

# A multi-objective genetic algorithm for a bi-objective facility location problem with partial coverage

Esra Karasakal<sup>1</sup> · Ahmet Silav<sup>1</sup>

Received: 13 February 2014 / Accepted: 31 May 2015 / Published online: 10 June 2015  
© Sociedad de Estadística e Investigación Operativa 2015

**Abstract** In this study, we present a bi-objective facility location model that considers both partial coverage and service to uncovered demands. Due to limited number of facilities to be opened, some of the demand nodes may not be within full or partial coverage distance of a facility. However, a demand node that is not within the coverage distance of a facility should get service from the nearest facility within the shortest possible time. In this model, it is assumed that demand nodes within the predefined distance of opened facilities are fully covered, and after that distance the coverage level decreases linearly. The objectives are defined as the maximization of full and partial coverage, and the minimization of the maximum distance between uncovered demand nodes and their nearest facilities. We develop a new multi-objective genetic algorithm (MOGA) called modified SPEA-II (mSPEA-II). In this method, the fitness function of SPEA-II is modified and the crowding distance of NSGA-II is used. The performance of mSPEA-II is tested on randomly generated problems of different sizes. The results are compared with the solutions of the most well-known MOGAs, NSGA-II and SPEA-II. Computational experiments show that mSPEA-II outperforms both NSGA-II and SPEA-II.

**Keywords** Multi-objective genetic algorithm · Facility location · Maximal coverage · Partial coverage · P-center

**Mathematics Subject Classification** 90C29 · 90B80 · 90C59

---

✉ Esra Karasakal  
kuktener@metu.edu.tr

<sup>1</sup> Industrial Engineering Department, Middle East Technical University, Dumlupınar Bulvarı, 06800 Ankara, Turkey

## 1 Introduction

Most facility location problems are multi-objective in nature and several objectives should be considered simultaneously. The maximal covering location problem (MCLP) is a well-known location problem where a fixed number of facilities are located to maximize the demand covered. In the classical MCLP, a demand node is defined as being covered if it is within a critical distance from at least one of the facilities, and not covered if it is outside the critical distance. Since the introduction by Church and ReVelle (1974), various extensions of MCLP have been developed in the literature (see Schilling et al. 1993; Berman et al. 2010 for reviews). The MCLP has many applications in public and private sectors, especially in locating emergency facilities (see Goldberg 2004 for a review).

The definition of coverage was extended to model situations where coverage (i.e., the level of service provided by the facility) does not change in a crisp way from “fully covered” to “not covered” at the critical distance. Karasakal and Karasakal (2004) introduced the partial coverage concept in a maximal covering location model. They assumed that the demand node was partially covered up to a maximum critical distance and fully covered within the minimum critical distance. Berman et al. (2003) modeled a covering location problem using a gradual concept. The coverage provided by the facility is assumed to decrease gradually after a certain distance.

A number of researchers have developed coverage models with more than one objective (see Farahani et al. 2010 for a review). Some of them considered backup coverage besides the first coverage (Hogan and ReVelle 1986; Storbeck and Vohra 1988). Church et al. (1991) developed a bi-objective maximal covering location model. In addition to the coverage objective, an objective that minimizes the sum of the weighted distance between uncovered demand and the nearest facility is formulated. Badri et al. (1998) proposed a goal programming model for locating fire stations, which considers objectives related to cost, demand, certain system requirements, technical and political strategies. Araz et al. (2007) developed a multi-objective coverage-based model. The maximization of the first coverage of demands, maximization of backup coverage and minimization of the total travel distance between uncovered demands and opened facilities are defined as objectives.

In MCLPs, full or partial coverage of all demand nodes may not always be possible due to limited number of facilities. However, in real life, especially emergency facilities such as ambulance, police, and fire stations should give service to all demand nodes including those that are not within the coverage distance of a facility. Another example may be in the location of Wi-Fi access points at large areas such as airports. People outside the coverage areas should go to the nearest access points. To take into account the quality of service to uncovered demand nodes, the maximum distance between uncovered demands and its nearest facilities (i.e., p-center objective) should also be minimized.

To the best of our knowledge, there are no studies that use the p-center objective to handle the uncovered demands in MCLPs. We present a bi-objective facility location model that considers both partial coverage and service to uncovered demand nodes. We assume that demand nodes within the predefined distance of opened facilities are fully covered and after that the distance coverage level decreases linearly. We

also assume that an uncovered demand can be served by its nearest facility when it is needed. Therefore, we define the second objective as the minimization of the maximum distance between any uncovered demand node and its nearest facility.

The MCLP is NP-hard (Megiddo et al. 1983). Thus, bi-objective MCLP addressed in the paper is also NP-hard. Many heuristic and metaheuristic methods have been developed to solve large-size MCLPs since it is cumbersome to solve large-size problems using exact methods (e.g., Karasakal and Karasakal 2004; Fazel Zarandi et al. 2011; ReVelle et al. 2008). In this paper, we develop a new multi-objective genetic algorithm (MOGA) to solve the bi-objective MCLP.

We develop a new multi-objective genetic algorithm (MOGA) called modified SPEA-II (mSPEA-II). We modify the fitness function of SPEA-II and use the crowding distance of NSGA-II. The performance of mSPEA-II is tested on randomly generated problems of different sizes. Results are compared with the solutions of NSGA-II and SPEA-II, since they are the representatives of the state-of-the-art MOGAs (Coello Coello et al. 2007, p. 274; Zhou et al. 2011; Durillo et al. 2009; Pohl and Lamonth 2008). Although there are many variations of these state-of-the-art algorithms, original versions are used in comparative studies of the newly developed MOGAs (Coello Coello et al. 2007, p. 98; Konak et al. 2006).

The organization of the paper is as follows. In the next section, we provide a brief review on the application of MOGAs to multi-objective facility location problems. In Sect. 3, we present the mathematical model. We develop the approach in Sect. 4. We report our computational experiments in Sect. 5 and present concluding remarks in the last section.

## 2 Literature review

MOGAs are developed to search for nondominated solutions in multi-objective optimization problems (see Ishibuchi et al. 2008 for a review). Deb (2001) categorizes MOGAs into two groups: non-elitist MOGAs and elitist MOGAs. Non-Dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al. 2002) and Strength Pareto Evolutionary Algorithm (SPEA-II) (Zitzler et al. 2001) are the best known elitist MOGAs in literature.

MOGAs are one of the most suitable approaches for multi-optimization problems (Alberto and Mateo 2011), because they generate population of solutions and find entire set of Pareto optimal solutions in a single run (Ghosh and Dehuri 2004).

Li et al. (2011) stated that in emergency facility location models, GAs are one of the most widely used techniques for solving problems. Aytug et al. (2003) listed the production and operation management problems that used GAs and evaluated the state of GA applications in these areas. Jaramillo et al. (2002) demonstrated that performance of GAs are more robust than Lagrangian heuristic in solving location problems.

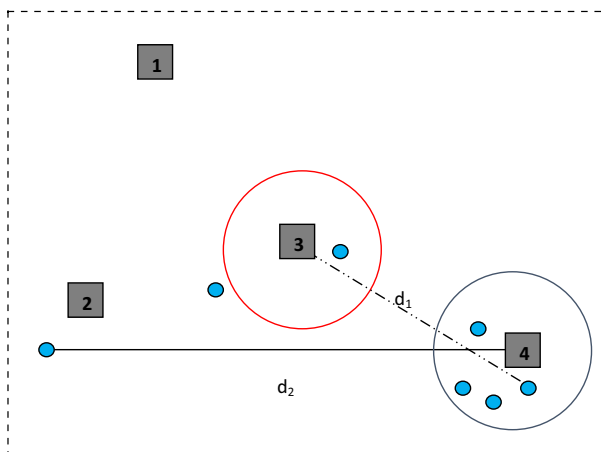
There are several studies that use GAs to solve coverage problems. Yigit et al. (2006) formulated uncapacitated facility location problem and used simulated annealing for local search and evolutionary approach for global search to obtain the desired solutions. Mahdavi et al. (2012) proposed a multi-facility location problem. They tried to find

the optimal locations of new facilities by minimizing the total weighted expected traveled rectilinear distances from the new facilities to those existing. They used genetic algorithm for large-size problems. [Gutjahr et al. \(2009\)](#) presented a multi-objective model for the public facilities. The objectives are based on weighted average of minimum and maximum coverage, minimizing the risk of tsunami for alternative locations and minimizing the cost, respectively. They used NSGA-II to solve the problem. [Bhattacharya and Bandyopadhyay \(2010\)](#) modeled a bi-objective facility location problem and solved the model by NSGA-II. [Li and Yeh \(2004\)](#) integrated the genetic algorithm with geographical information systems to solve a location problem whose objectives are the maximization of population coverage, the minimization of total transportation costs and the minimization of the proximity to roads. [Villegas et al. \(2006\)](#) modeled a bi-objective uncapacitated facility location problem whose objectives are the minimization of the total cost of assigning facilities to related demand nodes and fixed facility opening costs, and the maximization of coverage. They applied multi-objective evolutionary algorithms and an algorithm based on mathematical programming. [Yang et al. \(2007\)](#) proposed a model based on fuzzy multi-objective programming and a genetic algorithm. The achievement levels for fuzzy objectives were defined. The fuzzy multi-objective optimization model was converted into a single unified goal and a genetic algorithm was applied to the problem. [Medaglia et al. \(2009\)](#) formulated a bi-objective model to design a hospital waste management network. They integrated NSGA-II into mixed integer for the solution of the bi-objective problem.

