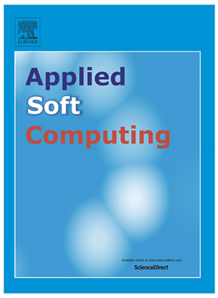
Journal Pre-proof



A GRASP method for the Bi-Objective Multiple Row Equal Facility Layout Problem

Nicolás R. Uribe, Alberto Herrán, J. Manuel Colmenar

PII: S1568-4946(24)00671-9
DOI: https://doi.org/10.1016/j.asoc.2024.111897
Reference: ASOC 111897

To appear in: *Applied Soft Computing*

Received date : 31 July 2023
Revised date : 17 May 2024
Accepted date : 17 June 2024

Revised Manuscript (Clean version)

A GRASP method for the Bi-Objective Multiple Row Equal Facility Layout Problem

Nicolás R. Uribe, Alberto Herrán, J. Manuel Colmenar*

Dept. Computer Sciences, Universidad Rey Juan Carlos
C/. Tulipán, s/n, Móstoles, 28933 (Madrid), Spain

Abstract

The Bi-Objective Multiple Row Equal Facility Layout Problem considers both quantitative and qualitative objectives that are very useful in many scenarios like the factory design. In this work, a new multi-objective GRASP approach is proposed which applies an ensemble of four different constructive methods followed by the combination of two local search procedures, improving the results from the state of the art. Due to the superiority of this proposal, a new dataset of larger problem instances is generated, providing detailed metrics of the obtained solutions.

Keywords: Metaheuristics, GRASP, Facility location, Row layout, Bi-objective optimization

1. Introduction

Facility Layout Problems (FLPs) are a well-known family of problems in the operational research area. The main objective of FLP is to find the optimal placement of different facilities in a given layout trying to minimize a particular objective function [1]. This family of problems has a number of applications in very different domains, such as manufacturing systems

*This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación (MCIN/AEI/10.13039/501100011033/FEDER, UE) under grant refs. PID2021-126605NB-I00 and RED2022-134480-T.

*Corresponding author

Email address: `nicolas.rodriguez@urjc.es`, `alberto.herran@urjc.es`, `josemanuel.colmenar@urjc.es`, (Nicolás R. Uribe, Alberto Herrán, J. Manuel Colmenar)

[2], delivery service [3], layout of logistics facilities [4], or urban and office planning [5].

The very first work addressing FLP was proposed in 1969 by [6], which studied the Single Row Facility Layout Problem (SRFLP). The main characteristic of this problem is that the layout where the facilities have to be placed is just a single row. However, there exist several variants of this problem when considering row layouts, depending on different features such as the number of rows, the area of the facilities, or the possibility of allowing spaces between facilities.

The majority of the works found in the literature consider a single objective function corresponding to or related to material handling cost between facilities [7, 8, 9, 10]. However, this quantitative metric is not the only one that could be taken into account. Some other qualitative metrics could be considered, like the closeness rating or the hazardous movement of material [11, 12]. In a realistic scenario of designing a factory, a decision maker could receive this way different alternative solutions with different values for the objective functions taken into account.

In this paper, we address a multi-objective problem from the FLP family, the Bi-Objective Multi-Row Equal Facility Layout Problem (BO-MREFLP). This problem considers two objective functions: the material handling cost and the closeness rating. The BO-MREFLP is a variant of the Multi-Row Equal Facility Layout Problem (MREFLP), where facilities have the same area and more than two rows are considered in the layout [9]. As it will be later detailed, the number of rows and the number of facilities that must be located in each row are also constraints of the BO-MREFLP.

We can find in the literature two different approaches for the BO-MREFLP. On the one hand, the optimization of a combined objective function using heuristics is applied in many works [13, 14, 15, 16, 17, 18, 19, 11]. On the other hand, a multi-objective optimization approach is only proposed in [11], which returns a number of non-dominated solutions as a result for each instance. This problem was firstly addressed in [13]. The author implemented a heuristic method that combines both approaches: the heuristic uses a set of non-dominated solutions, with the weights of each objective fixed. As a result, efficient layouts are given but the times are not reported for the unique instance studied, with six facilities. In [14], another heuristic based on the exchange move is proposed. This work studies two different instances, with six and eight facilities. Once again, efficient layouts are given, but execution times are not reported. A heuristic approach based on pairwise exchange is implemented in [15]. Three different instances with sizes six, eight and twelve facilities are studied, comparing their results with the previous papers, obtaining similar results, but in a single iteration of the algorithm. In [16], a heuristic with a construction phase and an improvement method is proposed. Four instances are studied with six, eight, twelve and thirty six facilities, improving the results of the previous papers. In addition, the execution time for the largest instance is reported. In [17], a heuristic approach is implemented, based on the proposals of [14] and [15]. Four instances with eight, twelve, fifteen, and twenty instances are studied. The authors detail each combination of weights they have used, the percentage improvement ratio, and execution times. Since this is the most detailed paper in terms of description of instances, we have taken it as a baseline in order to generate the new instances. In [19], a three-stage heuristic is proposed. The first stage consists of matrix normalization. The second stage consists of discovering the weights of the objectives. For this purpose, the Mean Weight Method (MWM), the Geometric Mean Weight Method (GMWM), the Standard Deviation Weight Method (SDWM), and the Critical Importance Through Inter Criteria Correlation Method (CRITICM) are implemented. The third stage is the resolution through their proposed methods. They have reported the results for each method but the time for the four studied instances with six, eight, twelve, and fifteen facilities. In [18], a metaheuristic approach is proposed. Specifically, a Simulated Annealing is implemented, which is run on four instances with six, eight, twelve, and fifteen facilities. The authors report efficient solutions and execution times, as well as a comparison with previous works. In [11], the authors have applied a weight-based approach through a Biogeography-Based Optimization (BBO) algorithm, using the weights proposed in the literature. Also, they have applied a multi-objective optimization approach through a NSBBO and a NSGA-II algorithms, where NSBBO stands for Non-dominated Sorting BBO. Efficient solutions are given for instances with six, eight, twelve, and fifteen facilities, but the execution times are not reported. Up to our knowledge, this is the only study where a set of non-dominated solutions is produced considering the material handling cost and the closeness rating as objectives. In this work, we use this last multi-objective approach to tackle the BO-MREFLP problem.

In this paper, we have designed a Greedy Randomized Adaptive Search Procedure (GRASP) that follows a multi-objective approach. The contributions of this proposal are both theoretical and practical. Firstly, we propose the combination of four different constructive methods to produce an initial set of diverse non-dominated solutions. The balance between the different methods is determined by input parameters whose value can be adapted. We have designed a new local search method that combines both a dominance-based approach and an alternation of objectives in the context of the tackled bi-objective problem. Again, this method allows different configurations, guided by parameters. In this way, it is possible to determine the width and depth of the search process. Finally, from the practical point of view, after tuning the parameters of our proposed method, we have improved the results from the state of the art. Hence, we provide a new set of larger instances, as well as the detailed results obtained with our proposal, and two evolutionary methods.

The structure of the paper is the following: Section 2 formally describes the target problem; Section 3 details the algorithmic approach we have used in this work; Section 4 shows the comparison between our results and the state of the art, proposing new instances and results for future comparisons; Section 5 analyzes the managerial implications and; finally, Section 6 draws the conclusions and future work.

2. Problem description

Most of the works in the literature dealing with facility layout problems only consider one quantitative objective involving the material handling cost or work flow between facilities. However, in a real-world scenario, there exist also other qualitative objectives, such as closeness rating, hazardous movement, or safety between facilities, which may affect the optimal layout. Then, the Bi-Objective Multi-Row Equal Facility Layout Problem (BO-MREFLP) takes into account both qualitative and quantitative objectives. Specifically, BO-MREFLP is a $\mathcal{NP}$-hard optimization problem that consists in finding an optimal arrangement of rectangular facilities in several rows considering both objectives, material handling cost (MHC) and closeness rating (CR), simultaneously [11].

Following the definitions of the previous works, CR represents different levels of proximity between facilities, which are desirable for an optimal layout. These levels are developed according to different criteria, such as facilities/departments using the same equipment/personnel, sequence of workflow, ease of communication, unsafe or unpleasant conditions, etc. Typically, five different levels of proximity between facilities can be used, indicating whether it is especially important (E), important (I), ordinary (O), unimportant (U) or undesirable (X) for departments/facilities to be close together [17]. In [15], the authors also consider another level of proximity, absolutely necessary (A). Then, the closeness rating is also computed as the total weighted sum of the center-to-center distances between each pair of facilities, being the corresponding weight between facilities a numerical score assigned to each level of proximity: A=5, E=4, I=3, O=2, U=1 and X=-1. However, we focus or work on the five levels of proximity (E=4, I=3, O=2, U=1 and X=-1), as proposed in [17], since this approach is followed by the current state of the art of the studied problem [11].

Futhermore, an instance of the BO-MREFLP consists of a set F of n facilities, the number of available rows, m, and two squared matrices W_{MHC} and W_{CR} ($|F| \times |F|$) of pairwise weights, w_{uv}, representing the material handling cost and the closeness rating between two facilities $u, v \in F$, respectively. Therefore, an instance I is defined by a 4-tuple $I = (F, m, W_{MHC}, W_{CR})$.

In the BO-MREFLP all the facilities have the same length L and height H, which, for the sake of simplicity, can be considered equal to 1. Hence, given a set of n facilities to be allocated in m rows, the number of facilities per row is fixed, and determined by $c = n/m$.

Let $\Pi = \{(i, j) : 1 \leq i \leq m, 1 \leq j \leq c\}$ be the set of all the available positions of the grid ($m \times c$) to allocate one solution (see Figure 1). A solution of this optimization problem can be described as a mapping $\varphi : F \rightarrow \Pi$ that assigns the set of facilities F to the corresponding layout with m rows and c columns. Specifically, for a facility $u \in F$, $\varphi(u) = (i, j)$ indicates that u is allocated in row i (with $1 \leq i \leq m$) and column j (with $1 \leq j \leq c$).

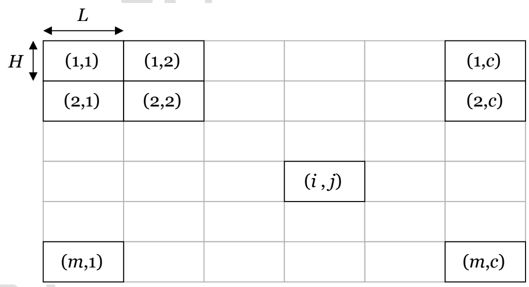


Figure 1: Grid with m rows and c columns to allocate n facilities of size $L \times H$.

Then, given a solution φ, both MHC and CR objectives are computed taking into account the pairwise weights and Manhattan distances between each pair of facilities u and v, located in positions (i, j) and (k, l) respectively, as it is shown in equations (1) and (2).

$$\mathcal{F}(\varphi, W) = \sum_{\substack{u,v \in F \\ u<v}} w_{uv} \cdot d_{uv}(\varphi) \tag{1}$$

$$d_{uv}(\varphi) = L \cdot |l - j| + H \cdot |k - i| \tag{2}$$

The BO-MREFLP consists in minimizing both objectives, $\mathcal{F}_1(\varphi) = \mathcal{F}(\varphi, W_{MHC})$ and $\mathcal{F}_2(\varphi) = \mathcal{F}(\varphi,\ W_{CR})$ at the same time. Hence, it can be mathematically formulated as shown in Equation (3), where Φ is the set of all feasible facility layouts.

$$\varphi^\star \leftarrow \underset{\varphi \in \Phi}{\arg\min} \left[\, \mathcal{F}_1(\varphi)\,,\, \mathcal{F}_2(\varphi) \,\right] \tag{3}$$

3. Greedy Randomized Adaptive Search Procedure

The Greedy Randomized Adaptive Search Procedure (GRASP) methodology [20] consists of two main steps: a first phase, where a feasible solution is constructed by iteratively adding new elements to an initial empty solution; and a second phase, which tries to improve the incumbent solution by some local search procedure. This methodology has been thoroughly used for bi-objective problems. Pareto dominance in GRASP has been considered in many different problems like scheduling [21, 22] and facility layout [23]. In this latter case, studying a very different scenario than the current problem, since it deals with continuous clearances between facilities . However, the majority of the GRASP proposals combine the objectives of the problem into one function to optimize in both phases, as in the recent [24]. Other approaches try to take advantage of the exploration of each one of the objectives separately in the improvement phase, joining the obtained solutions with a different method like Path Relinking [25, 26].

In this work, we adapt the GRASP methodology to solve the BO-MREFLP by considering the two objectives at the same time by means of the dominance in both phases of the methodology. Therefore, instead of having one solution, a set of non-dominated solutions will be considered.

3.1. Encoding of solutions

In order to explain the selected encoding for the solutions of the problem, we use an example instance with 15 facilities and 3 rows. We illustrate in Figure 2 the solution encoded in the form of a matrix, which is the one we selected for this work. As seen, the first row ($row1$) is accentuated in red, the second row ($row2$) in green, and the third row ($row3$) is delineated in blue. Within this matrix representation, the initial facility, numbered 15, is positioned in $row1$ and the first column. Conversely, facility 6 is located in $row3$ and occupies the fifth column. Notice that the rows are naturally differentiated in the matrix, as in the case of Figure 1.

row1	15	11	4	14	5
row2	3	13	8	12	10
row3	1	7	9	2	6

Figure 2: Matrix representing a solution with 15 facilities and 3 rows.

Similarly, a linear representation of the matrix will be equivalent, since the number of rows and the number of facilities per row is a data fixed by the instance. In fact, we use a linear representation for the chromosomes in the evolutionary methods that are described in Section 4.4.

3.2. Bi-objective GRASP

In a single-objective optimization problem, two solutions φ_a and φ_b can be directly compared attending to their objective function value. Specifically, in a minimization problem, φ_a is better than φ_b if $\mathcal{F}(\varphi_a) < \mathcal{F}(\varphi_b)$. However, in multi-objective problems this comparison is not as straightforward since it is necessary to check all the objective functions. Again, assuming that we minimize all the k objectives, a solution φ_a dominates another solution φ_b (denoted as $\varphi_a \prec \varphi_b$) if the following conditions are satisfied:

$$\begin{aligned} &\forall i \in \{1..k\} : \mathcal{F}_i(\varphi_a) \le \mathcal{F}_i(\varphi_b) \\ \wedge\ &\exists i \in \{1..k\} : \mathcal{F}_i(\varphi_a) < \mathcal{F}_i(\varphi_b) \end{aligned} \tag{4}$$

If there is no solution $\varphi \in \Phi$ which dominates $\varphi' \in \Phi$ it is said that φ' is a non-dominated or efficient solution. The Pareto Optimal Set (or simply Pareto Front) is the set of all non-dominated solutions φ' in the entire feasible search space Φ. Hence, solving a multi-objective optimization problem consists in generating an initial set of non-dominated solutions (ND) and iteratively improving it until the entire Pareto Front is found or a given termination condition is met.

