J Glob Optim (2008) 42:177–199
DOI 10.1007/s10898-007-9237-y

An interactive solution approach for a bi-objective semi-desirable location problem

Esra Karasakal · Deniz Nadirler

Received: 10 September 2006 / Accepted: 13 September 2007 / Published online: 12 October 2007

Abstract In this study, we consider a semi-desirable facility location problem in a continuous planar region considering the interaction between the facility and the existing demand points. A facility can be defined as semi-desirable if it has both undesirable and desirable effects to the people living in the vicinity. Our aim is to maximize the weighted distance of the facility from the closest demand point as well as to minimize the service cost of the facility. The distance between the facility and the demand points is measured with the rectilinear metric. For the solution of the problem, a three-phase interactive geometrical branch and bound algorithm is suggested to find the most preferred efficient solution. In the first two phases, we aim to eliminate the parts of the feasible region the inefficiency of which can be proved. The third phase has been suggested for an interactive search in the remaining regions with the involvement of a decision maker (DM). In the third phase, the DM is given the opportunity to use either an exact or an approximate procedure to carry out the search. The exact procedure is based on the reference point approach and guarantees to find an efficient point as the most preferred solution. On the other hand, in the approximate procedure, a hybrid methodology is used to increase the efficiency of the reference point approach. The approximate procedure can be used when the DM prefers to see locally efficient solutions so as to save computation time. We demonstrate the performance of the proposed method through example problems.

Keywords Location · Semi-desirable · Multiobjective decision making · Interactive approach

1 Introduction

A facility can be defined as semi-desirable (semi-obnoxious) if it has both undesirable and desirable effects to the people living in the vicinity. Since need for such facilities has been increasing rapidly, semi-desirable facilities have attracted the attention of many researchers

E. Karasakal (✉) · D. Nadirler
Industrial Engineering Department, Middle East Technical University, Ankara 06531, Turkey
e-mail: esra@ie.metu.edu.tr

during the last decade ([5] for a review of models suggested for semi-desirable facilities). Semi-desirable facility location problems balance public concerns and environmental requirements with the needs of facility planners in a sense that both protection from the facility and nearness to the facility are valued simultaneously.

There is an ever-growing problem of garbage disposal and location of dump sites. No one wants to live close to any solid waste disposal area that is often located far from city centers because of the danger of garbage gases, unpleasant odors and noise. However, keeping the cost of garbage collection low through transportation is one of the most important factors that municipalities often consider. Power plants and airports are other examples for semi-desirable facilities.

In this paper, we develop an interactive approach for the location of semi-desirable facilities with the motivation that inclusion of this type of facilities to our lives will continue to rise. To our knowledge, there are very few studies that aim to find either the set of efficient solutions for the semi-desirable facility location problem or a solution which optimizes a linear combination of the objectives. There is no interactive approach that explores the feasible region and find the most preferred location with the guidance of the decision maker (DM). Interactive approaches require the DM to provide preference information during the search process and provide information on the available solutions. This enhances the DM's understanding about both the problem and his/her preferences.

Searching a large region is not an easy task. If the feasible region can be reduced to those parts that contain highly preferred solutions, then it may become practical for the DM to use an interactive search procedure on the efficient frontier of the reduced region. In this paper, we develop an approach to reduce the feasible region by eliminating the parts the inefficiency of which can be proved and to search for the most preferred solution in the reduced feasible region.

Organization of the paper is as follows: In Sect. 2 we review the literature on semi-desirable facility location problems. We formulate the problem in Sect. 3 and develop the approach in Sect. 4. In Sect. 5, we present an interactive geometrical branch and bound method called Interactive Generalized Big Square Small Square (IGBSSS) method. We describe the implementation of mixed integer programming models used in IGBSSS in Sect. 6, and demonstrate the performance of the IGBSSS method through example problems in Sect. 7. Last section contains concluding remarks and directions for future research.

2 Literature review

Mehrez et al. [12] defined a single facility location problem on a square feasible region with maximin and minimax objectives using the rectilinear distances. They suggested an algorithm to find the optimal solution to the weighted maximin–minmax problem. They proved that the optimal points are either on the intersection points of any two lines forming the equirectilinear distances between any pair of demand points or boundary of the feasible region. Ohsawa [17] used the same objective functions but defined the problem on a polygon. He developed a polynomial-time algorithm for generating the analytical expressions of the efficient set and the tradeoff curve between the conflicting objectives. He characterized the efficient set and the trade-off curve using nearest and farthest-point Voronoi diagrams.

Romero-Morales et al. [20] developed a global optimization approach with a global objective function including two cost functions, the first of which is a nonincreasing function of distances measuring for the social cost of the facility and the second is a nondecreasing function measuring the transportation cost. These two functions are based on the total distance of the facility to all demand points. Their solution algorithm is based on the Big Square Small Square (BSSS) method suggested by Hansen et al. [8] with an improvement in bounding scheme. They obtained an upper bound using the lagrangian relaxation of some constraints.

Brimberg and Juel [3] proposed a solution approach to the semi-desirable facility problem with maximin and minisum objectives. In maximin objective Euclidean distances are used, whereas in minisum objective distances are assumed to be given by an arbitrary norm. For finding the efficient set they used two formulations the first of which is a parametric model where the sum of the weighted distances is minimized subject to constraints ensuring that the distance from the demand points must exceed some parameter and found the efficient set by varying the parameter. The second formulation combines the two objective functions and minimizes their weighted sum.

The maximin–minisum objective pair with rectilinear distance metric first appeared in the study of Melachrinoudis [14]. His solution method for the nonlinear nonconvex biobjective problem was based on partitioning the feasible region into n^2 subregions where n is the number of demand points as suggested by Drezner and Wesolowsky [7]. By partitioning, they eliminate the nonlinearity for each objective and have n^2 linear programs (LPs). Melachrinoudis and Xanthopulos [15] studied the maximin–minisum objective pair with Euclidean distances. They partitioned the feasible region into Voronoi polygons. The complete trajectory of efficient solutions was obtained by using the Karush–Kuhn–Tucker conditions along with the geometrical properties of Voronoi diagrams.

In another study, Brimberg and Juel [4] studied the semi-desirable facility problem with transportation cost function which is the weighted sum of all the distances to the facility, and the social cost function which is the minisum objective, where Euclidean distance is raised to a negative power. Their solution approach is based on the minimization of the weighted sum of the two objective functions. A trajectory of efficient points is defined by a system of differential equations. Skriver and Andersen [22] studied the same problem as Brimberg and Juel [4]. They proposed the biobjective adaptation of the BSSS algorithm.

Yapicioglu et al. [24] introduced a new model for the semi-obnoxious problem where transportation cost is represented with a weighted minisum function and obnoxious effects of the facility are represented by a distance-based piecewise function. They developed a heuristic solution approach based on particle swarm optimization.

3 Problem formulation

We define the problem of locating a single semi-desirable facility in a continuous planar region in the existence of several points representing living areas. We use two objectives to measure the desirable and undesirable aspects of the facility to be located.

Let N be the number of the existing demand points and $\boldsymbol{b^i} = (b_1^i, b_2^i)$ for $i = 1, \ldots, N$ be the coordinates of existing demand points, where $\boldsymbol{b^i} \in R^2$. Let S be the feasible region, which is a convex polygon in R^2 defined by k constraints:

$$S = \left\{\boldsymbol{x} \in R^2 : e_j x_1 + f_j x_2 \leq g_j \quad \text{for } j = 1, \ldots, k\right\}$$

where e_j, f_j and g_j are constants that define the linear constraints and k is the number of constraints required to define the convex polygon.

Let $L(\boldsymbol{x})$ be the minimum weighted distance of $\boldsymbol{x}$ from the demand points, and $W(\boldsymbol{x})$ be the total weighted distance between $\boldsymbol{x}$ and all demand points. The first objective function used to model the undesirable effects is maximin which maximizes the minimum weighted distance of the facility from the demand points.

$$L^*(S) = \max_{x \in S} L(\boldsymbol{x}) = \max_{(x_1,x_2)\in S} \left\{ \min_{i=1,\ldots,N} \theta_i \left(\left| x_1 - b_1^i \right| + \left| x_2 - b_2^i \right| \right) \right\}$$

where $\theta_i > 0$ is the weight associated with the ith demand point that can be defined as a decreasing function of the population.

The second objective function used to model the desirable effects is minisum which minimizes the total weighted distance of the facility from the demand points.

$$W^*(S) = \min_{x \in S} W(\boldsymbol{x}) = \min_{(x_1,x_2)\in S} \left\{ \sum_{i=1}^{N} \lambda_i \left(\left| x_1 - b_1^i \right| + \left| x_2 - b_2^i \right| \right) \right\}$$

where $\lambda_i > 0$ is the weight associated with the ith demand point which is defined as an increasing function of the population.

The distances are measured with the rectilinear metric. To model undesirable effects the use of rectilinear metric can be realistic depending on the application of concern [2,7,13,21,26]. For example, as indicated in [14], the unpleasant effects of a facility generally spread through rectangular isles in a factory including walls. Although the general trend is the utilization of euclidean metric to model undesirable effects, we believe that it may not be realistic to model the undesirable effects like noise and air pollution in the existence of various factors affecting the spread of such effects like wind, geographical barriers etc. Hence, the solution complexity caused by this metric may not be worth studying.

To model desirable effects, we measure service cost. For most of the facilities, service cost generally occurs through transportation cost and travel distance between facilities can be approximated by the rectilinear distance [14].

The mathematical model (*Maximin*-l^1) for the undesirable facility location problem with maximin objective has been proposed by Sayin [21].

$$\begin{aligned}
&(Maximin\text{-}l^1) && \\
&L^* = Max L && && (1)\\
&subject\ to && \\
&L \le \theta_i d_i && \text{for } i = 1, \ldots, N && (2)\\
&d_i = a_1^i + a_2^i && \text{for } i = 1, \ldots, N && (3)\\
&a_j^i - u_j^i = x_j - b_j^i && \text{for } i = 1, \ldots, N \quad j = 1, 2 && (4)\\
&a_j^i - o_j^i = b_j^i - x_j && \text{for } i = 1, \ldots, N \quad j = 1, 2 && (5)\\
&u_j^i \le M t_j^i && \text{for } i = 1, \ldots, N \quad j = 1, 2 && (6)\\
&o_j^i \le M(1 - t_j^i) && \text{for } i = 1, \ldots, N \quad j = 1, 2 && (7)\\
&e_j x_1 + f_j x_2 \le g_j && \text{for } j = 1, \ldots, k && (8)\\
&\boldsymbol{a^i}, \boldsymbol{u^i}, \boldsymbol{o^i} \ge 0 && \text{for } i = 1, \ldots, N &&\\
&\boldsymbol{t^i} \in \{0, 1\} && \text{for } i = 1, \ldots, N &&
\end{aligned}$$

where L is the minimum weighted distance of the facility from the demand points, d_i is the rectilinear distance of the facility from the ith demand point; a_j^i is the jth component of d_i; u_j^i, o_j^i are surplus variables associated with the ith demand point in the jth dimension; t_j^i is a binary variable, $t_j^i = 1$ if $x_j \le b_j^i$, $t_j^i = 0$, otherwise. M is a large number, which can be taken as twice of the longer side length of a rectangle that is large enough to contain the convex polygon determined by the constraint set (8).

The objective of (*Maximin-l^1*) is to maximize the minimum weighted distance of the facility from the demand points which is found by the constraint set (2). Constraint set (3) ensures that absolute distance is calculated as $d_i = a_1^i + a_2^i = \left|x_1 - b_1^i\right| + \left|x_2 - b_2^i\right|$. Since the objective is maximization, constraint sets (4)–(7) are used to guarantee one of the following inequalities holds as an equality: $a_j^i \geq \left(x_j - b_j^i\right)$ and $a_j^i \geq \left(-x_j + b_j^i\right)$; otherwise the program would be unbounded. Constraint set (8) defines the feasible region.

We construct a mathematical model with minisum objective as follows:

$$\begin{aligned}
&(Minisum\text{-}l^1) \\
&W^* = MinW && && \text{(9)} \\
&subject\ to \\
&W = \sum_{i=1}^{N} \lambda_i d_i && \text{for } i = 1, \ldots, N && \text{(10)} \\
&d_i = a_1^i + a_2^i && \text{for } i = 1, \ldots, N && \text{(11)} \\
&a_j^i \geq x_j - b_j^i && \text{for } i = 1, \ldots, N \quad j = 1, 2 && \text{(12)} \\
&a_j^i \geq b_j^i - x_j && \text{for } i = 1, \ldots, N \quad j = 1, 2 && \text{(13)} \\
&e_j x_1 + f_j x_2 \leq g_j && \text{for } j = 1, \ldots, k \\
&\boldsymbol{a^i} \geq 0 && \text{for } i = 1, \ldots, N
\end{aligned}$$

where W is the total weighted distance between the facility and the demand points.

The objective of (*Minisum-l^1*) is to minimize the total weighted distance of the facility from the demand points which is found by the constraint set (10). Constraint set (11) ensures that absolute distance is calculated as $d_i = a_1^i + a_2^i = \left|x_1 - b_1^i\right| + \left|x_2 - b_2^i\right|$. Constraint sets (12) and (13) guarantee the calculation of rectilinear distance. Since the objective is minimization, it is guaranteed that one of the constraint sets (12) and (13) holds as an equality. Thus, there is no need to control those constraints with binary variables. Since in the semi-desirable facility location problem we consider maximin and minisum objective functions simultaneously, the following biobjective mathematical model is formulated by combining models (*Maximin-l^1*) and (*Minisum-l^1*).

$$\begin{aligned}
&(Biobjective\text{-}l^1) \\
&L^* = MaxL && && \text{(14)} \\
&W^* = MinW && && \text{(15)} \\
&subject\ to \\
&W = \sum_{i=1}^{N} \lambda_i d_i && && \text{(16)} \\
&L \leq \theta_i d_i && \text{for } i = 1, \ldots, N && \text{(17)} \\
&d_i = a_1^i + a_2^i && \text{for } i = 1, \ldots, N && \text{(18)} \\
&a_j^i - u_j^i = x_j - b_j^i && \text{for } i = 1, \ldots, N \quad j = 1, 2 && \text{(19)} \\
&a_j^i - o_j^i = b_j^i - x_j && \text{for } i = 1, \ldots, N \quad j = 1, 2 && \text{(20)}
\end{aligned}$$

$$u^i_j \leq Mt^i_j \qquad \text{for } i = 1, \ldots, N \quad j = 1, 2 \tag{21}$$

$$o^i_j \leq M(1 - t^i_j) \qquad \text{for } i = 1, \ldots, N \quad j = 1, 2 \tag{22}$$

$$e_j x_1 + f_j x_2 \leq g_j \qquad \text{for } j = 1, \ldots, k \tag{23}$$

$$\boldsymbol{a^i}, \boldsymbol{u^i}, \boldsymbol{o^i} \geq 0 \qquad \text{for } i = 1, \ldots, N \tag{24}$$

$$\boldsymbol{t^i} \in \{0, 1\} \qquad \text{for } i = 1, \ldots, N \tag{25}$$

In multicriteria decision making (MCDM) terminology, $\boldsymbol{x} = (x_1, x_2)$ is a *decision vector*. The vector of objective function values $z(\boldsymbol{x}) = (L(\boldsymbol{x}), W(\boldsymbol{x}))$ belonging to $\boldsymbol{x}$ is named as an *objective vector*. Feasible region, S, containing decision vectors is called the *feasible decision region*. It is a subset of the *decision space*, R^2. *Feasible objective region* is defined as the image of the feasible decision region in the two objective functions. It is a subset of the *objective space*, R^2. Throughout the paper, words "feasible region" and "feasible decision region" will be used interchangeably.

Let $\boldsymbol{Z}$ define the *feasible objective region*: $\boldsymbol{Z} = \left\{z(\boldsymbol{x}) \in R^2 \middle| z(\boldsymbol{x}) = (L(\boldsymbol{x}), W(\boldsymbol{x})), \boldsymbol{x} \in \mathrm{S}\right\}$. A criterion vector $z(\boldsymbol{x})$ is said to be *dominated* by a criterion vector $z(\boldsymbol{x^h})$ if $L(\boldsymbol{x^h}) \geq L(\boldsymbol{x})$, $W(\boldsymbol{x^h}) \leq W(\boldsymbol{x})$ and $(L(\boldsymbol{x^h}), W(\boldsymbol{x^h})) \neq (L(\boldsymbol{x}), W(\boldsymbol{x}))$. A criterion vector $z(\boldsymbol{x})$ is said to be *nondominated* if there does not exist another feasible solution $\boldsymbol{x^h}$ such that $z(\boldsymbol{x^h})$ dominates $z(\boldsymbol{x})$; in this case the feasible solution $\boldsymbol{x}$ is said to be *efficient*. The set of all efficient solutions is called the efficient frontier.

Let S' be a subset of feasible decision region containing $\boldsymbol{x}$, i.e., $\boldsymbol{x} \in S' \subset S$. A solution $\boldsymbol{x}$ is *locally efficient* if and only if there does not exist another feasible solution $\boldsymbol{x^h} \in S'$ such that $z(\boldsymbol{x^h})$ dominates $z(\boldsymbol{x})$ [1].

Ideal objective vector of region S, $\boldsymbol{R}(S)$, is a point in the objective space, whose components are obtained by separately optimising each of the objective functions over region S, i.e., $R_1(S) = Max_{x \in S}\{L(\boldsymbol{x})\}$, $R_2(S) = Min_{x \in S}\{W(\boldsymbol{x})\}$. Nadir objective vector, $\boldsymbol{Q}(S)$, is a point whose components are obtained by anti-optimizing each of the objective functions in the efficient frontier of region S, i.e., $Q_1(S) = Min_{x \in E}\{L(\boldsymbol{x})\}$, $Q_2(S) = Max_{x \in E}\{W(\boldsymbol{x})\}$ where E represents the set of efficient solutions in S. Since nadir objective vectors are difficult to obtain in multobjective decision making problems, they are generally estimated by an approximation obtained from payoff tables [23] (p. 267). However, because the problem we consider is biobjective, we obtain exact nadir points from the payoff table constructed by solving $Max_{x \in S}\{L(\boldsymbol{x}) - \rho W(\boldsymbol{x})\}$ and $Min_{x \in S}\{W(\boldsymbol{x}) - \rho L(\boldsymbol{x})\}$ where ρ is a small positive constant that guarantees the generation of efficient solutions, which can be taken as less than both $1/R_1(S)$ and $1/Q_2(S)$.

4 Development of the approach

We adapt the new version of the BSSS method, Generalized Big Square Small Square (GBSSS) method [19] to the semi-desirable facility location problem defined on a convex polygon and we develop a method that explores the feasible region with the guidance of the DM. The main idea of BSSS and GBSSS is to eliminate some parts of the feasible region up to a prespecified precision. The elimination occurs when the inefficiency of a region is proved with the help of bounds. In the proposed method, we suggest an additional phase to GBSSS, in which we guide the DM in selecting the most preferred location based on her/his preferences. The aim of this phase is to guide the DM in searching the regions that are not proved to be inefficient to find the best solution. We propose the use of the reference point approach [25], which projects any point in the objective space to the efficient frontier of the region of concern. Theorem 1 shows how efficient solutions are generated by projecting solutions onto the efficient frontier.

Theorem 1 [10,25] *In a problem where all objectives are to be maximized, a solution* $\bar{z} \in Z$ *is nondominated if and only if there exists* $w_j^0 > 0$ *for each* j *and* $\rho > 0$*, such that* $\bar{z}$ *solves the problem* (P)

$$\underset{z \in Z}{Min} \left[\underset{j}{Max} \left(w_j^0 (G_j^0 - z_j) - \rho \sum_j z_j \right. \right]$$

where $\boldsymbol{G}^0$ *is a reference point.*

Corollary 1 *Given* $w_j^0 > 0$ *for each* j *and* $\rho > 0$*. If* $\boldsymbol{G}^0$ *is a feasible or an infeasible solution, optimal solution to* (P) *is nondominated.*

Corollary 2 *Let* (P^1) *denote the problem* (P) *with* $Z = Z^1$*, and* (P^2) *denote the problem* (P) *with* $Z = Z^2$*. Let* $\bar{\mathbf{z}}^1$ *and* $\bar{\mathbf{z}}^2$ *be the optimal solutions to* (P^1) *and* (P^2)*, respectively. Given a weight vector* $\mathbf{w}^0$ *and a reference point* $\boldsymbol{G}^0$, $\bar{z}_j^1 \leq \bar{z}_j^2$ *for all* j *with a strict inequality for at least one* j *if there exists* $\mathbf{z} \in Z^2$ *such that* $\mathbf{z} \geq \bar{\mathbf{z}}^1$ *and* $\mathbf{z} \neq \bar{\mathbf{z}}^1$.

The model that adapts Wierzbicki's [25] reference point method to our problem is called 'Achievement Scalarizing Program' (*ASP*). The *ASP* operates in the objective space and minimizes the maximum weighted deviation of objectives from the levels specified with a reference point. The program finds the closest efficient point to the reference point in the weighted Tchebycheff metric as defined below. The *ASP* is obtained by replacing objective functions (14) and (15) with a single objective and adding two constraints to those of (*Biobjective-*l^1) model.

$$(ASP)$$

$$Min\,[\alpha - \rho(L - W)] \tag{26}$$

$$subject\ to$$

$$\alpha \geq w_1^0 \left(\frac{G_1^0 - L}{R_1(S)} \right) \tag{27}$$

$$\alpha \geq w_2^0 \left(\frac{W - G_2^0}{Q_2(S)} \right) \tag{28}$$

Constraint sets (16)–(25) of (*Biobjective-*l^1)

where $\boldsymbol{G}^0$ is a reference point, α is the maximum deviation of the solution objective vector from the reference point, $\boldsymbol{w}^0 = (w_1^0, w_2^0)$ is a weight vector and ρ is a small positive constant that guarantees the generation of efficient solutions, which can be taken as less than both $1/R_1(S)$ and $1/Q_2(S)$.

Minimization of the objective function (26) ensures that a point which minimizes the maximum deviation from the levels specified with a reference point $\boldsymbol{G^o}$ is determined as the optimal solution. Both objectives are given weights by the DM considering their relative importance. Constraints (27) and (28) calculate the weighted Tchebycheff distance between the reference point and the solution vector, which are normalized using highest objective values (i.e., the ideal value of the maximin objective and the nadir value of the minisum objective) [18]. Since normalization with the highest objective values guarantees that differences between the reference point and the solution vector in all objectives are within the [0,1] range, the scale effect on the weights is avoided.

Since *ASP* is a mixed integer linear programming problem, computation time increases exponentially as the number of integer variables increases. This makes the *ASP* inefficient for large problems. To decrease the solution time, we use the idea suggested by Karaivanova et al. [9] for the solution of multiobjective mixed integer programs. They proposed the use of a two-phase continuous/integer method. In the first phase, the method operates in the relaxed continuous space parametrically to find a number of nondominated continuous solutions iteratively. Once the most preferred continuous solution is determined, the closest integer solution is found with the help of the *ASP*. The logic behind this approach is that it is reasonable to generate nondominated continuous solutions in the early iterations when the DM is searching regions far from the most preferred solution. However, when approaching the most preferred region, she/he needs more accuracy.

For the search in the nondominated continuous objective region, we use the reference direction approach [11]. We first solve the LP relaxation of the *ASP* to find an initial nondominated continuous solution, $\boldsymbol{y^o}$, and then search the nondominated continuous objective region by a parametric linear program. This method iteratively projects a line segment in the objective space onto the nondominated surface of the relaxed feasible objective region using Achievement Scalarizing Parametric Linear Program (*ASPLP*). *ASPLP* is obtained by replacing constraint sets (27) and (28) of *ASP* with the following constraints and defining $\boldsymbol{t^i}$ as nonnegative variable, i.e. $\boldsymbol{t^i} \geq 0$ for $i = 1, \ldots, N$.

$$\alpha \geq w_1^0 \left(\frac{y_1^0 + p\Delta d_1 - L}{R_1(S)} \right) \tag{29}$$

$$\alpha \geq w_2^0 \left(\frac{W - (y_2^0 + p\Delta d_2)}{Q_2(S)} \right) \tag{30}$$

where $\boldsymbol{y}^0$ is the reference point, $\boldsymbol{\Delta d}$ is the reference direction and $p = 1, \ldots, P$ (P is the desired number of points to be projected onto the efficient frontier in the determined direction). $\boldsymbol{\Delta d}$ and P are to be determined by the DM.

It should be noted that the continuous solution is very sensitive to the choice of M in constraint sets (21) and (22). Although, objective function (26) together with the constraint sets (16) and (28) forces constraints (19) and (20) to measure the rectilinear distance in the LP relaxation, when the integrality requirements are relaxed, both $\boldsymbol{u}$ and $\boldsymbol{o}$ are free to take positive values, which directly depend on the value of M. Therefore, value of M should be set to its possible minimum level.

5 Interactive Generalized Big Square Small Square (IGBSSS) method

In this section, we present the interactive geometrical branch and bound method, called IGBSSS method, for the solution of a single semi-desirable facility location problem. We define a region (i.e., either square or rectangle) that is large enough to contain the feasible region determined by the constraints and with sides parallel to the axis. Our algorithm consists of three phases. In the first two phases of the algorithm, at each branching we cut the region under consideration into four equal subregions by dividing it with two lines. For each axis, we find the maximum and minimum values of the region and calculate the midpoint.

Then we draw parallel lines to the axes passing through the midpoints. If the feasible region is neither square nor rectangle, some of the generated subregions may not intersect with the feasible region. We delete such subregions from further consideration. We pick a predetermined number of points from the region under consideration and keep the nondominated ones as incumbent points. We use them to check the efficiency of the subregions. We reduce the feasible region by eliminating the inefficient subregions. In the first phase, called rough cut phase, we use upper bounds on the optimal maximin objective function values, whereas in the second phase, called precise cut phase, we use the optimal maximin objective function values. In the last phase, we search the reduced feasible region with the involvement of the DM.

In our algorithm, we use the best bound search. Every time we look for a region to branch on, the region with the highest upper bound/optimal maximin objective value or the region with the lowest optimal minisum objective value is selected to be divided. We consider the difference from the second highest maximin objective value and the second lowest minisum objective value. We select the region that has the largest difference. The rationale behind is that the regions with best bounds are difficult to eliminate with incumbent points; hence it is reasonable to branch them first. Therefore, by best bound search, priority in branching is given to the subregions whose elimination is difficult. By doing this, we increase the chance to eliminate subregions with worse objective values without branching them much. The algorithm is as follows:

Phase 1: Rough cut phase In this phase, we use upper bound on the optimal maximin objective function value. However, since (*Minisum-l^1*) is a linear programming problem, we use the optimal minisum objective function value. If any incumbent point dominates this pair, i.e. the subregion under consideration is proved to be inefficient, it is discarded from further consideration.

Upper bounds for the optimal objective function value of (*Maximin-l^1*) are found solving both (*Maximin-l^1*) with demand points located inside the subregion and the following LP formulation [21]. Smallest upper bound is chosen as the upper bound on the optimal maximin objective.

$$\begin{array}{ll}
UB(Maximin\text{-}l^1) & \\
\bar{L}^* = Max\,\bar{L} & \\
subject\ to & \\
\bar{L} \le \sum_{f=1}^{F} \mu_f \theta_i d(\boldsymbol{v^f}, \boldsymbol{b^i}) & \text{for } i = 1, \ldots, N \\
\sum_{f=1}^{F} \mu_f = 1 & \\
\sum_{f=1}^{F} \mu_f v_j^f - x_j = 0 & \text{for } j = 1, 2 \\
e_j x_1 + f_j x_2 \le g_j & \text{for } j = 1, \ldots, k \\
\mu_f \ge 0 & \text{for } f = 1, \ldots, F
\end{array}$$

where $\boldsymbol{v^f}$ is the fth extreme point of the feasible region, μ_f is the weight associated to each extreme point, $\theta_i > 0$ is the weight associated with the ith demand point and $d(\boldsymbol{v^f}, \boldsymbol{b^i})$ is the rectilinear distance between $\boldsymbol{v^f}$ and $\boldsymbol{b^i}$. If the feasible region is not rectangle, extreme points of the rectangle containing the feasible region is used instead of the extreme points of the feasible region in the model *UB*(*Maximin-l^1*).

The purpose of this rough phase is to get rid of some subregions that can be eliminated with the help of an upper bound on the maximin objective since it is time consuming to find the optimal solution to (*Maximin-l^1*) especially for big samples of demand points.

Branching increases the chance of eliminating subregions because after every division, the upper bound on the maximin objective value decreases while the optimal minisum objective increases. At the same time, new incumbent points are selected from the subregions in each branching. Branching is performed until regions are reduced to a predetermined size.

Phase 2: Precise cut phase In this phase, the optimal maximin objective values are found by solving (*Maximin-l^1*) for the remaining subregions. Since the demand points, whose shortest weighted distance to the subregion under consideration is greater than the upper bound of that subregion on maximin objective, do not affect the optimal solution of (*Maximin-l^1*), we filter those points before solving (*Maximin-l^1*).

After finding the optimal value of the maximin objective for each subregion, we compare the ideal objective vectors of the subregions with the incumbent points for any possible further elimination. In this phase, we allow the DM to branch the remaining subregions further with the optimal values. Since the subregions are already divided up to a prespecified side length in phase 1, branching with optimal values is expected to take reasonable time.

Phase 3: Interactive search phase This phase of the algorithm is the beginning of the interactive search with the DM. In this part, we develop two procedures; one is exact and the other is an approximate procedure.

The exact procedure is based on the reference point approach which guarantees to find an efficient point as the most preferred solution. In this procedure, the remaining subregions after the first two phases are presented to the DM along with their ideal and nadir objective vectors. Each time a subregion is selected to be searched, the DM is asked to specify a weight vector and aspiration levels in both objectives (i.e. reference point) based on her/his preferences. Then the *ASP* is solved to project this reference point to the efficient frontier of the selected subregion with the specified weight vector. The solution found is efficient with respect to the subregion from which it is generated, but it may be dominated by the solutions in the other subregions (i.e. it is locally efficient). Therefore, each time a reference point is projected onto the efficient frontier of the selected subregion, we need to check whether there exist solutions in the other subregions which dominate the one at hand. For this check, we again use the idea of the *ASP* with which the solution at hand is projected onto the efficient frontier of the other subregions with the same weight vector specifed by the DM. If there is no solution dominating the one at hand, then the solution is proved to be nondominated (by Corollary 2). If any dominating solution or locally efficient solution is found, then the DM is given the chance to pass to the subregion from which it is produced and continue the search from that subregion. The DM continues to generate solutions in the same manner from the subregions she/he has selected by identifying new reference points and new weight vectors. The process is repeated until the DM has identified the most preferred solution.

The approximate procedure is also based on the reference point approach. This procedure can be used when the DM wants to see locally efficient solutions instead of efficient solutions in order to save computation time. In this procedure, the interactive search is carried out in the subregions that the DM selects. The procedure guarantees to find efficient solutions for the selected subregion (i.e. locally efficient solutions). However, the solution may be inefficient with respect to the other subregions. Each time a solution is found, it is compared to the other solutions already generated. Hence, the chance to obtain an inefficient final solution decreases considerably with these comparisons and the approach finds locally efficient solutions.

In the approximate procedure, the DM is asked to specify aspiration levels in both objectives (i.e. reference point) and a weight vector, each time a subregion is selected to be searched as in the exact method. In the beginning, the LP relaxation of the *ASP* is solved which minimizes the maximum deviation from the reference point. This program guarantees to find a nondominated continuous solution for the subregion under search. With this solution at hand, we present two ways to the DM. She/he may find the closest integer solution to the continuous one, or she/he can continue to search for other continuous solutions.

If the latter is selected, the DM is asked to specify a reference direction and continuous nondominated solutions closest to the points on this direction are found by solving the *ASPLP*. Once the DM is satisfied with a continuous solution, an integer solution closest to it is found with the *ASP* and kept as an objective vector of a candidate location point. This approach helps the DM to have a better understanding about the efficient frontier.

The details of the algorithm are presented in the Appendix.

6 Implementation of mixed integer programming models

The default MIP strategy settings of CPLEX intend to solve a vast majority of MIP models with the minimum solution time. However, difficult models which may benefit from the revision of performance measures of branch and bound algorithm exist [6]. Nadirler and Karasakal [16] found out that changing the strategies of the branch and bound algorithm to find better integer solutions with (MIPEMPHASIS = 1), fixing the pricing strategy with (DPRIIND=1), turning off cut constraints with (CUTS = NO) and branching on maximum infeasibility (VARSEL = 1) bring considerable improvement to (*Maximin-*l^1).

Since the binary structure of *ASP* is the same as that of (*Maximin-*l^1), we believe the use of the suggested strategy combination for (*Maximin-*l^1) will increase the efficiency of the model. With this idea, we have conducted an experimental study. The feasible region was defined in R^2, which is a 100 × 100 square defined by the constraints $0 \leq x_1 \leq 100$ and $0 \leq x_2 \leq 100$. We conducted three experiments using 1000, 3000 and 5000 demand points (10 randomly generated problems were solved in each experiment). The locations of the demand points were generated according to uniform distribution in the interval [0,100]. We assumed that both objectives were attributed with equal weights. We also assumed that the demand points have equal weights based on the results of the computational experiments given in [21] because the solution time of the weighted version of (*Maximin-*l^1) was found to be shorter compared to the unweighted one (i.e. equal weighted version). The reference point was assumed to be the ideal point in each problem. The runs were conducted on Pentium IV personal computer with 256 MB random access memory (RAM). The optimization models were solved in GAMS Version 20.2. The computer code that calls optimization programs was written in Borland C++ Builder Version 3. CPLEX Version 7.5 operating under GAMS Version 20.2 was used as the MIP solver. Table 1 reports the average CPU for each problem set.

As evident from Table 1, the solution time obtained for different sample sizes decreased considerably with the strategies as estimated and a great saving is achieved in the solution time of the *ASP* with the new branch and bound strategies.

7 Examples

In this section, we present two examples to show the performance of IGBSSS method. The first one illustrates the approximate procedure of the interactive search phase (Phase 3) on a small problem. The second one illustrates the exact procedure of Phase 3 on a large problem.

Table 1 Computational results

	CPU Time(s)		
Number of demand points	Default strategies	Proposed strategies	% Reduction in CPU time
1000	27.7	10.2	63
3000	164.5	50.5	69
5000	430	129	70

Example 1 Consider a single semi-desirable facility location problem in a 100×100 square. Suppose there are 6 demand points as given in Table 2 and weights associated with the demand points are 1 (i.e., $\lambda_i = 1$ and $\theta_i = 1$ for $i = 1, \ldots, 6$).

We assume that the DM has an underlying concave utility function, which we pretend that we do not know.

$$U = -\sum_{j=1}^{2} 0.5(R_j(S) - z_j)^2$$

where U is utility function, $R_j(S)$ is the jth coordinate of the ideal objective vector of the feasible region S and z_j is the jth coordinate of an objective vector.

Suppose the DM determines stopping side length for phases 1 and 2 as 12.5 and the number of feasible points generated from each region as 100.

At the end of phase 1, shaded regions shown in Fig. 1 are eliminated. The list of candidate efficient subregions (*LCES*) obtained at the end of phase 1 along with their upper bounds on the maximin objective and their optimal minisum objective values are given in Table 3. Table 4 presents the list of incumbent solutions (*LIS*) and their distance to the closest demand point, $L(\boldsymbol{x^h})$, and total distance to all demand points, $W(\boldsymbol{x^h})$.

In phase 2, the optimal maximin objective values for each subregion in the *LCES* (see Table 5) are found. Subregion 24 is eliminated because it is dominated by an element of *LIS,* $\boldsymbol{x}^8$ (see Table 4).

In phase 1, 90.63% of the feasible region, and in phase 2, 16.56% of the reduced feasible region have been proved to be inefficient and eliminated. At the end of two phases, only 7.84% of the feasible region has been left for further search (see Fig. 2).

In phase 3, subregions 20, 30 and 32 are combined into a single region and 3 regions are presented to the DM (see Fig. 2). For these regions, optimal locations with respect to maximin and minisum objectives are presented in Table 6. Table 6 also gives the ideal and nadir objective vectors of the regions.

Suppose that the DM wants to use the approximate procedure and wants to start the search from region 1. Assume that the DM specifies $\boldsymbol{G}^0 = (32, 390)$ as the reference

Table 2 Demand points

i	1	2	3	4	5	6
$\boldsymbol{b}^i$	(50,20)	(18,80)	(2,16)	(75,68)	(90,100)	(85,10)

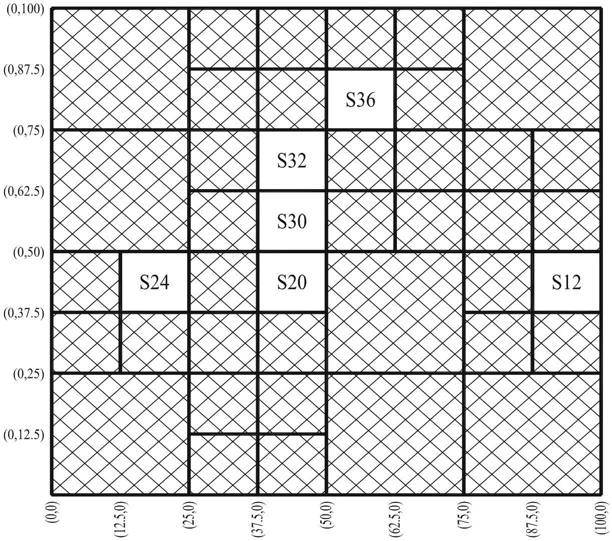

Fig. 1 Reduced feasible region at the end of phase 1

Table 3 List of candidate efficient solutions at the end of phase 1

Subregion (n)	$UB(S^n)$	$W^*(S^n)$
12	49.00	412.00
20	42.50	382.00
24	46.44	432.00
30	46.00	382.00
32	40.00	382.00
36	42.00	396.00

Table 4 List of incumbent solutions at the end of phase 1

Solution (h)	$\boldsymbol{x}^{\boldsymbol{h}}$	$L(\boldsymbol{x}^{\boldsymbol{h}})$	$W(\boldsymbol{x}^{\boldsymbol{h}})$
1	[50.00,56.50]	36.50	382.00
2	[48.75,56.25]	37.50	384.50
3	[47.50,56.25]	38.75	387.00
4	[46.25,56.25]	40.00	389.50
5	[45.00,57.50]	40.50	392.00
6	[42.50,56.25]	43.75	397.00
7	[41.25,56.25]	45.00	399.50
8	[40.00,56.50]	46.00	402.00
9	[97.50,43.75]	46.25	467.00
10	[98.75,43.75]	47.50	474.50
11	[100.00,43.75]	48.75	482.00

Table 5 Elimination in the list of candidate efficient solutions at phase 2

Subregion (n)	$L^*(S^n)$	$W^*(S^n)$	Status
12	49.00	412.00	Candidate
20	42.50	382.00	Candidate
24	*46.00*	*432.00*	*Dominated*
30	46.00	382.00	Candidate
32	40.00	382.00	Candidate
36	42.00	396.00	Candidate

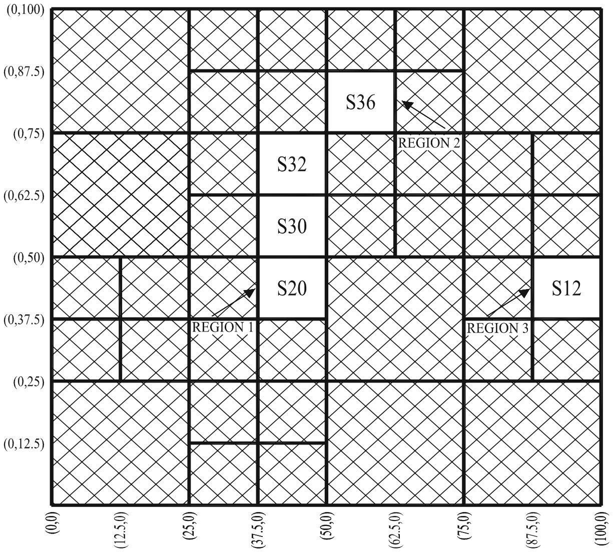

Fig. 2 Reduced feasible region at the end of phase 2

Table 6 Regions presented to the decision maker in phase 3

Region	Constraints	Optimal solution to (*Maximin-*l^1)	Optimal solution to (*Minisum-*l^1)	Ideal objective vector	Nadir objective vector
1	$37.5 \leq x_1 \leq 50$ $37.5 \leq x_2 \leq 75$	[37.5,53.5]	[50,69]	[46,382]	[26,407]
2	$50 \leq x_1 \leq 62.5$ $75 \leq x_2 \leq 87.5$	[52.5,87.5]	[50,75]	[42,396]	[32,436]
3	$87.5 \leq x_1 \leq 100$ $7.5 \leq x_2 \leq 50$	[100,44]	[87.5,37.5]	[49,412]	[30,482]

point and $\boldsymbol{w}^0 = (0.8, 0.2)$ as the weight vector. By solving the relaxed version of the *ASP*, the first continuous solution is found as $\boldsymbol{z}^0 = (36.5, 382)$ with the decision vector $\boldsymbol{x}^0 = (50, 56.5)$.

Suppose that the DM is not sure about the continuous solution and wants to see some alternative solutions in a direction that leads improvement in both objectives, $\boldsymbol{\Delta d} = (3.5, 2)$, with $P = 2$. The following feasible solutions are generated.

$$\text{For } p = 1, \quad z^1 = (39.87, 388.73) \quad x^1 = (50, 59.87)$$
$$\text{For } p = 2, \quad z^2 = (43.23, 395.47) \quad x^2 = (50, 63.23)$$

The most preferred continuous vector is $\boldsymbol{K} = z^1 = (39.87, 388.73)$. The closest nondominated integer vector is $\boldsymbol{C}^1 = (39.87, 388.73)$ with $\boldsymbol{x} = (46.63, 56.5)$. Since $\boldsymbol{C}^1$ is not dominated by any vector in the *LIS*, it is added to the list of candidate location points (*LCLP*) as a candidate location point.

Suppose that the DM wants to stop the search with $\boldsymbol{C}^1 = (39.87, 388.73)$ as the most preferred solution. Hence, the semi-desirable facility should be located at $\boldsymbol{x} = (46.63, 56.5)$. In this case the facility is 39.87 units far from the closest demand point, while it has a total distance of 388.73 units from all the demand points.

Note that the best solution that maximizes the underlying utility function is $L^* = 39$ and $W^* = 387$ with the decision vector $\boldsymbol{x} = (47.5, 56.5)$ and the utility value $U = -62.5$, which is very close to the most preferred solution found by the IGBSSS method in the criterion space.

Example 2 Consider a single semi-desirable facility location problem in a 100 × 100 square. Suppose that there are 2000 demand points uniformly generated in the interval [0,100] and weights associated with the demand points are 1(i.e., $\lambda_i = 1$ and $\theta_i = 1$ for $i = 1, \ldots, 2000$). We assume that the DM has an underlying linear utility function, which we pretend that we do not know.

$$U = 0.7\left(\frac{R_1(S) - z_1}{R_1(S) - Q_1(S)}\right) + 0.3\left(\frac{R_2(S) - z_2}{R_2(S) - Q_2(S)}\right)$$

U, $R_j(S)$ and z_j (for $j = 1, 2$) are as defined in Example 1 and $Q_j(S)$ (for $j = 1, 2$) is the jth coordinate of the nadir objective vector of the feasible region S.

Suppose the DM determines stopping side length for phase 1 as 12.5, phase 2 as 6.25, and the number of feasible points generated from each region as 100.

In this example, since the demand points are uniformly generated, function values of incumbent points are expected to be close to each other. Besides, the upper and lower bounds on the objective functions are expected to be similar in the regions. Hence, obviously, for any elimination, the feasible region should be divided more compared to the previous example. In addition, as a result of the uniformity in the demand points it will be difficult to eliminate big regions at once. Based on these, the resulting elimination pattern is expected to be more uniform and slower.

At the end of phase 1, shaded regions shown in Fig. 3 are eliminated.

In phase 2, the optimal maximin objective values for each subregion in the *LCES* are calculated and 21 subregions are eliminated since they are found to be dominated by solutions in the *LIS*.

Remaining subregions are further divided. With this further division, 80 additional subregions are generated. The optimal maximin and minisum objective values for each newly generated subregion are calculated. At this step, 62 of them are found to be dominated by the *LIS* and eliminated. Figure 4 shows the feasible region after the second elimination in Phase 2. Table 7 shows the percent elimination achieved in two phases. As seen from Fig. 4, the remaining subregions can be considered as 5 regions. The data related with the regions are given in Table 8.

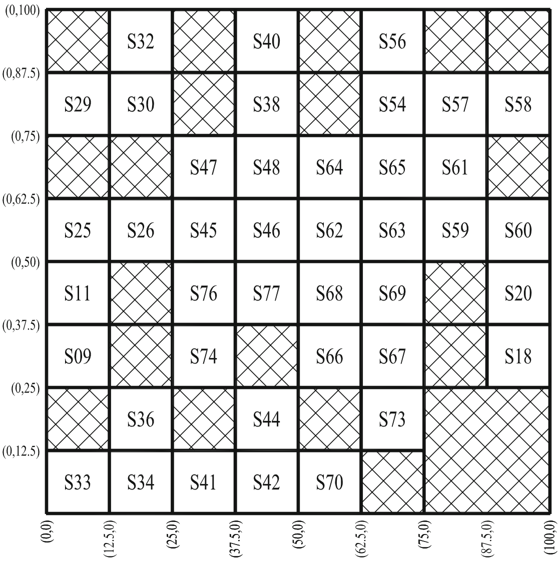

Fig. 3 Reduced feasible region at the end of phase 1

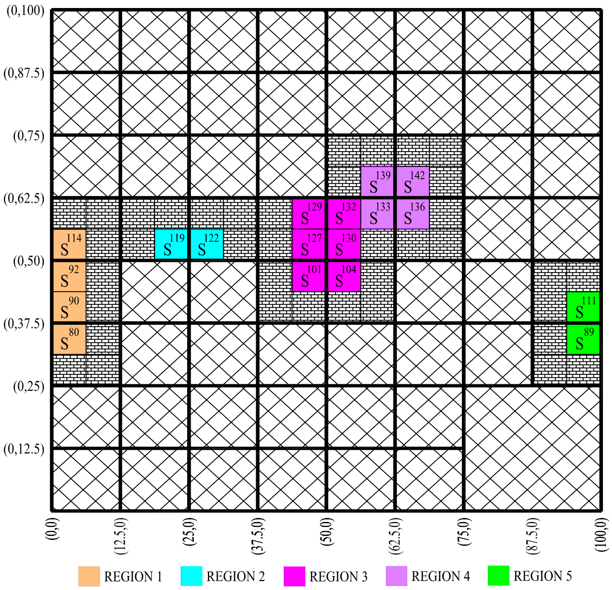

Fig. 4 Reduced feasible region at the end of phase 2

Table 7 Percent elimination achieved

Phase 1	Phase 2		Overall
	First elimination	Second elimination	
35.94	32.81	24.22	92.97

Table 8 Regions presented to the decision maker in phase 3

Region	Constraints	Optimal solution to (*Maximin-l^1*)	Optimal solution to (*Minsum-l^1*)	Ideal objective vector	Nadir objective vector
1	$0 \leq x_1 \leq 6.25$ $31.25 \leq x_2 \leq 56.25$	[0,36.73]	[6.25,49.56]	[6.09,139669.36]	[2.03,154828.16]
2	$18.75 \leq x_1 \leq 31.25$ $50 \leq x_2 \leq 56.25$	[24.9,51.57]	[31.25,50]	[4.87,107074.02]	[1.15,113024.49]
3	$43.75 \leq x_1 \leq 56.25$ $43.75 \leq x_2 \leq 62.5$	[51.53,45.89]	[51.25,49.56]	[3.53,99459.20]	[1.31,99758.55]
4	$56.25 \leq x_1 \leq 68.75$ $56.25 \leq x_2 \leq 68.75$	[62.71,62.61]	[56.25,56.25]	[4.37,100751.36]	[2.46,105298.47]
5	$93.75 \leq x_1 \leq 100$ $31.25 \leq x_2 \leq 43.75$	[100,36.78]	[93.75,43.75]	[5.75,138563.86]	[0.92, 152962.18]

Suppose that the DM wants to search the regions with the exact approach and start the search from region 1. Assume that the DM specifies $\boldsymbol{G}^0 = (6.0, 120000)$ as the reference point and $\boldsymbol{w}^0 = (0.9, 0.1)$ as the weight vector.

The closest nondominated integer vector to $\boldsymbol{G^o}$ in region 1 is found as $\boldsymbol{C^1} = (5.83, 154341.55)$ with $\boldsymbol{x} = (0.24, 36.72)$. $\boldsymbol{C^1}$ is not dominated by the *LIS*, so we project it onto the other regions by solving *ASP* to perform dominance check. Following solutions are generated: $\boldsymbol{C^2} = (4.86, 112598.92)$ from region 2, $\boldsymbol{C^3} = (3.52, 99758.95)$ from region 3, $\boldsymbol{C^4} = (4.36, 105404.36)$ from region 4, $\boldsymbol{C^5} = (5.75, 152962.18)$ from region 5. These solutions are not dominated by the *LIS*, thus they are added to the list of candidate nondominated vectors (*LCNV*).

As a result, since none of the projections dominates $\boldsymbol{C^1}$, $\boldsymbol{C^1}$ is proved to be nondominated and added to the *LCLP*.

Suppose that the DM would like to know whether $\boldsymbol{C^2}$ is nondominated or not. It is deleted from the *LCNV*. Let $\boldsymbol{G}^0 = \boldsymbol{C^2} = (4.86, 112598.92)$ and $w^0 = (0.9, 0.1)$. $\boldsymbol{C^2}$ is projected onto the other regions to perform dominance check. Following solutions are generated: $\boldsymbol{C^6} = (4.70, 149226.73)$ from region 1, $\boldsymbol{C^7} = (3.52, 99758.95)$ from region 3, $\boldsymbol{C^8} = (4.36, 105330.48)$ from region 4, $\boldsymbol{C^9} = (4.70, 150823.45)$ from region 5. These solutions are not dominated by the *LIS*, thus they are added to the *LCNV*.

As a result, since none of the projections dominates $\boldsymbol{C^2}$, $\boldsymbol{C^2}$ is proved to be nondominated and added to the *LCLP*.

Suppose the DM want to stop the search. The *LCLP* is presented to the DM. The elements of the *LCLP* are $\boldsymbol{C^1} = (5.83, 154341.55)$ and $\boldsymbol{C^2} = (4.86, 112598.92)$. The DM would prefer point $\boldsymbol{C^2}$ based on the assumed underlying utility function. Hence, the semi-desirable facility should be located in region 2 at $\boldsymbol{x} = (25.28, 51.18)$. In this case the facility is 4.86 units far from the closest demand point, while it has a total distance of 112598.92 units from all the demand points. Note that the IGBSSS method found the best solution that maximizes the underlying utility function ($\boldsymbol{U} = 0.26$).

8 Conclusion

In this study, we addressed a single facility location problem with maximin and minisum objectives in a continuous planar region where the distances are measured in the rectilinear metric. We proposed a three-phase interactive geometrical branch and bound algorithm, called IGBSS method, for the solution of the problem. In the first two phases, we aim to eliminate the parts of the feasible region the inefficiency of which can be proved. The third phase has been suggested for an interactive search in the remaining regions with the involvement of a DM.

In the third phase, the DM is given the opportunity to use either an exact or an approximate procedure to carry out the search. In the exact one, finding an efficient point at the end is guaranteed. This procedure is based on the reference point approach of Wierzbicki [25]. On the other hand, in the approximate procedure, a locally efficient solution can be presented at the end. In this procedure, a hybrid methodology [9] is used to increase the efficiency of the reference point approach. In the early iterations, we search the nondominated solutions of the reduced continuous problem using the reference direction approach suggested by Korhonen and Laakso [11]. In later iterations, when approaching the DM's most preferred region, we find integer nondominated solutions using the reference point approach. The approximate procedure can be used when the DM prefers to see locally efficient solutions to save computation time.

The first two phases of the algorithm was an adaptation of the GBSSS algorithm to the semi-desirable facility location problem with new bounding schemes. However, the third phase was completely new considering that there is no interactive approach suggested for the semi-desirable facility location problem in the literature. The solution approaches in the literature are either approximations that result in regions containing efficient points, or they are aimed at obtaining the complete efficient trajectory. Obviously, these approaches may cause information overload on the DM who may have difficulty in selecting the final location point. Proposed method gives the DM an opportunity to select a single -efficient or locally efficient- location point as the most preferred solution.

An area for future research should consider forbidden regions which allow modeling real location areas with geographical barriers. A new distance gauge that properly defines the spread of pollution should be investigated. Moreover, different criteria involving environmental considerations, such as geographical, climatic, should be incorporated to the problem of this area. Another area is the multi-facility version of the problem which will be useful in modeling the real life location problems.

Appendix

Interactive Generalized Big Square Small Square (IGBSSS) Algorithm

Notation

LCES	: List of candidate efficient subregions
LIS	: List of incumbent solutions
LCLP	: List of candidate location points
LCNV	: List of candidate nondominated vectors
SCES	: Sublist of candidate efficient subregions
n	: Index used to denote the subregion
i	: Index used to denote the demand point
r	: Largest value of index n
S^n	: Subregion n
$\boldsymbol{b^i}$	: Demand point i
N	: Number of demand points

θ_i :Weight associated with the ith demand point, a decreasing function of the population

λ_i :Weight associated with the ith demand point, an increasing function of the population

T : Number of points selected from a subregion in each branching.

$LFDP(S^n)$: List of filtered demand points for subregion S^n

$UB(S^n)$: Upper bound on optimal maximin function value in subregion S^n

$L^*(S^n)$: Optimal maximin function value in subregion S^n

$W^*(S^n)$: Optimal minisum function value in subregion S^n

$\boldsymbol{R}(S^n)$: Ideal objective vector of subregion S^n

$\boldsymbol{Q}(S^n)$: Nadir objective vector of subregion S^n

$d(\boldsymbol{b}^i, S^n)$: Smallest rectilinear distance between demand point $\boldsymbol{b^i} \notin S^n$ and subregion S^n

Phase 1: Rough cut phase: pruning with UB (*Maximin-l^1*) and (*Minisum-l^1*)

Step1.0 Initialization

- Define either a square or a rectangle that is large enough to contain the feasible region and with sides parallel to the axis and denote it by S^0.
- Ask the decision maker (DM) to specify the stopping side length, β (In theory, β can take on any value; however, we suggest using at least one eight of the longer side length of the rectangle for β).
- Set $m = 0, n = 0, r = 0$, $LIS = \{\}$ and $LCES = \{\}$.
- Put S^0 into the *LCES*

Step1.1 Branching and Pruning

- Let S^m be the selected subregion. Delete S^m from the *LCES*
- Pick T feasible points from S^m. Let these points be $\{\boldsymbol{x^1}, \boldsymbol{x^2}, \ldots, \boldsymbol{x^T}\}$
- Evaluate $L(\boldsymbol{x^t})$ and $W(\boldsymbol{x^t})$ for each of these T points

$$L(\boldsymbol{x^t}) = \left\{\min_{i=1,\ldots,N} \theta_i \left(\left|x_1^t - b_1^i\right| + \left|x_2^t - b_2^i\right|\right)\right\} \quad \text{for } t = 1, \ldots, T$$

$$W(\boldsymbol{x^t}) = \left\{\sum_{i=1}^{N} \lambda_i \left(\left|x_1^t - b_1^i\right| + \left|x_2^t - b_2^i\right|\right)\right\} \quad \text{for } t = 1, \ldots, T$$

- Add these points to the *LIS* after a dominance check.
  Add $(L(\boldsymbol{x^t}), W(\boldsymbol{x^t}))$ to the *LIS* if there does not exist another objective vector in the *LIS*, $(L(\boldsymbol{x^h}), W(\boldsymbol{x^h}))$ such that $L(\boldsymbol{x^t}) \leq L(\boldsymbol{x^h})$, $W(\boldsymbol{x^t}) \geq W(\boldsymbol{x^h})$ and $(L(\boldsymbol{x^t}), W(\boldsymbol{x^t})) \neq (L(\boldsymbol{x^h}), W(\boldsymbol{x^h}))$.
  Check whether any element of the *LIS*, $(L(\boldsymbol{x^h}), W(\boldsymbol{x^h}))$, is dominated by the newly added objective vector, $(L(\boldsymbol{x^t}), W(\boldsymbol{x^t}))$. If $L(\boldsymbol{x^h}) \leq L(\boldsymbol{x^t})$, $W(\boldsymbol{x^h}) \geq W(\boldsymbol{x^t})$ and $(L(\boldsymbol{x^t}), W(\boldsymbol{x^t})) \neq (L(\boldsymbol{x^h}), W(\boldsymbol{x^h}))$, delete $(L(\boldsymbol{x^h}), W(\boldsymbol{x^h}))$ from the *LIS*.
- Compare the *LCES* with the newly generated incumbent solutions. Delete S^n from the *LCES* if there exists a newly generated solution such that $UB(S^n) \leq L(\boldsymbol{x^t})$, $W^*(S^n) \geq W(\boldsymbol{x^t})$.
- Divide S^m into four equal subregions. Number the subregions from S^{r+1} to S^{r+4}. Set $r = r + 4$.

- Check if the newly generated subregions, $S^n(n = r+1, \ldots, r+4)$, have an intersection with the feasible region, i.e., $S^n \cap S^0 \neq \phi$. If $S^n \cap S^0 = \phi$, delete S^n from further consideration.
- For each newly generated subregion, S^n, solve *UB*(*Maximin-l^1*) *w*ith all existing demand points and (*Maximin-l^1*) with the demand points located inside the subregion. Choose the smallest optimal objective function value as the $UB(S^n)$.
- In order to find the optimal minisum objective value for newly generated subregion, S^n, solve (*Minisum-l^1*) with all existing demand points.
- Compare newly generated subregions, S^n, with the *LIS*. Add S^n to the *LCES* if $UB(S^n) > L(\boldsymbol{x^h})$ or $W^*(S^n) < W(\boldsymbol{x^h})$ for all $\boldsymbol{x^h}$ in the *LIS*.
- Check whether the side length of the largest subregion in *LCES* is less than β. If not, then go to Step 1.2. Otherwise, stop Phase 1, let $SCES = \{S^n | \, S^n \in LCES\}$, $LCES = \{\}$. and go to Phase 2.

Step 1.2 Selecting Where to Branch Next

- Select the subregion with maximum $UB(S^n)$ or minimum $W^*(S^n)$
- Return to Step 1.1.

Phase 2: Precise cut phase: pruning with (*Maximin-l^1*) and (*Minisum-l^1*)

Step 2.0 Initialization

- Ask the DM to specify the stopping side length, δ (In theory, δ can take on any value; however, we suggest using at least one sixteenth of the longer side length of the rectangle for δ).
- Set $LFDP(S^n) = \{\}$ for all n.

Step 2.1 Pruning with Optimal Values

- Calculate the smallest weighted rectilinear distances of the demand points $\boldsymbol{b^i} \notin S^n$ to subregion S^n in the *SCES*. Add the demand points $\boldsymbol{b^i} \in S^n$ as well as the demand points whose weighted distance to S^n is smaller than $UB(S^n)$ to the $LFDP(S^n)$, i.e., for all n and i, if $\boldsymbol{b^i} \in S^n$ or $\theta_i d(\boldsymbol{b^i}, S^n) \leq UB(S^n)$ for $\boldsymbol{b^i} \notin S^n$ then add $\boldsymbol{b^i}$ to the $LFDP(S^n)$.
- For each subregion S^n in the *SCES* solve (*Maximin-l^1*) with the demand points in the $LFDP(S^n)$.
- Compare the ideal objective vector of subregions S^n in the *SCES* with the *LIS*. Add S^n to the *LCES* if $L^*(S^n) > L(\boldsymbol{x^h})$ or $W^*(S^n) < W(\boldsymbol{x^h})$ for all $\boldsymbol{x^h}$ in the *LIS*.
- Check whether the side length of the largest subregion in *LCES* is less than δ. If not, go to Step 2.2. Otherwise, go to Step 2.3.

Step 2.2 Further Branching

- Set *SCES* = {}.
- Select the subregion with maximum $UB(S^n)$ or minimum $W^*(S^n)$.
- Let S^m be the selected subregion. Delete S^m from the *LCES*. Divide S^m into four equal subregions. Number the subregions from S^{r+1} to S^{r+4}. Set r=r+4.
- Check if the newly generated subregions, S^n, have an intersection with the feasible region, $S^n \cap S^0 \neq \phi$. If $S^n \cap S^0 \neq \phi$, add S^n to the *SCES*.
- For each subregion S^n in the *SCES* solve (*Minisum-l^1*) with all existing demand points. Go to Step 2.1.

Step 2.3 Stopping

- Combine the remaining subregions.

Phase 3: Interactive search

- Ask the DM which procedure she/he wants to use: exact or approximate.

A. Exact Procedure

Step 3.0 Initialization

- Set $LCLP = \{\}$ and $LCNV = \{\}$.

Step 3.1 Starting the Search and Finding a Nondominated Integer Solution

- Present each region with $\boldsymbol{R}(S^n)$, $\boldsymbol{Q}(S^n)$ to the DM. Ask the DM to choose a region.
- Ask the DM to specify her/his reference point $\boldsymbol{G}^0 = (G_1^0, G_2^0)$.
- Ask the DM which objective is important, and how much. Set the initial vector of weights accordingly $\boldsymbol{w}^0 = (w_1^0, w_2^0)$.
- Solve *ASP* for the region. Let the solution be $\boldsymbol{C}$. Check whether $\boldsymbol{C}$ is dominated by the *LIS* and *LCLP*. If so, then delete it from further consideration.

Step 3.2 Finding Nondominated Integer Solutions

- Project $\boldsymbol{C}$ to the other regions by solving *ASP* with $\boldsymbol{w}^0$. Compare the projections of $\boldsymbol{C}$ with solution $\boldsymbol{C}$. If none of them dominates $\boldsymbol{C}$ then $\boldsymbol{C}$ is proved to be a nondominated solution and added to the *LCLP*. Otherwise, delete $\boldsymbol{C}$ from further consideration.
- Compare the projections of $\boldsymbol{C}$ with *LIS* and *LCLP*. Put the nondominated ones to the *LCNV*.
- Ask the DM if she/he wants to continue the search from one of the solutions in *LCNV.* If so, repeat this step with the selected solution. Otherwise, go to Step 3.3.

Step 3.3 Stopping the Search

- Present the *LCLP* to the DM. Ask her/him if she/he wants to stop searching. If so, ask the DM to select one of the alternatives in the *LCLP* as the most preferred alternative and stop. Otherwise, go to Step 3.1.

B. Approximate procedure

Step 3.0 Initialization

- Set $LCLP = \{\}$.

Step 3.1 Starting the Search.

- Present each region with $\boldsymbol{R}(S^n)$, $\boldsymbol{Q}(S^n)$ to the DM. Ask the DM to choose a region.

Step 3.2 Finding a Starting Nondominated Continuous Solution

- Ask the DM to specify her/his reference point $\boldsymbol{G}^0 = (G_1^0, G_2^0)$.
- Ask the DM which objective is important, and how much. Set the initial vector of weights accordingly $\boldsymbol{w}^0 = (w_1^0, w_2^0)$.
- Solve the *ASPLP* to find a starting continuous solution closest to the reference point. Let the solution be $\boldsymbol{z}^0 = (z_1^0, z_2^0)$.
- If the DM likes $\boldsymbol{z}^0 = (z_1^0, z_2^0)$, then set $\boldsymbol{K} = \boldsymbol{z}^0$ and go to Step 3.4. Else, go to Step 3.3.

Step 3.3 Generating Alternative Nondominated Continuous Solutions

- Ask the DM to specify a reference direction, $\boldsymbol{\Delta d} = (\Delta d_1, \Delta d_2)$.
- Ask the DM the number of solutions that she/he wants to see. Let this number be P.

- Solve the *ASPLP* for $p = 1, \ldots, P$. Let the solutions be $z^1, \ldots, z^{\boldsymbol{P}}$.
- Present the solutions to the DM. Ask her/him if she/he wants to change her/his reference point. If so, return to Step 3.1. Otherwise, ask him to select one of the continuous solutions $z^1, \ldots, z^{\boldsymbol{P}}$. Let $\boldsymbol{K}$ be the selected continuous solution. Go to Step 3.4.

Step 3.4 Finding an Integer Nondominated Solution

- Solve the *ASP* to find the closest integer solution to solution $\boldsymbol{K}$. Let the solution be $\boldsymbol{C} = (C_1, C_2)$.
- Check if $\boldsymbol{C}$ is dominated by one of the elements of the *LIS* or the *LCLP*. If so, delete $\boldsymbol{C}$ from further consideration. Else, add it to the *LCLP*.
- Ask the DM if she/he wants to search the region under consideration further. If yes, return to Step 3.2. Otherwise go to Step 3.5.

Step 3.5 Stopping the Search

- Present the *LCLP* to the DM. Ask her/him if she/he wants to stop searching. If so, ask the DM to select one of the alternatives in the *LCLP* as the most preferred alternative and stop. Otherwise, return to Step 3.1.

References

1. Ahuja, R.K., Ergun, Ö., Orlin, J.B., Punnen, A.P.: A survey of very large-scale neighborhood search techniques. Discrete Appl. Math. **123**(1–3), 75–102 (2002)
2. Appa, G.M., Giannikos, I.: Is linear programming necessary for single facility location with maximin of rectilinear distance? J. Oper. Res. Soc. **45**, 97–107 (1994)
3. Brimberg, J., Juel, H.: On locating a semi-desirable facility on the continuous plane. Int. Trans. Oper. Res. **5**(1), 59–66 (1998a)
4. Brimberg, J., Juel, H.: A bicriteria model for locating a semi-desirable facility in the plane. Eur. J. Oper. Res. **106**, 144–151 (1998b)
5. Carrizosa, E., Plastria, F.: Location of semi-obnoxious facilities. Stud. Locational Anal. **12**, 1–27 (1999)
6. Cplex Using the Cplex Callable Library, Version 6.0, ILOG, Inc., NV, USA (1998)
7. Drezner, Z., Wesolowsky, G.O.: The location of an obnoxious facility with rectangular distances. J. Reg. Sci. **23**, 241–248 (1983)
8. Hansen, P., Peeters, D., Thisse, J.F.: On the location of an obnoxious facility. Sist. Urbani **3**, 299–317 (1981)
9. Karaivanova, J., Korhonen, P., Narula, S., Wallenius, J., Vassilev, V.: A reference direction approach to multiple objective integer linear programming. Eur. J. Oper. Res. **81**, 176–187 (1995)
10. Kaliszewski, I.: Quantitative Pareto Analysis by Cone Seperation Technique. Kluwer, Dordrecht (1994)
11. Korhonen, P., Laakso, J.: A visual interactive method for solving the multiple criteria problem. Eur. J. Oper. Res. **23**, 161–179 (1986)
12. Mehrez, A., Sinuany-Stern, Z., Stulman, A.: A single facility location problem with a weighted maximin–minimax rectilinear distance. Comput. Oper. Res. **12**, 51–60 (1985)
13. Mehrez, A., Sinuany-Stern, Z., Stulman, A.: An enhancement of the Drezner-Wesolowsky algorithm for single-facility location with maximin of rectilinear distance. J. Oper. Res. Soc. **37**, 971–977 (1986)
14. Melachrinoudis, E.: Bicriteria location of a semi-obnoxious facility. Comput. Ind. Eng. **37**, 581–593 (1999)
15. Melachrinoudis, E., Xanthopulos, Z.: Semi-obnoxious single facility location in Euclidean space. Comput. Oper. Res. **30**, 2191–2209 (2003)
16. Nadirler, D., Karasakal, E.: Mixed integer programming-based solution procedure for single-facility location with maximin of rectilinear distance. J. Oper. Res. Soc. (2007) (to appear)
17. Ohsawa, Y.: Bicriteria Euclidean location associated with maximin and minimax criteria. Nav. Res. Log. **47**, 581–592 (2000)
18. Özerol, G., Karasakal, E.: Interactive outranking approaches for multicriteria decision making problems with imprecise information. J. Oper. Res. Soc. (2007) (to appear)
19. Plastria, F.: GBSSS: The generalized big square small square method for planar single-facility location. Eur. J. Oper. Res. **62**, 163–174 (1992)

20. Romero-Morales, D., Carrizosa, E., Conde, E.: Semi-obnoxious location models: a global optimization approach. Eur. J. Oper. Res. **102**, 295–301 (1997)
21. Sayin, S.: A mixed integer programming formulation for the l-maximin problem. J. Oper. Res. Soc. **51**, 371–375 (2000)
22. Skriver, A.J.V., Andersen, K.A.: The bicriterion semi-obnoxious location (BSL) problem solved by an ε-approximation. Eur. J. Oper. Res. **146**, 517–528 (2003)
23. Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation and Application. John Wiley, New York (1986)
24. Yapicioglu, H., Smith, A.E., Dozier, G.: Solving the semi-desirable facility location problem using bi-objective particle swarm. Eur. J. Oper. Res. **177**, 733–749 (2007)
25. Wierzbicki, A.: The use of reference objective in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making, Theory and Application, pp. 468–486. Springer-Verlag, Berlin (1980)
26. White, D.J.: Rectilinear location revisited. J. Oper. Res. Soc. **47**, 181–187 (1996)