### 3 Formulation of the problem

We formulate a bi-objective model that maximizes total coverage of the facilities and minimizes the maximum distance between uncovered demand nodes and the nearest facilities.

In [Fig. 1](#), there are four alternative facilities which are numbered from one to four and one of them will be opened. Demand nodes are denoted by points and assumed



**Fig. 1** An example

to have equal weights. The circles around the facilities show the coverage areas of the facilities. If the maximum distance between the demand nodes and their nearest facilities is minimized, facility 3 is selected. The maximum distance to uncovered demand nodes is  $d_1$  and the number of uncovered demand nodes is 6. On the other hand if full and partial coverage is minimized, facility 4 is opened and the number of uncovered demand nodes is 3, but the maximum distance is  $d_2$  which is greater than  $d_1$ . These objectives generally conflict with each other in most of the real-life location problems.

Notation used in the model is as follows:

- $I$  set of demand nodes,
- $J$  set of potential facility sites,
- $a_i$  demand at node  $i$ ,
- $p$  number of facilities to be sited,
- $d_{ij}$  shortest travel distance separating node  $i$  from node  $j$ ,
- $T$  maximum partial coverage distance,
- $S$  maximum full coverage distance,
- $\alpha$  maximum distance between an uncovered demand node and its nearest facility.

$$M_i = \{j \mid d_{ij} \leq T\}$$

$$c_{ij} = \begin{cases} 1 & \text{if } d_{ij} \leq S, \\ f(d_{ij}) & \text{if } S \leq d_{ij} \leq T, \quad (0 < f(d_{ij}) < 1), \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if uncovered demand node } i \text{ is assigned to facility } j, \\ 0 & \text{otherwise,} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if demand node } i \text{ is not within } T \text{ of any facility,} \\ 0 & \text{otherwise,} \end{cases}$$

$$f_j = \begin{cases} 1 & \text{if a facility is sited at node } j, \\ 0 & \text{otherwise,} \end{cases}$$

$$s_{ij} = \begin{cases} 1 & \text{if demand at point } i \text{ is either partially or fully covered by facility } j, \\ 0 & \text{otherwise.} \end{cases}$$

The formulation of the bi-objective MCLP is as follows:

$$\begin{aligned} \text{Min } Z_1 &= \alpha \\ \text{Max } Z_2 &= \sum_{i \in I} \sum_{j \notin M_i} c_{ij} a_i s_{ij} \end{aligned}$$

s.t.

$$\sum_{j \in M_i} f_j + y_i \geq 1 \quad \forall i \in I, \tag{1}$$

$$\sum_{j \notin M_i} x_{ij} - y_i = 0 \quad \forall i \in I, \tag{2}$$

$$f_j - x_{ij} \geq 0 \quad \forall i \in I, \quad j \notin M_i, \tag{3}$$

$$\sum_{j \in J} f_j = p, \tag{4}$$

$$\sum_{j \notin M_i} x_{ij} \cdot d_{ij} \leq \alpha \quad \forall i \in I, \tag{5}$$

$$s_{ij} - f_j \leq 0 \quad \forall i \in I, \quad j \notin M_i, \tag{6}$$

$$\sum_{j \in M_i} s_{ij} \leq 1 \quad \forall i \in I, \tag{7}$$

$$x_{ij}, s_{ij} \in \{0, 1\} \quad \forall i \in I, \quad j \in J,$$

$$y_i \in \{0, 1\} \quad \forall i \in I,$$

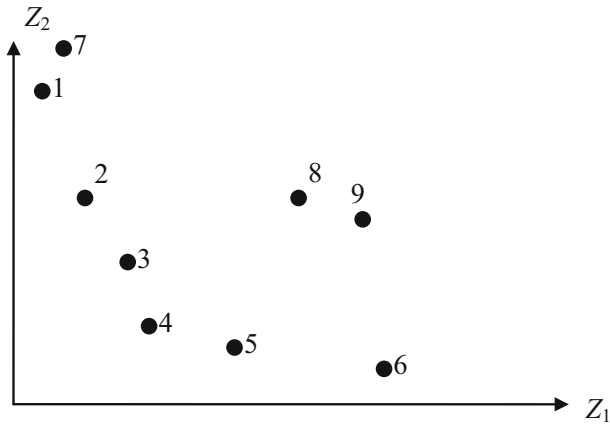
$$f_j \in \{0, 1\} \quad \forall j \in J.$$

The first objective minimizes the maximum distance between uncovered demand nodes and the nearest facilities. The second objective maximizes the total coverage including both partial and full coverage.

Constraint set (1) ensures that if there is not any opened facility within the distance  $T$  for a demand node, then that demand node is an uncovered demand node and the variable  $y$  is forced to be 1. Constraint set (2) is formulated to assign the uncovered demand to an opened facility, because if the distance between an opened facility and demand node  $i$  is less than  $T$ , then that demand node is covered and  $y_i$  is 0, so all the  $x_{ij}$  values are forced to be 0 for that demand node. However, if a demand node is not covered, then  $y_i$  is 1 and constraint set (2) requires the demand node be assigned to a facility. The constraint set (3) ensures that an uncovered demand node be assigned to a facility only if a facility is located at node  $j$ . Thus, if  $f_j$  is 0, this means that a facility is not opened at node  $j$ , so all the corresponding  $x_{ij}$  values are 0. If  $f_j$  is 1, then uncovered demand may be assigned to this facility, so some  $x_{ij}$  values may be 1. Constraint (4) restricts the number of opened facilities. Constraint set (5) calculates the maximum of the distances between uncovered demand nodes and their nearest facilities. The constraint sets (6) and (7) are peculiar to the partial coverage concept. If the  $j$ th facility is not opened, then all its  $s_{ij}$  values are forced to be 0 and the demand nodes will not be assigned to that facility, but if the  $j$ th facility is opened then the values  $s_{ij}$  may be 0 or 1 and may get a facility assigned to it or not. This will be determined together with the constraint set (7), because constraint set (7) allows any demand point to be covered by at most one of the facilities sited. If there is more than one facility covering a demand node, then the second objective determines which of them will provide service.

#### 4 A new multi-objective genetic algorithm (mSPEA-II)

NSGA-II proposed by Deb et al. (2002) incorporates elitism by allowing the best solutions from the parent and offspring pools of chromosomes to be retained. NSGA-II uses nondominated sorting instead of a fitness function to classify the entire population, and calculates crowding distance to maintain a good spread of solutions in the solution



**Fig. 2** A sample population

set. Nondominated sorting gives the same priority to all members in the same front. This may cause problems in converging the true Pareto front in cases where some members are dominated by many members of the other front, and others are dominated by a few members. A simple bi-objective problem where objectives are to be minimized is depicted in Fig. 2. There are two fronts. Solutions 7 and 9 are in the second front. Solution 7 is dominated only by solution 1. Solution 9 is dominated by three solutions, 3, 4 and 5. Since they are boundary solutions in the second front, their crowding distances are the same and very large, meaning that they have the same priority for selection. However, solution 7 should have more priority than solution 9, because solution 7 should be kept in the population to generate solutions having larger  $Z_2$  values and smaller  $Z_1$  values than that of solution 1 and converge the true Pareto-optimal front quickly.

The Strength Pareto Evolutionary Algorithm (SPEA-II) proposed by Zitzler et al. (2001) uses a regular population and an archive (external) set. New generated offsprings are stored in a regular set, whereas nondominated solutions are stored in an archive set. Fitness assignment, based on the domination concept and density estimation, is done for the union of these sets. The fitness of a solution is the sum of raw fitness and density fitness values. The strength of each member in the archive and regular populations is calculated. The greater strength means that it dominates more members in the population. The raw fitness of a solution is calculated by summing up the strength of all the solutions that dominate the given solution. Also, for each member in the union set, the distances to all other solutions are calculated and sorted in increasing order, and density fitness value is calculated to differentiate those solutions which have the same raw fitness values. Density fitness values are used in a truncation operation when the actual archive is greater than the fixed archive size. Calculating the density fitness values and using the truncation operation increase the complexity of the SPEA-II algorithm. Experiments demonstrate that the computational complexity of SPEA-II is considerably greater than that of NSGA-II (Eskandari et al. 2007).

To decrease the computational complexity of SPEA-II and keep good solutions to maintain the diversity and convergence to the true Pareto front, we propose a genetic

algorithm based on SPEA-II (mSPEA-II). In mSPEA-II, instead of density fitness,  $D(i)$ , the objective fitness values,  $\min O(i)$ , are calculated. For each minimization objective, the solutions are sorted in ascending order.  $\min O(i)$  is set to the minimum rank among all objectives. For instance, the objective function values of the bi-objective minimization problem depicted in Fig. 2 are given in Table 1.

These solutions are sorted according to both objectives as shown in Table 2.

As seen in Table 2,  $\min O(7)$  is 2, because it is in the second rank with respect to the first objective,  $Z_1$ , and in the ninth rank with respect to the second objective,  $Z_2$ , so the minimum is 2.

A fitness value for each solution is determined as follows:

$$F_1(i) = R(i) + \frac{\min O(i)}{\text{no. of solutions} + 1}, \tag{8}$$

where  $R(i)$  is the raw fitness value of a solution, calculated as in SPEA-II. Objective fitness value is divided by “total number of solutions + 1” to obtain a value  $< 1$ . All nondominated solutions have fitness values  $< 1$ . Two solutions that have the same raw fitness values are differentiated by their objective fitness values. Thus, more importance is given to boundary solutions. For some objectives, a solution may have worse

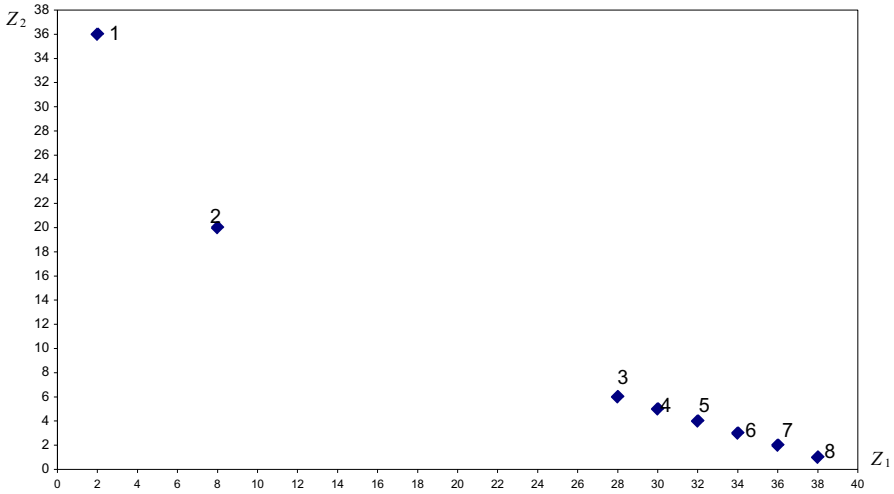
**Table 1** Objective function values of solutions

	$Z_1$	$Z_2$
1	2	18
2	6	14
3	10	10
4	12	6
5	16	4
6	25	2
7	4	22
8	20	14
9	23	12

**Table 2** Solutions sorted according to each objective

Solution	$Z_1$	Solution	$Z_2$
1	2	6	2
7	4	5	4
2	6	4	6
3	10	3	10
4	12	9	12
5	16	8	14
8	20	2	14
9	23	1	18
6	25	7	22





**Fig. 3** An illustrative example when all solutions are nondominated

objective values, but if it has a better objective value for at least one objective, then its objective fitness will be small and that solution can be kept. This method also decreases the run time of the algorithm.

SPEA-II has an archive of fixed size. If the nondominated solutions of the regular and archive populations are less than this fixed size, then dominated solutions are used to fill up the archive population. The dominated solutions are selected according to fitness values that are greater than 1. In this selection, if two solutions have the same raw fitness values, then the objective fitness values determine the selection. However, if the nondominated solutions are greater than the fixed archive size, then some members have to be discarded. In SPEA-II, this truncation is done according to density fitness values. In mSPEA-II, we calculate the objective fitness values instead of density fitness values to decrease the run time. However, using objective fitness values for truncation may not be suitable when all the solutions in the archive set are nondominated. An illustrative example problem is shown in Fig. 3.

In Fig. 3, all eight solutions are nondominated.  $\min O(7)$  and  $\min O(2)$  are 2. However, solution 2 is responsible for a big region to maintain convergence. Therefore, discarding solutions according to  $\min O(i)$  could avoid the convergence to the true Pareto front. If the algorithm is in a truncation operation, a niching strategy based on the crowding distances can be used to select solutions from less crowded regions. Crowding distance that is the average distance between two solutions on two sides of solution  $i$  along each objective is calculated for each solution as in NSGA-II. For all objectives, crowding distance values of boundary solutions are set to infinity. For intermediate solutions, the crowding distance values for each objective are calculated as follows (Deb et al. 2002):

$$d_i^m = \frac{(f_m^{I_{i+1}^m} - f_m^{I_{i-1}^m})}{(f_m^{\max} - f_m^{\min})}, \tag{9}$$

where  $I_i$  denotes the solution index of the  $i$ th member in the list that is obtained by sorting the values of objective  $m$  in ascending order, and  $f_m^{I_{i+1}^m}$  and  $f_m^{I_{i-1}^m}$  are the neighbors of the  $i$ th member according to the  $m$ th objective.  $f_m^{\max}$  and  $f_m^{\min}$  are the maximum and minimum values for the  $m$ th objective.

The crowding distance value of one member in a front is measured as follows:

$$C(i) = \sum_m d_{I_i^m}.$$

The new fitness value is calculated based on raw fitness and crowding fitness

$$F_2(i) = R(i) + \frac{1}{C(i) + 1}, \tag{10}$$

where  $C(i)$  is the crowding distance value of  $i$ .

To summarize, when filling up the mating pool, fitness values that are calculated based on raw and objective fitness are used for binary tournament if the archive size is less than the predefined fixed archive size. If the archive size is greater than the fixed archive size, then fitness values that are calculated based on raw and crowding fitness are used for binary tournament.

**The Algorithm**

- Step 1: Initialization: Generate an initial population  $P_0$  and create the empty archive set  $Q_0 = \{\}$ . Set  $t = 0$ .
- Step 2: Selection: Put all nondominated individuals in  $P_t$  and  $Q_t$  into  $Q_{t+1}$ .
- Step 3: If the size of  $Q_{t+1}$  exceeds a fixed archive size,  $N$ , then calculate  $F_2$  fitness value of each member and reduce  $Q_{t+1}$  using the truncation operator. Otherwise, calculate  $F_1$  fitness value of each solution and fill  $Q_{t+1}$  with dominated individuals in  $P_t$  and  $Q_t$ .
- Step 4: Termination: If  $t > T$  or another stopping criterion is satisfied, then set the nondominated individuals in  $Q_{t+1}$  as the resulting set. Stop.
- Step 5: Fill the mating pool by performing binary tournament selection on  $Q_{t+1}$ .
- Step 6: Variation: Perform crossover and mutation to the mating pool and set offspring population as the new population  $P_{t+1}$ . Set  $t = t + 1$  and go to Step 2.

**5 Computational experiments**

We first test the performance of mSPEA-II on six problem sets. Each problem set consists of nine problems and each problem consists of ten randomly generated problem instances. We run each problem instance five times with different seeds. The problem set properties are given in Table 3. To analyze the effect of the number of facility nodes and number of facilities to be sited on the performance of the algorithms, we define three values for both facility nodes and facilities to be sited in each problem set. For example, for problem set 1, the number of facility nodes is determined as 10, 25 and 50, and number of facilities to be sited is determined as three, five and seven. For each

**Table 3** Problem set characteristics

	Problem set 1	Problem set 2	Problem set 3	Problem set 4	Problem set 5	Problem set 6
No. of demand nodes	100	200	250	500	1000	1500
No. of facility nodes <sup>a</sup>	10, 25, 50	25, 50, 75	50, 75, 100	75, 100, 150	100, 150, 200	150, 200, 250
No. of facilities to be sited <sup>a</sup>	3, 5, 7	5, 7, 9	5, 7, 9	5, 7, 9	5, 7, 9	5, 7, 9
No. of iterations	100	200	500	1000	1500	2000
Full cov. dis.	10	10	10	10	20	20
Partial cov. dis.	20	20	20	20	40	40
Region length	200	200	200	200	400	400

<sup>a</sup> For each combination of these values, problem sets have problems defined as A, B, C, D, E, F, G, H and I. For example, problem 1A has ten facility nodes and three facilities to be sited

combination of these values, a problem is defined. For ten facility nodes and three, five and seven facilities to be sited, problems 1A, 1B and 1C are defined, respectively. Similarly in problems 1D, 1E and 1F, the number of facility nodes is 25 and numbers of facilities to be sited are three, five and seven, respectively. For 50 facility nodes, problems 1G, 1H and 1I have three, five and seven facilities to be sited, respectively.

The number of demand nodes changes among problem sets. The region length shows the side length of the plane where facility and demand nodes are generated. For instance,  $x - y$  coordinates are generated from a uniform distribution between 0 and 200 in the first four problem sets. The demand amounts are generated from a uniform distribution in interval  $[0, 500]$  for all problem sets. Full coverage and partial coverage distance are taken as 5 and 10% of the region size, respectively.

To analyze the effect of the number of iterations on the performance of the algorithms, we also change the number of iterations for problem set 4 with 100 facility nodes and five facilities to be sited. We define the number of iterations as 100, 200, 500, 1000 and 2000 while keeping other parameters the same.

We also analyze the effect of computation time on the performance of the algorithms. For problem set 6 with 200 facility nodes and five facilities to be sited, we define computation times as 30, 40, 50 and 60 s. We run the three algorithms and stop them after the same computation time.

## 5.1 Parameter settings

In the literature, different encodings have been used for the solutions (see [Mladenovic et al. 2007](#) for a review). In binary encoding, the number of potential facilities determines the size of the string and each binary gene represents whether the potential facility is to be sited or not ([Hosage and Goodchild 1986](#)). In this encoding, the number of opened facilities is penalized to avoid infeasibility. Computational results show that binary encoding does not result in good results even for small-size problems. In real parameter encoding, the number of facilities to be sited determines the size of the string and each gene represents the index of the opened facility ([Alcaraz et al. 2012](#)). We used

**Fig. 4** Representation of a solution

5	9	7
---	---	---

real parameter encoding, since it is simpler, efficient and never leads to infeasibility. Figure 4 shows an example string where string size is 3 and facilities 5, 9 and 7 are to be opened. The Euclidean distances are considered between the solutions. For SPEA-II, distances between members in the objective space are calculated for the density estimation. In NSGA-II and mSPEA-II, crowding distances are also calculated for the niching mechanism. Since the range of objectives is different, the objective function values are normalized using the estimated maximum possible values in the Pareto front.

We use uniform crossover with fixed pattern (Villegas et al. 2006; Michalewicz 1996, p.90). In the crossover mechanism, pattern {1, 2, 1, 2, . . .} is used. For example, pattern {1, 2, 1, 2} shows that the first and third genes of the first member, and the second and fourth genes of the second member are taken. However, after crossover, we may need to repair the mechanism. If we have same numbers in the string (solution), this means that a facility is opened twice which is not allowed, and one of these genes should be replaced randomly by another number. We choose archive size as 50 for each problem and MOGA.

As mutation mechanism, single point mutation is used (Mingjun et al. 2004; Iris and Asan 2012). A gene is randomly selected and it is mutated with a probability. After each mutation, the member is again checked for repair. After some preliminary experiments, we determine the mutation probability as 0.3 and the number of iterations as given in Table 3.

### 5.2 Performance metrics

To compare the performance of the algorithms, three performance metrics are used. These metrics are the hypervolume ratio (HVR), inverted generational distance (IGD) and percentage of found solutions. For the fifth and sixth problem sets, one additional metric, set coverage metric,  $C(A, B)$ , is also used. Since we do not know the true Pareto front in these sets, this measure is used to compare the algorithms' Pareto optimal sets with each other.

HVR is the extension of the hypervolume metric (Zitzler and Thiele 1998). Hypervolume metric calculates the total objective space dominated by a given set of nondominated solutions with respect to a predefined reference point. This metric evaluates both closeness and diversity of the results. The reference point can be determined as the nadir point or worse.

$$HV = \text{volume} \left( \bigcup_{i \in A} v_i \right), \tag{11}$$

where  $A$  is the nondominated front and  $v_i$  is the objective space dominated by solution  $i$  with respect to a reference point. HVR measures the ratio of the region enclosed by the nondominated set and the region enclosed by the true Pareto front.

The inverted generational distance (Bosman and Thiernes 2003) is the average Euclidean distance between nondominated solutions and their closest true nondominated front member. Hence, an algorithm with a small IGD is better. IGD is calculated as follows:

$$\text{IGD} = \frac{1}{|A|} \sum_{i \in A} \left( \min_{j \in \text{PF}} \|Z^i - Z^j\|_2 \right), \quad (12)$$

where  $A$  denotes the nondominated set generated by MOGA, PF denotes the true Pareto front set and  $\|Z^i - Z^j\|_2$  represents the Euclidean distance between  $Z^i$  and  $Z^j$ .

The third metric is the percentage of solutions generated by a MOGA in the true Pareto front. Since for the last three problem sets we do not know the true Pareto front, the estimated Pareto front is generated by combining the nondominated solutions of all algorithms. Thus, for the last three problem sets, this metric gives us the contribution of that algorithm when generating the estimated Pareto front.

Set coverage metric  $C(A, B)$  (Zitzler and Thiele 1998) calculates the fraction of  $B$  dominated by  $A$ .

$$C(A, B) = \frac{|\{b \in B\} | \exists a \in A : a \succ b|}{|B|}, \quad (13)$$

where  $A$  denotes the Pareto front generated by algorithm  $A$  and  $B$  denotes the Pareto front generated by algorithm  $B$ .  $C(A, B) = 1$  means that all members in  $B$  are dominated by  $A$ .  $C(A, B) = 0$  represents the situation in which no individual in  $B$  is dominated by  $A$ . The domination operator is not a symmetric operator. Thus,  $C(A, B)$  is not necessarily equal to  $1 - C(A, B)$ .

### 5.3 Computational results

Problem sets 1, 2 and 3 are solved by the  $\varepsilon$ -constraint method (Haimes et al. 1971) with GAMS 2.25 Cplex solver (Brooke et al. 1992) on a computer Intel Pentium IV with 1.8 GHz and 512 MB RAM. The true Pareto optimal front is found for these problem sets. However, problem sets 4, 5 and 6 cannot be solved by the  $\varepsilon$ -constraint method within 24 h. For these problems, all run results for the three MOGAs are combined and an estimated Pareto front is generated by combining the nondominated results of the three MOGAs.

For each problem instance, NSGA-II, SPEA-II and mSPEA-II are run five times using different seeds. For each run, the HVR, IGD and percentage of found solutions metrics are calculated. The average metric values and standard deviations of problems are given in Table 4. The values in the parenthesis in the second column of Table 4 correspond to the number of facility nodes and number of facilities to be sited.

A point a little worse than the nadir point is selected as the reference point for HVR measure. The range of HVR values is approximately between 0.90 and 1. For all problem sets, the average HVR values of mSPEA-II are greater than those of the other algorithms. Also, the standard deviations of the HVR values of mSPEA-II are very small and less than or equal to those of the other algorithms in 47 out of 54 problems. These results show that mSPEA-II approximates the true Pareto front better than the

**Table 4** Computational results

	NSGA-II						SPEA-II						mSPEA-II					
	HVR		IGD		% Of found sol.		HVR		IGD		% Of found sol.		HVR		IGD		% Of found sol.	
	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev
P. set 1																		
1A (10-3)	0.992	0.008	0.001	0.001	0.944	0.083	0.990	0.009	0.001	0.002	0.933	0.098	0.994	0.008	0.000	0.002	0.955	0.070
1B (10-5)	0.988	0.012	0.001	0.002	0.924	0.082	0.988	0.014	0.001	0.002	0.925	0.101	0.991	0.010	0.000	0.001	0.941	0.088
1C (10-7)	0.982	0.017	0.001	0.002	0.901	0.104	0.984	0.015	0.002	0.002	0.911	0.112	0.988	0.009	0.000	0.001	0.922	0.120
1D (25-3)	0.990	0.017	0.001	0.002	0.883	0.112	0.990	0.016	0.001	0.004	0.859	0.178	0.994	0.009	0.000	0.001	0.886	0.094
1E (25-5)	0.969	0.037	0.003	0.004	0.618	0.297	0.977	0.027	0.003	0.004	0.647	0.227	0.991	0.014	0.001	0.001	0.824	0.146
1F (25-7)	0.965	0.032	0.005	0.004	0.521	0.125	0.967	0.032	0.005	0.004	0.533	0.188	0.988	0.017	0.002	0.002	0.732	0.126
1G (50-3)	0.960	0.023	0.003	0.004	0.442	0.224	0.963	0.032	0.003	0.004	0.476	0.200	0.978	0.033	0.001	0.003	0.624	0.122
1H (50-5)	0.941	0.028	0.004	0.003	0.402	0.122	0.949	0.024	0.004	0.002	0.412	0.109	0.960	0.012	0.002	0.001	0.511	0.093
1I (50-7)	0.936	0.031	0.005	0.004	0.329	0.110	0.938	0.032	0.005	0.005	0.344	0.121	0.955	0.022	0.003	0.002	0.488	0.013
P. set 2																		
2A (25-5)	0.988	0.012	0.002	0.003	0.667	0.375	0.987	0.015	0.003	0.003	0.657	0.388	0.989	0.013	0.002	0.002	0.768	0.136
2B (25-7)	0.980	0.018	0.003	0.002	0.482	0.201	0.982	0.017	0.003	0.003	0.513	0.201	0.983	0.016	0.002	0.002	0.634	0.173
2C (25-9)	0.973	0.012	0.004	0.003	0.442	0.142	0.979	0.021	0.004	0.003	0.417	0.188	0.980	0.001	0.003	0.002	0.588	0.188
2D (50-5)	0.975	0.024	0.004	0.004	0.440	0.331	0.975	0.020	0.003	0.004	0.376	0.255	0.988	0.012	0.001	0.001	0.700	0.167
2E (50-7)	0.966	0.026	0.005	0.005	0.228	0.189	0.971	0.022	0.004	0.004	0.226	0.160	0.982	0.014	0.001	0.002	0.440	0.172
2F (50-9)	0.944	0.029	0.009	0.008	0.201	0.143	0.949	0.028	0.007	0.007	0.209	0.174	0.960	0.023	0.006	0.006	0.401	0.281
2G (75-5)	0.952	0.022	0.005	0.003	0.332	0.179	0.963	0.019	0.004	0.001	0.382	0.223	0.970	0.017	0.003	0.003	0.423	0.176
2H (75-7)	0.933	0.038	0.007	0.003	0.219	0.167	0.941	0.023	0.006	0.002	0.212	0.188	0.966	0.012	0.004	0.003	0.420	0.164
2I (75-9)	0.933	0.025	0.008	0.003	0.189	0.145	0.940	0.033	0.007	0.003	0.208	0.188	0.955	0.014	0.006	0.002	0.386	0.173

**Table 4** continued

	NSGA-II						SPEA-II						mSPEA-II					
	HVR		IGD		% Of found sol.		HVR		IGD		% Of found sol.		HVR		IGD		% Of found sol.	
	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev
P. set 3																		
3A (50-5)	0.978	0.014	0.003	0.002	0.523	0.182	0.982	0.018	0.001	0.002	0.588	0.158	0.989	0.011	0.001	0.001	0.668	0.123
3B (50-7)	0.965	0.016	0.005	0.002	0.401	0.199	0.974	0.017	0.002	0.002	0.445	0.194	0.981	0.016	0.000	0.002	0.589	0.156
3C (50-9)	0.951	0.022	0.006	0.003	0.305	0.225	0.968	0.021	0.004	0.004	0.324	0.255	0.978	0.017	0.002	0.002	0.558	0.154
3D (75-5)	0.961	0.031	0.004	0.005	0.433	0.277	0.973	0.020	0.002	0.002	0.540	0.245	0.983	0.016	0.001	0.002	0.594	0.195
3E (75-7)	0.951	0.030	0.006	0.007	0.242	0.219	0.967	0.017	0.003	0.004	0.321	0.163	0.978	0.015	0.001	0.002	0.548	0.145
3F (75-9)	0.942	0.035	0.009	0.018	0.220	0.204	0.956	0.023	0.006	0.007	0.288	0.188	0.973	0.021	0.004	0.002	0.522	0.156
3G (100-5)	0.958	0.024	0.006	0.005	0.388	0.187	0.966	0.019	0.004	0.004	0.411	0.122	0.978	0.011	0.004	0.003	0.542	0.123
3H (100-7)	0.944	0.028	0.009	0.006	0.225	0.210	0.955	0.021	0.006	0.005	0.301	0.144	0.975	0.015	0.004	0.003	0.500	0.188
3I (100-9)	0.934	0.033	0.014	0.005	0.188	0.234	0.944	0.024	0.010	0.006	0.221	0.157	0.974	0.016	0.004	0.004	0.488	0.196
P. set 4																		
4A (75-5)	0.965	0.016	0.002	0.003	0.223	0.123	0.986	0.011	0.002	0.005	0.423	0.164	0.989	0.005	0.000	0.000	0.681	0.112
4B (75-7)	0.962	0.015	0.004	0.001	0.211	0.223	0.982	0.015	0.002	0.005	0.389	0.144	0.987	0.011	0.001	0.001	0.643	0.103
4C (75-9)	0.958	0.017	0.005	0.002	0.156	0.144	0.980	0.020	0.003	0.004	0.312	0.135	0.982	0.010	0.001	0.002	0.622	0.142
4D (100-5)	0.960	0.023	0.003	0.002	0.208	0.209	0.976	0.015	0.003	0.007	0.387	0.148	0.984	0.015	0.001	0.001	0.631	0.176
4E (100-7)	0.960	0.022	0.005	0.004	0.122	0.151	0.979	0.013	0.003	0.003	0.316	0.180	0.985	0.009	0.001	0.001	0.616	0.137
4F (100-9)	0.943	0.026	0.006	0.003	0.118	0.123	0.966	0.015	0.006	0.003	0.224	0.189	0.980	0.010	0.002	0.004	0.605	0.127
4G (150-5)	0.923	0.023	0.019	0.004	0.204	0.121	0.934	0.015	0.007	0.002	0.351	0.115	0.945	0.013	0.002	0.002	0.568	0.099
4H (150-7)	0.914	0.026	0.020	0.004	0.108	0.119	0.931	0.017	0.006	0.001	0.345	0.127	0.942	0.014	0.003	0.001	0.553	0.111
4I (150-9)	0.911	0.030	0.021	0.006	0.098	0.135	0.923	0.018	0.008	0.001	0.336	0.134	0.940	0.016	0.007	0.001	0.554	0.109

Table 4 continued

	NSGA-II						SPEA-II						mSPEA-II					
	HVR		IGD		% Of found sol.		HVR		IGD		% Of found sol.		HVR		IGD		% Of found sol.	
	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev
P. set 5																		
5A (100-5)	0.959	0.021	0.002	0.001	0.272	0.124	0.969	0.027	0.001	0.001	0.421	0.199	0.982	0.024	0.000	0.001	0.712	0.123
5B (100-7)	0.952	0.016	0.003	0.001	0.241	0.167	0.962	0.018	0.001	0.002	0.442	0.213	0.975	0.015	0.001	0.001	0.665	0.102
5C (100-9)	0.943	0.019	0.004	0.002	0.233	0.142	0.961	0.013	0.002	0.002	0.388	0.228	0.972	0.012	0.001	0.001	0.643	0.112
5D (150-5)	0.947	0.033	0.003	0.002	0.238	0.186	0.965	0.033	0.001	0.002	0.415	0.212	0.974	0.033	0.001	0.002	0.682	0.143
5E (150-7)	0.947	0.022	0.005	0.004	0.236	0.091	0.950	0.023	0.002	0.002	0.346	0.184	0.970	0.018	0.001	0.001	0.547	0.157
5F (150-9)	0.931	0.012	0.003	0.002	0.226	0.060	0.952	0.021	0.002	0.001	0.311	0.086	0.962	0.021	0.001	0.001	0.553	0.100
5G (200-5)	0.915	0.023	0.010	0.002	0.234	0.089	0.919	0.017	0.010	0.001	0.323	0.076	0.956	0.019	0.002	0.003	0.592	0.123
5H (200-7)	0.905	0.012	0.012	0.003	0.212	0.077	0.912	0.014	0.011	0.002	0.314	0.175	0.949	0.012	0.002	0.002	0.533	0.134
5I (200-9)	0.903	0.015	0.013	0.003	0.201	0.065	0.911	0.020	0.012	0.002	0.301	0.124	0.946	0.012	0.002	0.003	0.511	0.135
P. set 6																		
6A (150-5)	0.966	0.019	0.003	0.001	0.224	0.113	0.979	0.018	0.002	0.002	0.532	0.109	0.988	0.011	0.001	0.000	0.698	0.093
6B (150-7)	0.950	0.017	0.004	0.002	0.212	0.142	0.977	0.017	0.003	0.002	0.421	0.133	0.980	0.015	0.001	0.000	0.674	0.083
6C (150-9)	0.944	0.015	0.006	0.002	0.178	0.140	0.963	0.019	0.004	0.003	0.396	0.112	0.971	0.017	0.002	0.001	0.682	0.113
6D (200-5)	0.963	0.019	0.003	0.004	0.192	0.143	0.983	0.013	0.002	0.004	0.527	0.201	0.988	0.012	0.001	0.001	0.725	0.170
6E (200-7)	0.943	0.023	0.007	0.005	0.167	0.094	0.969	0.013	0.002	0.002	0.363	0.143	0.985	0.011	0.001	0.001	0.568	0.168
6F (200-9)	0.932	0.014	0.008	0.004	0.161	0.102	0.941	0.022	0.005	0.003	0.388	0.122	0.953	0.017	0.002	0.003	0.557	0.112
6G (250-5)	0.913	0.024	0.013	0.010	0.112	0.082	0.916	0.022	0.012	0.008	0.332	0.124	0.948	0.016	0.006	0.002	0.622	0.082
6H (250-7)	0.901	0.028	0.014	0.009	0.104	0.103	0.909	0.026	0.013	0.007	0.342	0.092	0.941	0.014	0.006	0.004	0.602	0.121
6I (250-9)	0.883	0.029	0.019	0.012	0.101	0.087	0.885	0.029	0.018	0.006	0.322	0.123	0.939	0.015	0.009	0.006	0.563	0.095



others and the HVR values of mSPEA-II are more consistent than those of the other algorithms.

The IGD results are normalized values. Objective function values are scaled since their ranges are different. The IGD average values of mSPEA-II are better than those of the others. This shows that the generated nondominated solutions of mSPEA-II are closer to the true Pareto front than those of the other algorithms.

If the algorithms are compared based on the percentage of found solutions, mSPEA-II generates more nondominated solutions than the other algorithms. For small-size problems, such as in problem set 1 with ten facility nodes, the values are close to each other for all algorithms. However, for the other problems, mSPEA-II outperforms the other algorithms significantly. In problem sets 1–3, mSPEA-II finds 61.67 %, SPEA-II finds 46.95 % and NSGA-II finds 44.77 % of true Pareto front solutions on average. For problem sets 4–6, this metric gives the contribution of the algorithm to the estimated Pareto front, since the true Pareto fronts are unknown for these problem sets. In these problem sets, mSPEA-II finds 61.48 %, SPEA-II finds 36.91 % and NSGA-II finds 18.48 % of estimated Pareto front solutions on average.

We evaluate the effect of the parameters, namely the number of facility nodes and number of facilities to be sited on the performance of MOGAs in Table 4.

In each problem set for a given number of facility nodes, as the number of facilities to be sited increases, the number of efficient solutions increases and as a result the performance of all three MOGAs decreases. However, decrease in the performance of mSPEA-II is less compared to other MOGAs and mSPEA-II outperforms the other algorithms. For instance, in problems 3G and 3I, the number of facility nodes is 100 and the number of facilities to be sited are five and nine. Increasing the number of facilities to be sited from 5 (problem 3G) to 9 (problem 3I) decreases HVR values from 0.958 to 0.934 in NSGA-II, 0.966 to 0.944 in SPEA-II and 0.978 to 0.974 in mSPEA-II. The IGD values increase from 0.006 to 0.014 in NSGA-II, 0.004 to 0.010 in SPEA-II and remain the same (0.004) in mSPEA-II. In terms of percentage of found solutions, mSPEA-II results are significantly better than NSGA-II and SPEA-II when we increase the number of facilities to be sited. For the above problems, the percentage of found solutions decreases from 38.8 to 18.8 % in NSGA-II, 41.1 to 22.1 % in SPEA-II and 54.2 to 48.8 % in mSPEA-II.

When we analyze the effect of the number of facilities to be sited in large-size problems, we again observe that mSPEA-II performs better than NSGA-II and SPEA-II with respect to all performance measures. For example, in problem set 6 for problems with 150 facility nodes, if the number of facilities to be sited is changed from five (problem set 6A) to nine (problem set 6C), the percentage of found solutions decreases from 22.4 to 17.8 % in NSGA-II, 53.2 to 39.6 % in SPEA-II and 69.8 to 68.2 % in mSPEA-II.

We also analyze the effect of the number of facility nodes on the performance of MOGAs. For any problem set with a given number of facilities to be sited, increase in the number of facility nodes decreases the performance of all MOGAs. For instance, for problem set 1 with three facilities to be sited, increasing the number of facility nodes from 10 to 50 (problems 1A and 1G) decreases the percentage of found solutions from 94.4 to 44.2 % in NSGA-II, 93.3 to 47.6 % in SPEA-II and 95.5 to 62.4 % in mSPEA-II. The amount of decrease is less for mSPEA-II compared to those of other MOGAs.

The performance of mSPEA-II is also better in large-size problems. For instance, in problem set 6, with five facilities to be sited, increasing the number of facility nodes from 150 to 250 (problems 6A, 6G) decreases the percentage of found solutions from 22.4 to 11.2 % in NSGA-II, 53.2 to 33.2 % in SPEA-II and 69.8 to 62.2 % in mSPEA-II.

Problem 4D is solved with a different number of iterations. The estimated Pareto fronts of the problems are obtained by combining the results of all solutions generated by solving different MOGAs with different number of iterations, since the true Pareto front of problem set 4 is unknown. The results of the performance metrics are presented in Table 5.

In all problems, mSPEA-II outperforms NSGA-II and SPEA-II. With 100 iterations, NSGA-II outperforms SPEA-II. In the rest of the problem sets, SPEA-II outperforms NSGA-II with respect to HVR and the percentage of found solutions. The results show that when the number of iterations is increased from 100 to 2000, the performances of MOGAs improve on all metrics as expected. However, the amount of improvement decreases as the number of iterations increases.

To evaluate the performance of mSPEA-II with constant computation time, we run the three MOGAs for problem 6D at a predetermined time. The computation times are set as 30, 40, 50 and 60 s. Table 6 shows the results of each MOGA with different computation times. Since the true Pareto front is unknown for problem set 6, we combine all generated nondominated solutions and estimate the true Pareto front. The performance metric values are computed with respect to the estimated Pareto fronts. For each computation time, mSPEA-II outperforms both NSGA-II and SPEA-II with respect to all performance metrics. For instance, when computation time is set as 30 s, mSPEA-II finds 25.08 % of the estimated Pareto front; however, SPEA-II and NSGA-II generate 10.80 and 12.05 % of it, respectively. The results show that for 30 and 40 s, NSGA-II performs better than SPEA-II; for the rest, SPEA-II performs better than NSGA-II. As the computation time increases, the performance of MOGAs increases as expected.

An illustration of nondominated solutions found by the MOGAs and the generated true Pareto front in problem 3D is given in Fig. 5. The nondominated fronts are the first run results of MOGAs. Some solutions of the true Pareto front cannot be generated by any of the MOGAs.

The set coverage values for problem sets 5 and 6 are presented in Table 7. In the  $C(A, B)$  metric,  $A$  corresponds to a row of the table and  $B$  corresponds to a column of the table. This metric compares two algorithms.

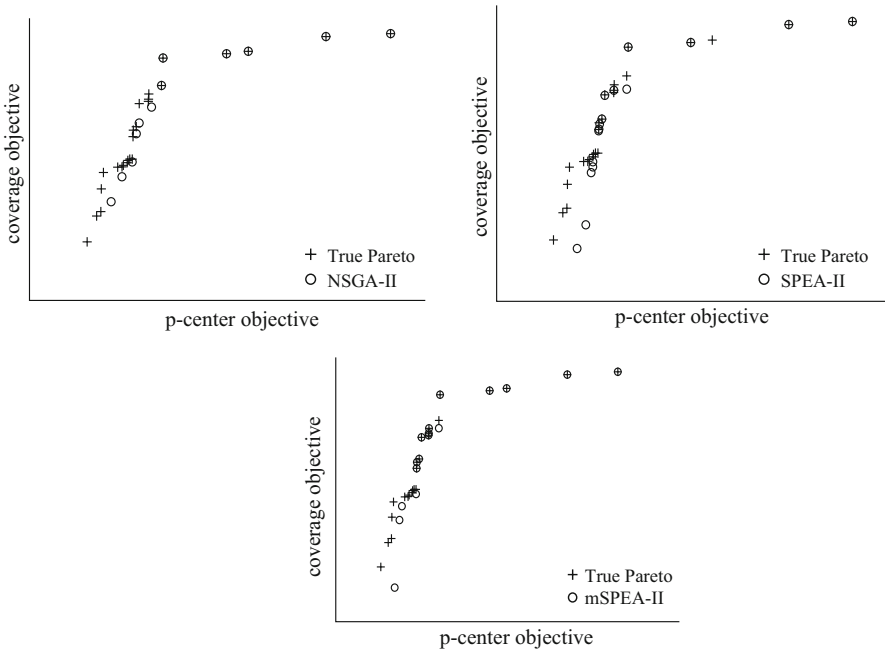
In problems sets 5 and 6, solutions found by NSGA-II dominates at most 5 % of solutions generated by SPEA-II and mSPEA-II. This shows that most of the solutions of NSGA-II have worse objective function values than those of SPEA-II and mSPEA-II. On the other hand, SPEA-II solutions dominate NSGA-II solutions between 15 % at the minimum and 64 % at the maximum in problem set 5. The same measure changes between 23 and 75 % in problem set 6. mSPEA-II's performance is better than that of SPEA-II, and the set coverage metric values for mSPEA-II against NSGA-II changes between 31 and 70 % in problem set 5 and between 28 and 85 % in problem set 6. When we compare SPEA-II and mSPEA-II, SPEA-II solutions dominate mSPEA-II solutions, 1 % at the minimum and 14 % at the maximum for problem sets 5 and 6. However, the set coverage metric values for mSPEA-II against SPEA-II changes

**Table 5** Results of problem 4D with different numbers of iterations

No. of iterations	NSGA-II			SPEA-II			mSPEA-II		
	HVR	IGD	% Of found solution	HVR	IGD	% Of found solution	HVR	IGD	% Of found solution
100	0.9218	0.0088	0.3040	0.9125	0.0095	0.2632	0.9303	0.0051	0.4823
200	0.9325	0.0060	0.2803	0.9330	0.0054	0.2815	0.9463	0.0032	0.5421
500	0.9417	0.0038	0.2825	0.9568	0.0035	0.3096	0.9688	0.0019	0.5546
1000	0.9596	0.0027	0.3084	0.9762	0.0032	0.3874	0.9842	0.0007	0.6312
1500	0.9612	0.0018	0.3552	0.9821	0.0020	0.4350	0.9903	0.0005	0.6508
2000	0.9615	0.0017	0.3560	0.9832	0.0019	0.4403	0.9910	0.0005	0.6572

**Table 6** Results of problem 6D with different computation times

	Computation time			
	30 s	40 s	50 s	60 s
<b>HVR</b>				
NSGA-II	0.9202	0.9356	0.9443	0.9524
SPEA-II	0.9188	0.9305	0.9462	0.9585
mSPEA-II	0.9388	0.9422	0.9612	0.9658
<b>IGD</b>				
NSGA-II	0.012	0.0086	0.0069	0.0058
SPEA-II	0.018	0.0098	0.0061	0.0043
mSPEA-II	0.009	0.0074	0.0039	0.0016
<b>% Of found solution</b>				
NSGA-II	0.1205	0.1586	0.1636	0.1752
SPEA-II	0.1080	0.1550	0.1885	0.2035
mSPEA-II	0.2508	0.3025	0.4068	0.5230



**Fig. 5** Nondominated fronts generated by NSGA-II, SPEA-II and mSPEA-II in problem set 3D

between 16 and 34% in problem set 5 and between 10 and 35% in problem set 6. Computational experiments show that mSPEA-II outperforms the other algorithms and the second best-performing algorithm is SPEA-II.

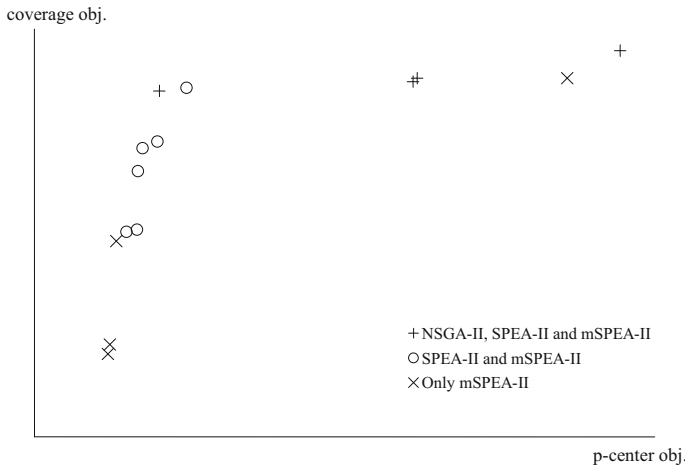
Also in any problem set with a given number of facility nodes, as the number of facilities to be sited increases, the average set coverage metric values of mSPAE-II against SPEA-II improve in all problems, and against NSGA-II improve in all

**Table 7** Average set coverage metric values for problem sets 5 and 6

Problem	SPEA-II	mSPEA-II	Problem	SPEA-II	mSPEA-II
<b>NSGA-II</b>					
5A	0.05	0.04	6A	0.02	0.01
5B	0.04	0	6B	0	0
5C	0.02	0.03	6C	0.03	0.03
5D	0.04	0.02	6D	0.01	0.01
5E	0.04	0	6E	0.01	0.02
5F	0.04	0.01	6F	0.05	0.03
5G	0.02	0.02	6G	0.04	0
5H	0	0	6H	0.02	0.01
5I	0	0	6I	0	0
Problem	NSGA-II	mSPEA-II	Problem	NSGA-II	mSPEA-II
<b>SPEA-II</b>					
5A	0.36	0.01	6A	0.42	0.14
5B	0.39	0.04	6B	0.36	0.12
5C	0.45	0.02	6C	0.40	0.13
5D	0.15	0.03	6D	0.23	0.03
5E	0.64	0.10	6E	0.75	0.11
5F	0.62	0.14	6F	0.52	0.11
5G	0.36	0.09	6G	0.48	0.08
5H	0.40	0.12	6H	0.63	0.04
5I	0.42	0.12	6I	0.70	0.05
Problem	NSGA-II	SPEA-II	Problem	NSGA-II	SPEA-II
<b>mSPEA-II</b>					
5A	0.42	0.22	6A	0.36	0.14
5B	0.58	0.28	6B	0.32	0.19
5C	0.54	0.34	6C	0.35	0.28
5D	0.31	0.16	6D	0.28	0.10
5E	0.70	0.26	6E	0.82	0.30
5F	0.66	0.31	6F	0.64	0.34
5G	0.54	0.20	6G	0.72	0.22
5H	0.58	0.26	6H	0.78	0.33
5I	0.64	0.28	6I	0.85	0.35

problems except problem sets 6B and 6C. This shows that as the string size increases, the performance of the mSPEA-II improves.

Figure 6 shows the estimated true Pareto front of a problem in problem 4E. There are 14 nondominated solutions in the estimated true Pareto front. This front is created by combining all run results of the three MOGAs. mSPEA-II generates all of them, while SPEA-II generates ten of them and NSGA-II only four.



**Fig. 6** Estimated Pareto front of a sample problem in problem set 4E

**Table 8** Wilcoxon test results of the HVR metric between NSGA-II and mSPEA-II

Problem	<i>p</i> value	Winner	Problem	<i>p</i> value	Winner	Problem	<i>p</i> value	Winner
1A	0.335	None	3A	0.003	mSPEA-II	5A	0	mSPEA-II
1B	0.329	None	3B	0.002	mSPEA-II	5B	0	mSPEA-II
1C	0.275	None	3C	0	mSPEA-II	5C	0	mSPEA-II
1D	0.236	None	3D	0	mSPEA-II	5D	0	mSPEA-II
1E	0	mSPEA-II	3E	0	mSPEA-II	5E	0	mSPEA-II
1F	0.001	mSPEA-II	3F	0	mSPEA-II	5F	0	mSPEA-II
1G	0	mSPEA-II	3G	0	mSPEA-II	5G	0	mSPEA-II
1H	0	mSPEA-II	3H	0	mSPEA-II	5H	0	mSPEA-II
1I	0	mSPEA-II	3I	0	mSPEA-II	5I	0	mSPEA-II
2A	0.318	None	4A	0	mSPEA-II	6A	0	mSPEA-II
2B	0.252	None	4B	0	mSPEA-II	6B	0	mSPEA-II
2C	0.173	None	4C	0	mSPEA-II	6C	0	mSPEA-II
2D	0	mSPEA-II	4D	0	mSPEA-II	6D	0	mSPEA-II
2E	0.001	mSPEA-II	4E	0	mSPEA-II	6E	0	mSPEA-II
2F	0.001	mSPEA-II	4F	0	mSPEA-II	6F	0	mSPEA-II
2G	0	mSPEA-II	4G	0	mSPEA-II	6G	0	mSPEA-II
2H	0	mSPEA-II	4H	0	mSPEA-II	6H	0	mSPEA-II
2I	0	mSPEA-II	4I	0	mSPEA-II	6I	0	mSPEA-II

To check whether a significant difference exists between mSPEA-II and the other two algorithms, we test the following hypothesis at 99% significance level. In a statistical test, only the HVR is used. The Wilcoxon non-parametric statistical test is performed, and *p* values are calculated to determine the better-performing algorithm as given in Tables 8 and 9.

**Table 9** Wilcoxon test results of the HVR metric between SPEA-II and mSPEA-II

Problem	<i>p</i> value	Winner	Problem	<i>p</i> value	Winner	Problem	<i>p</i> value	Winner
1A	0.428	None	3A	0.004	mSPEA-II	5A	0	mSPEA-II
1B	0.442	None	3B	0.004	mSPEA-II	5B	0	mSPEA-II
1C	0.364	None	3C	0.002	mSPEA-II	5C	0	mSPEA-II
1D	0.437	None	3D	0	mSPEA-II	5D	0	mSPEA-II
1E	0	mSPEA-II	3E	0	mSPEA-II	5E	0	mSPEA-II
1F	0.002	mSPEA-II	3F	0	mSPEA-II	5F	0	mSPEA-II
1G	0	mSPEA-II	3G	0.001	mSPEA-II	5G	0	mSPEA-II
1H	0	mSPEA-II	3H	0	mSPEA-II	5H	0	mSPEA-II
1I	0	mSPEA-II	3I	0	mSPEA-II	5I	0	mSPEA-II
2A	0.302	None	4A	0.009	mSPEA-II	6A	0.002	mSPEA-II
2B	0.451	None	4B	0.004	mSPEA-II	6B	0.005	mSPEA-II
2C	0.373	None	4C	0.011	None	6C	0.003	mSPEA-II
2D	0	mSPEA-II	4D	0.002	mSPEA-II	6D	0.004	mSPEA-II
2E	0.002	mSPEA-II	4E	0.003	mSPEA-II	6E	0.002	mSPEA-II
2F	0.003	mSPEA-II	4F	0	mSPEA-II	6F	0	mSPEA-II
2G	0.004	mSPEA-II	4G	0	mSPEA-II	6G	0	mSPEA-II
2H	0	mSPEA-II	4H	0	mSPEA-II	6H	0	mSPEA-II
2I	0	mSPEA-II	4I	0	mSPEA-II	6I	0	mSPEA-II

$$\begin{aligned}
 H_0 &: \mu_{\text{hvr}}^{\text{mSPEA-II}} = \mu_{\text{hvr}}^O \\
 H_1 &: \mu_{\text{hvr}}^{\text{mSPEA-II}} \neq \mu_{\text{hvr}}^O,
 \end{aligned}$$

where  $\mu_{\text{hvr}}^{\text{mSPEA-II}}$  is the mean of mSPEA-II and  $\mu_{\text{hvr}}^O$  is the mean of NSGA-II or SPEA-II.

There is not a statistically significant difference in problem sets 1A, 1B, 1C, 1D, 2A, 2B, 2C between NSGA-II and mSPEA-II at 99 % significance level. Actually, the size of these problems is very small and the results of all the algorithms are close to each other. Except for these seven problems, mSPEA-II performs significantly better than NSGA-II. Similarly, mSPEA-II outperforms SPEA-II significantly in all problems except these seven problems and problem 4C. However, for problem 4C, there is significant difference between SPEA-II and mSPEA-II at 95 % significance level.

Table 10 reports the average number of solutions in the true Pareto front for problem sets 1, 2 and 3 and run times of the algorithms for all problem sets. For each problem, mSPEA-II and NSGA-II run times are fairly close to each other. SPEA-II run times are greater than those of the other two algorithms for all problems.

We choose archive size as 50 for each problem and for each MOGA. In all problem sets, the generated nondominated solutions are less than 50 for each MOGA. Thus in SPEA-II, the truncation operation according to density fitness value is not used in these problems. Thus, the computational complexity of SPEA-II is not significantly higher

**Table 10** Average run times of algorithms (s)

Problem	True Pareto front	NSGA-II	SPEA-II	mSPEA-II	Problem	NSGA-II	SPEA-II	mSPEA-II
1A	12	1	1	1	4A	10	20	9
1B	14	1	1	1	4B	12	21	12
1C	13	1	1	1	4C	14	24	13
1D	54	1	1	1	4D	18	26	18
1E	49	1	1	1	4E	24	32	24
1F	66	1	2	1	4F	24	34	22
1G	225	2	2	2	4G	38	51	36
1H	254	2	2	2	4H	39	54	34
1I	301	2	3	2	4I	40	61	39
2A	109	2	2	2	5A	28	46	24
2B	156	2	2	2	5B	32	52	32
2C	177	2	3	2	5C	32	47	30
2D	494	2	3	2	5D	44	53	42
2E	670	2	3	2	5E	53	64	51
2F	822	3	3	3	5F	61	72	57
2G	1258	4	6	4	5G	73	88	70
2H	1547	4	6	4	5H	72	87	72
2I	1951	5	6	4	5I	76	91	75
3A	1422	5	7	5	6A	71	98	68
3B	1823	5	6	5	6B	75	107	72
3C	1786	5	6	5	6C	76	103	72
3D	2827	6	10	6	6D	83	100	77
3E	2037	8	13	8	6E	108	124	107
3F	2812	7	12	7	6F	114	141	108
3G	4832	13	18	12	6G	128	167	122
3H	5765	14	22	14	6H	136	171	126
3I	5841	15	23	15	6I	130	166	128

than that of NSGA-II and mSPEA-II as expected. Only the distance calculation of each solution increases the SPEA-II run times as compared to NSGA-II and mSPEA-II.

## 6 Conclusion

In this study, we formulate a bi-objective model that maximizes total coverage and minimizes the maximum distance between uncovered demands and their nearest facilities. The model is especially useful for emergency service location problems where all demand nodes are not covered by facilities.

We developed a MOGA, called mSPEA-II, to solve our model. In this algorithm, we modify the fitness assignment of SPEA-II and use the crowding distance of NSGA-II to eliminate the drawbacks of NSGA-II and SPEA-II in reaching the true Pareto



front. We tested the performance of mSPEA-II on randomly generated problems and compared the results with those of NSGA-II and SPEA-II. Our experimental results show that the mSPEA-II algorithm performs better than the others. For small problems, the results of these MOGAs are close to each other, but on average mSPEA-II performs better. As the number of facilities to be sited or the number of facility nodes increases, the computational complexity of the problems increases. This affects the performance of all MOGAs. However, decrease in the performance of mSPEA-II is less compared to those of NSGA-II and SPEA-II. Also, the run time performance of mSPEA-II is better than that of SPEA-II, and approximately the same as that of NSGA-II.

An area for future research direction may be to develop a preference-based MOGA to find the best solution for the DM. We may also apply mSPEA-II to other combinatorial optimization problems. Another research direction may be to extend the bi-objective model to handle capacity restrictions on the facilities.

## References

- Alberto I, Mateo PM (2011) A crossover operator that uses Pareto optimality in its definition. *TOP* 19(1):67–92
- Alcaraz J, Landete M, Monge JF (2012) Design and analysis of hybrid metaheuristics for the reliability p-median problem. *Eur J Oper Res* 222(1):54–64
- Araz C, Selim H, Ozkarahan I (2007) A fuzzy multi-objective covering-based vehicle location model for emergency services. *Comput Oper Res* 34:705–726
- Aytug H, Khouja M, Vergara FE (2003) Use of genetic algorithms to solve production and operations management problems: a review. *Int J Prod Res* 41(17):3955–4009
- Badri MA, Mortagy AK, Alsayed CA (1998) A multi-objective model for locating fire stations. *Eur J Oper Res* 110:243–260
- Berman O, Krass D, Drezner Z (2003) The gradual covering decay location problem on a network. *Eur J Oper Res* 151:474–480
- Berman O, Drezner Z, Krass D (2010) Generalized coverage: new developments in covering location models. *Comput Oper Res* 37:1675–1687
- Bhattacharya R, Bandyopadhyay S (2010) Solving conflicting bi-objective facility location problem by NSGA-II evolutionary algorithm. *Int J Adv Manuf Technol* 51:397–414
- Bosman PAN, Thiernes D (2003) The balance between proximity and diversity in multi-objective evolutionary algorithms. *IEEE Trans Evolut Comput* 7(2):174–188
- Brooke A, Kendrick D, Meeraus A (1992) GAMS: a user's guide, release 2.25. The Scientific Press, South San Francisco
- Church R, ReVelle C (1974) The maximal covering location problem. *Pap Reg Scie Assoc* 32:101–118
- Church R, Current J, Storbeck J (1991) A bicriterion maximal covering location formulation which considers the satisfaction of uncovered demand. *Dec Sci* 22(1):38–52
- Coello Coello C, Lamont GB, van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems. Springer, Berlin
- Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, New York
- Deb K, Pratap A, Agarwal S (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans Evolut Comput* 6(2):182–197
- Durillo JJ, Nebro AJ, Luna F, Alba E (2009) On the effect of the steady-state selection scheme in multi-objective genetic algorithms. In: Ehrgott M, Fonseca CM, Gandibleux X, Hao JK, Sevaux M (eds) Evolutionary multi-criteria optimization: 5th international conference. Springer, Nantes, pp 184–195
- Eskandari H, Geiger CD, Lamont GB (2007) FastPGA: a dynamic problem sizing approach for solving expensive multiobjective optimization problems. In: Obayashi S, Deb K, Poloni C, Hiroyasu T, Murata T (eds) Proceedings of evolutionary multi-criterion optimization (EMO 2007), Lecture notes in computer science, vol 4403. Springer, Berlin, pp 141–155
- Farahani RZ, SteadieSeifi M, Asgari N (2010) Multiple criteria facility location problems: a survey. *Appl Math Modell* 34:1689–1709

- Fazel Zarandi MH, Dvari S, Haddad Sisakht SA (2011) The large scale maximal covering location problem. *Sci Iran* 18(6):1564–1570
- Ghosh A, Dehuri S (2004) Evolutionary algorithms for multi-criterion optimization: a survey. *Int J Comput Inf Sci* 2(1):38–57
- Goldberg JB (2004) Operations research models for the deployment of emergency services vehicles. *EMS Manag J* 1:20–39
- Gutjahr WJ, Doerner KF, Nolz PC (2009) Multi-criteria location planning for public facilities in tsunami-prone coastal areas. *OR Spect* 31:651–678
- Haimes YY, Ladson LS, Wismer DA (1971) Bicriterion formulation of problems of integrated system identification and system optimization. *IEEE Trans Syst Man Cybern* 1(3):296–297
- Hogan K, Reville C (1986) Concepts and applications of backup coverage. *Manag Sci* 32(11):1434–1444
- Hosage CM, Goodchild MF (1986) Discrete space location-allocation solutions from genetic algorithms. *Ann Oper Res* 6:35–46
- Iris C, Asan SS (2012) A review of genetic algorithm applications in supply chain network design. In: Kahraman C (ed) *Computational intelligence systems in industrial engineering—with recent theory and applications*. Atlantic Press, Paris, pp 203–228
- Ishibuchi H, Tsukamoto N, Nojima Y (2008) Evolutionary many-objective optimization: a short review. In: *Proceedings of 2008 IEEE congress on evolutionary computation*. Hong Kong, pp 2424–2431
- Jaramillo JH, Bhadury J, Batta R (2002) On the use of genetic algorithms to solve location problems. *Comput Oper Res* 29:761–779
- Karasakal O, Karasakal E (2004) A maximal covering location model in the presence of partial coverage. *Comput Oper Res* 31:1515–1526
- Konak A, Coit D, Smith A (2006) Multi-objective optimization using genetic algorithms: a tutorial. *Reliab Eng Syst Saf* 91:992–1007
- Li X, Yeh AG (2004) Integration of genetic algorithms and GIS for optimal location search. *Int J Geograph Inf Sci* 19(5):581–601
- Li X, Zhou Z, Zhu X, Wyatt T (2011) Covering models and optimization techniques for emergency response facility location and planning: a review. *Math Methods Oper Res* 74:281–310
- Mahdavi I, Shiripour S, Amiri-Aref M, Otaghsara MM, Amiri NM (2012) Multi-facility location problems in the presence of a probabilistic line barrier: a mixed integer quadratic programming model. *Int J Prod Res* 50(15):3988–4008
- Medaglia AL, Villegas JG, Rodríguez-Coca DM (2009) Hybrid biobjective evolutionary algorithms for the design of a hospital waste management network. *J Heurist* 15:153–176
- Megiddo N, Zemel E, Hakimi SL (1983) The maximum coverage location problem. *SIAM J Algebr Discret Methods* 4(2):253–261
- Michalewicz Z (1996) *Genetic algorithms + data structures = evolution programs*. Springer, Berlin
- Mingjun J, Tang H, Guo J (2004) A single-point mutation evolutionary programming. *Inf Process Lett* 30(6):293–299
- Mladenovic N, Brimberg J, Hansen P, Moreno-Perez JA (2007) The p-median problem: a survey of meta-heuristic approaches. *Eur J Oper Res* 179:927–939
- Pohl AJ, Lamonth GB (2008) Multi-objective UAV mission planning using evolutionary computation. In: Mason SJ, Hill RR, Monch L, Rose O, Jefferson T, Fowler JW (eds) *Proceedings of the 2008 winter simulation conference*. WSC, Florida USA, pp 1268–1279
- ReVelle C, Scholssberg M, Williams J (2008) Solving the maximal covering location problem with heuristic concentration. *Comput Oper Res* 35(2):427–435
- Schilling D, Jayaraman V, Barkhi R (1993) A review of covering problems in facility location. *Locat Sci* 1:25–55
- Storbeck JE, Vohra R (1988) A simple trade-off model for maximal and multiple coverage. *Geograph Anal* 20(3):220–230
- Villegas JG, Palacios F, Medaglia AL (2006) Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example. *Ann Oper Res* 147:109–141
- Yang L, Jones BF, Yang S (2007) A fuzzy multi-objective programming for optimization of fire station locations through genetic algorithms. *Eur J Oper Res* 181:903–905
- Yigit V, Aydin ME, Turkbey O (2006) Solving large-scale uncapacitated facility location problems with evolutionary simulated annealing. *Int J Prod Res* 44(22):4773–4791
- Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm Evolut Comput* 1:32–49

- Zitzler E, Thiele L (1998) Multi-objective optimization using evolutionary algorithms—a comparative case study. In: Eiben AE, Back T, Schoenauer M, Schwefel HP (eds) Proceedings of the fifth international conference on parallel problem solving from nature, Lecture notes in computer science, vol 1498. Springer, Amsterdam, pp 292–301
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. In: Technical report 103. Computer engineering and networks laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland