



A reactive path relinking algorithm for solving the bi-objective p -Median and p -Dispersion problem

I. Lozano-Osorio¹ · J. Sánchez-Oro¹ · A. D. López-Sánchez² · A. Duarte¹

Accepted: 22 February 2023 / Published online: 31 March 2023
© The Author(s) 2023

Abstract

This paper deals with an interesting facility location problem known as the bi-objective p -Median and p -Dispersion problem ($BpMD$ problem). The $BpMD$ problem seeks to locate p facilities to service a set of n demand points, and the goal is to minimize the total distance between facilities and demand points and, simultaneously, maximize the minimum distance between all pairs of hosted facilities. The problem is addressed with a novel path relinking approach, called reactive path relinking, which hybridizes two of the most extended path relinking variants: interior path relinking and exterior path relinking. Additionally, the proposal is adapted to a multi-objective perspective for finding a good approximation of the Pareto front. Computational results prove the superiority of the proposed algorithm over the best procedures found in the literature.

Keywords Multi-objective combinatorial optimization problems · p -Median problem · p -Dispersion problem · Reactive path relinking

1 Introduction

Facility location problems (FLP) are a very-studied family of combinatorial optimization problems that consists in finding the *best* location to site certain number of facilities to efficiently serve all demand points. As is described in López-Sánchez et al. (2020), *the definition of “best” in the facility location literature varies, as it depends on the idiosyncrasies of each particular problem and its solution requirements*. Briefly, in the p -Median problem, first studied in the mid-sixties (Hakimi 1964, 1965), the objective function seeks to minimize the total distance between demand

points and their nearest facility; in the p -center problem, first solved in Minieka (1970), the objective function seeks to minimize the maximum distance between demand points and their nearest facilities; in the maximal covering location problem, first introduced in Church and Velle (1974), the objective consists in maximizing the total number of demand points covered, within a specific distance, by the facilities; or in the p -Dispersion Problem (Erkut 1990), the objective selects facilities as dispersed as possible. There exists many other variants of FLP that optimize different objectives. Traditionally, most of the FLPs has been widely studied not only theoretically but also practically, and many methods of resolution have been successfully proposed, exact and approximate algorithms.

FLPs become even more difficult when more than one objective function must be optimized at a time. This situation appears when companies, either public or private, deal with real-world FLPs since they are inherently multi-objective in nature. Indeed, besides the typical objective function, expressed as a sum of various component expenses (most simply as transportation and fixed costs), multi-objective approaches require for optimizing resource utilization and customer responsiveness, in addition to the standard economic objectives. Recent concerns regarding climate change or environmental objectives have been also considered. See Farahani et al. (2019) for a thorough review.

✉ J. Sánchez-Oro
jesus.sanchezoro@urjc.es

I. Lozano-Osorio
isaac.lozano@urjc.es

A. D. López-Sánchez
adlopsan@upo.es

A. Duarte
abraham.duarte@urjc.es

¹ Universidad Rey Juan Carlos, Computer Science and Statistics Department, Tulipán s/n. 28933 Móstoles, Madrid, Spain

² Department of Economics, Quantitative Methods, and Economic History, Universidad Pablo de Olavide, Ctra. de Utrera km 1. 41013, Sevilla, Spain

The optimization of such problems is hard since the objectives that must be optimized are usually in conflict with each other (Deb and Deb 2014). Indeed, when this happens, it is impossible to improve one objective without deteriorating another one, requiring multi-objective optimization (Sánchez-Oro et al. 2020). The majority of these problems appear when solving real-world challenges, because more than one perspective needs to be tackled simultaneously, such as the profit of the company, the benefits of the workers, or the satisfaction of the customers, among others (Franca et al. 2010; Sagrado et al. 2015; Pinto-Varela et al. 2011). The simple fact of considering more than one perspective at the same time justifies the emergence of a conflict between the different objectives.

In the last few years, the number of scientific papers that have addressed FLP from a multi-objective point of view has increased considerably. For instance, the p -Center problem and the p -Dispersion Problem are studied from a bi-objective optimization perspective in Tutunchi and Fathi (2019) and Pérez-Peló et al. (2019). However, in Karatas and Yakıcı (2018) and López-Sánchez et al. (2020), it is addressed a multi-objective FLP in which they considered three problems: the p -Center problem, the p -Median problem, and the maximal coverage location problem. Different objectives were simultaneously considered in Karatas (2017), where the coverage of the demand points is maximized while the total distance and balance workload of the facilities are minimized. Similarly, in Wang et al. (2018), it is maximized the total coverage and simultaneously minimized the total distance. There exist many other FLP variants depending on the objectives to optimize. For a recent survey of this family of problems, see Boonmee et al. (2017).

This paper is focused on the Bi-objective p -Median and p -Dispersion problem ($BpMD$ problem), a multi-objective facility location problem, which was first addressed in Sayyady (2012). Both objectives have been proven to be in conflict in Sayyady (2012) and Sayyady et al. (2015). Additionally, for a thorough analysis on the relation between several facility location problems from a multi-objective perspective see Erkut and Neuman (1989).

As stated in Sayyady et al. (2015), the location of traffic sensors in highway networks motivated the consideration of the p -Median and p -Dispersion problem simultaneously. Nevertheless, this is not the only realistic problem that fits in this bi-objective model. For instance, considering the location of any profitable chain of business in different sites of a region and being the business exactly the same, the goal would be to locate the businesses near to the customers but, at the same time, in order to do not compete one business to each other, it would be necessary to locate one business as far as possible to another one and here, relies the importance of solving the $BpMD$ problem. Sayyady et al. (2015) formulated the $BpMD$ problem as an integer pro-

gramming model and then, solved by using the ϵ -Constraint (ϵ -C). Furthermore, they proposed an alternative Integer Programming (IP) model and an iterative algorithm named the Incremental Algorithm. Then, they also used a Lagrangian heuristic procedure for solving the IP model for larger instances. Later on, Colmenar et al. (2018) proposed the first metaheuristic to solve the $BpMD$ problem using a Scatter Search algorithm with three improvement methods: a local search based on dominance, a local search that alternates both objectives and a local search that tries to minimize the normalized distance to an ideal point. The algorithms providing the best results for the $BpMD$ problem in the literature were proposed by Sayyady et al. (2015) and Colmenar et al. (2018); therefore, both algorithms have been considered in the competitive testing of Sect. 4. Furthermore, the comparison includes the well-known multi-objective evolutionary algorithms: MOEA/D, Multi-Objective Evolutionary Algorithm based on Decomposition (Zhang and Li 2007); NSGA-II, elitist Non-dominated Sorting Genetic Algorithm (Deb et al. 2002); and SPEA2, Strength Pareto Evolutionary Algorithm (Zitzler et al. 2001).

In this paper, we propose a novel approach based on the path relinking (PR) methodology is proposed, named reactive path relinking (RPR), which is based on the combination of the two most extended path relinking strategies, Interior and Exterior PR (IPR and EPR, respectively) to solve the $BpMD$ problem. EPR strategy was first introduced by Duarte et al. (2015) and was compared against the IPR strategy. The authors solved another interesting FLP known as Minimum Differential Dispersion Problem, whose objective is to minimize the difference between the sum of the maximum and the sum of the minimum distances between the demand points and the selected facilities. The novelty of the current proposal lies in the combination of both path relinking strategies (IPR and EPR) into a single one, called reactive path relinking. The proposed procedure will decide which of the strategies should be used depending on the pairs of solutions to combine. Computational results prove the superiority of the proposed algorithm over the state-of-the-art procedures when considering the set of instances previously used in the related literature. Specifically, the widely accepted multi-objective metrics have been considered in order to compare the procedures, emerging the RPR as the most competitive algorithm in both quality and computing time.

The main contributions of this work are the following:

- A novel metaheuristic algorithm based on the path relinking methodology has been proposed, the reactive path relinking (RPR) algorithm.
- The RPR algorithm has emerged to address a multi-objective facility location problem: the $BpMD$ problem.
- A complete analysis of the performance of the novel approach reactive path relinking has been included.

- A competitive testing is performed with the RPR algorithm and the state-of-the-art algorithms.
- The proposed algorithm and the obtained results are publicly available to ease further comparisons.¹

The rest of this paper is organized as follows. Section 2 describes the *BpMD* problem and includes some definitions related to the problem. Section 3 details the algorithm implemented to solve the problem under consideration. Section 4 includes the performance metrics used to test the quality of the algorithm and the computational experiments to check the validity of our proposal. Finally, Sect. 5 summarizes the paper and discusses future work.

2 Problem definition

Let N and F be two sets of locations, with $|N| = n$ representing the demand points and $|F| = m$ representing the candidate facilities. Without loss of generality, in this paper $N = F$ since our instances consider that demand points are also candidates to host facilities. It is worth mentioning that this assumption is not always considered, since in some cases N and F are disjoint sets. Let us define $d_{ij} \geq 0$ as the distance between i and j , with $i, j \in N$. This distance satisfies $d_{ij} \geq d_{ik} + d_{kj}, \forall i, j, k \in N$, and $d_{ii} = 0$. The goal of a facility location problem is to select a subset of nodes to host facilities, $S \subset N$ with $|S| = p$, in order to optimize one or more objective functions.

The p -Median problem (pMP) selects a subset of $S \subset N$ facilities, with $|S| = p$, with the aim of minimizing the total distance between demand points and its closest facility. Given a solution S , the objective function of p -Median is evaluated as:

$$f_{pM}(S) \leftarrow \sum_{j \in N \setminus S} \min_{i \in S} d_{ij}$$

Then, the objective of the p -Median problem is to find a solution S_{pM}^* with the minimum f_{pM} value. More formally,

$$S_{pM}^* \leftarrow \arg \min_{S \in \mathbb{P}} f_{pM}(S)$$

being \mathbb{P} the set of all possible combinations of facilities that can be conformed selecting p elements from the set of available locations N .

The p -Dispersion problem (pDP) seeks to maximize the distance between the p selected facilities, with the aim of distributing the facilities throughout the available space and avoiding to place facilities close to each other. Given a solution S , the objective function of the p -Dispersion problem is

Table 1 Evaluation of all the feasible solutions for the example depicted in Fig. 1

Facilities	f_{pM}	f_{pD}
(1,1);(1,4);(2,2)	3.83	1.41
(1,1);(1,4);(3,2)	3.24	2.24
(1,1);(1,4);(4,4)	3.65	3.00
(1,1);(2,2);(3,2)	4.47	1.00
(1,1);(2,2);(4,4)	3.24	1.41
(1,1);(3,2);(4,4)	3.83	2.24
(1,4);(2,2);(3,2)	3.65	1.00
(1,4);(2,2);(4,4)	2.41	2.24
(1,4);(3,2);(4,4)	3.24	2.24
(2,2);(3,2);(4,4)	3.65	1.00

evaluated as:

$$f_{pD}(S) \leftarrow \min_{i, j \in S, i \neq j} d_{ij}$$

The p -Dispersion problem then consists in finding a solution S_{pD}^* with the maximum f_{pD} value. Specifically,

$$S_{pD}^* \leftarrow \arg \max_{S \in \mathbb{P}} f_{pD}(S)$$

Therefore, the Bi-objective p -Median p -Dispersion problem, known as *BpMD* problem, is focused on solving two problems at the same time, the pMP and the pDP . It is needed to emphasize that the objective function of the pMP and the pDP are in conflict and, consequently, the improvement of one objective function leads the deterioration the other one. It is worth mentioning that a detailed integer programming model for the *BpMD* is included in Sayyady et al. (2015).

Next, an example with $N = \{1, 2, 3, 4, 5\}$ (with $|N| = n = 5$) and the aim to locate $p = 3$ facilities is included in order to show the conflict between the functions. Table 1 shows all possible solutions (10 combinations), the first column (named Facilities) shows the x and y coordinates of the chosen facilities, and columns 2 and 3 compute the values of the two objective functions of the two considered problems, pMP and pDP , respectively. For this example, the distance between any two points is calculated as the euclidean distance. The best objective function value for each problem isolated is highlighted in bold font.

Figures 1 and 2 represent the optimal location for the p -Median problem and the p -Dispersion problem, respectively. Facilities are represented with squares and demand points with circles. Considering the Euclidean distance, the optimal solution value for the p -Median problem is 2.41 units and the facilities must be located at points (1, 4), (2, 2), and (4, 4). On the contrary, the value of the objective function of the p -Dispersion problem is 2.24 units, when considering the same facilities. Analogously, the optimal solution value for the p -

¹ <https://grafo.etsii.urjc.es/BpMpD>.

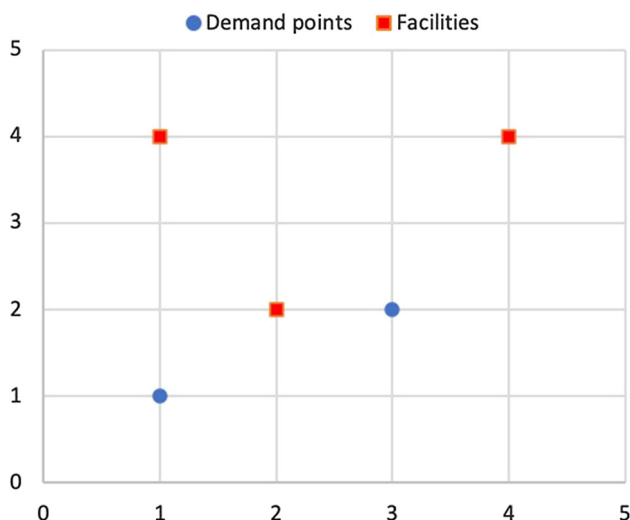


Fig. 1 $(f_{pM}^*, f_{pD}) = (2.41, 2.24)$

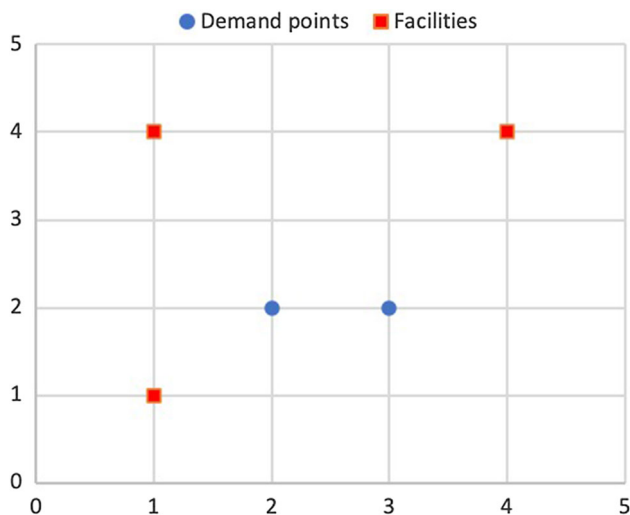


Fig. 2 $(f_{pM}, f_{pD}^*) = (3.65, 3.00)$

Dispersion problem is 3.00 units and the facilities should be located at points (1, 1), (1, 4), and (4, 4), while the evaluation of the objective function for the p -Median problem yields 3.65 units.

As this toy-example shows, both objectives are in conflict, that is, it might not be possible to find a single solution with the optimum value for f_{pM} and f_{pD} simultaneously, being impossible to improve one objective without deteriorating the other one.

Next, without loss of generality the $BpMD$ problem is formally defined as a minimization bi-objective combinatorial optimization problem:

$$\min_{S \in \mathbb{P}} [f_{pM}(S), -f_{pD}(S)]$$

where \mathbb{P} is called the feasible set and $f_{pM} : \mathbb{P} \rightarrow \mathbb{R}$ and $f_{pD} : \mathbb{P} \rightarrow \mathbb{R}$ are the two considered objective functions. The image of the feasible set is $\{(f_{pM}(S), -f_{pD}(S)) : S \in \mathbb{P}\}$ is named the objective space.

Let us briefly introduce the basic concepts of multi-objective definitions. Given two solutions S_1 and S_2 in \mathbb{P} :

- S_1 weakly dominates S_2 , denoted as $S_1 \leq S_2$, if and only if $f_{pM}(S_1) \leq f_{pM}(S_2)$ and $-f_{pD}(S_1) \leq -f_{pD}(S_2)$.
- S_1 dominates S_2 , denoted as $S_1 < S_2$, if and only if $S_1 \leq S_2$ and at least one objective function is strictly better than the other, that is, $f_{pM}(S_1) < f_{pM}(S_2)$ and $-f_{pD}(S_1) \leq -f_{pD}(S_2)$ or $f_{pM}(S_1) \leq f_{pM}(S_2)$ and $-f_{pD}(S_1) < -f_{pD}(S_2)$.
- S_1 strictly dominates S_2 , denoted as $S_1 \leq\leq S_2$, if and only if $f_{pM}(S_1) < f_{pM}(S_2)$ and $-f_{pD}(S_1) < -f_{pD}(S_2)$.

The goal of the $BpMD$ problem is to find a high-quality approximation of non-dominated solutions, also known as efficient solutions or Pareto set. It should bear in mind that the *Pareto set* contains only those solutions that are not dominated by any other solution. Notice that the image of the Pareto set is named the *Pareto front*. In order to achieve this goal, a metaheuristic procedure able to solve the $BpMD$ problem is proposed. The metaheuristic finds the best approximation of the Pareto front with high-quality solutions in a reasonable amount of time even for problems with a considerable size.

3 Algorithmic proposal

The methodology used to solve the $BpMD$ problem is based on a simple but efficient algorithm which has been successfully applied to solve a wide variety of combinatorial optimization problems, the path relinking (PR) metaheuristic (see Resende et al. (2010); Pérez-Peló et al. (2020); Campos et al. (2014) for relevant research in PR). This metaheuristic was first proposed in the framework of tabu search to integrate intensification and diversification strategies (Glover and Laguna 1998). The success of PR relies on exploring trajectories that connect pairs of high-quality solutions (hereinafter, the *initiating* and *guiding solutions*, S_i and S_g , respectively). The procedure generates new *intermediate solutions* that would be hopefully better than these pairs of connected solutions. Specifically, PR links S_i to S_g combining paths of solutions which in turn, generates new solutions on such paths that will share attributes from both solutions. Without loss of generality, combining two solutions S_i and S_g consists in including attributes of S_g in S_i iteratively until $S_i = S_g$.

In this paper, two different PR strategies, Interior and Exterior PR (IPR and EPR, respectively), are considered. Additionally, a novel approach based on the mixture of these two path relinking strategies is proposed. Both strategies have been deeply studied but in an isolated way by Duarte et al. (2015). The novelty of our proposal lies on the combination of both, IPR and EPR, into a single one, which is able to autonomously decide which of the strategies should be used, analyzing the pairs of solutions to combine.

3.1 Constructive method

As previously explained, the PR methodology requires a set of efficient solutions to combine. Although they can be constructed in a random manner, a greedy constructive procedure is proposed in order to form promising points of the search space. Of course, another way to obtain a good set of efficient solutions would be using multi-start algorithms such as GRASP (Greedy Randomized Adaptive Search Procedure), see Feo and Resende (1995). The constructive method is responsible for obtaining the initial set of efficient solutions. It uses a greedy criterion to generate solutions from scratch by using an iterative algorithm. At each iteration, the algorithm performs the best choice with respect to the objective function to optimize.

The $BpMD$ problem is a bi-objective optimization problem, so it is needed to optimize two objective functions that are in conflict. Therefore, to consider both objectives at the same time in the constructive procedure, this bi-objective optimization problem is turned into a single-objective optimization problem. There are several ways of combining two or more objectives in an optimization procedure. For example, Marti et al. (2015) presents a new approach for switching between different objectives. In this research, an aggregation of the considered objectives is proposed, resulting in a simple yet effective mechanism. More precisely, given a partial solution S' with $|S'| < p$ and a node $v \in N \setminus S'$, the proposed function is defined as $\beta \cdot f_{pM}(S' \cup \{v\}) - (1 - \beta) \cdot f_{pD}(S' \cup \{v\})$, where $\beta \in [0, 1]$ is a weight or proportion to the relative importance of the pMP objective.

In order to normalize both objective functions in the range $[0, 1]$, the following intervals are defined: $f_{pM}(S' \cup \{v\}) \in [\min_{pM}; \max_{pM}] = [0; \max d_{ij} : \forall i, j \in N \wedge i \neq j]$ and $f_{pD}(S' \cup \{v\}) \in [\min_{pD}; \max_{pD}] = [0; \max d_{ij}, \forall i, j \in N \wedge i \neq j]$, where \min_{pM} and \max_{pM} represent the minimum and the maximum values of f_{pM} , respectively, and similarly, \min_{pD} and \max_{pD} are the minimum and the maximum values of the f_{pD} , respectively. For the sake of convenience, the objective function of the pDP is transformed into a minimization problem by simply switching the sign, i.e., $-f_{pD}(S' \cup \{v\}) \in [-\max_{pD}; -\min_{pD}]$.

Therefore, given a partial solution S' , a candidate $v \in N \setminus S'$, and β , the considered greedy function is defined as

follows:

$$\begin{aligned}
 g(S' \cup \{v\}, \beta) &= \beta \frac{f_{pM}(S' \cup \{v\}) - \min_{pM}}{\max_{pM} - \min_{pM}} \\
 &\quad + (1 - \beta) \frac{-f_{pD}(S' \cup \{v\}) - (-\max_{pD})}{-\min_{pD} - (-\max_{pD})} \\
 &= \beta \frac{f_{pM}(S' \cup \{v\})}{\max_{pM}} \\
 &\quad - (1 - \beta) \frac{f_{pD}(S' \cup \{v\}) + \max_{pD}}{\max_{pD}} \tag{1}
 \end{aligned}$$

Algorithm 1 shows the pseudocode for the greedy construction used in this paper, which receives as input parameter the set of candidate elements able to host a facility and the step used to update β . It is worth mentioning that as a multi-objective optimization problem is being addressed, therefore, the output of the constructive procedure is a set of efficient solutions.

Algorithm 1 Greedy($N, step$)

```

1:  $ES \leftarrow \emptyset$ 
2:  $\beta \leftarrow 0$ 
3: while  $\beta \leq 1$  do
4:   for all  $v \in N$  do
5:      $S' \leftarrow \{v\}$ 
6:      $C \leftarrow N \setminus \{v\}$ 
7:     while  $|S'| \neq p$  do
8:        $v' \leftarrow \arg \min_{v \in C} g(S' \cup \{v\}, \beta)$ 
9:        $S \leftarrow S \cup \{v'\}$ 
10:       $C \leftarrow C \setminus \{v'\}$ 
11:    end while
12:    Insert&Update( $S', ES$ )
13:  end for
14:   $\beta \leftarrow \beta + step$ 
15: end while
16: return  $ES$ 

```

The algorithm starts with an initial set of efficient solutions, ES , that is empty and the parameter β is initialized to zero (steps 1 and 2). With the aim of increasing the diversity of the set of constructed solutions, the values of β are uniformly selected in the interval $[0, 1]$. Specifically, β is set from 0.0 to 1.0, where each iteration increases its value by adding the input parameter $step$ (see steps 3 to 15).

Greedy algorithms are mainly deterministic procedures, since they select in each step the best element to be included in the partial solution under construction. In order to further improve the diversification, it is proposed to construct several solutions where each one is initialized by considering a different element. In particular, if there are n elements, that means that at most, n different solutions could be constructed, (steps 4 to 13).

The method selects the first facility, v , to be added to the solution S' (step 5), creating the list of candidate nodes, C ,

with all nodes except the one that has been already included in the solution (step 6). The algorithm includes one node at each iteration until p nodes have been included in S , (steps 7–11). In each iteration, the most promising node from the list of candidates is selected, v' by using the aforementioned greedy function (see Eq. 1). Then, the selected node is included in the solution (step 9) and removed from the candidate list (step 10).

As a bi-objective optimization problem is being addressed, once that a feasible solution is found, it is needed to check whether it can be admitted in the set of efficient solutions (step 12) or not. Specifically, the function *Insert&Update* (S' , ES) determines if the solution S' is an efficient solution and, if so, it is inserted in the set of efficient solutions, removing all solutions of the set that are dominated by S' . Finally, before starting a new iteration, the value of β is updated in step 14. The greedy algorithm ends by returning the set of efficient solutions (step 16).

3.2 Local search method

Once an initial set of efficient solutions is attained, a second phase is considered. This phase consists in applying a local search to all the efficient solutions obtained. As a facility location problem is being solved, the neighborhood of a solution S , denoted as $\mathcal{N}(S)$, for the *BpMD* problem is defined as the set of solutions that can be obtained by exchanging a selected facility with any non-selected facility. Specifically,

$$\mathcal{N}(S) \leftarrow \{(S \setminus \{u\}) \cup \{v\} : u \in S, v \in N \setminus S\}$$

The proposed local search follows a first-improvement strategy. In particular, the method traverses $\mathcal{N}(S)$, performing the first move that leads to a better solution with respect to the considered objective. If so, the search is subsequently restarted to the neighborhood of the updated solution; otherwise, the method stops since no better solution can be found. It is important to bear in mind that two objective functions are being addressed. Then, there are different variants to consider (see Mladenović et al. 2007; Erkut 1990). The use of the weighted sum function that was previously defined in Eq. 1 to guide the search is considered again. Specifically, the procedure starts by considering $\beta = 0$, which means that only f_{pM} is improved (ignoring the other objective). Notice that after each move, the corresponding neighboring solution is tested whether it is admitted in the set of efficient solutions or not. This strategy is maintained until $\beta = 1.0$ which means that only f_{pD} is improving, ignoring f_{pM} .

Algorithm 2 shows the pseudocode of the proposed improvement strategy used in this paper. It receives three input parameters: an initial solution, S ; the set of efficient solutions, ES ; and the β parameter. As customary, the local search ends when no improvement is found (steps 2 to 11).

In each iteration, this method explores the aforementioned neighborhood (steps 4 to 10) by following the first improving strategy. Specifically, the exploration is halted once a better neighboring solution is found (steps 5 to 8). Notice that the comparison between S and S' is done by considering Eq. 1 (step 5). Notice that it is needed to test whether any solution explored qualifies to enter in ES or not (step 9). Finally, the local search algorithm returns a set of efficient solutions (step 12).

Algorithm 2 LocalSearch(S , ES , β)

```

1: improve  $\leftarrow$  true
2: while improve = true do
3:   improve  $\leftarrow$  false
4:   for all  $S' \in \mathcal{N}(S)$  and improve = false do
5:     if  $f(S', \beta) < f(S, \beta)$  then
6:        $S \leftarrow S'$ 
7:       improve  $\leftarrow$  true
8:     end if
9:     Insert&Update( $S'$ ,  $ES$ )
10:  end for
11: end while
12: return  $ES$ 

```

So far, our proposal is quite similar to GRASP. However, our way to construct solutions is always deterministic with the only diversification of changing the initial node to be added to the partial solution under construction. Therefore, n different solutions are obtained in the constructive phase. Once that a feasible solution is obtained, the local search is applied as it is done in GRASP.

3.3 Interior path relinking

Interior PR (IPR) creates a path that connects two high-quality feasible solutions, exploring new solutions while traversing the path. The rationale behind this is that if the initiating and guiding solutions are promising, then there is a high probability of finding good solutions in the path that connects them.

The path between S_i and S_g is created including in S_i elements that are in $S_g \setminus S_i$, exchanging them with those elements in $S_i \setminus S_g$. In other words, it consists in iteratively modifying S_i to become more similar to S_g in every step, until reaching S_g .

Figure 3 shows an example where the IPR is illustrated. The example considers a set of $|N| = 10$ nodes labelled from 1 to 10 and $p = 5$. Let $S_i = \{1, 2, 3, 4, 5\}$ and $S_g = \{1, 2, 6, 7, 8\}$ be the initiating and guiding solutions, respectively. At the first step of the IPR, element 3 is replaced by element 6, obtaining the intermediate solution $S_1 = \{1, 2, 6, 4, 5\}$, then, at the second step, element 4 is interchanged by element 7, resulting in other intermediate solution, $S_2 = \{1, 2, 6, 7, 5\}$, and at the last step, element 5 is

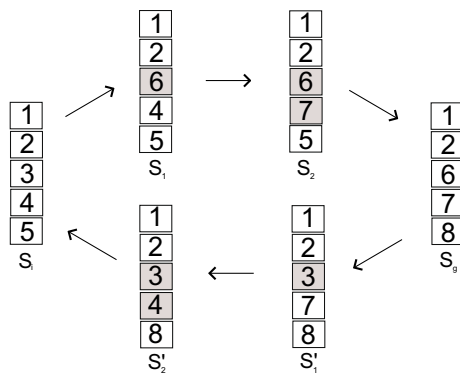


Fig. 3 Interior path relinking

substituted by element 8 obtaining, in such a way, the guiding solution, S_g . Similarly, to transform S_g into S_i , the first step of the IPR replaces element 6 by element 3, obtaining the intermediate solution $S'_1 = \{1, 2, 3, 7, 8\}$, then, at the second step, element 7 is interchanged by element 4, resulting in the solution $S'_2 = \{1, 2, 3, 4, 8\}$, and the last step substitutes element 8 by element 5 obtaining, in such a way, the initiating solution, S_i .

Notice that every solution found during the path is considered for entering in the set of efficient solutions.

3.4 Exterior path relinking

Exterior PR (EPR) (Duarte et al. 2015) follows the opposite idea of IPR. If the solutions in the efficient set are similar among them, then the path created with IPR will be rather short, thus leading to a small number of new solutions explored. This behavior results in maintaining the similarity among solutions in the efficient set. Then, it is necessary to include some diversity in the search, to explore a different region of the search space.

Given an initiating solution S_i and a guiding solution S_g , EPR iteratively includes in S_i elements that are not in S_g , with the aim of reaching new solutions which are diverse with respect to both S_i and S_g . This strategy will eventually lead the algorithm to explore a wider portion of the search space to increase the quality of the efficient set.

Figure 4 depicts an example where the EPR has been illustrated. Again, a set of $|N| = 10$ nodes is considered, labelled from 1 to 10 and $p = 5$. Let $S_i = \{1, 2, 3, 4, 5\}$ and $S_g = \{1, 2, 3, 6, 7\}$ be two pairs of solutions. The task is to remove the values $S_i \cap S_g = \{1, 2, 3\}$ with elements in the set $N \setminus S_i \cup S_g = \{8, 9, 10\}$. At the first step of the EPR, element 1 is replaced by element 8, getting the intermediate solution $S_1 = \{8, 2, 3, 4, 5\}$, at the second step, element 2 is interchanged by element 9, obtaining $S_2 = \{8, 9, 3, 4, 5\}$, at the third step, element 3 is substituted by element 10, resulting in the intermediate solution

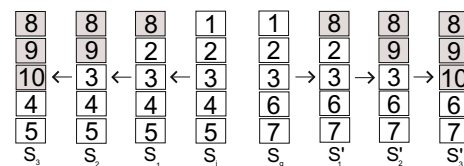


Fig. 4 Exterior path relinking

$S_3 = \{8, 9, 10, 4, 5\}$. Similarly, if you start with the solution S_g thought S_i , $S_g = \{1, 2, 3, 6, 7\}$, then the first step of the EPR interchanges element 1 with element 8, resulting in the intermediate solution $S'_1 = \{8, 2, 3, 6, 7\}$. The second step of the EPR replaces element 2 by element 9, getting the solution $S'_2 = \{8, 9, 3, 6, 7\}$. And finally, at the last step of the EPR, the element 3 is substituted by element 10, obtaining the intermediate solution $S'_3 = \{8, 9, 10, 6, 7\}$.

3.5 Reactive path relinking

One of the main contributions of this work is the proposal of a novel path relinking approach that selects IPR or EPR analyzing the initiating and guiding solutions. This algorithm, named reactive path relinking (RPR), combines the two strategies of the PR previously described, IPR and EPR.

RPR analyzes the similarity between the initiating S_i and guiding S_g solutions. If the similarity between them does not exceed a certain threshold k , then it is assumed that S_i and S_g are different enough to apply IPR with the aim of intensifying. Otherwise, EPR is applied to diversify the search. The combination of both strategies balances intensification and diversification with the aim of finding a high-quality set of efficient solutions. The value of k is an input parameter of the algorithm (see Sect. 4 for a detailed experimentation to select the best value) which represents the minimum number of facilities that S_i and S_g must have in common to be considered similar solutions. Algorithm 3 shows the pseudocode for the considered RPR algorithm.

The algorithm receives 3 input parameters: the set of efficient solutions, ES ; k , the threshold to consider that two solutions are similar or not; and the weight β . RPR starts by initializing the backup-copy of the efficient set (step 1) and the best found solution found during the complete execution of RPR (step 2). Additionally, the set T is initialized (step 3) and used to store the already combined solutions. Then, the main body of the algorithm is repeated while the generation of paths finds non-dominated intermediate solutions (steps 4 to 32). In each iteration, all elements are first shuffled in the efficient set to avoid prioritize in any search direction (step 5). Then, in order to increase the efficiency of the method, those solutions for which a path was generated in the previous iterations are discarded (step 8). After that, for non-explored pairs of solutions, their similarity is evalu-

Algorithm 3 $RPR(ES, k, \beta)$

```

1:  $ES' \leftarrow \emptyset$ 
2:  $S^* \leftarrow \emptyset$ 
3:  $T \leftarrow \emptyset$ 
4: while  $ES \neq ES'$  do
5:    $ES' \leftarrow \text{shuffle}(ES)$ 
6:   for all  $S_i \in ES'$  and  $(ES \setminus ES') = \emptyset$  do
7:     for all  $S_g \in ES'$  and  $S_i \neq S_g$  and  $(ES \setminus ES') = \emptyset$  do
8:       if  $\langle S_i, S_g \rangle \notin T$  then
9:         if  $|S_i \cap S_g| < k$  then  $\triangleright$  If solutions are different (less
            than  $k$  facilities in common), IPR
10:            $R \leftarrow S_i \setminus S_g$ 
11:            $A \leftarrow S_g \setminus S_i$ 
12:         else  $\triangleright$  If solutions are similar ( $k$  of more facilities in
            common), EPR
13:            $R \leftarrow S_i \cap S_g$ 
14:            $A \leftarrow N \setminus (S_i \cap S_g)$ 
15:         end if
16:          $S \leftarrow S_i$ 
17:         while  $R \neq \emptyset$  do
18:            $u \leftarrow \text{RND}(R)$ 
19:            $v \leftarrow \text{RND}(A)$ 
20:            $S \leftarrow S \setminus \{u\} \cup \{v\}$ 
21:            $R \leftarrow R \setminus \{u\}$ 
22:            $A \leftarrow A \setminus \{v\}$ 
23:            $\text{Insert\&Update}(S, ES)$ 
24:           if  $S^* = \emptyset$  or  $f_r(S^*) < f_r(S)$  then  $\triangleright$  for
             $r = pM, pD$ 
25:              $S^* = S$ 
26:           end if
27:         end while
28:          $T \leftarrow T \cup \langle S_i, S_g \rangle$ 
29:       end if
30:     end for
31:   end for
32: end while
33:  $\text{LocalSearch}(S^*, ES, \beta)$ 
34: return  $ES$ 

```

ated. If they are different enough (step 9), the IPR strategy is considered, being the set of nodes to remove, denoted by R , those which are in S_i but not in S_g (step 10), and the set of candidates nodes to be added those which are in S_g but not in S_i , denoted by A (step 11). Otherwise, EPR strategy is considered (step 12), conforming R with the nodes that S_i and S_g have in common (step 13) and A with those that are neither in S_i nor in S_g (step 14). Then, the solution used to traverse the path, S , is initialized with the initiating solution S_i (step 16).

The method iterates while there are nodes to be removed (steps 17–27). In each iteration, the nodes that will be removed and added to the solution are selected at random from R and A , respectively (steps 18–19). Then, the solution is updated (step 20), as well as the nodes to be removed (step 21) and the candidates to be added (step 22).

Considering that a multi-objective optimization problem is being solved, it must be checked if any constructed solution can be admitted in the efficient set (step 23). Moreover, it is tested if any of the solutions found during the generation of a path is the best based on f_r , or not (steps 24 to 26).

Once a specific path between S_i and S_g has been finished, the corresponding pair is stored in T to avoid exploring the same path in future iterations (step 28). At the end of RPR, the improvement method is applied with the aim of further improving the generated solution, updating, if required, the efficient set (step 33). Finally, the algorithm returns the update ES .

3.6 Computational complexity

This section is devoted to analyze the computational complexity of the proposed algorithm. In particular, each component of the complete algorithm is evaluated and, finally, the complexity of the algorithm is presented.

The first component of the proposed algorithm is the greedy constructive procedure. This method constructs a solution starting from each node $i \in N$ and, in each construction, the best node is selected to be included in each iteration, resulting in a complexity of $O(n \cdot p)$.

Then, for each constructed solution, the local search method is applied which, following a first improvement approach, requires to traverse the complete neighborhood in the worst case, resulting in swapping each selected element $s \in S$ with every available node in $V \setminus S$, resulting in a complexity of $O(n \cdot p)$.

These two phases, constructive and local search, are repeated during β/step iterations, thus having a computational complexity of $O(n \cdot p \cdot \beta/\text{step})$.

Once the initial set of non-dominated solutions, ES , is constructed, the RPR is applied. The RPR is executed for each pair of solutions in ES avoiding repeating pairs (i.e., the combination of S_2 and S_1 is not performed if the combination of S_1 and S_2 has been already executed). Therefore, the complexity of this method is $O(|ES| \cdot \log |ES| \cdot p)$. Notice that the inclusion of p in this equation is due to that, in the worst case, each combination performs p iterations (when the combined solutions are completely different in IPR or the same solution in EPR). At the end of each combination, the local search is applied, and then, RPR presents a total complexity of $O(|ES| \cdot \log |ES| \cdot p^2 \cdot n)$.

Finally, the computational complexity of the complete algorithm is the maximum between the complexity of generating the set of non-dominated solutions and the complexity of executing the RPR over the set of non-dominated solutions, that is the maximum between $O(n \cdot p \cdot \beta/\text{step})$ and $O(|ES| \cdot \log |ES| \cdot p^2 \cdot n)$, respectively.

4 Computational results

This section presents the computational results conducted on the proposed metaheuristic and a comparison against several of the best-known multi-objective algorithms. All experiments have been performed in an Intel Core i7-7700HQ (4 x 2.8 GHz) with 8GB RAM, and the algorithms were implemented using Java 9. The source code has been also made publicly available.²

The results of this section are divided into two parts: preliminary and final experiments. In the former, the influence of each part of the proposed algorithm is analyzed as well as the corresponding parameters setting, while the latter shows the performance of the best configuration for the proposed algorithm over the complete set of instances, configured with the best selected parameters. The quality of the obtained results is measured by comparing them with the best approximation of the Pareto front found (hereafter B)³ since the optimal Pareto front of the $BpMD$ problem is unknown.

To evaluate the performance of the proposal, three set of instances have been considered, denoted as *pmed set*, *D set*, and *kmedian set*. The first one consists of 40 instances of the well-known OR-library⁴ (Beasley 1990). The number of nodes ranges from 100 to 900, while the number of facilities to select is in the range [5, 200], having a large variety of combinations. The second set of instances consists of 9 medium-sized instances with $n = 250$ and $p = 25$, named as $D250$ and 10 instances with $n = 350$ and $p = 35$, named $D350$ previously solved by Colmenar et al. (2018) and by Sayyady et al. (2015). Last set of instances consists of 15 large-sized instances, the smallest one contains $n = 1000$ nodes, then, the next instances increment by 1000 nodes until reach $n = 5000$, each instance has 3 different p values [$n \cdot 0.05, n \cdot 0.15, n \cdot 0.25$]. They have been downloaded from UFLLib.⁵

Table 2 summarizes the main characteristics of each instance. To prevent the well-known over-fitting, a representative subset of instances (25% aprox.) with different characteristics is randomly selected.

Before starting with the computational results, it is needed to briefly describe how the quality of the algorithms is measured when dealing with multi-objective optimization problems. As it is well-known, the performance among multi-objective optimization algorithms cannot be done by comparing the objective function values, but comparing

Pareto fronts since multi-objective optimization obtains a set of efficient solutions instead of an unique solution as in single-objective optimization. Therefore, to compare different algorithms it is necessary to use metrics that evaluate the quality of each obtained Pareto front. It is needed to measure basically the cardinality of the Pareto front, the proximity of the obtained solutions to the best/optimal Pareto front, and the diversity of the solutions. In this work, several of the most extended multi-objective metrics have been considered (see Durillo and Nebro 2011 or Li and Yao 2019): the number of efficient solutions, coverage, spread, hypervolume, the ϵ -indicator, generational distance, and inverted generational distance.

The *number of efficient solutions in the Pareto front* ($|A|$) counts the number of efficient solutions found with the considered algorithm. The decision-maker usually prefers a larger number of efficient solutions. However, this metric does not have into account the quality of the solutions.

The *coverage metric*, $C(A, B)$, evaluates the proportion of solutions of the best/optimal Pareto front B that weakly dominates the solutions from the Pareto front obtained by the algorithm A . Let r be the number of objective functions, then the coverage is computed as:

$$C(A, B) = \frac{|\forall b \in B | \exists a \in A : f_r(a) \leq f_r(b) \forall r|}{|B|}$$

Therefore, the smaller the value of $C(A, B)$, the better the algorithm A is. Note that $0 \leq C(A, B) \leq 1$, if $C(A, B) = 0$ means that no solution in the best/optimal Pareto front is weakly dominated by the solutions obtained by algorithm A and $C(A, B) = 1$ means that all the solutions in the best/optimal Pareto front are weakly dominated by the solutions obtained by algorithm A .

The *spread* (Δ) measures the extent of spread by the set of computed solutions obtained by the Pareto front of algorithm A .

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{|A|-1} |d_i - \bar{d}|}{d_f + d_l + (|A| - 1)\bar{d}}$$

where d_i is the Euclidean distance between consecutive solutions in the Pareto front obtained with algorithm A , \bar{d} is the mean of these distances, and d_f and d_l are, respectively, the Euclidean distances to the extreme solutions of the best/optimal Pareto front in the objective space. The best possible value is $\Delta = 0$ which indicates a perfect spread of the solutions in the Pareto front obtained with the considered algorithm.

The *hypervolume* (HV) evaluates the volume in the objective space which is covered by the Pareto front obtained with the algorithm A . To calculate the HV , for each solution $a \in A$, a hypercube v_a is constructed with a reference point

² <https://grafo.etsii.urjc.es/BpMpD>.

³ The best approximation of the Pareto front, denoted B , is estimated with all the non-dominated solutions of the set, resulting from merging the solutions found by all the algorithms under comparison.

⁴ <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>.

⁵ <http://resources.mpi-inf.mpg.de/departments/d1/projects/benchmarks/Ufllib/packages.html>.

Table 2 Main characteristics of each considered instance

pmed instances			D_250 & D_350 instances			kmedian instances		
Instance	Nodes	Facilities	Instance	Nodes	Facilities	Instance	Nodes	Facilities
pmed1	100	5	D_250_1	250	25	kmedian1	1000	50
pmed2	100	10	D_250_2	250	25	kmedian2	1000	150
pmed3	100	10	D_250_3	250	25	kmedian3	1000	250
pmed4	100	20	D_250_5	250	25	kmedian4	2000	100
pmed5	100	33	D_250_6	250	25	kmedian5	2000	300
pmed6	200	5	D_250_7	250	25	kmedian6	2000	500
pmed7	200	10	D_250_8	250	25	kmedian7	3000	150
pmed8	200	20	D_250_9	250	25	kmedian8	3000	450
pmed9	200	40	D_250_10	250	25	kmedian9	3000	750
pmed10	200	67	D_350_1	350	35	kmedian10	4000	200
pmed11	300	5	D_350_2	350	35	kmedian11	4000	600
pmed12	300	10	D_350_3	350	35	kmedian12	4000	1000
pmed13	300	30	D_350_4	350	35	kmedian13	5000	250
pmed14	300	60	D_350_5	350	35	kmedian14	5000	750
pmed15	300	100	D_350_6	350	35	kmedian15	5000	1250
pmed16	400	5	D_350_7	350	35			
pmed17	400	10	D_350_8	350	35			
pmed18	400	40	D_350_9	350	35			
pmed19	400	80	D_350_10	350	35			
pmed20	400	133						
pmed21	500	5						
pmed22	500	10						
pmed23	500	50						
pmed24	500	100						
pmed25	500	167						
pmed26	600	5						
pmed27	600	10						
pmed28	600	60						
pmed29	600	120						
pmed30	600	200						
pmed31	700	5						
pmed32	700	10						
pmed33	700	70						
pmed34	700	140						
pmed35	800	5						
pmed36	800	10						
pmed37	800	80						
pmed38	900	5						
pmed39	900	10						
pmed40	900	90						

W (a vector of worst objective function values) and the solution a as the diagonal corners of the hypercube. Then, the hypervolume is the union of all hypercubes is found.

$$HV = \text{volume} \left(\bigcup_{a=1}^{|A|} v_a \right)$$

Then, the larger the HV value, the better the set of efficient solutions obtained by the algorithm.

The ϵ -indicator (ϵ) measures the smallest distance (the smallest ϵ value) needed to transform every point of the Pareto front obtained with the algorithm A in the closest point of the best/optimal Pareto front.

$$\epsilon = \inf\{\epsilon \in \mathbb{R} : \forall b \in B \exists a \in A \text{ such that } f_k(a) \leq \epsilon \cdot f_k(b) \forall k\}.$$

Therefore, the smaller the ϵ -indicator, the better.

The *generational distance* (GD) measures how far the solutions of the Pareto front obtained by the algorithm A are from those solutions in the best/optimal Pareto front (B).

$$GD = \frac{\sqrt{\sum_{i=1}^{|A|} d_i^2}}{|A|}$$

where d_i is the Euclidean distance between each of these solutions and the nearest solution of the best/optimal Pareto front. If $GD = 0$ indicates that all the solutions are in the best Pareto front.

The *inverted generational distance* (IGD) is an inversion of the generational distance metric with the aim of measuring the distance from the best/optimal Pareto front to the set of efficient solutions obtained by the algorithm A .

$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{|B|}.$$

The smallest the value of IGD, the better. Note that the only difference between GD and IGD is that while comparing, the IGD does not miss any part of best/optimal Pareto front.

Finally, the computing time (CPU) required by each algorithm is included.

As the considered problem is a multi-objective optimization problem, a set of non-dominated solutions (which trivially contains more than one solution) is generated by each algorithm. The results presented in Tables 3 to 7 report the average values across the set of instances used in each experiment, by performing a single execution of the compared procedures.

4.1 Algorithm parameter tuning

This section is oriented toward both demonstrating the effectiveness of the proposed strategies and tuning the parameters. The aim of the first preliminary experiment is to determine the number of constructions to calculate the initial set of efficient solutions. As Sect. 3.1 describes, the number of solutions built depends on the *step* value. Specifically, it has been considered $step \in \{1/50, 1/100, 1/150, 1/200\}$, which results in 51, 101, 151, and 201 solutions built. Table 3 shows the results obtained in this experiment, where the metrics described above are reported.

Analyzing the results of Table 3, as expected the larger the number of solutions built, the better performance in all metrics. Therefore, the more solutions, the better results. However, the average computing time increases considerably. Note that between the two intermediate values of the

column named *step*, values 1/100 and 1/150, there are no significant differences in the performance in all the metrics and the computing time can be substantially reduced. For this reason, a compromise between the quality of the performance metrics and the computational time could be selected. Thus, *step* is set to 1/100.

As described in Sect. 3.1, it is desirable to construct a rich and diverse set of efficient solutions. To this end, β is set to 0.00, 0.01, 0.02, ..., 0.98, 0.99, 1.00 since it was opted by the $step=1/100$. Note that, $\beta = 0.0$ optimizes f_{pD} meanwhile $\beta = 1.0$ optimizes f_{pM} . Of course, $\beta = 0.50$ considers that both objectives are equally important, f_{pM} and f_{pD} , see Eq. 1.

Once the number of constructions is set, Table 4 is analyzed. In the next experiment, the results obtained with the reactive path relinking when considering different values for k are shown. Specifically, the k value are set as $0.25 \times p$, $0.50 \times p$, and $0.75 \times p$, being p the number of facilities. Furthermore, in this Table 4, the IPR and the EPR are included in order to evaluate the actual contribution of the reactive mechanism. The associated results are summarized in Table 4, where the same metrics than in the above table are reported.

According to the results shown in Table 4, it is difficult to conclude the more appropriate value of k to select the best RPR variant. Nevertheless, the RPR with $k = 0.75 \times p$ achieves the best values for the coverage (0.38), the hypervolume (0.64), and IGD (6680.34). Additionally, it can be observed that the differences between the best values obtained with other strategies and the strategy of RPR with $k = 0.75 \times p$ are quite similar, obtaining no significant differences: the number of efficient solutions is practically equal (36.68 vs 36.06), the spread (0.95 vs 0.95), the ϵ -indicator (0.03 vs 0.03), and GD (1648.78 vs 1674.72). Furthermore, the computing time of RPR with $k = 0.75 \times p$ is the slightly faster (1014.91 vs 1067.66). Therefore, this variant is selected as the most competitive one.

Finally, with the aim of analyzing the impact of each part of the proposed algorithm, an additional experiment which consists of comparing the results obtained by the constructive procedure (C) is conducted, then coupled with the local search method (C+LS), and finally, the reactive path relinking (RPR). Table 5 shows the results obtained in this experiment.

Analyzing the coverage, the most relevant part of the algorithm is the RPR, which is able to reduce the coverage to almost 0. In this case, the local search procedure clearly outperforms the results obtained by the constructive procedure (0.77 versus 0.87). On the other hand, considering the hypervolume, C+LS considerably improves the results of the constructive procedure, while RPR is able to provide the final improvement that results in a highly competitive algorithm for the *BpMD*. Notice that in the ϵ -indicator metric the local search procedure is the one obtaining the most remarkable improvement. Additionally, the number of non-dominated

Table 3 Effect of varying the number of steps in the constructive procedure

<i>step</i>	<i>A</i>	<i>C</i> (<i>A</i> , <i>B</i>)	Δ	HV	ϵ	GD	IGD	CPU
1/50	5.62	0.10	0.96	0.40	0.13	3696.27	6784.83	124.99
1/100	7.18	0.03	0.97	0.43	0.07	3164.47	6713.45	240.86
1/150	7.31	0.03	0.97	0.43	0.06	3146.91	6713.45	350.65
1/200	8.18	0.00	0.97	0.46	0.00	2931.86	6704.58	462.50

Table 4 Comparison of the different proposed variants of path relinking

	<i>A</i>	<i>C</i> (<i>A</i> , <i>B</i>)	Δ	HV	ϵ	GD	IGD	CPU
IPR	36.68	0.40	0.95	0.64	0.03	1648.78	6681.52	1117.21
RPR ($k = 0.25 \times p$)	32.00	0.82	0.95	0.62	0.06	1798.44	6689.65	1014.91
RPR ($k = 0.50 \times p$)	31.93	0.74	0.95	0.63	0.04	1784.81	6687.09	1551.63
RPR ($k = 0.75 \times p$)	36.06	0.38	0.95	0.64	0.03	1674.72	6680.34	1067.66
EPR	32.18	0.80	0.96	0.62	0.05	1788.03	6688.96	1404.24

Table 5 Analysis of the effect of the construction phase, local search and path relinking in the proposed algorithm

<i>step</i>	<i>A</i>	<i>C</i> (<i>A</i> , <i>B</i>)	Δ	HV	ϵ	GD	IGD	CPU
C	7.18	0.87	0.97	0.50	0.28	3164.47	6713.46	240.86
C + LS	31.50	0.77	0.98	0.61	0.08	1810.21	6699.64	757.47
RPR	36.06	0.05	0.95	0.64	0.01	1674.72	6680.33	1067.66

solutions increases considerably when including the local search strategy. Therefore, it can be concluded that each part of the proposed algorithm is essential in the final RPR algorithm, each one of them providing different advantages that lead to increase the effectiveness of the algorithm.

4.2 Comparison to other algorithms

The most appropriate parameters to implement the RPR algorithm have been chosen considering the results obtained in the previous section. Specifically, $step = 1/100$ and $k = 0.75 \times p$. The performance of the proposed algorithm is compared with three of the most competitive evolutionary algorithms, MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition), NSGA-II (elitist Non-dominated Sorting Genetic Algorithm), and SPEA2 (Strength Pareto Evolutionary Algorithm), the Scatter Search algorithm proposed by Colmenar et al. (2018) and the ϵ -C method, see Sayyady et al. (2015). In all the remaining tables, the number between parenthesis close to the name of the instance set indicates the number of instances in that set. Before presenting the results, all algorithms will be briefly described.

MOEA/D, first proposed in Zhang and Li (2007), decomposes the multi-objective optimization problem into single-objective optimization subproblems using a set of even spread weight vectors. All these subproblems are simultaneously solved in a single run to approximate the set of efficient solutions. In MOEA/D, the neighborhood relations among these subproblems are defined based on the distances between their weight vectors. The optimal solutions of two

neighboring subproblems should be very similar. Each subproblem is optimized by using information only from its neighboring subproblems.

NSGA-II, proposed by Deb et al. (2002), starts by generating an initial population of solutions that will be sorted based on non-domination. To sort each solution, a rank (fitness) value is assigned based on the front in which they belong to. Then, a crowding distance is assigned to each solution in the front. The crowding distance is a measure related to the density of solutions around each solution, i.e., how close a solution is to its neighbors. Next, individual solutions are selected by using a binary tournament selection with crowded-comparison-operator. The selected population generates offspring from the crossover and mutation operators, and this new population with the current population and current offsprings is sorted again based on non-domination and only the best N individuals are selected, being N the population size. The selection is based on rank and on crowding distance on the last front.

SPEA2 proposed by Zitzler et al. (2001) is an improvement version of the SPEA that resolves its weakness: fitness assignment, density estimation and archive truncation. SPEA2 uses a regular population and an external archive to keep the elite solutions. The algorithm starts by generating an initial population of solutions and an empty archive. Next, all non-dominated population solutions are copied to the archive and any dominated solutions or duplicates are removed from the archive. If the size of the archive exceeds a predefined limit, further archive members are deleted by a clustering technique which preserves the characteristics of the non-

dominated front. Afterward, fitness values are assigned to both archive and population members and again individual solutions are selected by using a binary tournament selection. The selected population generates offspring from the crossover and mutation operators.

Note that the solution representation for these three evolutionary algorithms consists of an integer array of p elements, where each element indicates the selected facility (notice that the order of the selected facilities is not relevant). These algorithms require one main input parameter, the population size. This input parameters have been tuned with the aim of having a fair comparison in terms of computing time, resulting in a population size of 100 solutions. The stopping criterion for these three evolutionary algorithms is set to 3600s. Finally, the remaining parameters have been set to the default and recommended values by the MOEA Framework library used to implement these algorithms (Hadka 2015).

The Scatter Search (SS) procedure reported in Colmenar et al. (2018) is another evolutionary algorithm that starts by randomly constructing a set of solutions to generate the initial population. Then, they are improved with three different local search methods, which are based on the first improvement strategy. The reference set update method maintains a set of not only the best solutions (in terms of the objective function values) but also according to their diversity. The subset generation method and the combination method considered in this Scatter Search follow a typical implementation.

Finally, it has been included an exact procedure based on the ϵ -C strategy, with the aim of providing a near-optimal approximation of the Pareto front. Since this algorithm is very computationally demanding, the computational effort required to generate the optimal Pareto front makes it not suitable for this kind of problems, but it can be used to analyze how far from the best Pareto front are the proposed algorithms in a short computing time. The ϵ -C method used in this work has been adapted from Sayyady et al. (2015) but including a time limit of 7200s and, therefore, not all instances are solved to optimality.

Now, the results obtained by the algorithms are shown in Table 6 that summarizes the average results provided when solving the whole set of instances of the OR-library. In particular, the instances are grouped by the number of nodes in steps of 100. The summary results presented in the last row prove the superiority of our proposal (RPR) over the five analyzed algorithms (MOEA/D, NSGA-II, SPEA 2, SS and ϵ -C). It can be observed that the RPR is the algorithm that reaches the largest number of efficient solutions on average. Specifically, almost 32 efficient solutions, followed by ϵ -C which is able to find more than 17 efficient solutions on average. The coverage indicates that the percentage of solutions of the best Pareto front, B , that dominates the solutions from the Pareto front obtained by the RPR is only 13%, meanwhile that percentage is more than 95% for the MOEA/D, NSGA-II

and SPEA2, more than 97% for the Scatter Search, and more than 75% for the ϵ -C method, which demonstrates that most of the solutions of algorithm RPR are part of the best Pareto front B . To measure the extent of the spread achieved by the Pareto front approximation obtained by each algorithm, the Scatter Search finds the best value 0.99 and RPR obtains 0.96, but both values are very similar. On the other hand, the average size of the space covered by the RPR is 70.08%. Regarding the average ϵ -C, the smallest distance needed to transform every solution of the Pareto front obtained by the RPR in the closest solution of the best Pareto front is 0.03 and the average values of the ϵ -C for the remaining algorithms are much more larger. Finally, RPR obtains the best values for the GD and IGD, which indicate how far the solutions are from the best Pareto front.

If we now analyze the results depending on the instance size, we can clearly see that the time required by RPR scales better with the size, while the other algorithms quickly reach the time limit. Regarding the other metrics, the behavior is similar independently of the instance size, emerging RPR as the most competitive algorithm.

Furthermore, in order to facilitate future comparisons, all Pareto fronts of $D250$ and $D350$ instances are available in Fig. 5 and Fig. 6, and all individual results per instance are detailed in Tables 10–17 of “Appendix A”.

The second set of instances considered in this paper are those reported in Colmenar et al. (2018), that is, set $D250$ and $D350$ with 9 and 10 instances, respectively. The associated average results are shown in Table 7. Analyzing the results, these instances seem to be easier to solve for the exact procedure ϵ -C, so they can be used to illustrate how far from the best approximation of the Pareto Front are the heuristic algorithms included in this comparison. Although in terms of coverage and hypervolume value, ϵ -C algorithm is able to obtain consistently better values, it is worth mentioning that the number of solutions in the non-dominated set obtained by RPR is considerably larger than the ones obtained by the other algorithms, including ϵ -C. Furthermore, half of the solutions provided by the RPR are included in the best approximation of the Pareto front regarding the coverage metric. Additionally, the hypervolume obtained by the RPR is very similar to the one obtained by ϵ -C, which is also an indicator of the quality of the front obtained by RPR. Regarding the metrics that measure the distance to the closest solution in the best approximation front, ϵ -indicator, GD, and IGD, it can be seen that RPR presents the best values, indicating that, in those cases in which its solutions are covered by the best approximation front, they are still really close to the best solution. Indeed, RPR presents better results than ϵ -C in this metrics, since when ϵ -C is not able to find the best solution, it remains far from it due to the constraint on the computational time. Finally, the computing time produced by RPR is an order of magnitude lower than the one required by ϵ -C, highlight-

Table 6 Comparison among all the algorithms with the pmed testbed

Instance	Algorithm	$ A $	$C(A, B)$	Δ	HV	ϵ	GD	IGD	CPU
100 (5)	MOEA/D	19.00	0.98	1.05	0.39	0.39	1097.84	4118.11	3600.00
	NSGA-II	31.00	0.94	0.98	0.42	0.38	892.52	4105.91	3600.00
	SPEA2	32.20	0.95	0.98	0.43	0.36	868.58	4109.11	3600.00
	SS	9.20	1.00	0.99	0.28	0.62	1529.14	4210.91	750.00
	ϵ -C	16.80	0.86	0.96	0.47	0.26	1299.29	4056.51	1531.54
	RPR	34.00	0.07	0.97	0.72	0.04	845.96	3725.91	7.83
200 (5)	MOEA/D	12.00	1.00	1.05	0.29	0.65	1548.26	4797.16	3600.00
	NSGA-II	29.20	0.96	1.00	0.38	0.49	1126.30	4817.76	3600.00
	SPEA2	27.20	0.99	0.99	0.38	0.53	1137.47	4803.16	3600.00
	SS	9.40	0.98	0.99	0.22	0.79	1704.11	4828.96	750.00
	ϵ -C	19.60	0.71	0.91	0.58	0.15	1435.12	4712.76	1531.54
	RPR	40.60	0.12	0.97	0.75	0.04	878.15	4394.36	69.42
300 (5)	MOEA/D	10.40	0.98	1.05	0.27	0.66	1814.80	5154.93	3600.00
	NSGA-II	20.60	0.97	0.98	0.31	0.60	1717.87	5186.13	3600.00
	SPEA2	21.00	0.98	0.99	0.29	0.63	1595.33	5180.53	3600.00
	SS	8.80	1.00	1.00	0.16	0.74	2159.51	5533.33	750.00
	ϵ -C	19.60	0.72	0.93	0.53	0.26	1494.28	5090.53	2447.38
	RPR	39.60	0.10	0.97	0.73	0.03	1030.23	4694.13	305.09
400 (5)	MOEA/D	6.20	0.96	1.05	0.19	0.76	2743.31	5411.69	3600.00
	NSGA-II	17.20	0.96	0.99	0.25	0.69	2249.31	5425.09	3600.00
	SPEA2	18.00	0.96	0.98	0.25	0.70	2022.47	5421.69	3600.00
	SS	6.40	0.97	0.99	0.13	0.89	2521.49	5467.89	750.00
	ϵ -C	17.20	0.75	0.93	0.49	0.19	1800.94	5322.89	4001.62
	RPR	31.20	0.18	0.95	0.69	0.03	1239.91	4948.49	804.20
500 (5)	MOEA/D	4.60	0.93	1.04	0.19	0.83	3392.10	5971.73	3600.00
	NSGA-II	16.60	0.98	0.97	0.27	0.70	2142.81	5950.73	3600.00
	SPEA2	14.80	0.94	0.98	0.26	0.74	2401.71	5967.13	3600.00
	SS	6.40	0.97	0.99	0.16	0.92	2920.58	6281.13	750.00
	ϵ -C	18.00	0.75	0.96	0.49	0.26	1811.09	5866.33	4610.01
	RPR	27.00	0.12	0.98	0.71	0.03	1402.81	5455.53	1954.18
600 (5)	MOEA/D	4.80	0.93	1.04	0.18	0.84	3349.03	6071.69	3600.00
	NSGA-II	15.80	0.95	0.98	0.25	0.73	2511.50	6076.89	3600.00
	SPEA2	14.80	0.96	0.98	0.23	0.77	2465.68	6074.69	3600.00
	SS	5.40	1.00	1.00	0.12	0.94	2982.91	6108.49	750.00
	ϵ -C	16.40	0.79	0.96	0.48	0.25	2043.23	5998.49	4383.84
	RPR	26.20	0.09	0.95	0.69	0.03	1530.84	5598.89	4006.49
700 (4)	MOEA/D	4.75	1.00	1.04	0.20	0.74	4197.21	7074.90	3600.00
	NSGA-II	16.50	0.95	0.98	0.32	0.59	2630.93	7094.65	3600.00
	SPEA2	17.75	0.93	0.98	0.31	0.60	3104.33	7108.90	3600.00
	SS	6.50	0.95	0.99	0.19	0.85	3824.58	7204.15	750.00
	ϵ -C	15.75	0.74	0.94	0.51	0.21	2425.16	6985.40	5672.23
	RPR	26.25	0.20	0.93	0.65	0.02	1879.80	6817.40	3557.06
800 (3)	MOEA/D	6.00	0.90	1.05	0.24	0.66	4003.54	8653.99	3600.00
	NSGA-II	20.33	0.89	0.94	0.39	0.49	2884.31	8700.66	3600.00
	SPEA2	17.33	0.94	0.96	0.32	0.59	3672.50	8682.32	3600.00
	SS	7.67	0.93	0.98	0.19	0.79	4147.74	8690.99	750.00

Table 6 continued

Instance	Algorithm	A	C(A, B)	Δ	HV	ϵ	GD	IGD	CPU
900 (3)	ϵ -C	22.00	0.75	0.95	0.57	0.10	2756.42	8561.66	6687.81
	RPR	32.00	0.15	0.97	0.63	0.03	2360.79	8480.99	2049.54
	MOEA/D	5.33	1.00	1.04	0.28	0.63	4328.03	8735.03	3600.00
	NSGA-II	19.00	0.92	0.95	0.41	0.48	2914.03	8769.37	3600.00
	SPEA2	15.00	0.87	0.97	0.35	0.58	3031.94	8756.70	3600.00
	SS	6.67	0.94	0.99	0.22	0.79	3904.50	8804.37	750.00
Summary (40)	ϵ -C	10.33	0.62	0.94	0.58	0.20	3551.73	8631.37	5044.19
	RPR	27.33	0.22	0.92	0.69	0.03	2383.08	8588.70	3217.34
	MOEA/D	8.45	0.96	1.05	0.25	0.69	2787.76	5952.33	3600.00
	NSGA-II	20.90	0.95	0.98	0.33	0.58	2028.01	5965.03	3600.00
	SPEA2	20.20	0.95	0.98	0.31	0.61	2124.67	5963.36	3600.00
	SS	7.43	0.97	0.99	0.18	0.82	2713.59	6086.41	750.00
	ϵ -C	17.45	0.75	0.94	0.52	0.22	1951.12	5868.96	3579.70
	RPR	31.90	0.13	0.96	0.70	0.03	1409.76	5564.13	1644.12

Table 7 Comparison among all the algorithms with the testbeds D_250 and D_350

Instance	Algorithm	A	C(A, B)	Δ	HV	ϵ	GD	IGD	CPU
250 (9)	MOEA/D	12.11	0.95	1.01	0.47	0.46	354.95	1190.80	3600.00
	NSGA-II	17.89	0.93	0.99	0.50	0.43	302.61	1198.46	3600.00
	SPEA2	21.22	0.96	0.99	0.56	0.34	275.97	1201.35	3600.00
	SS	8.78	0.94	0.98	0.48	0.32	464.41	1295.36	750.00
	ϵ -C	8.11	0.00	0.99	0.75	0.07	439.13	1178.58	950.43
	RPR	39.22	0.65	0.98	0.73	0.08	210.10	1188.46	150.60
350 (10)	MOEA/D	9.27	0.96	1.00	0.38	0.58	525.77	1543.79	3600.00
	NSGA-II	20.36	0.98	0.99	0.42	0.52	364.07	1551.16	3600.00
	SPEA2	22.55	0.99	0.99	0.47	0.46	339.75	1554.25	3600.00
	SS	8.91	0.90	0.98	0.43	0.35	602.57	1695.21	1650.00
	ϵ -C	7.82	0.00	0.98	0.74	0.05	583.48	1526.89	2022.16
	RPR	48.64	0.69	0.98	0.71	0.08	245.15	1537.16	608.44
Summary (19)	MOEA/D	10.55	0.95	1.01	0.42	0.52	448.90	1384.95	3600.00
	NSGA-II	19.25	0.95	0.99	0.46	0.48	336.41	1392.45	3600.00
	SPEA2	21.95	0.97	0.99	0.51	0.41	311.05	1395.45	3600.00
	SS	8.85	0.91	0.98	0.45	0.33	540.40	1515.27	1220.36
	ϵ -C	7.95	0.00	0.98	0.75	0.06	518.52	1370.15	1516.61
	RPR	44.40	0.67	0.98	0.72	0.08	229.38	1380.25	391.57

Table 8 Comparison among all the algorithms with the testbeds kmedian where the ϵ -C is able to obtain solutions

	A	C(A, B)	Δ	HV	ϵ	GD	IGD	CPU
MOEA/D	3.00	0.38	1.00	0.29	0.70	224146.38	359819.45	3600.00
NSGA-II	4.17	0.58	1.00	0.32	0.66	191338.38	362277.95	3600.00
SPEA2	4.67	0.69	1.00	0.39	0.57	197144.20	363908.95	3600.00
SS	1.67	0.83	1.00	0.14	0.82	317525.50	410873.28	3600.00
ϵ -C	5.33	0.63	1.09	0.32	1.22	294216.00	417664.28	3600.00
RPR	7.83	0.26	1.00	0.31	0.66	147239.65	401851.45	3600.00

Table 9 Comparison among all the algorithms with the testbeds kmedian

Instance	Algorithm	$ A $	$C(A, B)$	Δ	HV	ϵ	GD	IGD	CPU
1000 (3)	MOEA/D	3.67	0.08	1.00	0.27	0.72	173941.79	296555.84	3600.00
	NSGA-II	5.33	1.00	1.00	0.16	0.82	142943.16	300331.51	3600.00
	SPEA2	6.67	0.33	1.00	0.34	0.62	133090.68	300326.84	3600.00
	SS	1.67	1.00	1.00	0.05	0.88	268572.93	340187.84	3600.00
	RPR	9.33	0.14	1.00	0.22	0.72	108241.74	331050.51	3600.00
2000 (3)	MOEA/D	2.33	0.42	0.99	0.11	0.89	274350.97	423082.80	3600.00
	NSGA-II	3.00	0.17	1.00	0.28	0.71	239733.61	424224.13	3600.00
	SPEA2	2.67	0.72	1.00	0.29	0.68	261197.72	427490.80	3600.00
	SS	1.67	0.67	1.00	0.05	0.92	366478.06	481558.46	3600.00
	RPR	6.33	0.18	1.00	0.15	0.83	186237.55	472652.13	3600.00
3000 (3)	MOEA/D	2.33	0.33	1.00	0.07	0.94	351719.25	522714.48	3600.00
	NSGA-II	1.33	0.67	1.00	0.19	0.79	440792.89	524267.81	3600.00
	SPEA2	1.33	0.33	1.00	0.19	0.80	440973.44	525090.81	3600.00
	SS	1.33	0.67	1.00	0.01	0.96	493445.71	586263.48	3600.00
	RPR	4.33	0.23	1.00	0.09	0.89	283194.81	581405.81	3600.00
4000 (3)	MOEA/D	2.33	0.42	0.99	0.16	0.86	427414.82	627533.45	3600.00
	NSGA-II	3.00	0.56	1.00	0.25	0.77	396050.94	640942.12	3600.00
	SPEA2	5.00	0.60	1.00	0.24	0.77	307279.63	645066.12	3600.00
	SS	1.67	0.67	0.67	0.01	0.70	508217.08	554933.04	3600.00
	RPR	4.67	0.11	1.00	0.04	0.96	317071.74	673989.45	3600.00
5000 (3)	MOEA/D	2.67	0.67	1.00	0.04	7.21	442607.60	731,899.31	3600.00
	NSGA-II	1.67	0.67	1.00	0.01	8.61	591911.87	746946.64	3600.00
	SPEA2	2.67	1.00	1.00	0.00	8.44	527963.92	746332.64	3600.00
	SS	1.33	0.33	1.00	0.02	0.97	632507.01	751754.64	3600.00
	RPR	5.00	0.11	1.00	0.03	0.96	333125.13	752550.64	3600.00
Summary (15)	MOEA/D	2.67	0.38	1.00	0.13	2.12	334006.89	520357.18	3600.00
	NSGA-II	2.87	0.61	1.00	0.18	2.34	362286.49	527342.44	3600.00
	SPEA2	3.67	0.60	1.00	0.21	2.26	334101.08	528861.44	3600.00
	SS	1.53	0.67	0.93	0.03	0.89	453844.16	542939.49	3600.00
	RPR	5.93	0.16	1.00	0.11	0.87	245574.20	562329.71	3600.00

ing the suitability of RPR for providing fast and high-quality solutions for the tackled problem. A similar behavior to those obtained in the previous table can be emphasized, the scalability of RPR when analyzing the computing time. In the case of the exact procedure, the increase in computing time makes it unaffordable for solving larger instances.

With the aim of testing the algorithm when dealing with a more challenging set of instances, we have included the kmedian testbed, where the number of nodes ranges from 1000 to 5000. This experiment is divided into two different tables. On the one hand, Table 8 includes all the instances for which the ϵ -C algorithm is able to finish, even without guaranteeing optimality, since the time limit has been set to 3600 s. On the other hand, Table 9 shows the results obtained in those instances in which the exact procedure ϵ -C is not even able to start since the memory requirements are too high.

Analyzing the results in Table 8, the ϵ -C is only able to provide solutions for 6 out of 15 of the kmedian instances. These instances seem to be harder to solve, since the number of solutions $|A|$ is close to 5, being RPR able to generate a more populated set of non-dominated solutions. Additionally, RPR is able to reach the smallest value of coverage, indicating that it is able to reach the largest number of solutions belonging to the reference set of non-dominated solutions. In terms of hypervolume, RPR also provides competitive results, as well as in the remaining metrics in the considered time horizon of 3600 s.

Table 9 shows the results where ϵ -C is not considered, since it is not able even to generate a single feasible solution. As it can be seen, results are rather similar than the previous ones. However, it is important to remark that, in this case, RPR is able to obtain the best results in more metrics, highlighting the scalability of the proposal. In par-

ticular, RPR is able to generate, on average, approximately 6 solutions in each non-dominated set, while the remaining methods are not able to reach 4 solutions. The value of 0.16 in the coverage metric indicates that most of the solutions provided by RPR belongs to the reference set, and the value of ϵ -indicator or GD, for instance, suggests that the non-dominated set of solutions of RPR are better distributed along the search space. Analyzing the results per instance size, RPR maintains similar performance in all the metrics, even improving the ϵ -indicator metric while increasing the size of the instance.

5 Conclusions

Nowadays, most of real-world optimization problems are modeled considering more than one objective at a time and usually those objectives are in conflict among them, being impossible to improve one objective without deteriorating at least one of the other. Facility locations are interesting combinatorial optimization problems that are being recently addressed from a multi-objective perspective. This manuscript is focused on the Bi-objective p -Median p -Dispersion problem ($BpMD$ problem) that seeks to minimize the total distance between facilities and demand points and, at the same time to maximize the minimum distance between all pairs of facilities.

To address the $BpMD$ problem, a new path relinking strategy has been proposed. The algorithm combines two different approaches: interior and exterior. The reactive path relinking has been designed in an efficient way and is able to decide which should use depending on the pairs of solutions to combine. This algorithm is able to find high-quality Pareto front approximations in short computing times, becoming a competitive algorithm when comparing the state of the art. Furthermore, computational experiments show the effectiveness of our proposed algorithm for solving relatively large instances of this bi-objective optimization problem in comparison with the other considered algorithms: MOEA/D, NSGA-II, SPEA2, Scatter Search and ϵ -C method, being able to obtain better Pareto fronts.

Therefore, our results establish the first benchmarks not only for this problem but also for the applications of this novel proposed methodology, the reactive path relinking, to other multi-objective combinatorial optimization problems.

A future line of research is to adapt the reactive path relinking algorithm to solve other combinatorial optimization problems, whether single-objective or multi-objective nature. A new strategy to prove would be the adaptive reactive path relinking that will allow the algorithm to decide the k parameter to consider to combine every pair of solutions according their similarity or dissimilarity. Furthermore, it would be interesting to test another constructive metaheuristic

as GRASP that includes randomization when selecting one node from a restricted candidate list containing the most promising nodes. Another interesting methodology to test would be the Fixed Set Search (FSS) algorithm, see Jovanovic et al. (2022), where a population of solutions is generated with GRASP and the elements appearing with more frequency in a subset of the most promising solutions is fixed for the next generation of solutions. As the authors hold, the FSS can be viewed as a metaheuristic that adds a learning mechanism to the GRASP.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. J. Sánchez-Oro, A. Duarte and I. Lozano-Osorio acknowledge the support from the Spanish Ministry of “Ciencia, Innovación y Universidades” (MCIN/AEI/10.13039/501100011033/FEDER, UE) under grant ref. PID2021-126605NB-I00 and PID2021-125709OA-C22, the “Comunidad de Madrid” and “Fondos Estructurales” of the European Union with grant reference S2018/TCS-4566. A.D. López-Sánchez acknowledges support from the Spanish Ministry of “Economía, Industria y Competitividad” through Projects PID2019-104263RB-C41, and from Junta de Andalucía, FEDER-UPO Research & Development Call, reference number UPO-1263769.

Data availability Enquiries about data availability should be directed to the authors and publicly available to ease further comparisons in <https://grafo.etsii.urjc.es/BpMpD>.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Human and animals rights This article does not contain any studies with human participants performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A

See Figs. 5, 6 and Tables 10, 11, 12, 13, 14, 15, 16 and 17.

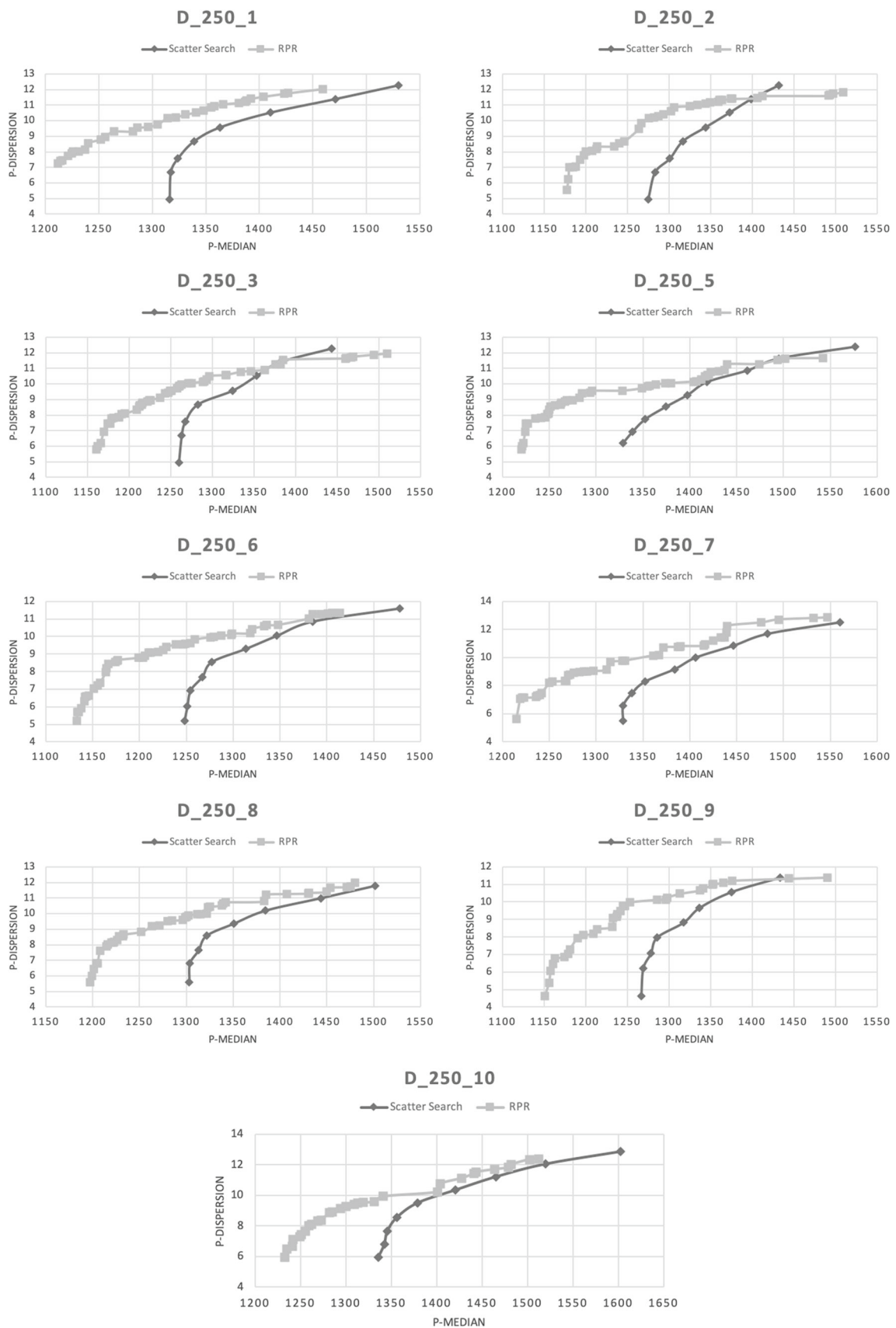


Fig. 5 Pareto fronts obtained by scatter search and RPR

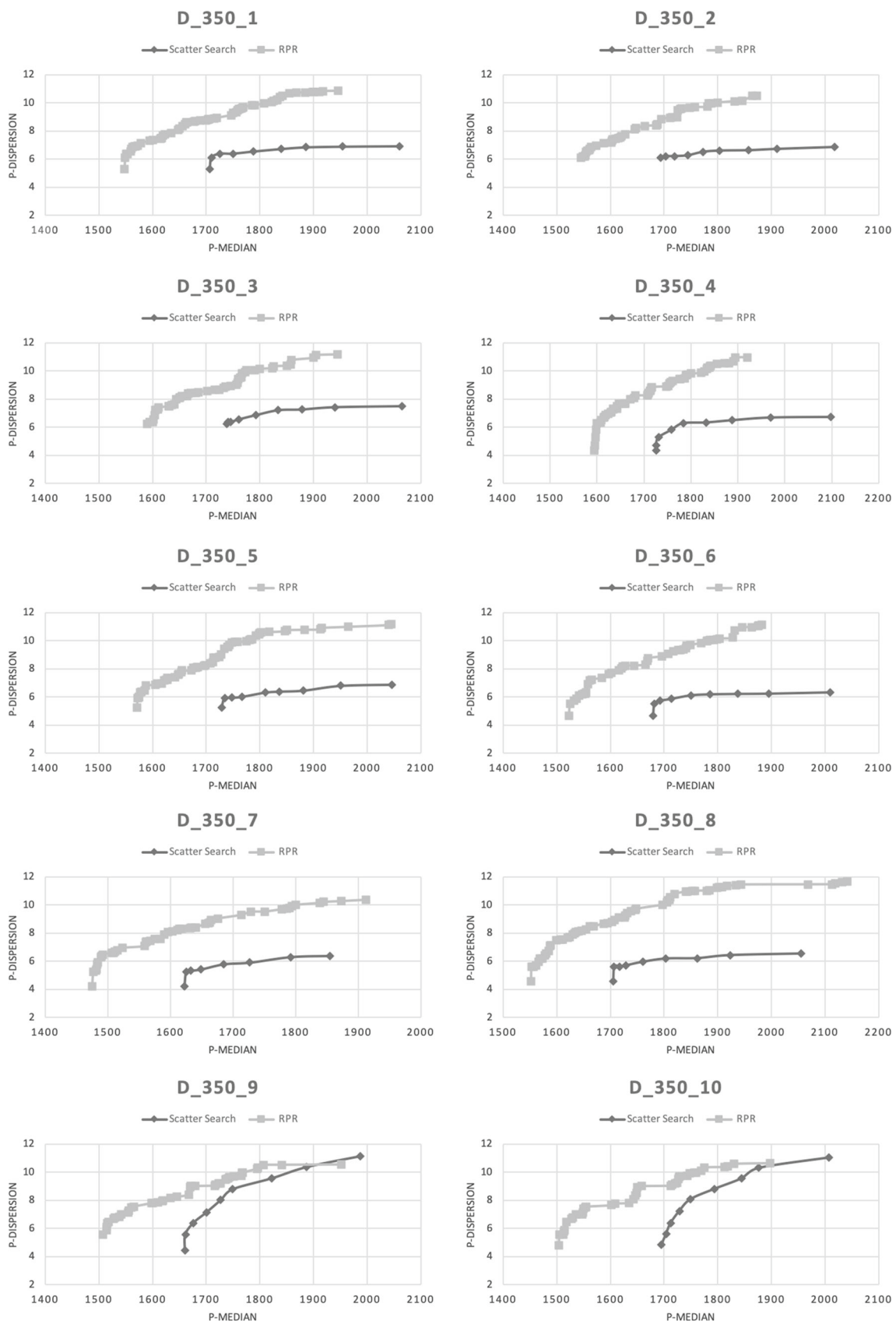


Fig. 6 Pareto fronts obtained by scatter search and RPR

Table 10 Number of efficient solutions in the Pareto front. Detailed results for every instance and every algorithm