Algorithm 1 shows our multi-objective GRASP proposal to tackle this problem. The algorithm has 5 parameters: α, which controls the randomness/greediness of the constructive procedures; B, a set of percentages β_i which determine the use of each constructive method; n_c, the total number of iterations of the constructive phase; γ, which controls the balance of iterations among the local search methods in the improvement phase; and n_{ls}, the total number of iterations of the improvement phase. The algorithm begins in step 1 by populating an initial set of non-dominated solutions ND using all the available constructive procedures (Section 3.3). Then, steps 2 to 5 try to approximate ND to the Pareto Front by sequentially applying two different local search strategies. A first one, `DBLS`, which stands for *Dominance Based Local Search* (Section 3.4.1), executed $\gamma \cdot n_{ls}$ times, and a second one, `AOLS`, which stands for *Alternate Objectives Local Search* (Section 3.4.2), executed $(1-\gamma) \cdot n_{ls}$ times. The final set of non-dominated solutions is returned in step 6.

Algorithm 1: BO-GRASP$(\alpha, B, n_c, \gamma, n_{ls})$

```
1  ND ← Constructive(α, B, n_c)         ▷ Section 3.3
2  for i = 1 to γ · n_ls do
3  └ ND ← DBLS(ND)                      ▷ Section 3.4.1
4  for i = 1 to (1 − γ) · n_ls do
5  └ ND ← AOLS(ND)                      ▷ Section 3.4.2
6  return ND
```

Under this scheme, we will test in Section 4.2 another variant of this proposal which modifies the order where `DBLS` and `AOLS` local searches are executed. For the shake of brevity, we do not include here the pseudo-code, since it is almost the same as Algorithm 1 but executing steps 4 and 5 before steps 2 and 3.

Notice that a set of non-dominated solutions ND is returned after the execution of each method in steps 1, 3 and 5 of Algorithm 1. So, in order to maintain ND, every time a new solution φ' is reached in any of these methods, a procedure `Update` is called. If some solution $\varphi \in ND$ dominates

φ', the `Update` procedure discards φ'. If there are no solutions $\varphi \in ND$ that dominate φ', the `Update` procedure incorporates φ' into ND, and checks if there exist other solutions $\varphi \in ND$ dominated by φ', removing all dominated solutions from ND.

3.3. Constructive methods

In this work, we propose the combination of four different constructive strategies C_1, C_2, C_3, and C_4, based on the constructive procedures described in [9] for the MREFLP. C_1 is a *Greedy-Random* constructive strategy whose aim is to minimize the *MHC* objective. C_2 is similar to C_1 but interchanging the phases from *Greedy-Random* to *Random-Greedy*, as proposed in [27]. Similarly to C_1 and C_2, C_3 and C_4 minimize the *CR* objective under the *Greedy-Random* and *Random-Greedy* approaches.

In order to favor the diversification of the search, all C_1 to C_4 begin randomly filling the first column of a new solution φ_p. Then, they iteratively include new facilities, one at a time, to the first available position (i, j) of the partial solution with less than n facilities. With the aim of building good quality solutions, all the methods select the facility u to be included in each iteration according to the greedy function g shown in Equation (5). Notice that this function is similar to the objective function shown in Equation (1) but only computing the contribution of the new facility u to the partial solution φ_p. Following their above definition, C_1 and C_2 will use $g(\varphi_p, W_{MHC}, u)$ while C_3 and C_4 will use $g(\varphi_p, W_{CR}, u)$.

$$g(\varphi_p, W, u) = \sum_{v \in \varphi_p} w_{uv} \cdot d_{uv}(\varphi_p) \tag{5}$$

As stated before, the proposed constructive method will consider the four constructive strategies. It is important to keep in mind that our constructive methods are based on the GRASP methodology. Methods $C1$ and $C3$ use a *Greedy-Random* scheme, where the candidate list is made up of candidates that have exceeded a certain threshold determined by the best and worst solution and α. In this case, α changes the threshold to select candidates in a more greedy or random way. Methods $C2$ and $C4$ follow a *Random-Greedy* scheme, where the candidate list is generated by elements selected at random. In this case, the α parameter controls the size of the candidate list. For both schemes, we need to evaluate the candidate elements using the greedy function defined in Equation (5). Algorithm 2 shows the pseudocode of the proposed constructive procedure to populate the initial ND set of solutions. It receives 3 parameters: α, which controls the randomness($\alpha = 0$)/greediness($\alpha = 1$) of the constructive procedures; B, a set of percentages β_i which determine the use of each constructive method; and n_c, the total number of iterations of the constructive phase. The algorithm starts by creating in step 1 an empty set of non-dominated solutions *ND*. Then, it begins a loop, from step 2 to step 5 where, for each constructive method C_i, builds $n_c \cdot \beta_i$ solutions using the α parameter (steps 3 and 4) to update the current set of non-dominated solutions in step 5. Finally, the algorithm returns *ND* in step 6. The final value for each β_i parameter, together with the value of α balancing the randomness/greediness of all C_i methods, will be experimentally selected, as explained in Section 4.2.

Algorithm 2: CONSTRUCTIVE(α, B, n_c)

1 $ND \leftarrow \emptyset$
2 **for** $\beta_i \in B$ **do**
3   **repeat** $n_c \cdot \beta_i$ **times**
4         $\varphi \leftarrow C_i(\alpha)$
5         $ND \leftarrow \texttt{Update}(ND, \varphi)$
6 **return** *ND*

Figure 3 graphically shows the sets of non-dominated solutions (*ND*) that can be obtained by combining the four constructive methods using $\alpha = 0$ or $\alpha > 0$. Right upper corner shows the set *ND* which is expected to be obtained when using all the C_i methods with $\alpha = 0$. Since $\alpha = 0$ means a purely random strategy for generating new solutions, the solutions in this set will have poor quality in both *MHC* and *CR* objectives. However, by increasing the value of α the solutions generated with C_1/C_2 or C_3/C_4 will tend to improve the solution quality regarding *MHC* or *CR*, respectively. Moreover, the higher the α value, the more concentrated will be the solutions at the edges of the *ND* set, which could mean a loss of diversity on the initial *ND* set of solutions. Hence, as we will see in the experimental part, values of α around 0.5 achieve a good compromise between the desired quality and diversity on the initial set of *ND* solutions.

3.4. Local search

The second stage in a GRASP method consists in improving the solution generated in the constructive phase by applying a local search procedure,

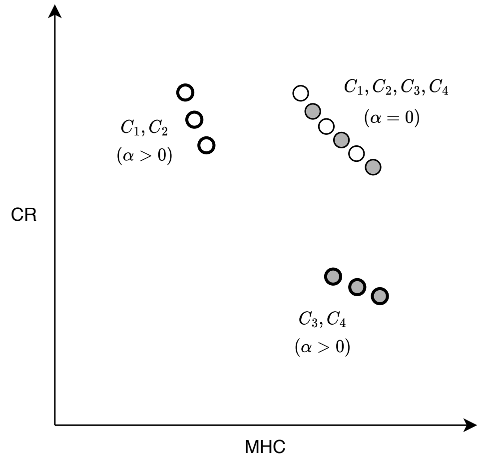


Figure 3: Sets of non-dominated solutions produced by the constructive methods C_i with $\alpha = 0$ or $\alpha > 0$.

which systematically explores the neighborhood of the incumbent solution. To this end, we define the neighborhood based on *exchange*. Figure 4 shows the effect of an *exchange* move, where two facilities u and v (`A` and `I` in the example) interchange their positions in the layout φ, resulting in a new neighbor solution φ'. We denote this move as $\varphi' \leftarrow exchange(\varphi, u, v)$.

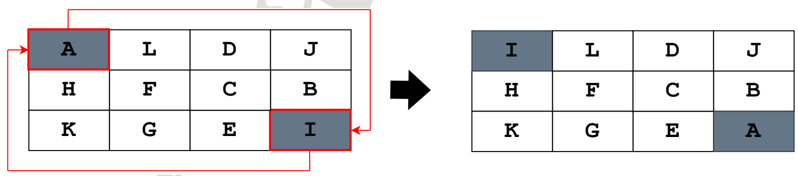


Figure 4: Effect of an *exchange* move. Original solution (left) and the resulting layout after the *exchange* move (right).

Once a neighborhood is created with the *exchange* move, there exist two typical strategies to explore it: *best improvement* and *first improvement*. The *best improvement* strategy explores all of the solutions in the neighborhood by a fully deterministic procedure, and the best move (i.e., the one that leads to a solution φ' with minimum associated cost) is applied at each iteration.

On the other hand, the *first improvement* strategy tries to avoid the time complexity of the *best improvement* strategy by exploring the neighborhood and performing the first move that enhances the current best cost. In this work, we adopt a *hybrid* strategy, which combines the *best* and *first improvement* strategies [9]. Hence, it selects one facility $u \in F$ at random, and then it explores all the possible *exchange* moves with the rest of facilities $v \in F$, $v \neq u$. If no improvement is encountered, the method continues exploring all the possible *exchange* moves from a new facility $u' \in F$, $u' \neq u$. Otherwise, the move $\varphi' \leftarrow exchange(\varphi, u, v)$ is applied and the local search continues exploring a new neighborhood from φ'.

As shown in Algorithm 1, our GRASP proposal for this bi-objective problem uses two different approaches in the local search stage, one based on dominance between solutions (`DBLS`) and other one based on a mono-objective local search along each objective (`AOLS`). Next, we describe both procedures.

3.4.1. Dominance-based local search

The first local search algorithm is based on the concept of dominance since it only performs moves that lead to solutions not dominated by the current one [28]. Algorithm 3 shows the pseudo-code of the `DBLS` local search. It tries to improve the incoming non-dominating set of solutions *ND* by exploring the neighborhoods generated with *exchange* moves using the *hybrid* strategy explained above.

The algorithm starts by initializing the best non-dominating set of solutions $ND^\star$ with *ND*. Then, it iterates for each solution $\varphi \in ND$ through steps 2 to 20. Steps 3 and 4 initialize φ^* and *improve*. Then, the inner loop executes while some improvement can be made on $ND^\star$ (steps 5 to 20). Before exploring the *exchange* neighborhood, we do copy $ND^\star$ into ND' in step 7, and select $\varphi^\star$ as the current solution for the exploration, φ', in step 8. The `GetFacilities` method used in step 9 of the algorithm returns all the facilities of a set (F), which are stored in S_v. In step 10, the `Shuffle` method creates a copy of a list (S_v) and returns it in randomly arranged order (S_u). We employ these lists (S_v and S_u) to execute every potential *exchange* between the facilities. After a neighbor solution φ'' is generated in step 13, ND' is *updated* with φ'' in step 14, since there could be other solutions in ND' also dominated by φ''. Besides, if φ'' dominates φ^* (step 15), then $\varphi^\star$ is updated with φ'' (step 16). Notice that this solution will be the starting solution in the next iteration of the outer loop, generating a new trajectory for the explored solution. Once all the possible *exchange* moves from v are computed, the algorithm checks in step 17 if the set $ND^\star$ has changed, which means that new non-dominated solutions have been added to ND' in step 14. In that case, the incumbent set of non-dominated solutions $ND^\star$ is updated in step 19, and the process continues exploring a new neighborhood from solution φ' (steps 18 and 20). Finally, the algorithm returns the best set of non-dominated solutions $ND^\star$ in step 21.

Algorithm 3: DOMINANCE BASED LOCAL SEARCH `DBLS`(ND)

1 $ND^\star \leftarrow ND$
2 **for** $\varphi \in ND$ **do**
3     $\varphi^* \leftarrow \varphi$
4     $improve \leftarrow$ `true`
5   **while** $improve$ **do**
6         $improve \leftarrow$ `false`
7         $ND' \leftarrow ND^\star$
8         $\varphi' \leftarrow \varphi^\star$
9         $S_v \leftarrow$ `GetFacilities`(F)
10         $S_u \leftarrow$ `Shuffle`(S_v)
11   **for** $u \in S_u$ **do**
12   **for** $v \in S_v \wedge v \neq u$ **do**
13                 $\varphi'' \leftarrow exchange(\varphi', u, v)$
14                 $ND' \leftarrow$ `Update`(ND', φ'')
15   **if** $(\varphi'' \prec \varphi^\star)$ **then**
16                     $\varphi^\star \leftarrow \varphi''$
17   **if** $ND' \neq ND^\star$ **then**
18                 $improve \leftarrow$ `true`
19                 $ND^\star \leftarrow ND'$
20   **break**
21 **return** $ND^\star$

3.4.2. Alternate objectives local search

The second local search algorithm follows a different approach than `DBLS`. Instead of considering both $\mathcal{F}_1$ and $\mathcal{F}_2$ objectives at the same time to check if the move is accepted, `AOLS` executes two independent single-objective local searches considering either $\mathcal{F}_1$ or $\mathcal{F}_2$, following the *hybrid* approach described in [9]. In this method, the *first improvement* and *best improvement* strategies are combined. Algorithm 4 shows the pseudo-code of the AOLS local search. Again, it starts by initializing the best set of non-dominated solutions $ND^\star$ with ND. Then, it iterates for each solution $\varphi \in ND$ through steps 2 to 6. In steps 3 and 4 AOLS tries to improve the solution φ with the hybrid LocalSearch procedure described in [9], attending to either $\mathcal{F}_1$ or $\mathcal{F}_2$, respectively. Then, steps 5 and 6 update $ND^\star$ with the obtained solutions. Finally, the algorithm returns the set of non-dominated solutions in step 7.

Algorithm 4: ALTERNATE OBJECTIVES LOCAL SEARCH AOLS(ND)

```
1 ND* ← ND
2 for φ ∈ ND do
3     φ1 ← LocalSearch(φ, F1)
4     φ2 ← LocalSearch(φ, F2)
5     ND* ← Update(ND*, φ1)
6     ND* ← Update(ND*, φ2)
7 return ND*
```

3.4.3. Ensemble of methods

As shown in Algorithm 1, we propose the combination of the two local search methods by means of a γ parameter. This idea allows a different exploration of the search space more adequate to the multi-objective version of the problem than to the single-objective one.

Figure 5 graphically shows the effect of the combination of DBLS and AOLS procedures executed over the same initial set of non-dominated solutions ND generated using all the constructive procedures with $\alpha > 0$. Figure 5(a) shows how the initial ND set will evolve when using DBLS. Since this method is based on dominance, it will tend to improve both objectives at the same time populating the central part of the ND set. On the other hand, Figure 5(b) shows how the AOLS will improve one of the objectives (*MHC* or *CR*) populating the more extreme parts of the ND set. Therefore, the combination of both DBLS and AOLS procedures should be a good strategy to get a populated set of quality non-dominated solutions without losing the solutions located at the edges, as suggested in Figure 5(c).

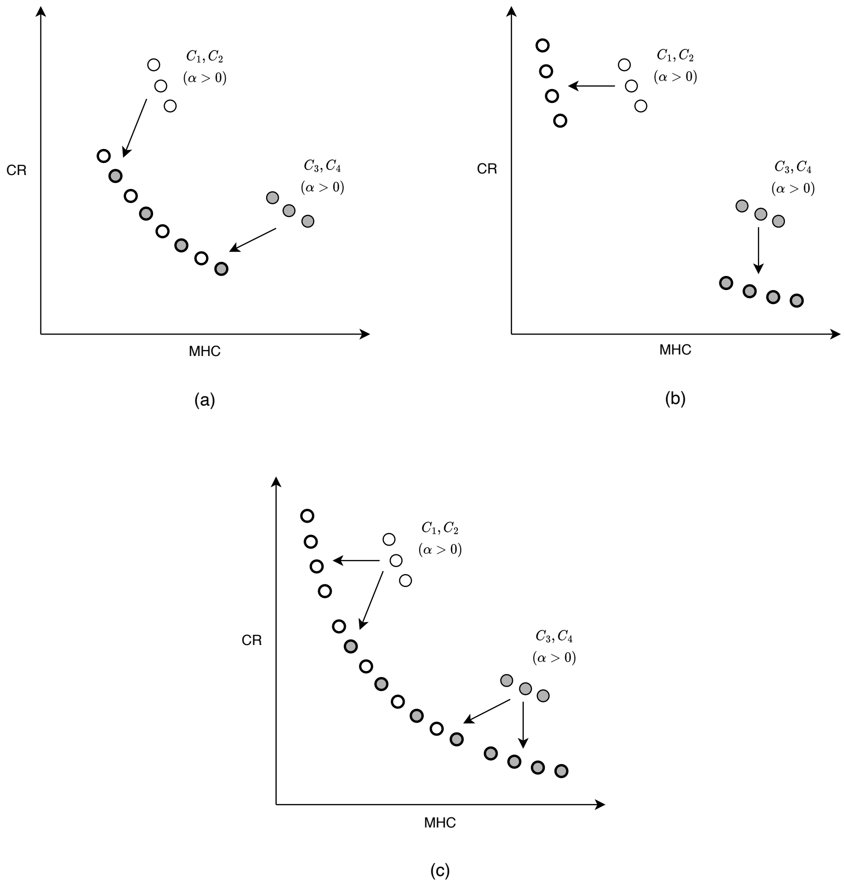


Figure 5: Effect of the DBLS (a) and AOLS (b) procedures executed over the same initial set of non-dominated solutions ND generated using all the constructive procedures with $\alpha > 0$; and (c), the expected set ND combining both DBLS and AOLS procedures.

4. Computational results

This section is devoted to show the experimental results of our proposal. In this regard, since the previous work only studied 4 instances [11], we have extended the dataset with 56 additional instances, making a total number of 60 studied instances. We consider these instances to be useful to the scientific community for future research.

In order to speed up our algorithms, we use the efficient local search proposed in [8]. This efficient local search has been adapted to both objective functions, which, combined with the different mechanisms of the new multi-objective approach, allowed us to save, on average, 60% of computation time in the benchmark instances, reaching 80% in the largest instances. We omit these experiments for the sake of space. Notice that the efficient computation is not a contribution of this work, but the application of a previous work to a new problem.

Our algorithms have been implemented in Java 17 and the experiments have been executed on a Windows 10 laptop provided with an Intel i7 1065G7 processor running at 1.3GHz with 16GB of RAM. All the code, instances, and detailed results are available in `https://grafo.etsii.urjc.es/en/BO-MREFLP`.

4.1. Generation of new instances

As stated in Section 1, many single-objective problems from the FLP family have been studied in the literature. Therefore, in order to enlarge the dataset for the BO-MREFLP, we have taken a set of instances from the MREFLP, which already included data for the *MHC* objective, and we have incorporated them an additional matrix for the *CR*.

More precisely, we expanded 56 instances from [9] with sizes up to 60. To this aim, we studied the *CR* matrices of the 4 instances from [11] and found out that their values were randomly generated using the interval $[-1, 4]$, considering these values as a penalty to minimize as stated in [17]. A similarly created matrix can be found in that work, where the previous instance with size 8 appears for the very first time. Therefore, we added randomly generated matrices using integer numbers in $[-1, 4]$ for the *CR* objective ($\mathcal{F}_2$) to the 56 selected instances.

We represent in Table 1 the features of the whole set of 60 instances. For each one, the number of rows (m), the number of columns (c), and total number of facilities to be located (n) is presented. Instances *previous6*, *previous8*, *previous12* and *previous15* correspond to the instances studied in [11], while the rest of the instances are the expanded ones. It is worth noticing that this problem requires $c = n/m$ facilities to be located in each row.

4.2. Preliminary experiments

To determine the best configuration and parameter values of our proposal, we have used an automatic configuration tool on a representative subset of instances which we call benchmark instances. In this regard, we have selected 9 representative instances (15%) using the method proposed in [29] with a 90% PCA ratio and the suggested features for both the *MHC* and the *CR* matrices. The selected benchmark instances are highlighted in bold font in Table 1.

Once the benchmark instances are selected, the values for the parameters of the proposed algorithm were also automatically obtained. In this way, we used `irace`, a tool based on the iterated F-race method, able to obtain the best parameter values according to this statistical process [30, 31]. For each parameter configuration, the quality of the execution will be measured with the hypervolume which, as we will explain in Section 4.3, is the only multi-objective metric that does not require an additional reference set of solutions.

As we describe in Algorithm 1, our proposal requires parameter values for n_c and n_{ls}. These parameters are not fixed by `irace` but for us because, since they influence the execution time by incrementing the number of iterations of the algorithm, `irace` will always try to make them as high as possible. Hence, in order to limit the execution time, we set $n_c = 100$ and $n_{ls} = n$, where n is the size of the instance.

The parameters configured by `irace` are shown in Table 2. As seen, the first column (**Parameter**) indicates the parameter to configure, the second column (**Type**) is the type of parameter, the third column (**Range**) is the range of values of the parameter, and the fourth column (**Constraint**) indicates if the parameter has any additional constraint. Notice that all our parameters are real values in [0.00, 1.00] using two decimal symbols but the *LS* parameter. We use parameter α to configure how greedy or random we select candidates in our GRASP. Furthermore, there is a constraint named $R1$ for the parameters β_1 to β_4. This constraint means that the sum of these parameters must be equal to 1. The reason behind this constraint is that β_i are the percentage of use of each constructive. As we explained before, we

1

Table 1: Instances and their features. Instances included in the training set are depicted in bold.

Instance	m	c	n	Instance	m	c	n
previous6	**2**	**3**	**6**	previous12	3	4	12
previous8	2	4	8	previous15	3	5	15
A-10-10	2	5	10	A-20-30	4	5	20
A-10-20	2	5	10	A-20-40	4	5	20
A-10-30	2	5	10	**A-20-50**	**4**	**5**	**20**
A-10-40	2	5	10	A-20-60	4	5	20
A-10-50	2	5	10	A-20-70	4	5	20
A-10-60	2	5	10	A-20-80	4	5	20
A-10-70	**2**	**5**	**10**	A-20-90	4	5	20
A-10-80	2	5	10	N-20	4	5	20
A-10-90	2	5	10	O-20	4	5	20
0-10	2	5	10	S-20	4	5	20
Y-10	2	5	10	Y-20	4	5	20
A-12-10	2	6	12	A-25-10	5	5	25
A-12-20	2	6	12	A-25-20	5	5	25
A-12-30	2	6	12	A-25-30	5	5	25
A-12-40	2	6	12	A-25-40	5	5	25
A-12-50	2	6	12	A-25-50	5	5	25
A-12-60	2	6	12	**A-25-60**	**5**	**5**	**25**
A-12-70	**2**	**6**	**12**	A-25-70	5	5	25
A-12-80	2	6	12	A-25-80	5	5	25
A-12-90	2	6	12	A-25-90	5	5	25
S-12	2	6	12	S-25	5	5	25
Y-12	2	6	12	Y-25	5	5	25
N-15	3	5	15	Y-30	5	6	30
0-15	**3**	**5**	**15**	Y-35	5	7	35
S-15	3	5	15	Y-40	5	8	40
Y-15	**3**	**5**	**15**	**Y-45**	**5**	**9**	**45**
A-20-10	4	5	20	**Y-50**	**5**	**10**	**50**
A-20-20	4	5	20	Y-60	6	10	60

will use all the constructive methods, so these percentages have to sum 1. More formally:

$$\sum_{i=1}^{|\beta|} \beta_i = 1$$

We also use a parameter denoted as γ, which balances the use of the two available types of local search methods (see Section 3.2). For parameter LS we have two possible values: DA or AD. DA stands for dominance-based local search (DBLS) followed by alternate objectives local search(AOLS). AD is the other way around. Taking one value or another indicates the order in which the local search will be executed (see Section 3).

Table 2: Overview of the parameters obtained with irace

Parameter	Type	Range	Constraint
α	real	[0.00, 1.00]	-
β_1	real	[0.00, 1.00]	R1
β_2	real	[0.00, 1.00]	R1
β_3	real	[0.00, 1.00]	R1
β_4	real	[0.00, 1.00]	R1
γ	real	[0.00, 1.00]	-
LS	*categorical*	[DA, AD]	-

Table 3 shows the results provided by irace. Notice that each row of the table corresponds to a configuration of parameter values generated by irace. These configurations are listed from best to worst according to irace. As seen in the table, α takes values that are closely related to the values for the β parameters. For the two highest values of α, very close, and shown in configurations #1 and #3, the values of the four different β are also similar. In the same way, configurations #2 and #4, with smaller values of α, present a similar distribution of the β values. In all the cases, the participation of all the constructive methods is necessary, according to the obtained β values. It is also observed that γ reaches high values, which means that the algorithm mostly depends on the intensification performed by the DBLS search. However, some iterations of AOLS are always needed, since this method performs a wider exploration in the edges of the front. Finally, in the last column we can observe that the order AD is preferable to DA, which means that a first phase expanding the front followed by an intensification is preferred to the other way around.

Table 3: Overview of the configurations produced from `irace`

Config.	$\boldsymbol{\alpha}$	$\boldsymbol{\beta_1}$	$\boldsymbol{\beta_2}$	$\boldsymbol{\beta_3}$	$\boldsymbol{\beta_4}$	$\boldsymbol{\gamma}$	**LS**
#1	0.43	0.03	0.46	0.32	0.19	0.92	AD
#2	0.14	0.29	0.27	0.13	0.31	0.90	AD
#3	0.44	0.03	0.44	0.33	0.20	0.92	AD
#4	0.19	0.12	0.19	0.12	0.57	0.75	AD

Since the proposed four configurations are different, we will use all of them for both the comparison with the state of the art, and the solution of the proposed new instances. From this point on, we denote configuration $\#i$ in Table 3 as $GRASP_i$.

4.3. Comparison with state of the art

In this section we compare the sets of non-dominated solutions obtained with the four GRASP configurations of our proposal shown in the previous section, with the results from [11]. Firstly, we will perform a graphical comparison of the obtained solutions. In this way, the plots will denote the results from [11] as *NSBBO* and *NSGA-II*, since those are the algorithms that the previous authors proposed. Our solutions will be denoted as $GRASP_i$ for each configuration. The plots will show both objective functions, *MHC* and *CR*, in x-axis and y-axis respectively. It is worth noticing that, in the next four instances, our four configurations for the algorithm have several solutions in common. As result, the plots will show some overlapped solutions.

Figure 6 shows the resulting non-dominated solutions for instance *previous6*. In this case, the previous NSBBO and NSGA-II algorithm obtain the same solutions. For this reason, they are denoted as *Previous* in the graphic. In the same way, our four GRASP configurations reach the same solutions, but for $GRASP_3$ where solution (96, 76) is not found. As seen in the plot, the front formed by the GRASP solutions dominates the front from the previous work.

Figure 7 shows the resulting non-dominated solutions for instance *previous8*. In this case, the previous NSBBO and NSGA-II have different solutions and are denoted in that way. Just as in the previous instance, our four configurations for the GRASP approach share some of the solutions. Let solutions

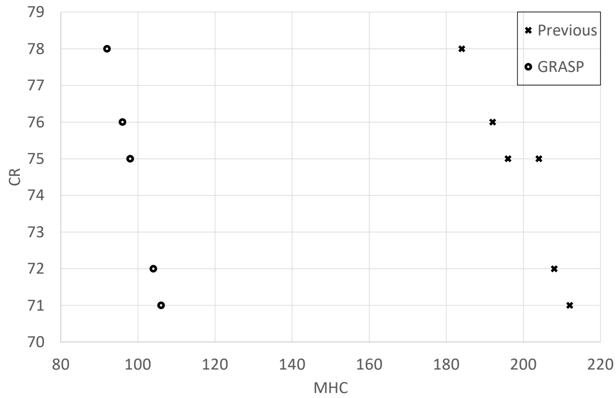


Figure 6: Results for instance *previous6*.

(179, 202), (199, 193), (207, 194), (213, 188) and (217, 178) be denoted as A, B, C, D and E, respectively. *GRASP$_1$* reaches solutions A, D and E; *GRASP$_2$* reaches solutions A, B, D and E; *GRASP$_3$* reaches solutions A and E; and *GRASP$_4$* reaches solutions A, C and E. Due to the overlapping of solutions, we have assigned different symbols to each one of our configurations for a better understanding. Again, the solutions obtained with the GRASP methods dominate the solutions from the literature.

Figure 8 shows the resulting non-dominated solutions for instance *previous12*. Here, the previous NSBBO and NSGA-II produced four and three solutions, respectively. As in the previous instances, our four configurations for the algorithm have several solutions in common that are overlapped between them. However, we have separated the fronts since the majority of the solutions are not overlapped. Again, the solutions from the literature are dominated by our results.

Finally, Figure 9 shows the resulting non-dominated solutions for instance *previous15*. For this instance, the previous NSBBO and NSGA-II produced six and nine solutions, respectively. Just as in the previous instances, our algorithms outperform the results from the literature.

In the graphic representation it is not possible to determine which one of our GRASP configurations is the best. However, it is clear that the methods from the literature, NSBBO and NSGA-II, have difficulties in reaching good quality solutions, since all of them are dominated by the solutions from our

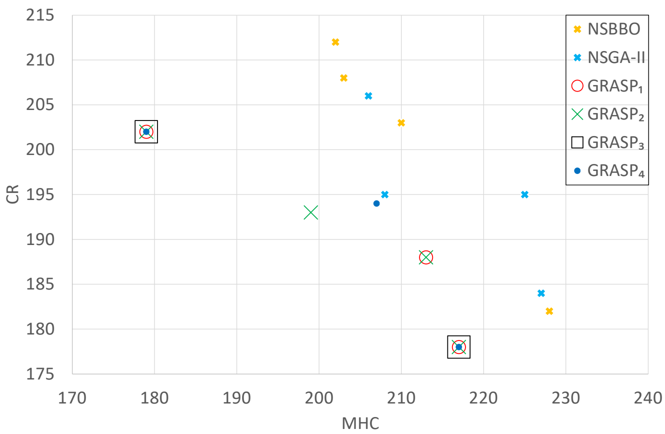


Figure 7: Results for instance *previous8*.

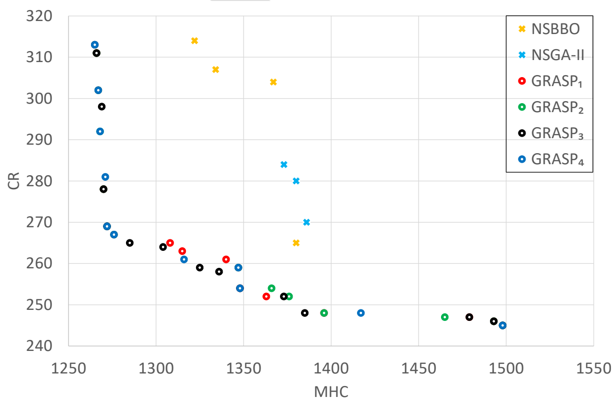


Figure 8: Results for instance *previous12*.

