bound algorithm strongly depends on its ability to find good lower and upper bounds. In this section we introduce an effective procedure for computing an initial lower-bound for the MINLP Problem **P1** given by (3)–(7). This procedure can also be useful in the implementation of the branching rule of the branch-and-bound algorithm discussed in the previous section. To describe this method, we first introduce the following Nonlinear Programming Problem:

$$\textbf{NLP}(I): \quad \text{Maximize} \quad \sum_{k\in I} z_k \tag{64}$$

$$\text{subject to} \quad 0 \leq z_k \leq U_k, \quad \forall k \in I, \tag{65}$$

$$z_k = \theta \sum_{j\in N} u_j\left(1 - \alpha d_{jk}^{\beta}\right)\frac{z_k d_{jk}^{-\gamma}}{\sum_{i\in I} z_i d_{ji}^{-\gamma}}, \quad \forall k \in I, \tag{66}$$

where initially, we have $I \equiv N$. Let $\bar{z}$ be a KKT point [16] for this nonlinear program. If

$$\bar{z}_k \geq z_{\min} \quad \text{or} \quad \bar{z}_k = 0 \quad \text{for all } k \in N, \tag{67}$$

then consider the set of variables $\bar{y}_k$ given by

$$\bar{y}_k = \begin{cases} 1 & \text{if } \bar{z}_k \geq z_{\min}, \\ 0 & \text{if } \bar{z}_k = 0, \ \forall k \in N. \end{cases} \tag{68}$$

Hence $(\bar{z}, \bar{y})$ is a feasible solution to Problem **P1** and provides a lower bound $\sum_{k\in N} \bar{z}_k$ for the branch-and-bound algorithm. If $\bar{z}$ does not satisfy (67), consider the set of violations

$$I_V = \{k \in I : 0 < \bar{z}_k < z_{\min}\}. \tag{69}$$

Note that for each $I \subseteq N$, $|I_V| = 0$ if and only if $\bar{z} = (\bar{z}_I, 0)$ and $\bar{y}$ is a feasible solution for Problem **P1**, where $\bar{z}_I$ is a KKT point for Problem NLP(I) and $\bar{y}$ is given by (68). The idea of the procedure is to compute a finite number of KKT points of Problems NLP(I) such that $|I|$ strictly reduces each time until we attain $|I_V| = 0$. To accomplish this, we update the set I at each iteration by a simple formula related to the values of the variables in the KKT solution computed for the previous iteration. The steps of this procedure are stated below:

Finite Lower Bounding Procedure (LB)

Initial Step: Set $I = N$ and compute a KKT point for NLP(I). If $\bar{z}$ satisfies Condition (67), go to Exit. Otherwise, reset I as the set of indices of the most positive V components of $\bar{z}$, where V is defined by (9).

General Step: Find a KKT point for NLP(I). If $\bar{z} = (\bar{z}_I, 0)$ satisfies Condition (67), go to Exit. Otherwise, update $I \leftarrow I - \{k^*\}$, where $k^* \in \arg\min_{k\in I} \bar{z}_k$. Repeat.

Exit: $\bar{z} = (\bar{z}_I, 0)$ and $\bar{y}$ given by (68) is a feasible solution to Problem **P1** and provides the lower bound $\sum_{k\in N} \bar{z}_k$.

This procedure has two updating rules for the set I, which we have found efficient in practice, although other updating rules may be used as well. Each iteration of the method requires the computation of a KKT point for the nonlinear program NLP(I), for which we have utilized the well-known code MINOS [17].

This technique can also be used to find new incumbent solutions and respective lower bounds within Algorithm B&B. Consider an active node r, with optimal solution $(z^r, y^r, \phi^r, \psi^r)$ to Problem LP(Ω^r). If this solution is feasible to Problem **P2** then the node is fathomed and the lower bound is updated. In practice, the feasibility check is implemented by using some tolerances, and rounding errors may fail to recognize that a feasible solution for the MINLP is at hand. We therefore applied Procedure LB to alleviate this impediment. The implementation of the branching rule with tolerance checks and using Procedure LB is described next.

Implementation of the Branching Rule

Given: Active node r, with optimal solution $(z^r, y^r, \phi^r, \psi^r)$ to Problem LP(Ω^r). Let L denote the current total number of nodes generated and v^* the current lower bound.

Partitioning of Ω^r:

Step 1 (*Branching on the Binary Variables*):
If $|y^r_k| < 10^{-3}$ or $|y^r - 1| < 10^{-3}$ for all $k \in N$, then y^r is declared binary (rounded if necessary); go to Step 2. Otherwise, find

$$k^* \in \arg\min_{k\in N} |y^r_k - 0.5|$$

and partition Ω^r into two subsets Ω^{R+1} and Ω^{R+2} based on restricting y_{k^*} equal to 0 and 1, respectively, within each subset. Exit this rule.

Step 2 (*Feasible Solution*):

(i) If $|\phi_k^r \psi_k^r - 1.0| \leq 10^{-3}$ for all $k \in N$, then (z^r, y^r) is declared as feasible solution for the MINLP and we set $\nu^* \leftarrow \max\{\nu^*, \sum_{k\in N} z_k^r\}$. Exit this rule.

(ii) If $|\phi_k^r \psi_k^r - 1.0| \leq 10^{-1}$ for all $k \in N$, then apply Procedure LB to obtain a feasible solution $(\bar{z}^r, \bar{y}^r)$ for the MINLP and set $\nu^* \leftarrow \max\{\nu^*, \sum_{k\in N} z_k^r\}$. Exit this rule. Otherwise, go to Step 3 with the previous solution $(z^r, y^r, \phi^r, \psi^r)$.

Step 3 (*Branching on the Continuous Variables*):

Let

$$j^* \in \arg\max_{j\in N} |\phi_k^r \psi_k^r - 1.0|,$$

and partition Ω^r into two subsets Ω^{R+1} and Ω^{R+2} based on bounding ϕ_{j^*} within the intervals $[l_{j^*}^r, \phi_{j^*}^r]$ and $[\phi_{j^*}^r, u_{j^*}^r]$, respectively, within each subset. Exit this rule.

6 Computational Experience

In this section we report some computational experience with the proposed algorithm discussed in Sects. 4 and 5. All the tests have been performed on a Pentium IV (Intel) with Hyperthreading, 3.0 GHz CPU, 2 GB RAM computer, using the operating system Linux. The branch-and-bound method was implemented in the General Algebraic Modeling System (GAMS) language (Rev 118 Linux/Intel) [18] and the LP solver CPLEX (Version 9.1) [19] has been used to compute the upper bounds required at each node. The Procedure LB described in Sect. 5 was implemented by using the solver MINOS (Version 5.51) [17]. Both Procedure LB and the branch-and-bound algorithm B&B were tested using a set of instances. The instances refer to a region of 100 by 100 length units, and differ with respect to the number of centers (we assume that centers coincide with candidate sites), the location of the centers, and the number of potential users at each center. We consider two sets of problems, one with 10-center instances and the other with 20-center instances (abbreviated Prob10c# and Prob20c#, for $\# = 1, \ldots, 10$). The center coordinates were determinated by generating random numbers uniformly over the interval [0, 100]. The number of potential users at each center was generated uniformly over the discrete interval [10, 100]. The minimum level of demand ($z_{\min}$) that a facility must satisfy was assumed to be 100 for the 10-center instances and 200 for the 20-center instances, and the size of facility per user (θ) was assumed to be 1. The following calibration parameters were considered: $\alpha = \frac{1}{\max\{d_{ij}:\forall i,j\in N\}}$, $\beta = 1$ and $\gamma = 1$. These values are within the range generally identified for such parameters. Furthermore, the optimality tolerance ϵ in the branch-and-bound algorithm was set to 10^{-2}.

For the sake of comparison we also solved all the problems using the well-known commercial code BARON [14] (version 22.7.2) with its default options. We note that this latter procedure was developed for determining global optima of nonlinear and mixed-integer nonlinear programs.

Table 2 Computational results for Procedure LB

Problem	Init	IT	LB	Opt
Prob10c1	284.000	4	284.1923	300.1812
Prob10c2	325.000	4	342.8279	383.0505
Prob10c3	231.000	4	231.000	250.1113
Prob10c4	250.000	4	294.6108	294.6108
Prob10c5	384.000	4	405.4829	405.4829
Prob10c6	307.000	5	338.6385	341.2855
Prob10c7	440.000	4	504.9324	504.9324
Prob10c8	454.000	6	504.5034	515.7262
Prob10c9	252.000	3	329.0000	329.0000
Prob10c10	428.000	4	541.6011	541.6011
Prob20c1	717.000	4	790.3244	790.3244
Prob20c2	681.000	4	699.8012	702.1358
Prob20c3	823.000	3	922.6824	922.6824
Prob20c4	702.000	5	730.2478	744.9917
Prob20c5	737.000	5	763.6925	783.2329
Prob20c6	815.000	4	878.5657	888.5200
Prob20c7	947.000	5	986.3995	996.2900
Prob20c8	773.000	4	848.6399	860.1542
Prob20c9	786.000	4	843.2991	853.9300
Prob20c10	755.000	4	839.8316	839.8316

The computational results of the performances of the Procedure LB, Algorithm B&B, and BARON (with default options) are displayed in Tables 2, 3, and 4, respectively. In these tables, Init, LB, and Opt denote the value of the objective function at the feasible solutions given by Proposition 4.1, Procedure LB, and the branch-and-bound algorithm B&B (global maximum). Furthermore, IT is the number of iterations required by the Procedure LB, and Nodes and CPU represent the number of nodes enumerated and the CPU time consumed by Algorithm B&B and BARON up to termination. Finally, BestNode expresses the number of the node where the solution of value Opt was found, and ITpivot represents the total number of pivotal operations performed by Algorithm B&B.

Table 2 reports the experiments with Procedure LB. The numerical results show that this method is always able to find a good lower bound with a small effort. In fact, the procedure found a global optimum in 40% of the tests and always provided an initial lower bound that is close to the global optimum (with 98.2% of optimality). Furthermore, it required a few ($\leq$6 and typically 4) number of iterations to find an initial lower bound (recall that each iteration of the algorithm essentially amounts to finding a KKT point by using MINOS).

The numerical results with the proposed branch-and-bound algorithm B&B are displayed in Table 3. This algorithm always found a global optimum in a reasonable amount of effort. The incorporation of Procedure LB improved the efficiency of the

Table 3 Computational results for the branch-and-bound algorithm

Problem	Opt	Nodes	BestNode	ITpivot	CPU	$z_{\min}$	V
Prob10c1	300.1812	320	320	8312	27.0	100	439
Prob10c2	383.0505	368	368	9519	30.4	100	489
Prob10c3	250.1113	158	158	3225	10.6	100	367
Prob10c4	294.6108	112	0	2452	8.0	100	419
Prob10c5	405.4829	202	0	4717	14.4	100	521
Prob10c6	341.2855	246	0	5903	18.2	100	512
Prob10c7	504.9324	406	0	11447	30.6	100	613
Prob10c8	515.7262	1179	1179	37130	93.1	100	700
Prob10c9	329.0000	152	0	3684	11.1	100	423
Prob10c10	541.3822	598	0	15634	45.9	100	655
Prob20c1	790.3244	3790	0	128150	199.2	200	1090
Prob20c2	702.1358	990	826	40142	48.4	200	926
Prob20c3	922.6824	4370	0	140102	219.6	200	1184
Prob20c4	744.9917	908	768	38113	44.8	200	1027
Prob20c5	783.2329	2748	2248	113805	136.4	200	1076
Prob20c6	888.5200	3944	3174	158050	193.2	200	1094
Prob20c7	996.2900	4156	3218	177294	207.2	200	1333
Prob20c8	860.1542	3292	2498	129225	162.4	200	1111
Prob20c9	853.9300	4192	3708	166612	205.2	200	1121
Prob20c10	839.8316	2698	0	107029	132.8	200	1152

branch-and-bound method significantly. As stated before, the Procedure LB usually finds a good quality lower bound that allows to prune the enumerative tree effectively. Furthermore, the use of this procedure in the branching rule, as explained in Sect. 5, enabled the algorithm to detect new lower bounds that would not otherwise have been found readily. It is also interesting to note that whenever the initial lower bound was smaller than the optimal value, the branch-and-bound algorithm usually detected the optimal solution toward the end of the tree search (see the columns Nodes and BestNode).

Table 4 reports the experimental results for solving the same test problems with BARON. The results show that the proposed branch-and-bound algorithm is more efficient than BARON, particularly for the 20-center instances proposed. The average CPU time consumed by BARON for the 10 and 20 centers problems were 41.32 and 17709.98, respectively, as compared with 28.91 and 154.92 for the branch-and-bound algorithm B&B. It is important to note, however, that BARON is a general purpose code, although it implements several sophisticated preprocessing and model tightening strategies.

Table 4 Computational results for BARON

Problem	Opt	Nodes	BestNode	CPU	$z_{\min}$	V
Prob10c1	300.1812	906	361	9.3	100	439
Prob10c2	383.0505	1975	19	20.5	100	489
Prob10c3	250.1113	874	739	7.9	100	367
Prob10c4	294.6108	770	631	8.8	100	419
Prob10c5	405.4829	2367	388	30.7	100	521
Prob10c6	341.2855	1682	514	21.0	100	512
Prob10c7	504.9324	3556	3106	49.5	100	613
Prob10c8	515.7262	9741	3034	161.3	100	700
Prob10c9	329.0000	1360	1036	14.6	100	423
Prob10c10	541.3822	5849	5347	89.7	100	655
Prob20c1	790.3244	187579	134785	11922.6	200	1090
Prob20c2	702.1358	62537	57997	3531.7	200	926
Prob20c3	922.6824	203342	26722	14520.2	200	1184
Prob20c4	744.9917	72503	67213	4869.2	200	1027
Prob20c5	783.2329	135395	1	7523.3	200	1076
Prob20c6	888.5200	367025	364564	20104.4	200	1094
Prob20c7	996.2900	879006	49726	64080.3	200	1333
Prob20c8	860.1542	334286	317224	17713.0	200	1111
Prob20c9	853.9300	331976	141382	19586.2	200	1121
Prob20c10	839.8316	188816	87967	13249.1	200	1152

7 Conclusions

In this paper, we have introduced a facility location model with distance-sensitive and size-sensitive demands. The purpose of the model is to determine the optimal number, location, and size of facilities so as to maximize the total expected demand attracted to the facilities. The model assumes that the demand for service increases as the distance to the facility decreases, and it increases with the size of the facility. The size of a facility is endogenously determined and is proportional to the number of users visiting the facility. This feature makes the model nonlinear (in addition to being mixed-integer) and, hence, difficult to solve. The model also considers that facilities must satisfy a minimal level of demand (facilities are not economically viable below this level).

A branch-and-bound algorithm has been designed for finding a global minimum to this optimization problem. This procedure is based on a reformulation and polyhedral outer approximation of the nonlinear constraints of the problem. The tree search procedure is based on partitioning the domain of the original variables and incorporates effective techniques for obtaining upper bounds at each node of the enumeration tree. Theoretical convergence of the proposed branch-and-bound algorithm has been established. An efficient procedure for computing lower bounds has also been integrated within the algorithm, while recognizing finite precision computations, and has been

demonstrated to significantly reduce the total search in the tree for the computation of a global minimum. Computational experience was reported, which demonstrates that the algorithm is efficient and performs in general better than the well-known commercial code BARON in practice, reducing the effort required by 30.03% and 99.13% on average, for the 10 and 20 center problems, respectively.

References

1. Daskin, M.: Network and Discrete Location: Models, Algorithms and Applications. Wiley, New York (1995)
2. Current, J.D.M., Schilling, D.: Discrete network location models. Facility Location: Applications and Theory, pp. 81–118 (2002)
3. Holmberg, K.: Facility location problems with spatial interaction. In: Floudas, Pardalos (eds.) Encyclopedia of Optimization, 2nd edn., pp. 982–989. Springer, Berlin (2009)
4. ReVelle, C., Eiselt, H.: Location analysis: a synthesis and survey. Eur. J. Oper. Res. **165**(1), 1–19 (2005)
5. Tuy, H.: Global optimization in location problems. In: Floudas, Pardalos (eds.) Encyclopedia of Optimization, 2nd edn., pp. 1354–1359. Springer, Berlin (2009),
6. Krarup, J., Pruzan, P.M.: Simple plant location problem: survey and synthesis. Eur. J. Oper. Res. **12**(1), 36–81 (1983)
7. Mirchandani, P.: The p-median problem and generalizations. Discrete Location Theory **1**, 55–117 (1990)
8. Perl, J., Ho, P.: Public facilities location under elastic demand. Transp. Sci. **24**(2), 117–136 (1990)
9. OKelly, M.: Spatial interaction based location-allocation models. Spatial analysis and location-allocation models, pp. 302–326 (1987)
10. Eiselt, H., Laporte, G.: The maximum capture problem in a weighted network. J. Reg. Sci. **29**(3), 433–439 (1989)
11. Plastria, F., Carrizosa, E.: Optimal location and design of a competitive facility. Math. Program. **100**(2), 247–265 (2004)
12. Berman, O., Krass, D.: Locating multiple competitive facilities: spatial interaction models with variable expenditures. Ann. Oper. Res. **111**(1), 197–225 (2002)
13. Aboolian, R., Berman, O., Krass, D.: Competitive facility location and design problem. Eur. J. Oper. Res. **182**(1), 40–62 (2007)
14. Sahinidis, N., Tawarmalani, M.: BARON: The GAMS Solver Manual. GAMS Development Corporation, Washington, DC, pp. 9–20 (2004)
15. Cascetta, E.: Transportation Systems Analysis: Models and Applications, 2nd edn. Springer, Berlin (2009)
16. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms, 3rd edn. Wiley-Interscience New York (2006)
17. Murtagh, B., Saunders, M., Murray, W., Gill, P., Raman, R., Kalvelagen, E.: MINOS-NLP. Systems Optimization Laboratory, Stanford University, Palo Alto, CA
18. Brooke, A., Kendrick, D., Meeraus, A., Raman, R.: GAMS—a user's guide. GAMS Development Corporation (1998)
19. CPLEX, I.: 11.0 Users Manual. ILOG SA, Gentilly, France (2008)