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed1	16	40	42	10	17	30
pmed2	22	35	41	12	19	50
pmed3	26	38	37	11	19	33
pmed4	16	23	23	6	17	35
pmed5	15	19	18	7	12	22
pmed6	23	65	59	18	22	62
pmed7	20	45	36	12	24	54
pmed8	7	19	25	10	14	40
pmed9	6	9	11	4	22	28
pmed10	4	8	5	3	16	19
pmed11	20	51	51	15	19	54
pmed12	14	30	31	11	19	54
pmed13	10	16	16	11	17	47
pmed14	5	2	3	3	24	28
pmed15	3	4	4	4	19	15
pmed16	14	46	53	12	25	49
pmed17	11	33	27	13	21	43
pmed18	3	3	5	2	17	31
pmed19	2	2	3	3	13	21
pmed20	1	2	2	2	10	12
pmed21	8	43	44	14	18	40
pmed22	10	29	21	10	21	40
pmed23	3	4	5	4	23	26
pmed24	1	5	3	3	17	18
pmed25	1	2	1	1	11	11
pmed26	11	46	42	14	22	42
pmed27	7	24	25	7	14	42
pmed28	4	6	3	2	14	22
pmed29	1	1	2	2	17	15
pmed30	1	2	2	2	15	10
pmed31	12	34	43	14	16	35
pmed32	5	27	25	10	18	37
pmed33	1	3	1	1	16	19
pmed34	1	2	2	1	13	14
pmed35	10	37	35	13	26	40
pmed36	5	21	16	9	21	37
pmed37	3	3	1	1	19	19
pmed38	7	34	28	12	13	34

Table 10 continued

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed39	7	20	14	6	9	31
pmed40	2	3	3	2	9	17
Average	8.45	20.90	20.20	7.43	17.45	31.90
D_250_1	17	15	22	8	8	32
D_250_2	10	25	24	9	7	37
D_250_3	10	17	14	9	8	49
D_250_5	12	21	26	9	8	45
D_250_6	14	21	22	9	7	46
D_250_7	10	10	15	9	9	40
D_250_8	12	22	24	9	9	42
D_250_9	10	19	22	8	8	32
D_250_10	14	11	22	9	9	30
D_350_1	11	9	23	9	7	66
D_350_2	7	23	26	9	7	42
D_350_3	10	18	19	9	7	46
D_350_4	13	23	24	9	8	54
D_350_5	7	26	28	9	8	57
D_350_6	11	22	24	9	8	46
D_350_7	9	16	24	8	8	46
D_350_8	12	20	19	9	9	59
D_350_9	8	23	19	9	8	41
D_350_10	7	22	21	9	8	39
Average	10.74	19.11	22.00	8.84	7.95	44.68
kmedian1	6	10	11	2	-	12
kmedian2	1	5	8	1	-	11
kmedian3	4	1	1	2	-	5
kmedian4	4	5	4	2	-	7
kmedian5	2	2	3	2	-	7
kmedian6	1	2	1	1	-	5
kmedian7	2	2	2	2	-	5
kmedian8	3	1	1	1	-	6
kmedian9	2	1	1	1	-	2
kmedian10	4	6	10	2	-	9
kmedian11	1	2	2	1	-	3
kmedian12	2	1	3	2	-	2
kmedian13	4	3	6	2	-	8
kmedian14	2	1	1	1	-	3
kmedian15	2	1	1	1	-	4
Average	2.67	2.87	3.67	1.53	-	5.93

Table 11 Coverage metric. Detailed results for every instance and every algorithm

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed1	0.88	0.70	0.74	1.00	0.53	0.33
pmed2	1.00	1.00	1.00	1.00	0.89	0.00
pmed3	1.00	1.00	1.00	1.00	0.89	0.03
pmed4	1.00	1.00	1.00	1.00	1.00	0.00
pmed5	1.00	1.00	1.00	1.00	1.00	0.00
pmed6	1.00	0.80	0.93	0.89	0.18	0.53
pmed7	1.00	1.00	1.00	1.00	0.71	0.09
pmed8	1.00	1.00	1.00	1.00	0.79	0.00
pmed9	1.00	1.00	1.00	1.00	0.95	0.00
pmed10	1.00	1.00	1.00	1.00	0.94	0.00
pmed11	0.90	0.94	0.90	1.00	0.37	0.24
pmed12	1.00	0.90	1.00	1.00	0.32	0.28
pmed13	1.00	1.00	1.00	1.00	1.00	0.00
pmed14	1.00	1.00	1.00	1.00	0.96	0.00
pmed15	1.00	1.00	1.00	1.00	0.95	0.00
pmed16	0.79	0.87	0.91	1.00	0.44	0.47
pmed17	1.00	0.91	0.89	0.85	0.48	0.40
pmed18	1.00	1.00	1.00	1.00	0.94	0.03
pmed19	1.00	1.00	1.00	1.00	1.00	0.00
pmed20	1.00	1.00	1.00	1.00	0.90	0.00
pmed21	0.75	0.93	0.77	0.93	0.50	0.33
pmed22	0.90	0.97	0.90	0.90	0.43	0.28
pmed23	1.00	1.00	1.00	1.00	0.91	0.00
pmed24	1.00	1.00	1.00	1.00	1.00	0.00
pmed25	1.00	1.00	1.00	1.00	0.91	0.00
pmed26	0.64	0.74	0.93	1.00	0.68	0.29
pmed27	1.00	1.00	0.88	1.00	0.43	0.10
pmed28	1.00	1.00	1.00	1.00	0.93	0.05
pmed29	1.00	1.00	1.00	1.00	1.00	0.00
pmed30	1.00	1.00	1.00	1.00	0.93	0.00
pmed31	1.00	0.85	0.74	1.00	0.69	0.37
pmed32	1.00	0.96	0.96	0.80	0.39	0.43
pmed33	1.00	1.00	1.00	1.00	0.88	0.00
pmed34	1.00	1.00	1.00	1.00	1.00	0.00
pmed35	0.90	0.76	0.89	1.00	0.69	0.28
pmed36	0.80	0.90	0.94	0.78	0.62	0.16
pmed37	1.00	1.00	1.00	1.00	0.95	0.00

Table 11 continued

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed38	1.00	0.76	0.82	0.83	0.85	0.32
pmed39	1.00	1.00	0.79	1.00	0.11	0.29
pmed40	1.00	1.00	1.00	1.00	0.89	0.06
Average	0.96	0.95	0.95	0.97	0.75	0.13
D_250_1	1.00	1.00	1.00	0.88	0.00	0.63
D_250_2	1.00	0.76	1.00	0.89	0.00	0.51
D_250_3	0.70	0.88	0.93	1.00	0.00	0.41
D_250_5	1.00	1.00	0.96	1.00	0.00	0.71
D_250_6	1.00	1.00	1.00	0.89	0.00	0.57
D_250_7	1.00	0.70	0.87	1.00	0.00	0.73
D_250_8	0.83	1.00	1.00	0.89	0.00	0.81
D_250_9	1.00	1.00	0.95	0.88	0.00	0.59
D_250_10	1.00	1.00	0.91	1.00	0.00	0.87
D_350_1	1.00	1.00	1.00	0.89	0.00	0.58
D_350_2	1.00	1.00	0.96	1.00	0.00	0.60
D_350_3	1.00	1.00	1.00	0.89	0.00	0.54
D_350_4	1.00	1.00	1.00	0.89	0.00	0.91
D_350_5	1.00	1.00	1.00	0.89	0.00	0.61
D_350_6	1.00	1.00	1.00	0.89	0.00	0.59
D_350_7	0.67	0.75	0.92	0.88	0.00	0.80
D_350_8	1.00	1.00	1.00	0.89	0.00	0.69
D_350_9	0.88	1.00	1.00	0.89	0.00	0.76
D_350_10	1.00	1.00	1.00	0.89	0.00	0.77
Average	0.95	0.95	0.97	0.92	0.00	0.67
kmedian1	0.00	1.00	0.73	1.00	–	0.25
kmedian2	0.00	1.00	0.25	1.00	–	0.18
kmedian3	0.25	1.00	0.00	1.00	–	0.00
kmedian4	0.25	0.00	0.50	1.00	–	0.14
kmedian5	1.00	0.00	0.67	1.00	–	0.00
kmedian6	0.00	0.50	1.00	0.00	–	0.40
kmedian7	1.00	0.00	1.00	1.00	–	0.20
kmedian8	0.00	1.00	0.00	1.00	–	0.00
kmedian9	0.00	1.00	0.00	0.00	–	0.50
kmedian10	0.25	0.17	0.80	1.00	–	0.33
kmedian11	0.00	0.50	0.00	1.00	–	0.00
kmedian12	1.00	1.00	1.00	0.00	–	0.00
kmedian13	1.00	0.00	1.00	1.00	–	0.00
kmedian14	0.00	1.00	1.00	0.00	–	0.33
kmedian15	1.00	1.00	1.00	0.00	–	0.00
Average	0.38	0.61	0.60	0.67	–	0.16

Table 12 Spread metric. Detailed results for every instance and every algorithm

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed1	1.03	0.99	0.99	0.99	0.91	0.95
pmed2	1.12	0.96	0.98	0.98	0.93	0.99
pmed3	1.11	0.99	0.97	1.01	1.00	0.95
pmed4	1.01	0.97	0.98	0.98	0.97	0.97
pmed5	0.97	0.98	0.97	0.99	0.97	0.98
pmed6	1.14	0.98	0.92	0.99	0.89	0.95
pmed7	1.11	0.98	1.01	0.99	0.93	1.03
pmed8	0.98	1.03	1.01	0.99	0.88	0.96
pmed9	1.01	1.03	1.00	1.00	0.91	0.95
pmed10	1.00	1.00	1.00	1.00	0.92	0.97
pmed11	1.20	0.96	0.99	0.99	0.98	0.98
pmed12	1.06	0.96	0.97	0.99	0.91	0.96
pmed13	1.00	0.99	0.98	1.01	0.89	0.94
pmed14	1.01	1.00	1.00	1.00	0.91	0.97
pmed15	1.00	1.00	1.00	1.00	0.98	0.99
pmed16	1.15	0.98	1.00	0.99	0.95	0.98
pmed17	1.10	0.96	0.93	0.98	0.87	0.91
pmed18	0.99	1.00	1.00	1.00	0.92	0.95
pmed19	1.00	1.00	1.00	1.00	0.95	0.94
pmed20	1.00	1.00	1.00	1.00	0.96	0.97
pmed21	1.08	0.92	0.94	0.94	0.97	0.92
pmed22	1.10	0.97	0.98	1.01	0.92	0.96
pmed23	1.00	0.99	1.00	1.00	1.00	0.98
pmed24	1.00	1.00	1.00	1.00	0.96	1.01
pmed25	1.00	0.96	1.00	1.00	0.94	1.04
pmed26	1.14	0.95	0.97	1.00	1.04	0.93
pmed27	1.05	0.94	0.91	0.98	0.93	0.90
pmed28	1.00	1.00	1.00	1.00	0.93	0.93
pmed29	1.00	1.00	1.00	1.00	0.95	0.97
pmed30	1.00	1.00	1.00	1.00	0.96	1.02
pmed31	1.16	0.96	0.92	0.96	0.94	0.96
pmed32	1.01	0.97	0.98	1.02	0.93	0.89
pmed33	1.00	1.01	1.00	1.00	0.91	0.93
pmed34	1.00	1.00	1.00	1.00	0.97	0.93
pmed35	1.14	0.93	0.96	0.95	0.96	0.97
pmed36	1.02	0.90	0.93	0.99	0.93	0.94
pmed37	1.00	1.00	1.00	1.00	0.95	0.98

Table 12 continued

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed38	1.13	0.93	0.92	0.99	0.98	0.92
pmed39	1.06	0.91	0.98	0.99	0.91	0.87
pmed40	0.94	1.00	1.00	0.99	0.94	0.98
Average	1.05	0.98	0.98	0.99	0.94	0.96
D_250_1	1.00	0.99	0.99	0.98	0.99	0.97
D_250_2	1.00	0.99	0.99	0.98	0.95	0.97
D_250_3	1.01	0.99	0.99	0.97	0.99	0.98
D_250_5	1.00	0.99	0.99	0.96	0.98	0.99
D_250_6	1.01	0.99	0.99	0.99	0.99	0.97
D_250_7	1.01	1.00	0.99	0.98	0.99	0.98
D_250_8	1.01	0.99	1.00	0.98	1.00	0.98
D_250_9	1.01	0.99	1.01	0.98	0.99	0.99
D_250_10	0.99	0.99	0.99	0.98	0.99	0.98
D_350_1	1.00	1.00	0.99	0.97	0.97	0.98
D_350_2	1.02	0.99	0.99	0.97	0.97	0.99
D_350_3	1.00	0.99	0.99	0.98	0.98	0.99
D_350_4	1.01	0.99	0.99	0.98	0.99	0.99
D_350_5	1.00	0.99	0.98	0.98	0.97	0.98
D_350_6	1.02	0.99	0.99	0.97	0.98	0.97
D_350_7	1.01	1.00	1.00	0.98	0.99	0.98
D_350_8	1.00	0.99	1.00	0.98	0.99	0.96
D_350_9	1.00	0.99	0.99	0.98	0.98	1.01
D_350_10	1.00	0.99	0.99	0.98	0.98	0.99
Average	1.01	0.99	0.99	0.98	0.98	0.98
kmedian1	1.00	1.00	1.00	0.99	–	1.00
kmedian2	1.00	1.00	1.00	1.00	–	0.99
kmedian3	1.00	1.00	1.00	1.00	–	1.00
kmedian4	1.00	1.00	1.00	0.99	–	1.00
kmedian5	0.98	1.00	1.00	1.00	–	1.00
kmedian6	1.00	1.00	1.00	1.00	–	1.00
kmedian7	1.00	1.00	1.00	1.00	–	1.00
kmedian8	1.00	1.00	1.00	1.00	–	1.00
kmedian9	1.00	1.00	1.00	1.00	–	1.00
kmedian10	1.00	1.00	1.00	1.00	–	1.00
kmedian11	1.00	1.00	1.00	1.00	–	1.00
kmedian12	0.96	1.00	1.00	0.00	–	1.00
kmedian13	1.00	1.00	1.00	1.00	–	1.00
kmedian14	1.00	1.00	1.00	1.00	–	1.00
kmedian15	1.00	1.00	1.00	1.00	–	1.00
Average	1.00	1.00	1.00	0.93	–	1.00

Table 13 Hypervolume metric. Detailed results for every instance and every algorithm

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed1	0.54	0.59	0.59	0.41	0.63	0.63
pmed2	0.51	0.52	0.53	0.38	0.63	0.71
pmed3	0.60	0.67	0.68	0.41	0.73	0.80
pmed4	0.27	0.30	0.32	0.21	0.36	0.70
pmed5	0.01	0.01	0.01	0.00	0.01	0.77
pmed6	0.48	0.63	0.62	0.34	0.64	0.64
pmed7	0.52	0.60	0.55	0.38	0.70	0.72
pmed8	0.32	0.44	0.49	0.37	0.67	0.73
pmed9	0.12	0.21	0.22	0.02	0.57	0.80
pmed10	0.00	0.02	0.00	0.00	0.34	0.87
pmed11	0.54	0.70	0.70	0.43	0.62	0.72
pmed12	0.39	0.46	0.42	0.19	0.54	0.60
pmed13	0.27	0.37	0.30	0.09	0.58	0.68
pmed14	0.14	0.03	0.05	0.08	0.54	0.78
pmed15	0.00	0.00	0.00	0.00	0.37	0.87
pmed16	0.46	0.63	0.63	0.25	0.64	0.65
pmed17	0.45	0.54	0.51	0.34	0.61	0.63
pmed18	0.06	0.08	0.09	0.06	0.56	0.64
pmed19	0.00	0.00	0.00	0.00	0.34	0.67
pmed20	0.00	0.00	0.00	0.00	0.32	0.88
pmed21	0.43	0.66	0.65	0.40	0.55	0.67
pmed22	0.42	0.49	0.45	0.31	0.63	0.67
pmed23	0.10	0.17	0.20	0.11	0.59	0.69
pmed24	0.00	0.04	0.00	0.00	0.39	0.70
pmed25	0.00	0.00	0.00	0.00	0.29	0.84
pmed26	0.47	0.71	0.68	0.37	0.65	0.71
pmed27	0.33	0.43	0.44	0.17	0.48	0.62
pmed28	0.08	0.12	0.04	0.04	0.52	0.62
pmed29	0.00	0.00	0.00	0.00	0.32	0.62
pmed30	0.00	0.00	0.00	0.00	0.42	0.87
pmed31	0.41	0.65	0.65	0.38	0.56	0.67
pmed32	0.36	0.55	0.53	0.33	0.58	0.65
pmed33	0.05	0.09	0.05	0.05	0.56	0.62
pmed34	0.00	0.00	0.00	0.00	0.32	0.64
pmed35	0.40	0.63	0.62	0.36	0.66	0.67
pmed36	0.32	0.43	0.35	0.22	0.54	0.61
pmed37	0.00	0.10	0.00	0.00	0.50	0.62
pmed38	0.33	0.58	0.50	0.27	0.52	0.68
pmed39	0.51	0.58	0.51	0.39	0.67	0.71

Table 13 continued

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed40	0.00	0.06	0.06	0.00	0.56	0.66
Average	0.25	0.33	0.31	0.18	0.52	0.70
D_250_1	0.50	0.44	0.57	0.44	0.77	0.74
D_250_2	0.37	0.57	0.52	0.51	0.75	0.72
D_250_3	0.53	0.55	0.53	0.48	0.74	0.76
D_250_5	0.44	0.46	0.56	0.40	0.72	0.69
D_250_6	0.45	0.51	0.56	0.46	0.74	0.73
D_250_7	0.34	0.41	0.48	0.51	0.73	0.73
D_250_8	0.57	0.51	0.61	0.54	0.78	0.76
D_250_9	0.52	0.57	0.68	0.51	0.80	0.80
D_250_10	0.47	0.50	0.56	0.45	0.72	0.66
D_350_1	0.31	0.34	0.41	0.44	0.72	0.71
D_350_2	0.30	0.34	0.42	0.29	0.65	0.63
D_350_3	0.34	0.40	0.39	0.44	0.75	0.74
D_350_4	0.44	0.44	0.53	0.50	0.77	0.72
D_350_5	0.26	0.34	0.46	0.40	0.67	0.65
D_350_6	0.44	0.50	0.51	0.44	0.75	0.73
D_350_7	0.49	0.57	0.59	0.48	0.81	0.75
D_350_8	0.44	0.39	0.45	0.45	0.72	0.71
D_350_9	0.37	0.50	0.47	0.46	0.77	0.71
D_350_10	0.40	0.43	0.49	0.43	0.79	0.74
Average	0.42	0.46	0.52	0.45	0.75	0.72
kmedian1	0.54	0.36	0.55	0.05	–	0.33
kmedian2	0.00	0.00	0.27	0.00	–	0.20
kmedian3	0.28	0.11	0.19	0.09	–	0.14
kmedian4	0.32	0.48	0.53	0.02	–	0.29
kmedian5	0.02	0.11	0.22	0.00	–	0.05
kmedian6	0.00	0.26	0.13	0.14	–	0.12
kmedian7	0.00	0.20	0.14	0.01	–	0.16
kmedian8	0.15	0.14	0.18	0.00	–	0.11
kmedian9	0.07	0.24	0.25	0.01	–	0.00
kmedian10	0.47	0.56	0.51	0.00	–	0.09
kmedian11	0.00	0.20	0.20	0.00	–	0.02
kmedian12	0.00	0.00	0.00	0.02	–	0.00
kmedian13	0.00	0.03	0.00	0.01	–	0.06
kmedian14	0.11	0.00	0.01	0.05	–	0.01
kmedian15	0.00	0.00	0.00	0.00	–	0.03
Average	0.13	0.18	0.21	0.03	–	0.11

Table 14 ϵ -indicator metric. Detailed results for every instance and every algorithm

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed1	0.16	0.10	0.10	0.43	0.03	0.08
pmed2	0.21	0.37	0.31	0.56	0.08	0.08
pmed3	0.22	0.13	0.15	0.54	0.09	0.04
pmed4	0.51	0.43	0.38	0.66	0.28	0.00
pmed5	0.84	0.85	0.84	0.93	0.80	0.00
pmed6	0.29	0.05	0.06	0.60	0.04	0.05
pmed7	0.37	0.22	0.37	0.59	0.06	0.04
pmed8	0.65	0.49	0.41	0.60	0.08	0.06
pmed9	0.85	0.72	0.72	0.98	0.17	0.02
pmed10	1.09	0.95	1.09	1.19	0.42	0.02
pmed11	0.28	0.04	0.06	0.49	0.30	0.02
pmed12	0.46	0.43	0.50	0.81	0.28	0.05
pmed13	0.70	0.58	0.66	0.50	0.12	0.00
pmed14	0.81	0.97	0.94	0.91	0.17	0.03
pmed15	1.05	1.00	1.00	1.00	0.42	0.05
pmed16	0.30	0.06	0.08	0.73	0.04	0.06
pmed17	0.33	0.30	0.37	0.63	0.05	0.03
pmed18	0.94	0.91	0.91	0.94	0.12	0.01
pmed19	1.10	1.05	1.00	1.00	0.27	0.00
pmed20	1.15	1.15	1.15	1.15	0.47	0.03
pmed21	0.33	0.05	0.05	0.52	0.35	0.03
pmed22	0.42	0.40	0.49	0.67	0.08	0.05
pmed23	0.89	0.81	0.78	0.89	0.13	0.00
pmed24	1.12	0.94	1.00	1.12	0.27	0.00
pmed25	1.38	1.29	1.38	1.38	0.48	0.05
pmed26	0.33	0.08	0.11	0.60	0.14	0.07
pmed27	0.54	0.46	0.46	0.82	0.24	0.02
pmed28	0.91	0.87	0.96	0.96	0.16	0.02
pmed29	1.21	1.14	1.14	1.14	0.28	0.00
pmed30	1.19	1.09	1.19	1.19	0.42	0.06
pmed31	0.37	0.10	0.05	0.57	0.19	0.04
pmed32	0.57	0.29	0.33	0.64	0.17	0.03
pmed33	0.95	0.90	0.95	0.95	0.18	0.02
pmed34	1.08	1.08	1.08	1.23	0.32	0.00
pmed35	0.39	0.10	0.16	0.59	0.03	0.02
pmed36	0.60	0.48	0.60	0.78	0.15	0.04
pmed37	1.00	0.89	1.00	1.00	0.13	0.02
pmed38	0.51	0.21	0.36	0.70	0.38	0.02
pmed39	0.38	0.30	0.45	0.60	0.10	0.03

Table 14 continued

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed40	1.00	0.94	0.94	1.06	0.13	0.04
Average	0.69	0.58	0.61	0.82	0.22	0.03
D_250_1	0.38	0.51	0.29	0.35	0.04	0.06
D_250_2	0.52	0.31	0.39	0.27	0.06	0.07
D_250_3	0.40	0.39	0.41	0.31	0.12	0.04
D_250_5	0.49	0.46	0.31	0.35	0.07	0.08
D_250_6	0.47	0.40	0.35	0.34	0.09	0.06
D_250_7	0.63	0.56	0.47	0.28	0.11	0.09
D_250_8	0.38	0.45	0.30	0.28	0.07	0.07
D_250_9	0.41	0.34	0.21	0.34	0.05	0.07
D_250_10	0.44	0.44	0.36	0.33	0.05	0.13
D_350_1	0.65	0.62	0.55	0.32	0.06	0.05
D_350_2	0.64	0.59	0.51	0.44	0.08	0.09
D_350_3	0.59	0.55	0.57	0.33	0.05	0.06
D_350_4	0.52	0.51	0.38	0.29	0.07	0.07
D_350_5	0.70	0.61	0.44	0.34	0.05	0.11
D_350_6	0.50	0.44	0.41	0.33	0.04	0.07
D_350_7	0.48	0.38	0.34	0.36	0.05	0.10
D_350_8	0.49	0.57	0.51	0.31	0.06	0.07
D_350_9	0.60	0.45	0.47	0.34	0.02	0.11
D_350_10	0.57	0.52	0.44	0.39	0.05	0.07
Average	0.52	0.48	0.40	0.33	0.06	0.08
kmedian1	0.45	0.59	0.38	0.78	–	0.59
kmedian2	1.00	1.00	0.69	0.97	–	0.73
kmedian3	0.70	0.88	0.80	0.88	–	0.84
kmedian4	0.68	0.50	0.42	0.92	–	0.69
kmedian5	0.98	0.89	0.77	1.02	–	0.93
kmedian6	1.00	0.73	0.86	0.82	–	0.86
kmedian7	1.05	0.80	0.85	0.91	–	0.81
kmedian8	0.85	0.85	0.81	0.99	–	0.87
kmedian9	0.93	0.73	0.73	0.98	–	1.00
kmedian10	0.51	0.41	0.39	1.00	–	0.90
kmedian11	1.00	0.77	0.78	1.03	–	0.98
kmedian12	1.06	1.13	1.14	0.07	–	1.00
kmedian13	1.18	0.97	1.18	0.96	–	0.93
kmedian14	0.89	1.03	0.95	0.94	–	0.99
kmedian15	19.56	23.82	23.18	1.00	–	0.96
Average	2.12	2.34	2.26	0.89	–	0.87

Table 15 Generational distance metric. Detailed results for every instance and every algorithm

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed1	1840.31	1276.50	1259.61	2276.92	1939.05	1502.49
pmed2	1085.73	874.16	812.73	1425.89	1389.51	765.18
pmed3	988.04	908.91	873.90	1484.54	1307.97	948.96
pmed4	959.39	843.97	827.27	1579.43	1036.96	637.37
pmed5	615.73	559.08	569.39	878.93	822.97	375.82
pmed6	1939.57	1353.84	1426.20	2145.29	2279.50	1407.37
pmed7	1396.11	1102.49	1084.74	1796.33	1604.92	1001.64
pmed8	1849.87	1175.75	1079.75	1568.02	1682.35	936.99
pmed9	1397.00	1156.68	1058.14	1681.10	916.81	684.37
pmed10	1158.76	842.74	1038.53	1329.83	692.01	360.40
pmed11	2128.70	1628.41	1576.39	2515.04	2079.48	1605.68
pmed12	1976.38	1541.21	1461.15	2094.22	2014.24	1405.10
pmed13	1528.50	1278.95	1261.48	2553.81	1601.56	886.68
pmed14	1719.92	2647.42	2180.86	2146.24	973.84	735.35
pmed15	1720.51	1493.38	1496.75	1488.26	802.29	518.32
pmed16	2700.20	1913.10	1804.43	2678.76	2656.81	1958.24
pmed17	2443.23	1671.92	1801.96	2174.58	2387.76	1667.45
pmed18	2964.91	2979.79	2280.37	3562.42	1660.10	1169.25
pmed19	2475.23	2469.22	2010.92	1991.86	1183.27	785.07
pmed20	3133.00	2212.54	2214.66	2199.81	1116.79	619.54
pmed21	4102.86	2078.16	2152.87	3156.16	2662.03	2285.21
pmed22	3084.42	2065.38	2233.83	2921.87	2874.68	1993.50
pmed23	2849.23	2592.11	2244.13	2443.50	1366.25	1202.49
pmed24	3663.00	1639.59	2117.72	2843.37	1032.65	863.52
pmed25	3261.00	2338.79	3260.00	3238.00	1119.83	669.33
pmed26	3487.64	2140.78	2266.90	3106.65	3080.21	2392.74
pmed27	3595.75	2189.88	2166.63	3348.43	3287.50	2136.92
pmed28	2433.74	1977.82	2773.98	3356.29	1709.65	1368.42
pmed29	3779.00	3801.00	2676.40	2654.48	1107.04	1011.25
pmed30	3449.00	2448.00	2444.47	2448.71	1031.76	744.87
pmed31	3369.68	2465.50	2346.81	3242.81	3743.74	2608.97
pmed32	4615.17	2374.35	2322.69	3302.49	3056.54	2277.78
pmed33	5014.00	2986.95	5064.00	4959.00	1582.95	1560.63
pmed34	3790.00	2696.91	2683.82	3794.00	1317.40	1071.83
pmed35	3856.33	2680.10	2639.29	3563.52	3404.25	2709.75
pmed36	5000.78	2792.59	2939.20	3520.70	3263.21	2696.92
pmed37	3153.50	3180.24	5439.00	5359.00	1601.79	1675.71

Table 15 continued

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed38	4906.76	2728.05	2973.71	3693.78	3697.99	2963.34
pmed39	4034.86	2788.39	2917.82	4123.19	4577.15	2498.70
pmed40	4042.46	3225.66	3204.30	3896.53	2380.05	1687.21
Average	2787.76	2028.01	2124.67	2713.59	1951.12	1409.76
D_250_1	301.02	322.30	274.81	489.62	444.84	231.98
D_250_2	385.58	245.65	252.95	457.27	470.64	214.28
D_250_3	374.87	291.67	323.88	449.26	422.97	182.68
D_250_5	360.17	278.97	253.85	472.35	451.93	198.35
D_250_6	312.02	259.16	255.89	438.12	446.92	183.09
D_250_7	391.97	395.16	330.23	468.02	439.45	211.63
D_250_8	352.72	266.51	257.68	464.86	415.79	201.48
D_250_9	377.71	281.37	260.53	466.82	422.32	223.04
D_250_10	338.49	382.70	273.92	473.40	437.32	244.41
D_350_1	479.62	534.43	335.88	609.02	623.58	214.63
D_350_2	598.78	334.26	314.38	601.41	623.14	258.09
D_350_3	516.85	388.21	373.50	611.60	629.62	253.49
D_350_4	445.29	343.84	341.33	612.62	584.62	234.88
D_350_5	615.44	322.68	318.35	611.83	605.08	228.32
D_350_6	474.54	340.18	324.46	594.99	570.10	248.89
D_350_7	495.69	378.97	313.57	600.79	538.24	240.25
D_350_8	461.35	361.64	365.94	603.13	559.30	228.30
D_350_9	538.30	326.91	359.91	588.53	561.52	259.56
D_350_10	578.79	336.83	344.94	597.17	561.52	265.12
Average	442.06	336.39	309.26	537.41	516.26	227.50
kmedian1	200134.79	157649.17	150661.68	393170.71	–	158578.93
kmedian2	240771.34	108851.12	87021.15	281918.30	–	83224.22
kmedian3	80919.26	162329.19	161589.22	130629.77	–	82922.07
kmedian4	348937.47	313863.41	352431.35	558586.91	–	292549.95
kmedian5	246385.39	243655.99	201040.75	282386.12	–	149342.36
kmedian6	227730.05	161681.43	230121.07	258461.15	–	116820.35
kmedian7	608698.68	604969.57	610557.23	676403.99	–	427152.31
kmedian8	244457.03	426763.05	424318.05	482584.08	–	194814.74
kmedian9	202002.04	290646.03	288045.03	321349.07	–	227617.37
kmedian10	495149.42	406708.23	318972.69	784146.15	–	366940.86
kmedian11	500770.02	358673.59	357924.04	557704.05	–	320,801.74
kmedian12	286325.02	422771.00	244942.15	182801.03	–	263472.64
kmedian13	556465.40	636088.62	452559.75	865121.97	–	432153.26
kmedian14	418772.33	622599.01	617124.00	620031.04	–	358881.35
kmedian15	352585.08	517048.00	514208.00	412368.04	–	208340.77
Average	334006.89	362286.49	334101.08	453844.16	–	245574.20

Table 16 Inverted generational distance metric. Detailed results for every instance and every algorithm

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed1	5910.71	5910.71	5910.71	6015.71	5842.71	5818.71
pmed2	4304.75	4235.75	4251.75	4398.75	4177.75	4092.75
pmed3	4453.75	4453.75	4465.75	4704.75	4414.75	4249.75
pmed4	3650.66	3641.66	3641.66	3674.66	3619.66	3091.66
pmed5	2270.67	2287.67	2275.67	2260.67	2227.67	1376.67
pmed6	7837.73	7888.73	7837.73	7919.73	7758.73	7823.73
pmed7	5791.76	5750.76	5775.76	5907.76	5705.76	5638.76
pmed8	4691.73	4779.73	4692.73	4680.73	4560.73	4491.73
pmed9	3360.75	3357.75	3395.75	3333.75	3263.75	2749.75
pmed10	2303.84	2311.84	2313.84	2302.84	2274.84	1267.84
pmed11	7715.73	7732.73	7760.73	8015.73	7646.73	7695.73
pmed12	6688.59	6705.59	6691.59	6686.59	6593.59	6639.59
pmed13	4653.69	4770.69	4695.69	6290.69	4588.69	4410.69
pmed14	3740.76	3740.76	3767.76	3703.76	3666.76	2983.76
pmed15	2975.86	2980.86	2986.86	2969.86	2956.86	1740.86
pmed16	8179.68	8241.68	8196.68	8509.68	8142.68	8197.68
med17	7130.62	7152.62	7223.62	7238.62	6998.62	7009.62
pmed18	5114.67	5110.67	5077.67	5034.67	4978.67	4813.67
pmed19	3499.67	3491.67	3478.67	3445.67	3414.67	2875.67
pmed20	3133.83	3128.83	3131.83	3110.83	3079.83	1845.83
pmed21	9275.71	9202.71	9213.71	9683.71	9115.71	9137.71
pmed22	8729.71	8733.71	8736.71	8711.71	8578.71	8604.71
pmed23	4927.71	4912.71	4958.71	4883.71	4869.71	4655.71
pmed24	3663.72	3656.72	3665.72	4887.72	3575.72	2983.72
pmed25	3261.82	3247.82	3260.82	3238.82	3191.82	1895.82
pmed26	9957.73	9928.73	9987.73	10245.73	9908.73	9938.73
pmed27	8351.62	8388.62	8347.62	8337.62	8288.62	8309.62
pmed28	4819.60	4804.60	4799.60	4746.60	4680.60	4555.60
pmed29	3779.65	3801.65	3781.65	3750.65	3704.65	3100.65
pmed30	3449.85	3460.85	3456.85	3461.85	3409.85	2089.85
pmed31	10095.69	10095.69	10113.69	10633.69	10088.69	10085.69
pmed32	9398.63	9444.63	9461.63	9428.63	9235.63	9317.63
pmed33	5014.61	5027.61	5064.61	4959.61	4918.61	4780.61
pmed34	3790.67	3810.67	3795.67	3794.67	3698.67	3085.67
pmed35	10562.69	10486.69	10563.69	10707.69	10372.69	10399.69
pmed36	9955.64	10113.64	10043.64	10005.64	9979.64	9933.64
pmed37	5443.63	5501.63	5439.63	5359.63	5332.63	5109.63

Table 16 continued

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed38	11144.74	11,085.74	11160.74	11247.74	11103.74	11134.74
pmed39	9527.68	9639.68	9567.68	9670.68	9359.68	9450.68
pmed40	5532.68	5582.68	5541.68	5494.68	5430.68	5180.68
Average	5952.33	5965.03	5963.36	6086.41	5868.96	5564.13
D_250_1	1213.67	1224.67	1232.67	1315.74	1198.67	1211.67
D_250_2	1174.71	1175.71	1182.71	1274.89	1156.71	1176.71
D_250_3	1151.66	1157.66	1163.66	1259.52	1144.66	1160.66
D_250_5	1218.68	1236.68	1231.68	1328.09	1212.68	1219.68
D_250_6	1142.64	1161.64	1155.64	1247.57	1126.64	1132.64
D_250_7	1224.74	1224.74	1238.74	1328.15	1211.74	1214.74
D_250_8	1197.73	1203.73	1203.73	1302.78	1186.73	1196.73
D_250_9	1159.70	1163.70	1162.70	1266.12	1149.70	1150.70
D_250_10	1233.66	1237.66	1240.66	1335.38	1219.66	1232.66
D_350_1	1568.68	1565.68	1564.68	1706.22	1542.68	1546.68
D_350_2	1555.67	1566.67	1557.67	1693.30	1535.67	1544.67
D_350_3	1594.74	1602.74	1599.74	1738.14	1578.74	1589.74
D_350_4	1578.70	1586.70	1614.70	1725.65	1570.70	1593.70
D_350_5	1598.64	1588.64	1597.64	1728.93	1561.64	1570.64
D_350_6	1543.69	1537.69	1540.69	1679.01	1515.69	1522.69
D_350_7	1464.74	1487.74	1496.74	1621.78	1461.74	1474.74
D_350_8	1559.61	1566.61	1568.61	1704.34	1544.61	1551.61
D_350_9	1501.75	1506.75	1516.75	1660.17	1494.75	1506.75
D_350_10	1507.76	1526.76	1519.76	1694.86	1494.76	1503.76
Average	1378.48	1385.38	1388.90	1505.82	1363.59	1373.75
kmedian1	488,462.57	495659.57	493967.57	553918.57	–	537846.57
kmedian2	240,771.47	243005.47	245423.47	281918.47	–	271541.47
kmedian3	160,433.48	162329.48	161589.48	184726.48	–	183763.48
kmedian4	696,428.67	699489.67	704269.67	787811.67	–	764522.67
kmedian5	345089.32	344,552.32	348081.32	398402.32	–	393722.32
kmedian6	227,730.40	228630.40	230121.40	258461.40	–	259711.40
kmedian7	859662.40	855,393.40	862908.40	954856.40	–	947076.40
kmedian8	422,881.45	426763.45	424318.45	482584.45	–	475294.45
kmedian9	285,599.59	290646.59	288045.59	321349.59	–	321846.59
kmedian10	985,828.58	992980.58	1005136.58	1107094.58	–	1094360.58
kmedian11	500,770.53	507074.53	506145.53	557704.53	–	555109.53
kmedian12	396001.25	422771.25	423916.25	0.00	–	372498.25
kmedian13	1105190.30	1,101,192.30	1107665.30	1222864.30	–	1219656.30
kmedian14	592,194.41	622599.41	617124.41	620031.41	–	621424.41
kmedian15	498313.21	517048.21	514208.21	412,368.21	–	416571.21
Average	520,357.18	527342.44	528861.44	542939.49	–	562329.71

Table 17 CPU time. Detailed results for every instance and every algorithm

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed1	3600.00	3600.00	3600.00	750.00	116.91	1.69
pmed2	3600.00	3600.00	3600.00	750.00	102.55	4.23
pmed3	3600.00	3600.00	3600.00	750.00	162.63	3.81
pmed4	3600.00	3600.00	3600.00	750.00	39.11	9.86
pmed5	3600.00	3600.00	3600.00	750.00	10.08	19.54
pmed6	3600.00	3600.00	3600.00	750.00	2562.56	4.68
pmed7	3600.00	3600.00	3600.00	750.00	2166.09	10.27
pmed8	3600.00	3600.00	3600.00	750.00	1725.86	29.52
pmed9	3600.00	3600.00	3600.00	750.00	1082.85	87.43
pmed10	3600.00	3600.00	3600.00	750.00	120.32	215.20
pmed11	3600.00	3600.00	3600.00	750.00	1942.85	11.39
pmed12	3600.00	3600.00	3600.00	750.00	3122.69	37.70
pmed13	3600.00	3600.00	3600.00	750.00	3783.52	182.76
pmed14	3600.00	3600.00	3600.00	750.00	2602.36	455.56
pmed15	3600.00	3600.00	3600.00	750.00	785.48	837.99
pmed16	3600.00	3600.00	3600.00	750.00	3889.25	21.72
pmed17	3600.00	3600.00	3600.00	750.00	4135.74	56.34
pmed18	3600.00	3600.00	3600.00	750.00	5253.96	397.17
pmed19	3600.00	3600.00	3600.00	750.00	4976.48	1155.27
pmed20	3600.00	3600.00	3600.00	750.00	1752.63	2390.48
pmed21	3600.00	3600.00	3600.00	750.00	3538.66	33.20
pmed22	3600.00	3600.00	3600.00	750.00	6920.24	81.87
pmed23	3600.00	3600.00	3600.00	750.00	4802.91	993.01
pmed24	3600.00	3600.00	3600.00	750.00	4305.88	3141.25
pmed25	3600.00	3600.00	3600.00	750.00	3482.32	5521.56
pmed26	3600.00	3600.00	3600.00	750.00	6020.31	41.73
pmed27	3600.00	3600.00	3600.00	750.00	5489.43	137.98
pmed28	3600.00	3600.00	3600.00	750.00	4503.29	1966.87
pmed29	3600.00	3600.00	3600.00	750.00	2920.57	5878.63
pmed30	3600.00	3600.00	3600.00	750.00	2985.58	12007.19
pmed31	3600.00	3600.00	3600.00	750.00	5776.80	59.84
pmed32	3600.00	3600.00	3600.00	750.00	5535.60	169.07
pmed33	3600.00	3600.00	3600.00	750.00	5672.08	3401.64
pmed34	3600.00	3600.00	3600.00	750.00	5704.44	10597.65
pmed35	3600.00	3600.00	3600.00	750.00	6850.11	80.67
pmed36	3600.00	3600.00	3600.00	750.00	7402.39	232.95
pmed37	3600.00	3600.00	3600.00	750.00	5810.91	5835.00
pmed38	3600.00	3600.00	3600.00	750.00	5190.86	124.52

Table 17 continued

Instance	MOEA/D	NSGA-II	SPEA2	SS	ϵ -C	RPR
pmed39	3600.00	3600.00	3600.00	750.00	4293.78	304.18
pmed40	3600.00	3600.00	3600.00	750.00	5647.92	9223.30
Average	3600.00	3600.00	3600.00	750.00	3579.7031	1644.12
D_250_1	3600.00	3600.00	3600.00	750.00	656.17	128.55
D_250_2	3600.00	3600.00	3600.00	750.00	620.38	129.28
D_250_3	3600.00	3600.00	3600.00	750.00	789.03	188.02
D_250_4	3600.00	3600.00	3600.00	750.00	595.00	149.67
D_250_5	3600.00	3600.00	3600.00	750.00	991.99	179.95
D_250_6	3600.00	3600.00	3600.00	750.00	2239.39	152.64
D_250_8	3600.00	3600.00	3600.00	750.00	1022.35	158.40
D_250_9	3600.00	3600.00	3600.00	750.00	561.53	146.44
D_250_10	3600.00	3600.00	3600.00	750.00	1078.05	122.42
D_350_1	3600.00	3600.00	3600.00	1650.00	1947.16	742.52
D_350_2	3600.00	3600.00	3600.00	1650.00	1907.66	657.00
D_350_3	3600.00	3600.00	3600.00	1650.00	1602.20	587.17
D_350_4	3600.00	3600.00	3600.00	1650.00	2342.34	613.34
D_350_5	3600.00	3600.00	3600.00	1650.00	2967.38	579.26
D_350_6	3600.00	3600.00	3600.00	1650.00	1355.43	701.97
D_350_7	3600.00	3600.00	3600.00	1650.00	917.40	458.42
D_350_8	3600.00	3600.00	3600.00	1650.00	1749.88	806.78
D_350_9	3600.00	3600.00	3600.00	1650.00	2674.02	448.30
D_350_10	3600.00	3600.00	3600.00	1650.00	2798.09	489.62
Average	3600.00	3600.00	3600.00	1223.68	1516.60	391.57
kmedian1	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian2	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian3	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian4	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian5	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian6	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian7	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian8	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian9	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian10	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian11	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian12	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian13	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian14	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
kmedian15	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00
Average	3600.00	3600.00	3600.00	3600.00	3600.00	3600.00

References

- Beasley JE (1990) OR-library: distributing test problems by electronic mail. *J Oper Res Soc* 41:1069–1072. <https://doi.org/10.2307/2582903>
- Boonmee C, Arimura M, Asada T (2017) Facility location optimization model for emergency humanitarian logistics. *Int J Disaster Risk Reduct* 24:485–498. <https://doi.org/10.1016/j.ijdrr.2017.01.017>
- Campos V, Martí R, Sánchez-Oro J, Duarte A (2014) GRASP with path relinking for the orienteering problem. *J Oper Res Soc* 65:1800–1813. <https://doi.org/10.1057/jors.2013.156>
- Church R, Velle CR (1974) The maximal covering location problem. *Papers Reg Sci Assoc* 32:101–118. <https://doi.org/10.1111/j.1435-5597.1974.tb00902.x>
- Colmenar JM, Hoff A, Martí R, Duarte A (2018) Scatter search for the bi-criteria p-median p-dispersion problem. *Progress Artif Intell* 7:31–40. <https://doi.org/10.1007/s13748-017-0132-6>
- Deb K (2014) Multi-objective optimization. In: Burke EK, Kendall G (eds) *Search methodologies: introductory tutorials in optimization and decision support techniques*, pp. 403–449. Springer, Boston, MA. <https://doi.org/10.1201/9781315183176-4>
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6:182–197. <https://doi.org/10.1109/4235.996017>
- Duarte A, Sánchez-Oro J, Resende MG, Glover F, Martí R (2015) Greedy randomized adaptive search procedure with exterior path relinking for differential dispersion minimization. *Inf Sci* 296:46–60. <https://doi.org/10.1016/j.ins.2014.10.010>
- Durillo JJ, Nebro AJ (2011) Jmetal: a java framework for multi-objective optimization. *Adv Eng Softw* 42:760–771. <https://doi.org/10.1016/j.advengsoft.2011.05.014>
- Erkut E (1990) The discrete p-dispersion problem. *Eur J Oper Res* 46:48–60. [https://doi.org/10.1016/0377-2217\(90\)90297-o](https://doi.org/10.1016/0377-2217(90)90297-o)
- Erkut E, Neuman S (1989) Analytical models for locating undesirable facilities. *Eur J Oper Res* 40:275–291. [https://doi.org/10.1016/0377-2217\(89\)90420-7](https://doi.org/10.1016/0377-2217(89)90420-7)
- Farahani RZ, Fallah S, Ruiz R, Hosseini S, Asgari N (2019) Or models in urban service facility location: a critical review of applications and future developments. *Eur J Oper Res* 276:1–27. <https://doi.org/10.1016/j.ejor.2018.07.036>
- Feo TA, Resende MGC (1995) Greedy randomized adaptive search procedures. *J Global Optim* 6:109–133. <https://doi.org/10.1007/bf01096763>
- Franca RB, Jones EC, Richards CN, Carlson JP (2010) Multi-objective stochastic supply chain modeling to evaluate tradeoffs between profit and quality. *Int J Prod Econ* 127:292–299. <https://doi.org/10.1016/j.ijpe.2009.09.005>
- Glover F, Laguna M (1998). Tabu search. In: Du D-Z, Pardalos PM (eds) *Handbook of combinatorial optimization*, Vol 1–3, pp 2093–2229. Springer, Boston. https://doi.org/10.1007/978-1-4419-7997-1_17
- Hadka D (2015) Moea framework-a free and open source java framework for multiobjective optimization. version 2.11. <http://www.moeaframework.org>
- Hakimi SL (1964) Optimum locations of switching centers and the absolute centers and medians of a graph. *Oper Res* 12:450–459. <https://doi.org/10.1287/opre.12.3.450>
- Hakimi SL (1965) Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Oper Res* 13:462–475. <https://doi.org/10.1287/opre.13.3.462>
- Jovanovic R, Sanfilippo AP, Voß S (2022) Fixed set search applied to the multi-objective minimum weighted vertex cover problem. *J Heuristics* 28:481–508. <https://doi.org/10.1007/s10732-022-09499-z>
- Karatas M (2017) A multi-objective facility location problem in the presence of variable gradual coverage performance and cooperative cover. *Eur J Oper Res* 262:1040–1051. <https://doi.org/10.1016/j.ejor.2017.04.001>
- Karatas M, Yakıcı E (2018) An iterative solution approach to a multi-objective facility location problem. *Appl Soft Comput* 62:272–287. <https://doi.org/10.1016/j.asoc.2017.10.035>
- Li M, Yao X (2019) Quality evaluation of solution sets in multiobjective optimisation: a survey. *ACM Comput Surv*. <https://doi.org/10.1145/3300148>
- López-Sánchez AD, Sánchez-Oro J, Laguna M (2020) A new scatter search design for multiobjective combinatorial optimization with an application to facility location. *INFORMS J Comput*. <https://doi.org/10.1287/ijoc.2020.0966>
- Martí R, Campos V, Resende MG, Duarte A (2015) Multiobjective grasp with path relinking. *Eur J Oper Res* 240:54–71. <https://doi.org/10.1016/j.ejor.2014.06.042>
- Minieka E (1970) The m-center problem. *SIAM Rev* 12:138–139
- Mladenović N, Brimberg J, Hansen P, Moreno-Pérez JA (2007) The p-median problem: a survey of metaheuristic approaches. *Eur J Oper Res* 179:927–939. <https://doi.org/10.1016/j.ejor.2005.05.034>
- Pérez-Peló S, Sánchez-Oro J, Duarte A (2020) Finding weaknesses in networks using greedy randomized adaptive search procedure and path relinking. *Expert Syst* 37:e12540. <https://doi.org/10.1111/exsy.12540>
- Pérez-Peló S, Sánchez-Oro J, López-Sánchez AD, Duarte A (2019) A multi-objective parallel iterated greedy for solving the p-center and p-dispersion problem. *Electronics* 8:1440. <https://doi.org/10.3390/electronics8121440>
- Pinto-Varela T, Barbosa-Póvoa APF, Novais AQ (2011) Bi-objective optimization approach to the design and planning of supply chains: economic versus environmental performances. *Comput Chem Eng* 35:1454–1468. <https://doi.org/10.1016/j.compchemeng.2011.03.009>
- Resende M, Martí R, Gallego M, Duarte A (2010) Grasp and path relinking for the max-min diversity problem. *Comput Oper Res* 37:498–508. <https://doi.org/10.1016/j.cor.2008.05.011>
- Sagrado J, Águila IM, Orellana FJ (2015) Multi-objective ant colony optimization for requirements selection. *Empir Softw Eng* 20:577–610. <https://doi.org/10.1007/s10664-013-9287-3>
- Sánchez-Oro J, López-Sánchez AD, Colmenar JM (2020) A general variable neighborhood search for solving the multi-objective open vehicle routing problem. *J Heuristics* 26:423–452. <https://doi.org/10.1007/s10732-017-9363-8>
- Sayyady F (2012) Mathematical models and algorithms for the location of sensors on a traffic network. Ph.D. thesis. <http://www.lib.ncsu.edu/resolver/1840.16/8131>
- Sayyady F, Tutunchi GK, Fathi Y (2015) p-median and p-dispersion problems: a bi-criteria analysis. *Comput Oper Res* 61:46–55. <https://doi.org/10.1016/j.cor.2015.02.007>
- Tutunchi GK, Fathi Y (2019) Effective methods for solving the bi-criteria p-center and p-dispersion problem. *Comput Oper Res* 101:43–54. <https://doi.org/10.1016/j.cor.2018.08.009>
- Wang S-C, Lin C-C, Chen T-C, Hsiao HC (2018) Multi-objective competitive location problem with distance-based attractiveness for two facilities. *Comput Electric Eng* 71:237–250. <https://doi.org/10.1016/j.compeleceng.2018.07.013>
- Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11:712–731. <https://doi.org/10.1109/tevc.2007.892759>
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. ETH Zurich. <https://doi.org/10.3929/ethz-a-004284029>