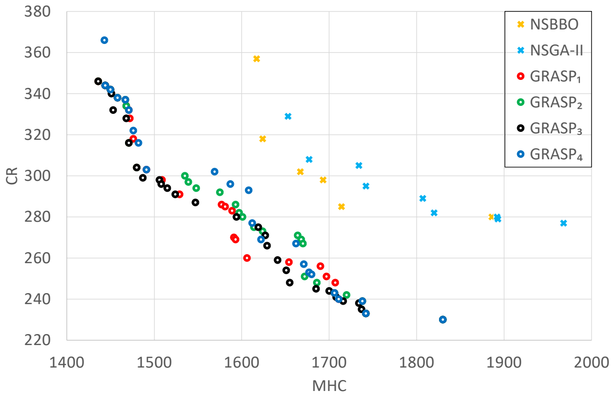


Figure 9: Results for instance *previous15*.

GRASP configurations. In addition, both NSBBO and NSGA-II algorithms have reported very few solutions. On average, our set of non-dominated solutions has similar size of both previous algorithms in instances *previous6* and *previous8*, which are the smallest of the study, and has nearly three times the size of both previous algorithms in instances *previous12* and *previous15*. As seen, the larger the instance, the larger the set of non-dominated solutions generated by our proposed methods.

The evaluation of these findings is also conducted with traditional multi-objective metrics [25]. For these metrics, except hypervolume, it is mandatory to have a reference set [32, 33]. Since this set is not available for the current set of instances, we have created it. Hence, for each instance, we obtain a set of non-dominated solutions per algorithm, denoted as the name of the algorithm. Then, we obtain the set of non-dominated solutions from the union of all these sets. Therefore, this Ref set only contains non-dominated solutions:

$$Ref \leftarrow \texttt{Update}(\texttt{GRASP}_1 \cup \texttt{GRASP}_2 \cup \texttt{GRASP}_3 \cup \texttt{GRASP}_4 \cup \texttt{NSBBO} \cup \texttt{NSGA}-\texttt{II})$$

Let Ref be a set of reference solutions and S be a set of non-dominated solutions, $|Ref|$ and $|S|$ correspond to the size of each set, respectively. Then, we can define the studied metrics as follows.

The coverage metric ($\mathrm{C}(S_1, S_2)$) measures the proportion of solutions in a set S_2 that are weakly dominated by solutions in set S_1. Equation (6) shows the definition considering the reference set and a given set S. Notice that $C(Ref, S) = 1$ indicates that all the solutions from S are *weakly dominated* by solutions from Ref.

$$\mathrm{C}(Ref, S) = \frac{|\{s \in S \mid \exists r \in Ref : r \text{ weakly dominates } s\}|}{|S|} \tag{6}$$

The hypervolume (HV) measures the volume of the objective space that is dominated by the set of non-dominated solutions and bounded by a reference point. The reference point is typically chosen to be a point that is worse than any feasible solution in all objectives. The larger the hypervolume, the better the quality of the set, as it indicates that a larger portion of the objective space is covered. Equation (7) shows the formal definition, where Volume denotes the volume calculation, $\bigcup_{s \in S}$ is the union of all the hypercubes formed by each solution s in the set S and the reference point r, and $\mathrm{Hypercube}(s, r)$ is the hypercube formed by the solution s and the reference point r.

$$\mathrm{HV}(S) = \mathrm{Volume}\left(\bigcup_{s \in S} \mathrm{Hypercube}(s, r)\right) \tag{7}$$

The epsilon $(\epsilon(S_1, S_2))$ calculates the smallest value of ϵ such that each solution in S_2 is weakly dominated by ϵ times some solution in S_1. Therefore, it measures how much it would take to make a set worse in the objective space to dominate another set. It provides a quantifiable measure of the degree to which one set of non-dominated solutions is better or worse than another. Equation (8) shows the definition considering the reference set and a given set S, where $r \leq \epsilon \cdot s$ means each objective of solution r is less than or equal to ϵ times the corresponding objective of solution s.

$$\epsilon(Ref, S) = \inf\{\epsilon \in \mathbb{R} : \forall s \in S, \exists r \in Ref \text{ such that } r \leq \epsilon \cdot s\} \tag{8}$$

The Generational Distance $(\mathrm{GD}(Ref, S))$ measures the distance from a set of non-dominated solutions S to the reference set. It is a measure of how close the solutions generated by the algorithm are to the reference solutions. Equation (9) shows the definition, where $\mathrm{dist}(s, Ref)$ is the minimum distance from a solution s in set S to the nearest point on the *Ref*, and p is the order of the norm used for the distance calculation.

$$\text{GD}(Ref, S) = \left(\frac{1}{|S|} \sum_{s \in S} \text{dist}(s, Ref) \right)^{1/p} \quad (9)$$

The Inverted Generational Distance (IGD(Ref, S)) measures the distance from the points on the reference set to a set of non-dominated solutions S. It can be calculated as an inverse of GD, as shown in Equation (10).

$$\text{IGD}(Ref, S) = \text{GD}(S, Ref) \quad (10)$$

The Inverted Generational Additive Distance (IGD$^+$) is a variant of IGD. While IGD focuses on the minimum distance from each point on the reference set to the nearest solution in the other set, IGD$^+$ considers the sum of all distances from each point on the reference set to its nearest neighbor in the other set. This approach makes IGD$^+$ more sensitive to outliers and provides a comprehensive measure of the overall distribution of the solutions. Equation (11) shows the definition, where dist(r, S) is the minimum distance from a solution r in *Ref* to the nearest point on the set of non-dominated solutions S.

$$\text{IGD}^+(Ref, S) = \frac{1}{|Ref|} \sum_{r \in Ref} \text{dist}(r, S) \quad (11)$$

The spread ($\Delta(S, Ref)$) measures the distribution and range of solutions across the set of non-dominated solutions S. Specifically, it is designed to assess the extent to which the obtained solutions cover the range of the reference set, and the distribution uniformity of these solutions along the set. Equation (12) shows the definition, where d_i is the Euclidean distance between consecutive solutions in S, $\bar{d}$ is the average of these distances, and d_f and d_l are the distances from the extreme solutions of the obtained set to the extremes of the Ref.

$$\Delta(S, Ref) = \frac{\sum_{i=1}^{|S|-1} |d_i - \bar{d}| + d_f + d_l}{\sum_{i=1}^{|S|-1} d_i + d_f + d_l} \quad (12)$$

In addition to these typical multi-objective metrics, we have included in the analysis the number of solutions of each set, denoted as *Size*, and the execution time spent in seconds. We have calculated those metrics using *jMetal* [34]. In all the metrics but *HV* and *Size*, the lower the value, the better the results.

Table 4 shows the results of the GRASP approaches in relation to the state-of-the-art methods. As seen, all the GRASP configurations obtain better results than the previous algorithms in all the metrics. Besides, it can be observed that the $GRASP_4$ configuration reaches better values than the other GRASP configurations for 7 out the 8 metrics. Since these results could be counterintuitive considering the outcomes of `irace` it is worth noticing that `irace` only considered a representative subset of the instances and used the hypervolume as the metric to compare (see Section 4.2). Therefore, due to the use of this metric and the fact that the final configurations are very similar, it is possible that any of them could present better performance when executed on different instances, as happens in this case.

Table 4: Overview for comparative analysis using metrics against the state-of-the-art.

Algorithm	C	HV	ϵ	GD	IGD	IGD^+	Δ	Size
$GRASP_1$	0.24	**0.50**	**0.08**	218.50	182.88	795.24	0.96	13.83
$GRASP_2$	0.26	0.49	**0.08**	221.49	182.88	795.18	**0.95**	13.63
$GRASP_3$	0.26	0.49	**0.08**	221.52	182.85	795.29	**0.95**	13.94
$GRASP_4$	**0.21**	0.49	**0.08**	**216.87**	**182.57**	**793.66**	**0.95**	**14.06**
NSBBO	1.00	0.14	2.04	405.42	204.07	875.45	0.98	5.00
NSGA-II	0.99	0.13	2.09	420.28	207.20	892.54	0.97	5.00

The detailed results for each instance and algorithm can be found in Appendix A, where the instances are distributed according to their size. In addition, we have included Table A.7, where we detail the results in the state-of-the-art instances.

4.4. Results for the new instances

As it will be shown in this section, the number of solutions obtained for the new instances is much higher than in the comparison with the state of the art. Besides, our algorithms obtain sets of solutions whose intersection is not empty. Therefore, a graphical comparison is difficult in this case. Hence, the analysis of results for the new set of instances will be performed using the multi-objective metrics described in the previous section: coverage (C), hypervolume (HV), epsilon (ϵ), generational distance (GD), inverse generational distance (IGD), additive inverse generational distance (IGD^+), and spread (Δ).

Since we have no access to the algorithms from [11], we have included in this experiment two classical multi-objective evolutionary algorithms: NSGA-II [35] and SPEA2 [36]. These algorithms were implemented using *jMetal* in

Table 5: Overview of the configurations produced from `irace` for the NSGA-II and SPEA2 implementations.

Config.	**Algorithm**	C_p	M_p
#1	NSGA-II	0.51	0.28
#2	NSGA-II	0.73	0.28
#3	NSGA-II	0.50	0.28
#1	SPEA2	0.93	0.12
#2	SPEA2	0.80	0.21
#3	SPEA2	0.90	0.09

Java, the same programming language used to implement the GRASP approach, adapting the codification and genetic operators to this problem. In particular, the chromosome stored a linear representation of the facilities, and the genetic operators included repair methods to conserve the permutation form. To be fair in the comparison, we run these algorithms in the same machine as the GRASP configurations, and we also executed `irace` to obtain the best parameter values for these algorithms fixing the population size and number of generations to 100 and 1000, respectively, to avoid long execution times. In addition, each algorithm has been executed 30 times. Table 5 shows the obtained parameter values. These configurations are listed from best to worst according to `irace`. The third and fourth columns denote the crossover operator probability (C_p), and the mutation operator probability (M_p). For the experimental comparison, we have chosen the best configuration for each algorithm.

Table 6 shows the average values of the aforementioned metrics obtained with the four GRASP configurations described in Section 4.2 and the two evolutionary methods, highlighting the best results in bold font. As expected, all the GRASP configurations have similar results. However, GRASP$_4$ obtains the best values for all metrics. Notice that size values obtained by the evolutionary methods are restricted by the population size, fixed to 100. Overall, all configurations of our proposed algorithm obtain competitive results and there are no big differences between them.

Again, the performance of the GRASP configurations is not aligned with the results given by `irace`. However, the reason is the same as explained in the previous section.

In summary, the combination of algorithmic elements under the proposed

Table 6: Average performance indicators for all the instances.

Algorithm	C	HV	ϵ	GD	IGD	IGD$^+$	Δ	Size	Time(s)
$GRASP_1$	0.55	0.63	0.09	352.77	327.09	3433.20	**0.95**	62.02	3.93
$GRASP_2$	0.56	0.63	0.09	352.00	327.14	3433.57	**0.95**	62.23	3.90
$GRASP_3$	0.55	0.63	0.09	351.76	327.07	3432.76	**0.95**	62.28	3.91
$GRASP_4$	**0.50**	**0.64**	**0.07**	**348.36**	**326.76**	**3430.27**	**0.95**	**63.31**	**3.40**
NSGA-II	0.58	0.62	0.11	491.50	328.29	3445.70	0.96	30.52	9.78
SPEA2	0.72	0.56	0.17	519.23	330.63	3467.50	0.96	26.05	9.78

GRASP method is able to consistently obtain the best results in large instances of this problem. This way, the method is both robust and scales well with the size of the instances. Again, the detailed results for each instance and algorithm can be found in Appendix A. The correctness of the obtained solutions was verified with the model proposed in [11].

4.5. Statistical analysis

The aim of this section is to conduct a statistical analysis to compare the four configurations of our GRASP approach and the NSGA-II and SPEA2 implementations. To do so, we have used the previous metrics obtained for the instances whose detailed results are shown in Appendix A. For the comparative assessment of these algorithms, a multi-algorithm multi-instance Bayesian analysis, described in [37] and [38], has been employed. This analysis facilitates the ranking of algorithms based on the quality of the obtained solutions, providing a ranking of algorithms based on a designated probability distribution derived after examining the outcomes. Hence, the probability of each algorithm's potential to outperform its counterparts is calculated. This metric is referred to as the "probability of winning", which is the percentage chance for each algorithm to obtain better results than the others, with a credible interval between 5% and 95%. For the sake of brevity, we have rounded the probability of winning using two decimals.

The results of this analysis are depicted in Figure 10. The absence of overlapping between the SPEA2 and NSGA-II algorithms in Figure 10 (a)-(g) is notably evident, which means that NSGA-II will always obtain better results than SPEA2. Furthermore, it should be highlighted that there is no overlapping between the $GRASP_4$ and NSGA-II, and $GRASP_3$ and NSGA-II. In addition, there are small overlaps between $GRASP_1$ and NSGA-II, and $GRASP_2$ and NSGA-II in Figure 10 (a), (b) and (g), meaning that the performance of both algorithms is indistinguishable for certain instances, which, as seen in the detailed results, is due to the smallest instances. Since there are no overlaps between GRASP_4 and the other GRASP configurations, the conclusion is that the performance of GRASP_4 will be better given a new instance.

In summary, the GRASP configurations outperform the NSGA-II and SPEA2 algorithms. Among the different GRASP configurations, the GRASP_4 reached the highest probability since, as shown in the experimental experience, it obtained the best results.

5. Managerial implications

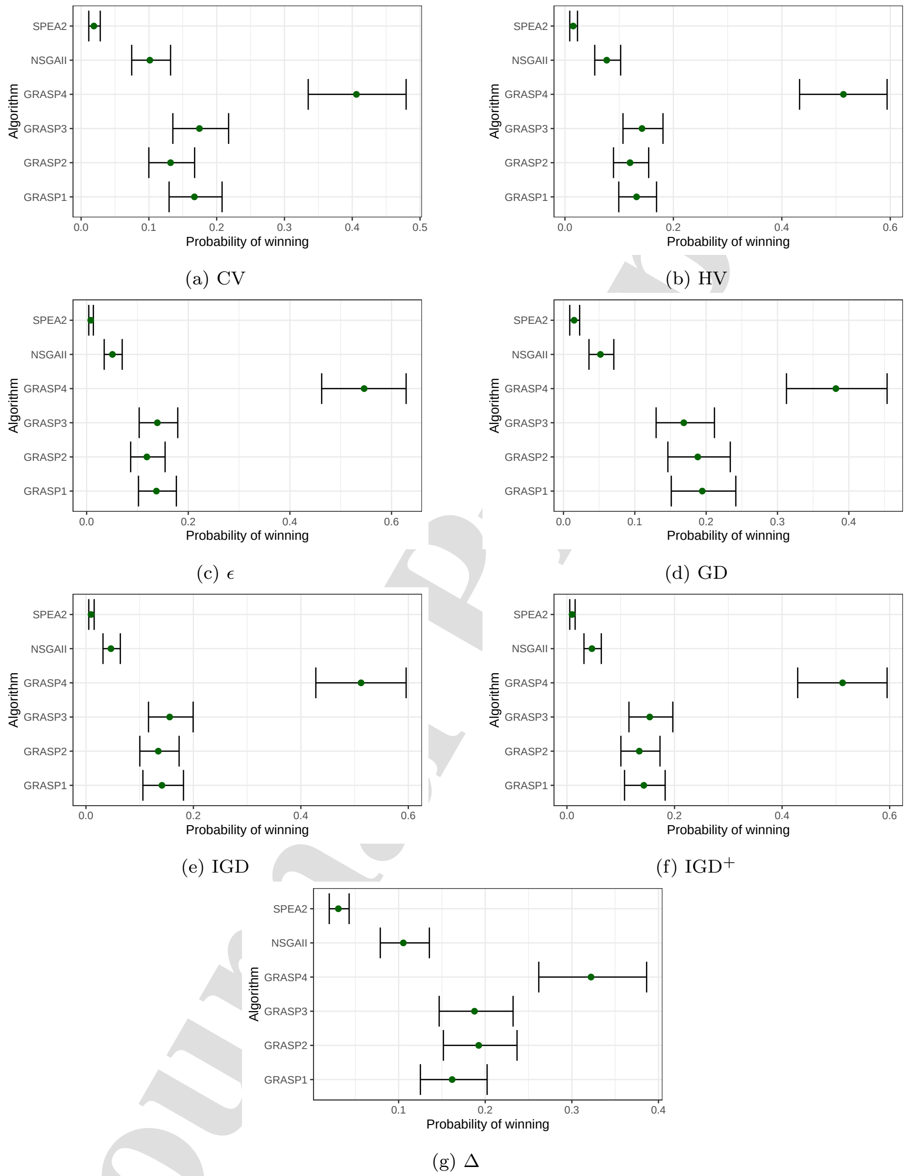
The design of a facility layout may involve different objectives due to the nature of the facilities at hand. In this work, we selected the material handling cost (MHC) and the closeness rating (CR) as the two objectives to be considered, since they represent different aspects, many times opposed, of the design process. On the one hand, MHC represents the flow of material between different machines. This way, the higher the value, the better to be close. On the other hand, CR represents the adequacy level of two machines to be close. In this regard, two machines that may present a higher MHC value could also present a high value of CR, meaning that they should be placed far away due to their heat generation, power consumption, or other safety measures.

Our proposed algorithm is able to both generate different solutions for large instances and to produce a high number of non-dominated solutions. These features give the decision maker a clear added value, since larger projects could be studied, and, once the solutions were presented, many different possibilities could be considered. In this way, the decision maker will be able to select the best solution according to any other additional criteria.

6. Conclusions and future work

The Bi-Objective Multiple Row Equal Facility Layout Problem (BO-MREFLP) is an interesting problem from the FLP family which takes into account both quantitative and qualitative objectives. Therefore, many real-life applications can be aligned with this problem.

In this work, we tackle the BO-MREFLP from a multi-objective point of view, considering both the material handling cost and the closeness rating objective functions. To this aim, we have designed an ensemble of constructive methods able to produce a diverse set of non-dominated solutions under



(a) CV (b) HV (c) ϵ (d) GD (e) IGD (f) IGD^{+} (g) Δ

Figure 10: Credible intervals for the probability of reaching the best results.

a GRASP methodology. The solutions are improved by means of a new local search method that combines dominance and alternation between both objectives, under a hybrid *best-improvement first-improvement* strategy.

Since the previous work in this problem provided a small set of instances, we have expanded 56 instances taken from the single-objective problem with matrices for the second objective, generating a new benchmark dataset for the research community.

The thorough experimental experience proved that our proposal, after an automatic tuning of the parameters, is able to produce much better results than the state of the art in the previous set of instances. Besides, detailed results are provided for the new set of instances, including some of the classical metrics for the multi-objective problems. Our experimental results, algorithms, and benchmarks could also be incorporated into other multi-objective problems of the FLP family.

However, the main limitation of our work is the computation time spent evaluating the dominated solutions that will not be part of the final set of dominated solutions. Although we have included an efficient computation of both objective functions, some heuristics could be developed to reduce the search space in the improvement phase, therefore, reducing the total computation time. In this regard, the use of the crowding distance or clustering techniques to select the solutions to be improved could help reducing the computational effort.

Acknowledgements

This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación (MCIN/AEI/10.13039/501100011033/FEDER, UE) under grant refs. PID2021-126605NB-I00 and RED2022-134480-T.

References

[1] H. Hosseini-Nasab, S. Fereidouni, S. M. T. Fatemi Ghomi, M. B. Fakhrzad, Classification of facility layout problems: a review study, The International Journal of Advanced Manufacturing Technology 94 (1) (2018) 957–977.

[2] M. Besbes, M. Zolghadri, R. Costa Affonso, F. Masmoudi, M. Haddar, A methodology for solving facility layout problem considering barriers:

genetic algorithm coupled with A* search, Journal of Intelligent Manufacturing 31 (3) (2020) 615–640.

[3] M. F. Anjos, M. V. Vieira, Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions, European Journal of Operational Research 261 (1) (2017) 1 – 16.

[4] S. Safarzadeh, H. Koosha, Solving an extended multi-row facility layout problem with fuzzy clearances using GA, Applied Soft Computing 61 (2017) 819–831.

[5] M. Rubio-Sánchez, M. Gallego, F. Gortázar, A. Duarte, GRASP with path relinking for the single row facility layout problem, Knowledge-Based Systems 106 (2016) 1–13.

[6] D. M. Simmons, One-dimensional space allocation: An ordering algorithm, Operations Research 17 (5) (1969) 812–826.

[7] J. Guan, G. Lin, H.-B. Feng, Z.-Q. Ruan, A decomposition-based algorithm for the double row layout problem, Applied Mathematical Modelling 77 (2020) 963–979.

[8] A. Herrán, J. Manuel Colmenar, A. Duarte, An efficient variable neighborhood search for the space-free multi-row facility layout problem, European Journal of Operational Research 295 (3) (2021) 893–907.

[9] N. R. Uribe, A. Herrán, J. M. Colmenar, A. Duarte, An improved grasp method for the multiple row equal facility layout problem, Expert Systems with Applications 182 (2021) 115184.

[10] L. Tang, Z. Li, J.-K. Hao, Solving the single row facility layout problem by K-Medoids Memetic Permutation Group, IEEE Transactions on Evolutionary Computation (2022) 1–1.

[11] D. Singh, S. Ingole, Multi-objective facility layout problems using BBO, NSBBO and NSGA-II metaheuristic algorithms, International Journal of Industrial Engineering Computations (2019) 239–262.

[12] Z. Kalita, D. Datta, G. Palubeckis, Bi-objective corridor allocation problem using a permutation-based genetic algorithm hybridized with a local search technique, Soft Computing 23 (3) (2019) 961–986.

[13] M. J. Rosenblatt, The facilities layout problem: a multi-goal approach, International Journal of Production Research 17 (1979) 323–332.

[14] K. N. Dutta, S. Sahu, A multigoal heuristic for facilities design problems: MUGHAL, International Journal of Production Research 20 (1982) 147–154.

[15] J. C. Fortenberry, J. F. Cox, Multiple criteria approach to the facilities layout problem, International Journal of Production Research 23 (1985) 773–782.

[16] C. M. Harmonosky, G. K. Tothero, A multi-factor plant layout methodology, International Journal of Production Research 30 (1992) 1773–1789.

[17] C.-W. Chen, A design approach to the multi-objective facility layout problem, International Journal of Production Research 37 (1999) 1175–1196.

[18] R. Matai, S. Singh, M. Mittal, Modified simulated annealing based approach for multi objective facility layout problem, International Journal of Production Research 51 (2013) 4273–4288.

[19] S. Singh, V. Singh, An improved heuristic approach for multi-objective facility layout problem, International Journal of Production Research 48 (4) (2010) 1171–1194.

[20] T. Feo, M. Resende, A probabilistic heuristic for a computationally difficult set covering problem, Operations Research Letters 8 (1989) 67–71.

[21] J. E. C. Arroyo, A. A. de Souza Pereira, A GRASP heuristic for the multi-objective permutation flowshop scheduling problem, The International Journal of Advanced Manufacturing Technology 55 (5) (2011) 741–753.

[22] H. C. Gomes, F. de Assis das Neves, M. J. F. Souza, Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations, Computers & Operations Research 44 (2014) 92–104.

[23] X. Wan, X. Zuo, X. Li, X. Zhao, A hybrid multiobjective GRASP for a multi-row facility layout problem with extra clearances, International Journal of Production Research 60 (3) (2022) 957–976.

[24] A. Estrada-Padilla, C. Gómez-Santillán, H. J. Fraire-Huacuja, L. Cruz-Reyes, N. Rangel-Valdez, M. L. Morales-Rodríguez, H. J. Puga-Soberanes, GRASP/δ: An efficient algorithm for the multi-objective portfolio optimization problem, Expert Systems with Applications 211 (2023) 118647.

[25] I. Lozano-Osorio, J. Sanchez-Oro, A. D. Lopez-Sanchez, A. Duarte, A reactive path relinking algorithm for solving the bi-objective p-median and p-dispersion problem, Soft Computing 27 (12) (2023) 8029–8059.

[26] V. Pérez-Piqueras, P. B. López, J. A. Gámez, GRASP-based hybrid search to solve the multi-objective requirements selection problem, in: International Conference on Optimization and Learning, Springer, 2022, pp. 189–200.

[27] J. J. Pantrigo, R. Martí, A. Duarte, E. G. Pardo, Scatter search for the cutwidth minimization problem, Annals of Operations Research 199 (2012) 285–304.

[28] J. Colmenar, R. Martí, A. Duarte, Multi-objective memetic optimization for the bi-objective obnoxious p-median problem, Knowledge-Based Systems 144 (2018) 88–101.

[29] R. Martín-Santamaría, S. Cavero, A. Herrán, A. Duarte, J. M. Colmenar, A practical methodology for reproducible experimentation: an application to the Double-row Facility Layout Problem, Evolutionary Computation (2022) 1–35.

[30] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, T. Stützle, The irace package: Iterated racing for automatic algorithm configuration, Operations Research Perspectives 3 (2016) 43–58.

[31] T. Stützle, M. López-Ibáñez, Automated algorithm configuration and design, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 959–982.

[32] J. D. Knowles, L. Thiele, E. Zitzler, A tutorial on the performance assessment of stochastic multiobjective optimizers, Vol. 214, ETH Zurich, 2006.

[33] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, V. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.

[34] J. J. Durillo, A. J. Nebro, jmetal: A java framework for multi-objective optimization, Advances in Engineering Software 42 (10) (2011) 760–771.

[35] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[36] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization, in: K. Giannakoglou, et al. (Eds.), Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), International Center for Numerical Methods in Engineering (CIMNE), 2002, pp. 95–100.

[37] B. Calvo, J. Ceberio, J. A. Lozano, Bayesian inference for algorithm ranking analysis, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 324–325.

[38] B. Calvo, O. M. Shir, J. Ceberio, C. Doerr, H. Wang, T. Bäck, J. A. Lozano, Bayesian performance analysis for black-box optimization benchmarking, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 1789–1797.

Appendix A. Detailed results

Table A.7, Table A.8, Table A.9, Table A.10, Table A.11, Table A.12, and Table A.13 show the detailed results for the complete set of instances using the multi-objective metrics explained in Section 4.4. For the sake of brevity, two decimal digits are shown in the figures of the tables. In some of the cases, an instance has the same values for the 4 configurations of our proposal; however, we highlight in bold some of these values. The values highlighted in bold are the best values, but considering more decimals.

Table A.7: Comparison among the 4 different configurations of our GRASP proposal, an NSGA-II and a SPEA2 algorithms, for the set of previous instances. Best values for the metrics are highlighted with **bold**.

Instance	Algorithm	C	HV	ϵ	GD	IGD	IGD$^+$	Δ	Size	Time(s)
previous6	$GRASP_1$	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	**0.96**	**5.00**	0.00
	$GRASP_2$	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	0.96	**5.00**	0.00
	$GRASP_3$	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	0.96	**5.00**	**0.00**
	$GRASP_4$	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	0.96	**5.00**	**0.00**
	NSGA-II	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	0.96	**5.00**	1.00
	SPEA2	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	0.96	**5.00**	1.00
previous8	$GRASP_1$	0.10	0.13	0.22	153.89	120.66	269.81	0.92	3.37	0.00
	$GRASP_2$	0.15	0.11	0.23	159.05	120.82	270.17	0.90	3.20	0.00
	$GRASP_3$	0.07	0.10	0.24	165.78	120.45	269.32	**0.90**	2.90	0.00
	$GRASP_4$	**0.07**	0.10	0.25	161.68	120.45	269.32	0.91	3.07	0.00
	NSGA-II	0.09	0.18	**0.06**	**126.60**	**120.34**	**269.08**	0.95	**4.90**	1.00
	SPEA2	0.12	**0.19**	0.08	129.02	120.45	269.32	0.94	4.70	1.00
previous12	$GRASP_1$	0.39	0.83	0.06	390.04	308.16	1301.18	0.98	12.57	0.02
	$GRASP_2$	0.38	0.84	0.06	395.77	307.97	1300.41	0.98	12.20	**0.02**
	$GRASP_3$	0.41	0.84	0.06	393.93	308.09	1300.95	0.98	12.33	0.02
	$GRASP_4$	**0.33**	**0.84**	**0.05**	**377.91**	307.94	1300.31	**0.97**	**13.23**	0.02
	NSGA-II	0.37	0.83	0.06	396.80	**307.93**	**1300.24**	0.97	12.20	1.00
	SPEA2	0.44	0.83	0.07	427.41	307.98	1300.46	0.98	10.50	1.00
previous15	$GRASP_1$	0.51	0.67	0.06	274.90	231.93	1490.05	**0.96**	34.37	0.07
	$GRASP_2$	0.56	0.67	0.06	275.98	231.95	1490.22	0.96	34.13	0.07
	$GRASP_3$	0.62	0.67	0.07	**271.20**	232.09	1490.97	0.96	**35.53**	0.07
	$GRASP_4$	**0.49**	**0.68**	**0.05**	272.73	**231.16**	**1485.10**	0.96	34.93	**0.07**
	NSGA-II	0.67	0.66	0.09	276.28	233.11	1497.56	0.97	34.20	1.00
	SPEA2	0.91	0.62	0.13	328.00	234.30	1505.38	0.97	25.10	1.00

Table A.8: Comparison among the 4 different configurations of our GRASP proposal, an NSGA-II and a SPEA2 algorithms, for a set of instances with size [6, 10]. Best values for the metrics are highlighted with **bold**.

Instance	Algorithm	C	HV	ϵ	GD	IGD	IGD$^+$	Δ	Size	Time(s)
previous6	GRASP$_1$	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	**0.96**	**5.00**	0.00
	GRASP$_2$	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	0.96	**5.00**	0.00
	GRASP$_3$	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	0.96	**5.00**	**0.00**
	GRASP$_4$	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	0.96	**5.00**	**0.00**
	NSGA-II	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	0.96	**5.00**	1.00
	SPEA2	**0.00**	**0.35**	**0.00**	**55.16**	**53.63**	**119.91**	0.96	**5.00**	1.00
previous8	GRASP$_1$	0.10	0.13	0.22	153.89	120.66	269.81	0.92	3.37	0.00
	GRASP$_2$	0.15	0.11	0.23	159.05	120.82	270.17	0.90	3.20	0.00
	GRASP$_3$	0.07	0.10	0.24	165.78	120.45	269.32	**0.90**	2.90	0.00
	GRASP$_4$	**0.07**	0.10	0.25	161.68	120.45	269.32	0.91	3.07	0.00
	NSGA-II	0.09	0.18	**0.06**	**126.60**	**120.34**	**269.08**	0.95	**4.90**	1.00
	SPEA2	0.12	**0.19**	0.08	129.02	120.45	269.32	0.94	4.70	1.00
A-10-10	GRASP$_1$	0.38	0.31	0.41	47.65	37.66	83.54	0.96	3.57	0.01
	GRASP$_2$	0.40	0.34	0.37	44.08	37.46	83.10	0.95	4.20	**0.01**
	GRASP$_3$	0.37	0.33	0.37	43.90	37.29	82.74	0.95	4.13	0.01
	GRASP$_4$	0.29	**0.38**	**0.29**	**41.20**	**37.07**	**82.25**	**0.95**	**4.57**	0.01
	NSGA-II	0.24	0.33	0.37	46.17	37.62	83.44	0.96	3.83	1.00
	SPEA2	**0.22**	0.32	0.38	48.40	37.58	83.37	0.96	3.40	1.00
A-10-20	GRASP$_1$	0.16	0.60	0.09	46.61	41.07	116.16	**0.96**	7.23	0.01
	GRASP$_2$	**0.11**	**0.61**	**0.07**	46.03	**40.91**	**115.72**	0.97	**7.37**	0.01
	GRASP$_3$	0.20	0.59	0.13	47.28	41.49	117.36	0.97	7.07	**0.01**
	GRASP$_4$	0.13	0.61	0.07	**45.94**	40.91	115.72	0.97	**7.37**	0.01
	NSGA-II	0.11	0.60	0.12	46.73	41.38	117.04	0.97	7.20	1.00
	SPEA2	0.19	0.59	0.15	47.89	41.61	117.69	0.97	6.93	1.00
A-10-30	GRASP$_1$	0.14	0.60	0.06	33.32	**25.62**	**81.03**	0.88	9.27	**0.01**
	GRASP$_2$	0.15	0.60	0.07	34.01	**25.62**	**81.03**	0.87	8.73	0.01
	GRASP$_3$	0.11	0.60	0.06	33.33	**25.62**	**81.03**	0.88	9.17	0.01
	GRASP$_4$	0.15	0.60	0.06	33.44	**25.62**	**81.03**	0.87	9.13	0.01
	NSGA-II	**0.09**	**0.61**	**0.04**	**32.62**	**25.62**	**81.03**	**0.86**	**9.57**	1.00
	SPEA2	0.17	0.60	0.05	33.03	25.68	81.22	0.88	9.43	1.00
A-10-40	GRASP$_1$	0.31	0.64	0.09	22.67	19.17	73.91	0.85	13.17	0.01
	GRASP$_2$	0.21	0.64	0.11	23.65	19.16	73.85	0.85	12.30	0.01
	GRASP$_3$	0.23	0.64	0.11	23.17	19.21	74.04	0.85	12.63	**0.01**
	GRASP$_4$	0.19	0.65	0.09	22.15	19.21	74.07	**0.83**	13.60	0.01
	NSGA-II	**0.08**	**0.66**	**0.07**	**22.00**	**19.03**	**73.35**	0.84	**13.70**	1.00
	SPEA2	0.16	0.65	0.11	23.06	19.08	73.56	0.85	12.60	1.00
A-10-50	GRASP$_1$	0.34	0.50	0.10	**25.38**	23.24	82.27	0.89	**11.93**	0.01
	GRASP$_2$	0.36	0.50	0.12	25.69	23.25	82.29	0.89	11.70	0.01
	GRASP$_3$	0.31	0.50	0.12	26.27	23.19	82.09	0.89	11.27	0.01
	GRASP$_4$	0.31	0.50	0.10	25.76	23.22	82.18	0.89	11.60	**0.01**
	NSGA-II	**0.15**	**0.53**	**0.08**	25.47	**23.07**	**81.67**	0.89	11.73	1.00
	SPEA2	0.34	0.50	0.13	26.22	23.17	82.01	**0.89**	11.20	1.00
A-10-60	GRASP$_1$	0.52	0.55	0.15	59.83	**54.70**	**140.52**	0.93	6.43	0.01
	GRASP$_2$	0.59	0.55	0.14	**58.13**	54.76	140.69	0.93	**6.80**	0.01
	GRASP$_3$	0.54	0.54	0.15	61.14	54.83	140.85	0.93	6.13	0.01
	GRASP$_4$	0.54	0.55	0.14	60.20	54.75	140.67	**0.93**	6.40	**0.01**
	NSGA-II	**0.36**	**0.57**	**0.12**	60.05	54.77	140.71	0.94	6.23	1.00
	SPEA2	0.51	0.56	0.14	59.09	55.02	141.34	0.94	6.57	1.00
A-10-70	GRASP$_1$	0.35	0.63	0.07	39.16	32.38	119.80	**0.92**	11.27	0.01
	GRASP$_2$	0.35	0.63	0.07	38.70	32.41	119.92	0.93	11.50	0.01
	GRASP$_3$	0.33	0.63	0.07	39.45	32.44	120.02	0.93	11.07	**0.01**
	GRASP$_4$	0.23	0.64	0.06	38.02	**32.32**	**119.57**	0.93	11.80	0.01
	NSGA-II	**0.13**	**0.65**	**0.05**	**37.03**	32.44	120.01	0.94	**12.23**	1.00
	SPEA2	0.20	0.64	0.06	38.69	32.46	120.08	0.93	11.40	1.00
A-10-80	GRASP$_1$	0.42	0.43	0.22	59.80	50.34	123.31	0.98	4.70	0.01
	GRASP$_2$	0.48	0.44	0.20	62.18	50.31	123.22	0.98	4.40	0.01
	GRASP$_3$	0.45	0.42	0.19	59.43	50.19	122.93	0.98	4.73	0.01
	GRASP$_4$	0.33	0.44	0.16	57.67	50.21	122.98	**0.96**	5.13	**0.01**
	NSGA-II	**0.06**	**0.51**	**0.06**	**55.07**	**50.12**	**122.76**	0.97	**5.40**	1.00
	SPEA2	0.13	0.50	0.12	57.25	50.15	122.85	0.97	5.03	1.00
A-10-90	GRASP$_1$	**0.00**	**0.33**	**0.00**	**77.83**	**76.26**	**152.52**	**0.97**	**4.00**	0.01
	GRASP$_2$	**0.00**	**0.33**	**0.00**	**77.83**	**76.26**	**152.52**	**0.97**	**4.00**	0.01
	GRASP$_3$	**0.00**	**0.33**	**0.00**	**77.83**	**76.26**	**152.52**	**0.97**	**4.00**	0.01
	GRASP$_4$	**0.00**	**0.33**	**0.00**	**77.83**	**76.26**	**152.52**	**0.97**	**4.00**	**0.00**
	NSGA-II	**0.00**	**0.33**	**0.00**	**77.83**	**76.26**	**152.52**	**0.97**	**4.00**	1.00
	SPEA2	**0.00**	**0.33**	**0.00**	**77.83**	**76.26**	**152.52**	**0.97**	**4.00**	1.00

Table A.9: Comparison among the 4 different configurations of our GRASP proposal, an NSGA-II and a SPEA2 algorithms, for a set of instances with size [10, 12]. Best values for the metrics are highlighted with **bold**.

Instance	Algorithm	C	HV	ϵ	GD	IGD	IGD$^+$	Δ	Size	Time(s)
O-10	GRASP$_1$	0.29	0.76	0.05	**241.55**	213.59	926.79	0.97	**17.43**	**0.01**
	GRASP$_2$	0.21	0.76	0.04	252.25	213.61	926.87	**0.97**	15.80	0.01
	GRASP$_3$	0.17	0.76	0.04	248.01	213.40	925.96	0.97	16.27	0.01
	GRASP$_4$	0.18	0.77	**0.03**	245.75	**213.28**	**925.43**	0.97	16.83	0.01
	NSGA-II	**0.10**	**0.77**	0.04	253.99	213.82	927.78	0.98	15.47	1.00
	SPEA2	0.20	0.76	0.05	257.07	213.97	928.45	0.98	15.03	1.00
Y-10	GRASP$_1$	0.16	0.69	0.07	604.76	**554.60**	**2260.93**	0.99	14.77	**0.01**
	GRASP$_2$	0.14	0.69	0.07	601.79	**554.60**	**2260.93**	0.99	14.90	0.01
	GRASP$_3$	0.17	0.69	0.07	601.85	**554.60**	**2260.93**	0.99	14.90	0.01
	GRASP$_4$	0.13	**0.69**	**0.06**	594.59	**554.60**	**2260.93**	**0.99**	15.27	0.01
	NSGA-II	**0.11**	0.69	0.07	**586.60**	**554.60**	**2260.93**	0.99	**15.70**	1.00
	SPEA2	0.26	0.68	0.08	613.30	555.19	2263.33	0.99	14.47	1.00
A-12-10	GRASP$_1$	0.68	0.35	0.39	71.79	64.53	117.66	0.94	3.30	**0.02**
	GRASP$_2$	0.70	0.32	0.45	71.21	64.61	117.82	**0.93**	3.43	0.02
	GRASP$_3$	0.66	0.36	0.38	**64.69**	**63.97**	**116.56**	0.95	**3.87**	0.02
	GRASP$_4$	0.55	**0.46**	0.33	72.33	63.99	116.59	0.93	3.20	0.02
	NSGA-II	0.53	0.39	0.35	69.71	64.48	117.49	0.95	3.53	1.00
	SPEA2	**0.52**	0.43	**0.32**	73.69	64.31	117.18	0.94	3.13	1.00
A-12-20	GRASP$_1$	0.69	0.55	0.12	42.95	35.54	121.99	0.95	10.17	0.02
	GRASP$_2$	0.67	0.55	0.12	43.32	35.51	121.94	0.94	9.93	**0.02**
	GRASP$_3$	0.67	0.55	0.13	42.61	35.58	122.17	0.95	10.30	0.02
	GRASP$_4$	0.65	0.55	0.11	42.35	**35.45**	**121.70**	**0.93**	**10.37**	0.02
	NSGA-II	**0.53**	**0.57**	**0.10**	**42.28**	35.60	122.25	0.94	**10.37**	1.00
	SPEA2	0.70	0.53	0.16	45.22	36.23	124.43	0.94	9.27	1.00
A-12-30	GRASP$_1$	0.56	0.61	0.12	56.18	44.33	151.06	**0.93**	9.13	**0.02**
	GRASP$_2$	0.58	0.60	0.13	56.47	44.43	151.42	0.94	9.30	0.02
	GRASP$_3$	0.49	0.61	0.13	57.13	44.33	151.12	0.94	8.97	0.02
	GRASP$_4$	**0.45**	**0.62**	**0.11**	**53.90**	**44.06**	**150.14**	0.94	**10.00**	0.02
	NSGA-II	0.48	0.61	0.13	56.90	44.44	151.48	0.95	8.93	1.00
	SPEA2	0.61	0.61	0.14	58.60	44.41	151.36	0.95	8.30	1.00
A-12-40	GRASP$_1$	0.64	0.67	0.10	44.21	35.58	139.79	0.93	12.90	**0.02**
	GRASP$_2$	0.60	0.67	0.09	43.47	35.51	139.51	0.93	13.50	0.02
	GRASP$_3$	0.69	0.67	0.09	42.85	35.52	139.58	0.92	13.80	0.02
	GRASP$_4$	0.59	0.68	**0.08**	**42.76**	**35.17**	**138.22**	**0.92**	**13.87**	0.02
	NSGA-II	**0.48**	**0.68**	0.09	46.09	35.59	139.87	0.93	12.03	1.00
	SPEA2	0.73	0.65	0.12	45.95	36.03	141.57	0.94	12.00	1.00
A-12-50	GRASP$_1$	0.51	0.67	0.07	33.70	28.72	126.72	0.89	18.73	**0.03**
	GRASP$_2$	0.59	0.66	0.08	33.14	28.87	127.39	**0.89**	19.23	0.03
	GRASP$_3$	0.58	0.66	0.08	33.21	28.71	126.69	0.89	18.97	0.03
	GRASP$_4$	0.49	**0.67**	**0.06**	**32.66**	**28.65**	**126.41**	0.89	**19.67**	0.03
	NSGA-II	**0.36**	0.67	0.08	34.41	28.71	126.67	0.90	17.83	1.00
	SPEA2	0.71	0.64	0.10	35.45	29.25	129.07	0.90	16.97	1.00
A-12-60	GRASP$_1$	0.44	0.55	0.12	**45.02**	37.33	130.24	**0.93**	**9.90**	0.02
	GRASP$_2$	0.37	0.56	0.13	46.20	**37.27**	**130.04**	0.93	9.40	0.02
	GRASP$_3$	0.45	0.56	0.13	46.37	37.37	130.37	0.94	9.37	0.02
	GRASP$_4$	0.33	0.57	**0.11**	45.52	37.28	130.06	0.94	9.60	**0.02**
	NSGA-II	**0.28**	**0.57**	0.13	49.75	37.32	130.19	0.94	8.43	1.00
	SPEA2	0.44	0.55	0.16	50.20	37.36	130.36	0.94	8.10	1.00
A-12-70	GRASP$_1$	0.47	0.27	0.24	53.34	44.75	137.75	0.95	8.27	0.02
	GRASP$_2$	0.55	0.26	0.28	53.19	44.89	138.20	0.95	8.17	0.02
	GRASP$_3$	0.50	0.27	0.25	52.93	44.82	137.95	0.95	8.33	0.02
	GRASP$_4$	0.41	0.28	**0.23**	54.69	**44.53**	**137.06**	0.95	7.83	**0.02**
	NSGA-II	**0.40**	**0.30**	0.26	**51.97**	45.07	138.72	0.95	**8.57**	1.00
	SPEA2	0.62	0.26	0.32	53.05	45.56	140.23	**0.95**	8.40	1.00
A-12-80	GRASP$_1$	0.61	0.65	0.08	57.80	49.72	192.18	0.92	13.33	0.02
	GRASP$_2$	0.58	0.66	0.08	56.07	49.68	191.99	0.93	13.97	**0.02**
	GRASP$_3$	**0.55**	0.66	0.07	54.62	49.69	192.04	0.92	14.77	0.02
	GRASP$_4$	0.55	0.66	0.08	**54.55**	49.63	191.82	0.93	**14.80**	0.02
	NSGA-II	0.56	**0.67**	**0.07**	56.25	**49.60**	**191.67**	**0.92**	13.83	1.00
	SPEA2	0.72	0.64	0.10	60.05	49.82	192.54	0.93	12.30	1.00
A-12-90	GRASP$_1$	0.39	0.48	0.17	77.04	69.09	199.99	0.95	7.37	**0.02**
	GRASP$_2$	0.38	0.48	**0.14**	76.43	68.98	199.67	**0.95**	7.37	0.02
	GRASP$_3$	0.60	0.44	0.20	77.21	69.12	200.08	0.95	7.43	0.02
	GRASP$_4$	0.46	0.46	0.16	**73.96**	**68.93**	**199.52**	0.95	**7.97**	0.02
	NSGA-II	0.47	0.48	0.17	78.83	69.20	200.33	0.95	7.10	1.00
	SPEA2	**0.38**	**0.49**	0.18	81.45	69.18	200.25	0.95	6.70	1.00

Table A.10: Comparison among the 4 different configurations of our GRASP proposal, an NSGA-II and a SPEA2 algorithms, for a set of instances with size [12, 15]. Best values for the metrics are highlighted with **bold**.

Instance	Algorithm	C	HV	ϵ	GD	IGD	IGD$^+$	Δ	Size	Time(s)
previous12	GRASP$_1$	0.39	0.83	0.06	390.04	308.16	1301.18	0.98	12.57	0.02
	GRASP$_2$	0.38	0.84	0.06	395.77	307.97	1300.41	0.98	12.20	**0.02**
	GRASP$_3$	0.41	0.84	0.06	393.93	308.09	1300.95	0.98	12.33	0.02
	GRASP$_4$	**0.33**	**0.84**	**0.05**	**377.91**	307.94	1300.31	**0.97**	**13.23**	0.02
	NSGA-II	0.37	0.83	0.06	396.80	**307.93**	**1300.24**	0.97	12.20	1.00
	SPEA2	0.44	0.83	0.07	427.41	307.98	1300.46	0.98	10.50	1.00
S-12	GRASP$_1$	**0.39**	0.72	0.10	709.96	598.65	2449.18	0.99	12.83	0.02
	GRASP$_2$	0.62	0.71	0.10	681.69	599.00	2450.59	0.99	14.23	0.02
	GRASP$_3$	0.41	**0.72**	0.08	692.36	**598.39**	**2448.15**	**0.99**	13.43	0.02
	GRASP$_4$	0.45	0.72	**0.08**	**667.39**	598.41	2448.23	0.99	**14.57**	**0.02**
	NSGA-II	0.40	0.72	0.10	732.27	599.41	2452.29	0.99	12.30	1.00
	SPEA2	0.68	0.69	0.13	755.74	599.90	2454.32	0.99	11.63	1.00
Y-12	GRASP$_1$	0.42	0.69	0.08	677.18	569.51	2638.10	0.99	16.13	0.02
	GRASP$_2$	0.44	0.69	0.09	681.64	569.73	2639.10	0.99	15.93	0.02
	GRASP$_3$	0.43	0.69	0.07	671.47	569.42	2637.68	0.99	16.33	0.02
	GRASP$_4$	0.43	**0.70**	**0.07**	**650.85**	**568.99**	**2635.70**	**0.99**	**17.33**	**0.02**
	NSGA-II	**0.35**	0.69	0.08	686.07	569.42	2637.74	0.99	15.63	1.00
	SPEA2	0.54	0.66	0.13	750.00	570.87	2644.42	0.99	13.53	1.00
N-15	GRASP$_1$	0.65	0.58	0.07	240.76	207.94	1222.36	0.95	**31.13**	0.07
	GRASP$_2$	0.63	0.58	0.08	242.24	208.05	1222.98	**0.95**	30.97	0.06
	GRASP$_3$	0.56	0.59	0.07	**240.66**	207.62	1220.53	0.96	31.03	0.07
	GRASP$_4$	**0.48**	**0.59**	**0.06**	240.84	**206.91**	**1216.31**	0.96	30.73	**0.06**
	NSGA-II	0.68	0.57	0.10	253.68	209.55	1231.79	0.96	28.37	1.00
	SPEA2	0.85	0.53	0.13	282.59	211.08	1240.98	0.96	23.03	1.00
O-15	GRASP$_1$	0.53	0.71	0.05	513.11	445.07	2672.15	0.98	30.27	0.06
	GRASP$_2$	0.57	0.71	0.05	508.32	444.79	2670.36	0.98	30.73	0.07
	GRASP$_3$	0.60	0.71	0.05	493.59	444.67	2669.66	0.98	32.77	0.07
	GRASP$_4$	**0.46**	**0.72**	**0.04**	**489.85**	**444.17**	**2666.67**	**0.98**	**33.03**	**0.06**
	NSGA-II	0.51	0.71	0.05	505.26	444.98	2671.51	0.98	31.20	1.00
	SPEA2	0.86	0.65	0.14	601.51	448.61	2693.25	0.98	22.50	1.00
previous15	GRASP$_1$	0.51	0.67	0.06	274.90	231.93	1490.05	**0.96**	34.37	0.07
	GRASP$_2$	0.56	0.67	0.06	275.98	231.95	1490.22	0.96	34.13	0.07
	GRASP$_3$	0.62	0.67	0.07	**271.20**	232.09	1490.97	0.96	**35.53**	0.07
	GRASP$_4$	**0.49**	**0.68**	**0.05**	272.73	**231.16**	**1485.10**	0.96	34.93	**0.07**
	NSGA-II	0.67	0.66	0.09	276.28	233.11	1497.56	0.97	34.20	1.00
	SPEA2	0.91	0.62	0.13	328.00	234.30	1505.38	0.97	25.10	1.00
S-15	GRASP$_1$	0.31	0.77	0.04	1019.93	900.73	4572.97	0.99	21.20	0.06
	GRASP$_2$	0.45	0.75	0.07	**977.26**	900.93	4574.05	0.99	**23.10**	0.06
	GRASP$_3$	0.26	0.78	0.04	1006.36	900.55	4572.10	0.99	21.73	0.06
	GRASP$_4$	**0.23**	**0.78**	**0.03**	995.02	**900.32**	**4570.88**	**0.99**	22.23	**0.06**
	NSGA-II	0.33	0.75	0.06	1039.01	901.48	4576.71	0.99	20.23	1.00
	SPEA2	0.85	0.67	0.14	1125.49	904.16	4590.29	0.99	17.93	1.00
Y-15	GRASP$_1$	0.35	0.67	0.05	649.04	603.97	3750.20	0.98	35.03	0.07
	GRASP$_2$	0.44	0.67	0.05	631.44	603.93	3749.95	0.98	37.07	0.08
	GRASP$_3$	0.39	0.67	0.05	644.27	604.03	3750.56	0.98	35.80	0.07
	GRASP$_4$	**0.35**	**0.67**	**0.04**	**623.29**	**603.44**	**3746.96**	0.98	**37.97**	**0.07**
	NSGA-II	0.41	0.67	0.06	650.19	604.13	3751.27	**0.98**	35.03	1.00
	SPEA2	0.78	0.62	0.12	729.29	606.14	3763.57	0.99	28.37	1.00

Table A.11: Comparison among the 4 different configurations of our GRASP proposal, an NSGA-II and a SPEA2 algorithms, for a set of instances with size 20. Best values for the metrics are highlighted with **bold**.

Instance	Algorithm	C	HV	ϵ	GD	IGD	IGD$^+$	Δ	Size	Time(s)
A-20-10	GRASP$_1$	0.70	0.69	0.07	**125.76**	110.98	613.25	0.98	**28.40**	0.25
	GRASP$_2$	0.73	0.68	0.09	129.65	111.30	615.05	**0.97**	26.90	0.23
	GRASP$_3$	0.70	0.68	0.08	129.21	111.31	615.05	0.97	26.97	0.23
	GRASP$_4$	**0.66**	**0.69**	**0.07**	126.09	**110.77**	**612.04**	0.97	28.20	**0.22**
	NSGA-II	0.79	0.66	0.12	141.41	113.36	626.44	0.97	22.87	1.00
	SPEA2	0.97	0.57	0.22	175.22	116.71	644.92	0.98	16.07	1.00
A-20-20	GRASP$_1$	0.74	0.66	0.07	96.07	79.87	479.45	0.93	32.67	0.25
	GRASP$_2$	0.74	0.66	0.07	95.70	79.73	478.62	0.94	32.90	0.25
	GRASP$_3$	0.73	0.66	0.07	95.34	79.54	477.42	0.93	33.07	**0.24**
	GRASP$_4$	**0.70**	**0.67**	**0.06**	**94.78**	**79.37**	**476.43**	**0.93**	**33.53**	0.24
	NSGA-II	0.79	0.64	0.11	103.07	81.37	488.36	0.94	28.37	1.00
	SPEA2	0.99	0.55	0.20	126.75	84.22	505.61	0.96	19.53	1.00
A-20-30	GRASP$_1$	0.68	0.72	0.06	**92.23**	79.05	505.87	0.94	**38.40**	0.27
	GRASP$_2$	0.73	0.72	0.06	93.66	79.19	506.78	0.94	37.47	0.26
	GRASP$_3$	0.76	0.72	0.06	93.77	79.22	506.91	0.94	37.50	0.26
	GRASP$_4$	**0.66**	**0.73**	**0.05**	93.48	**79.01**	**505.55**	**0.94**	37.53	**0.25**
	NSGA-II	0.85	0.69	0.10	102.15	80.36	514.26	0.95	31.60	1.00
	SPEA2	0.98	0.58	0.21	127.14	83.59	534.90	0.97	21.13	1.00
A-20-40	GRASP$_1$	0.74	0.66	0.07	82.30	70.18	426.71	**0.93**	33.93	0.24
	GRASP$_2$	0.69	0.67	0.07	**81.42**	70.41	428.19	0.94	**34.03**	0.23
	GRASP$_3$	0.71	0.66	0.08	83.98	70.29	427.46	0.93	32.40	0.23
	GRASP$_4$	**0.63**	**0.68**	**0.07**	81.70	**70.16**	**426.61**	0.94	33.83	**0.22**
	NSGA-II	0.82	0.64	0.11	89.48	70.89	431.08	0.94	28.63	1.00
	SPEA2	0.99	0.53	0.22	112.54	73.46	446.74	0.95	19.30	1.00
A-20-50	GRASP$_1$	0.72	0.65	0.07	106.23	93.28	553.03	0.93	33.30	0.23
	GRASP$_2$	**0.66**	**0.66**	0.06	105.53	93.31	553.23	0.93	33.57	**0.22**
	GRASP$_3$	0.70	0.65	0.07	107.37	**93.01**	**551.44**	0.93	32.67	0.23
	GRASP$_4$	0.73	0.66	**0.06**	**104.21**	93.43	553.96	**0.93**	**34.63**	0.23
	NSGA-II	0.79	0.63	0.13	115.03	93.99	557.23	0.94	28.43	1.00
	SPEA2	0.99	0.52	0.23	136.96	96.49	572.05	0.96	20.73	1.00
A-20-60	GRASP$_1$	0.67	0.69	0.06	105.29	94.26	617.86	0.96	38.90	0.26
	GRASP$_2$	0.65	0.68	0.07	105.30	94.14	617.06	**0.95**	39.07	0.26
	GRASP$_3$	0.66	0.69	0.06	106.42	94.34	618.40	0.95	38.20	0.25
	GRASP$_4$	**0.62**	**0.69**	**0.06**	**104.45**	**94.11**	**616.89**	0.95	**39.57**	**0.25**
	NSGA-II	0.90	0.65	0.11	112.75	94.93	622.28	0.96	34.10	1.00
	SPEA2	0.97	0.57	0.20	144.06	96.41	631.94	0.96	21.50	1.00
A-20-70	GRASP$_1$	0.72	0.60	0.13	168.36	159.41	749.36	0.97	22.37	0.20
	GRASP$_2$	0.67	0.61	0.12	169.24	159.03	747.42	0.97	21.90	0.20
	GRASP$_3$	0.77	0.59	0.14	170.95	159.66	750.44	0.97	21.63	0.20
	GRASP$_4$	**0.66**	**0.62**	**0.12**	**166.69**	**158.84**	**746.58**	**0.97**	**22.77**	**0.19**
	NSGA-II	0.82	0.57	0.18	177.83	160.50	754.60	0.97	19.90	1.00
	SPEA2	0.99	0.44	0.29	216.91	163.47	768.32	0.98	13.80	1.00
A-20-80	GRASP$_1$	0.82	0.68	0.07	125.02	115.58	655.81	0.96	30.73	0.22
	GRASP$_2$	0.70	0.69	0.06	125.54	115.26	654.00	0.96	30.33	0.21
	GRASP$_3$	**0.66**	0.69	0.07	129.32	115.42	654.92	0.96	29.07	0.21
	GRASP$_4$	0.67	**0.69**	**0.06**	**122.36**	**115.18**	**653.57**	**0.96**	**32.03**	**0.21**
	NSGA-II	0.77	0.66	0.11	134.45	115.88	657.53	0.96	26.70	1.00
	SPEA2	0.95	0.55	0.21	164.10	118.20	670.62	0.97	18.77	1.00
A-20-90	GRASP$_1$	0.73	0.68	0.06	141.39	133.09	796.53	**0.96**	34.77	0.24
	GRASP$_2$	0.71	0.69	0.06	144.75	133.00	796.02	0.96	33.27	**0.23**
	GRASP$_3$	0.72	0.69	0.06	141.22	132.94	795.62	0.96	34.87	0.25
	GRASP$_4$	**0.64**	**0.70**	**0.05**	**140.15**	**132.66**	**793.91**	0.96	**35.30**	0.24
	NSGA-II	0.82	0.67	0.09	151.97	133.80	800.79	0.96	30.60	1.00
	SPEA2	0.99	0.58	0.20	187.75	135.80	812.72	0.97	20.67	1.00
N-20	GRASP$_1$	**0.63**	**0.72**	0.07	447.10	385.19	2690.63	0.97	42.10	0.27
	GRASP$_2$	0.79	0.70	0.08	443.81	386.75	2702.00	0.97	43.03	0.25
	GRASP$_3$	0.67	0.71	0.07	447.12	386.73	2701.45	0.97	42.00	0.25
	GRASP$_4$	0.67	0.72	**0.07**	**438.88**	**384.80**	**2687.85**	**0.97**	**43.87**	**0.23**
	NSGA-II	0.82	0.67	0.12	491.56	390.14	2725.52	0.97	35.80	1.00
	SPEA2	0.99	0.57	0.21	623.48	396.82	2772.27	0.98	22.67	1.00
O-20	GRASP$_1$	0.56	0.71	0.05	**742.02**	689.52	5734.00	0.98	**66.50**	0.34
	GRASP$_2$	0.54	0.71	0.05	754.72	690.10	5738.77	0.98	64.37	0.32
	GRASP$_3$	0.47	0.72	0.05	758.43	690.11	5738.82	**0.98**	63.60	0.32
	GRASP$_4$	**0.41**	**0.72**	**0.04**	760.93	**689.13**	**5730.77**	0.98	62.90	**0.29**
	NSGA-II	0.76	0.66	0.12	841.43	694.25	5773.35	0.98	51.63	1.00
	SPEA2	1.00	0.57	0.20	1107.13	703.89	5853.46	0.99	31.10	1.00
S-20	GRASP$_1$	0.67	0.75	0.06	**1353.56**	1283.60	9491.77	0.99	**52.13**	0.28
	GRASP$_2$	0.61	0.75	**0.06**	1372.09	1283.56	9491.56	**0.99**	50.27	0.27
	GRASP$_3$	0.61	0.75	0.06	1376.71	1283.84	9493.58	0.99	50.27	0.27
	GRASP$_4$	**0.57**	**0.76**	0.06	1362.70	**1283.44**	**9490.55**	0.99	50.90	**0.25**
	NSGA-II	0.77	0.72	0.10	1451.72	1285.81	9508.07	0.99	44.90	1.00
	SPEA2	0.98	0.59	0.20	1991.74	1293.47	9564.54	0.99	25.30	1.00
Y-20	GRASP$_1$	0.70	0.71	0.06	762.53	740.87	5414.69	0.98	54.13	0.30
	GRASP$_2$	0.71	0.71	0.06	**756.83**	740.96	5415.54	0.98	**55.00**	0.30
	GRASP$_3$	0.69	0.72	**0.06**	760.97	740.82	5414.49	0.98	54.43	0.29
	GRASP$_4$	**0.66**	**0.72**	0.06	763.29	**740.57**	**5412.63**	**0.98**	54.03	**0.26**
	NSGA-II	0.83	0.68	0.11	815.10	743.23	5432.09	0.98	47.43	1.00
	SPEA2	1.00	0.57	0.21	1127.08	749.21	5475.95	0.99	25.73	1.00

Table A.12: Comparison among the 4 different configurations of our GRASP proposal, an NSGA-II and SPEA2 algorithms, for a set of instances with size 25. Best values for the metrics are highlighted with **bold**.

Instance	Algorithm	C	HV	ϵ	GD	IGD	IGD$^+$	Δ	Size	Time(s)
A-25-10	$GRASP_1$	0.70	0.71	0.06	154.77	132.50	953.05	0.96	46.27	**0.64**
	$GRASP_2$	0.73	0.71	0.06	153.36	132.67	954.33	0.96	47.00	0.67
	$GRASP_3$	0.73	0.71	0.06	153.72	132.36	952.15	**0.96**	**47.10**	0.73
	$GRASP_4$	**0.66**	**0.72**	**0.05**	**152.99**	**131.77**	**947.81**	0.96	**47.10**	0.69
	NSGA-II	0.84	0.68	0.12	177.63	135.83	977.03	0.97	35.73	2.00
	SPEA2	0.99	0.59	0.22	216.88	139.53	1003.83	0.97	25.10	2.00
A-25-20	$GRASP_1$	**0.68**	0.72	0.05	**140.05**	119.58	864.69	**0.95**	**48.87**	0.68
	$GRASP_2$	0.72	0.71	0.05	141.85	120.03	867.97	0.96	47.50	**0.64**
	$GRASP_3$	0.73	0.71	0.06	141.69	119.97	867.48	0.95	47.77	0.66
	$GRASP_4$	0.69	**0.72**	**0.04**	140.46	**119.32**	**862.75**	0.95	47.93	0.67
	NSGA-II	0.87	0.68	0.10	151.31	121.92	881.62	0.96	41.93	2.00
	SPEA2	0.99	0.58	0.19	190.72	126.54	914.93	0.97	28.03	2.00
A-25-30	$GRASP_1$	0.72	0.72	0.05	136.69	122.55	1011.62	0.94	65.13	0.82
	$GRASP_2$	0.79	0.71	0.05	136.06	122.73	1013.17	0.95	66.07	0.77
	$GRASP_3$	**0.69**	0.72	0.05	**135.65**	122.39	1010.35	0.95	**66.13**	0.79
	$GRASP_4$	0.70	**0.72**	**0.04**	138.09	**122.36**	**1010.11**	**0.94**	63.73	**0.76**
	NSGA-II	0.90	0.67	0.13	157.99	125.03	1032.06	0.95	50.30	2.00
	SPEA2	0.99	0.59	0.21	189.70	127.59	1053.33	0.97	35.77	2.00
A-25-40	$GRASP_1$	0.77	0.71	0.05	130.80	115.18	921.01	0.94	61.50	0.78
	$GRASP_2$	0.76	0.71	0.05	130.23	115.14	920.60	**0.94**	61.77	0.78
	$GRASP_3$	**0.73**	0.71	0.05	127.34	115.14	920.64	0.94	64.70	0.80
	$GRASP_4$	0.73	**0.71**	**0.04**	**126.54**	**114.47**	**915.32**	0.94	**64.93**	**0.74**
	NSGA-II	0.86	0.67	0.11	145.01	116.74	933.29	0.95	49.73	2.00
	SPEA2	0.99	0.58	0.21	182.55	120.16	960.85	0.96	32.50	2.00
A-25-50	$GRASP_1$	0.70	0.74	0.04	144.06	129.97	1107.11	**0.95**	68.30	0.81
	$GRASP_2$	**0.70**	0.74	0.04	**141.26**	130.00	1107.38	0.96	**70.67**	0.80
	$GRASP_3$	0.73	0.74	0.05	143.88	130.15	1108.68	0.95	68.23	0.79
	$GRASP_4$	0.73	**0.74**	**0.04**	142.97	**129.88**	**1106.34**	0.95	69.23	**0.77**
	NSGA-II	0.81	0.71	0.10	159.35	131.59	1120.86	0.96	55.60	2.00
	SPEA2	1.00	0.62	0.18	202.15	133.95	1140.99	0.97	36.07	2.00
A-25-60	$GRASP_1$	0.72	0.64	**0.06**	140.69	**129.09**	**1001.48**	0.95	57.90	0.76
	$GRASP_2$	0.76	0.64	0.06	139.92	129.42	1003.99	0.95	58.50	0.73
	$GRASP_3$	0.73	0.64	0.06	138.32	129.40	1003.89	0.95	59.83	0.77
	$GRASP_4$	**0.70**	**0.65**	0.06	**138.03**	129.15	1001.94	**0.95**	**60.13**	**0.70**
	NSGA-II	0.84	0.60	0.14	156.07	131.49	1020.11	0.96	47.77	2.00
	SPEA2	0.99	0.50	0.24	190.88	134.23	1041.42	0.97	33.50	2.00
A-25-70	$GRASP_1$	0.72	0.70	0.05	**162.97**	152.12	1229.14	0.96	**64.33**	0.77
	$GRASP_2$	0.73	0.70	0.05	163.38	152.19	1229.70	**0.96**	63.87	0.73
	$GRASP_3$	0.78	0.70	0.05	163.91	152.08	1228.78	0.96	63.97	0.75
	$GRASP_4$	**0.69**	**0.71**	**0.04**	163.03	**151.73**	**1226.00**	0.96	64.30	**0.72**
	NSGA-II	0.80	0.68	0.10	178.69	153.06	1236.82	0.96	53.43	2.00
	SPEA2	0.97	0.57	0.20	223.54	156.34	1263.31	0.97	35.90	2.00
A-25-80	$GRASP_1$	0.67	0.65	0.04	**161.61**	149.04	1101.55	0.94	**54.20**	0.70
	$GRASP_2$	0.63	0.65	0.05	163.45	149.13	1102.21	**0.94**	53.13	**0.69**
	$GRASP_3$	0.58	0.66	0.04	162.57	149.14	1102.29	0.94	53.63	0.72
	$GRASP_4$	**0.55**	**0.66**	**0.03**	162.22	**148.88**	**1100.40**	0.94	54.03	0.70
	NSGA-II	0.87	0.61	0.12	169.26	150.59	1112.97	0.95	48.93	2.00
	SPEA2	1.00	0.52	0.22	203.94	153.36	1133.50	0.97	34.50	2.00
A-25-90	$GRASP_1$	0.72	0.71	0.06	185.30	174.36	1172.89	0.97	43.07	0.66
	$GRASP_2$	0.79	0.71	0.07	187.95	174.45	1173.41	0.97	42.07	**0.61**
	$GRASP_3$	0.75	0.71	0.06	188.59	174.22	1171.87	**0.96**	41.70	0.63
	$GRASP_4$	**0.67**	**0.72**	**0.05**	**182.52**	**173.99**	**1170.31**	0.97	**44.33**	0.63
	NSGA-II	0.82	0.67	0.13	219.79	175.77	1182.29	0.97	31.40	2.00
	SPEA2	0.98	0.58	0.22	267.68	177.81	1195.97	0.98	21.83	2.00
S-25	$GRASP_1$	0.65	0.73	0.05	1863.24	1761.50	16535.96	0.99	82.07	0.88
	$GRASP_2$	0.71	0.73	0.05	1846.18	1761.08	16531.86	0.99	83.37	0.87
	$GRASP_3$	0.66	0.73	0.05	**1818.57**	1761.37	16534.47	0.99	**85.77**	0.90
	$GRASP_4$	**0.58**	**0.74**	**0.04**	1844.97	**1760.58**	**16527.13**	0.99	83.30	**0.83**
	NSGA-II	0.88	0.68	0.09	2105.42	1764.65	16565.47	**0.99**	64.10	2.00
	SPEA2	0.99	0.59	0.21	2597.16	1772.78	16642.44	0.99	44.47	2.00
Y-25	$GRASP_1$	0.66	0.70	0.04	742.59	710.95	8044.04	0.98	125.20	1.21
	$GRASP_2$	0.64	0.70	0.05	758.68	711.15	8046.28	0.98	120.27	1.11
	$GRASP_3$	0.73	0.70	0.05	750.76	711.28	8047.75	0.98	122.53	1.17
	$GRASP_4$	**0.63**	**0.70**	**0.03**	**739.54**	**710.73**	**8041.50**	**0.98**	**126.57**	**1.07**
	NSGA-II	0.92	0.65	0.14	997.73	715.19	8091.98	0.98	68.80	2.00
	SPEA2	0.99	0.57	0.21	1188.99	719.90	8145.13	0.99	51.67	2.00

Table A.13: Comparison among the 4 different configurations of our GRASP proposal, an NSGA-II and SPEA2 algoritms, for a set of instances with size [30, 60]. Best values for the metrics are highlighted with **bold**.

Instance	Algorithm	C	HV	ϵ	GD	IGD	IGD$^+$	Δ	Size	Time(s)
Y-30	GRASP$_1$	0.78	0.70	0.05	**898.88**	867.81	10121.44	0.98	**133.97**	2.44
	GRASP$_2$	**0.69**	0.71	0.04	909.34	867.75	10120.64	0.98	131.00	2.51
	GRASP$_3$	0.70	**0.71**	0.05	908.59	867.40	10116.37	0.98	131.53	2.49
	GRASP$_4$	0.75	0.71	**0.04**	900.54	**867.27**	**10115.02**	0.98	133.73	**2.31**
	NSGA-II	0.93	0.66	0.12	1211.48	870.53	10153.28	**0.98**	73.23	6.00
	SPEA2	1.00	0.57	0.21	1317.43	877.22	10230.61	0.99	63.87	6.00
Y-35	GRASP$_1$	0.74	0.71	0.04	940.69	942.52	13375.02	0.98	212.37	6.50
	GRASP$_2$	0.81	0.71	0.04	948.47	942.41	13373.59	0.98	209.20	6.35
	GRASP$_3$	0.78	0.71	0.04	932.60	942.23	13370.79	0.98	216.70	6.57
	GRASP$_4$	**0.71**	**0.72**	**0.03**	**930.01**	**942.13**	**13369.32**	0.98	**217.57**	**5.68**
	NSGA-II	0.91	0.67	0.10	1560.43	946.16	13426.59	**0.97**	77.20	14.00
	SPEA2	1.00	0.58	0.21	1398.07	952.80	13520.23	0.99	96.57	14.00
Y-40	GRASP$_1$	0.76	0.73	0.05	1030.63	1015.21	15805.11	0.98	245.97	11.89
	GRASP$_2$	0.75	0.73	0.05	1024.53	1015.35	15806.96	0.98	248.60	11.11
	GRASP$_3$	**0.72**	0.74	0.05	1022.23	1015.25	15805.84	0.98	249.67	11.75
	GRASP$_4$	0.78	**0.74**	**0.04**	**1008.37**	**1014.60**	**15796.04**	0.98	**255.60**	**10.26**
	NSGA-II	0.92	0.68	0.10	1847.17	1018.27	15852.70	**0.98**	76.30	30.00
	SPEA2	0.99	0.61	0.19	1615.54	1022.17	15913.41	0.99	99.37	30.00
Y-45	GRASP$_1$	0.79	0.75	0.03	1046.40	1022.35	20295.11	0.98	396.43	27.65
	GRASP$_2$	0.79	0.75	0.03	1052.36	1021.84	20285.54	0.98	391.07	26.40
	GRASP$_3$	0.71	0.75	0.03	1054.00	1021.82	20285.05	0.98	390.63	26.36
	GRASP$_4$	**0.70**	**0.75**	**0.03**	**1031.65**	**1021.80**	**20284.63**	0.98	**406.80**	**23.68**
	NSGA-II	0.96	0.69	0.11	2374.61	1025.56	20359.99	**0.97**	76.47	60.00
	SPEA2	0.97	0.64	0.19	2071.06	1028.97	20427.33	0.98	100.00	60.00
Y-50	GRASP$_1$	0.78	0.73	0.04	1249.93	1230.99	25869.12	0.98	445.40	44.21
	GRASP$_2$	0.78	0.73	0.04	1250.06	1231.49	25880.32	0.98	445.37	43.43
	GRASP$_3$	0.75	0.73	0.04	**1236.48**	1230.57	25860.75	0.98	**455.43**	45.31
	GRASP$_4$	**0.70**	**0.74**	**0.03**	1248.77	**1229.63**	**25840.84**	0.98	446.87	**38.93**
	NSGA-II	0.93	0.67	0.10	2988.72	1234.06	25934.83	**0.98**	77.87	112.00
	SPEA2	1.00	0.61	0.17	2634.89	1238.40	26026.08	0.98	100.00	112.00
Y-60	GRASP$_1$	0.80	0.73	0.03	1264.65	1236.28	31890.60	0.98	662.90	130.33
	GRASP$_2$	0.80	0.73	0.03	1242.15	1236.11	31886.21	0.98	**689.57**	131.71
	GRASP$_3$	0.72	0.73	0.03	1257.66	1235.67	31875.45	0.98	670.80	129.41
	GRASP$_4$	**0.69**	**0.74**	**0.03**	**1240.99**	**1235.27**	**31864.93**	0.98	687.97	**110.87**
	NSGA-II	0.96	0.67	0.11	3710.10	1239.74	31980.61	**0.97**	76.93	300.00
	SPEA2	0.99	0.62	0.18	3251.85	1244.69	32107.54	0.97	100.00	300.00

A GRASP method for the Bi-Objective Multiple Row Equal Facility Layout Problem

Highlights

- Ensemble of constructive methods to provide diversity of initial solutions.
- Combination of local search procedures to improve the non-dominated front.
- Improvement of the state of the art with all GRASP configurations.
- Study a of new set of larger instances created from the single-objective problem.

Nicolás R. Uribe: Conceptualization, Software, Writing- Original draft preparation, Writing- Reviewing and Editing; **Alberto Herrán:** Conceptualization, Methodology, Software, Writing- Original draft preparation, Writing- Reviewing and Editing; **J. Manuel Colmenar:** Conceptualization, Methodology, Writing- Original draft preparation, Writing- Reviewing and Editing.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: