

## Research Article

# Analyzing the Performance of a Hybrid Heuristic for Solving a Bilevel Location Problem under Different Approaches to Tackle the Lower Level

**Sayuri Maldonado-Pinto, Martha-Selene Casas-Ramírez,  
and José-Fernando Camacho-Vallejo**

*Facultad de Ciencias Físico-Matemáticas, Universidad Autónoma de Nuevo León, Avenida Universidad s/n,  
66450 San Nicolás de los Garza, NL, Mexico*

Correspondence should be addressed to Martha-Selene Casas-Ramírez; [martha.casasrm@uanl.edu.mx](mailto:martha.casasrm@uanl.edu.mx)

Received 19 March 2016; Accepted 18 May 2016

Academic Editor: Mónica A. López-Campos

Copyright © 2016 Sayuri Maldonado-Pinto et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The problem addressed here is a combinatorial bilevel programming problem called the uncapacitated facility location problem with customer's preferences. A hybrid algorithm is developed for solving a battery of benchmark instances. The algorithm hybridizes an evolutionary algorithm with path relinking; the latter procedure is added into the crossover phase for exploring the trajectory between both parents. The proposed algorithm outperforms the evolutionary algorithm already existing in the literature. Results show that including a more sophisticated procedure for improving the population through the generations accelerates the convergence of the algorithm. In order to support the latter statement, a reduction of around the half of the computational time is obtained by using the hybrid algorithm. Moreover, due to the nature of bilevel problems, if feasible solutions are desired, then the lower level must be solved for each change in the upper level's current solution. A study for illustrating the impact in the algorithm's performance when solving the lower level through three different exact or heuristic approaches is made.

## 1. Introduction

Location theory is an area of operational research which has attracted many researchers due to the existing necessity to abstract real day-to-day situations into mathematical models. One of the main problems is the facility location problem, which stems from Weber's problems, which consists in determining the point that minimizes the sum of the Euclidean distances from that point to all other given points. Further explanation is detailed in [1]. In the facility location problem, there is a set of customers that are distributed in a predefined space. They desire that their demands of a particular service or product are met by one or more facilities. The problem is to determine where to locate facilities and how customers should be allocated in order to minimize location and distribution costs associated with that particular decision. This situation is known as the classic facility location problem (FLP).

The FLP has led to several extensions. For example, the simple plant location problem (SPLP) arises when the facility

has an infinite capacity and can meet all the demands of any customer; the model for this problem was proposed in [2]. A taxonomy of location models including relevant issues and a classification is provided in [3]. Another variant appears when a new objective function is considered as in [4]. It includes the most popular objectives functions of location models and penalizes the distance between the customer and the facility according to the position occupied by the customer. In other words, a customer will have different penalties depending on how near or far it is regarding the facility.

Having an ordering problem within a location one increases its complexity in both the formulation and the methodology of solution. The Discrete Ordered Median Problem (DOMP) was introduced in [5], in which two formulations are proposed: as an integer linear program and as an integer nonlinear program. Then, in [6], an alternative integer linear programming formulation for the DOMP is proposed. A comparison with the existing ones is made showing that the proposed formulation is strengthened.

Moreover, some properties regarding optimal solutions that allow the elimination of a subset of variables are found. Taking advantage of the properties, a branch and bound algorithm was developed and it was used for solving a set of benchmark instances.

In [7], an extension of the DOMP called the ordered capacitated facility location (OCFL) problem is proposed and it was modeled from three different points of view. In the first model, the customer's demand can be divided; in the second model, fixed costs for locating facilities are considered; and in the third model, the shipping and locating costs are taken into account in the objective function. Also, in the latter model, they examine three approaches for incorporating shipping costs: the costs are paid (i) by the customers, (ii) by the distribution centers, or (iii) by the logistics provider. To consider the abovementioned approaches, they used the objective function proposed in [8]. The locating and shipping costs are ordered before the evaluation of the objective function is made. Two formulations of the model are proposed and some improvements for particular cases are introduced.

A different perspective regarding an ordering within the location problem can be achieved by considering the customer's preferences towards the facilities. The first paper which considered the customer's preferences in the simple plant location problem (SPLP) is [9], in which they assumed the location of a single facility and added restrictions to ensure that the preferences were considered. Furthermore, they presented different ways to include this set of constraints and proposed a greedy heuristic based on branch and bound for solving the problem. The problem is known as the simple plant location problem with order (SPLPO). It is shown in [10] that this problem and all its generalizations are classified as NP-Hard.

The consideration of customer's preferences may be seen as an optimization problem within the constraints of the location problem. These kind of problems can be modeled with bilevel programming. The first bilevel model for the SPLPO was proposed in [11]. The bilevel model is reformulated as a single-level problem by introducing pseudo-Boolean functions. Then, valid bounds are obtained through a relaxation of the lower level problem. Additional assumptions were made during that research: the optimal solution of follower is unique for any arbitrary solution of leader and all values of the customer's preferences are different.

Most articles that addressed the bilevel model of the SPLPO reformulate the bilevel model into a single-level one and solve the reformulation. For example, for the aforementioned purposes in [12], valid inequalities are used and in [13] a combinatorial formulation is considered. The main motivation for avoiding solving the bilevel version of the problem relies in the difficulty of solving an optimization problem for each configuration of facilities located.

Therefore, it is convenient to emphasize that it is extremely important to obtain the optimal solution of the lower level problem in order to obtain a bilevel feasible solution. For the uncapacitated facility location bilevel problem (UFLBP), the lower level consists of an allocation problem, which can be solved in an efficient manner by an optimizer. Moreover, due to the nature of the allocation problem,

alternative exact methods can be easily adapted for solving it. For instance, in [14, 15], an exact method based on the ordered matrix of preferences is considered for solving the bilevel version of UFLBP. However, the construction of efficient exact methods cannot be always obtained for solving the lower level problem; it clearly depends on its structure.

For clarifying the latter idea, some papers in which the lower level is solved in a heuristic manner are described. For example, in [16], a new formulation for the ring star problem as a bilevel model is proposed, in which they considered the existence of a leader and two independent followers. One of the followers must solve a travelling salesman problem and the use of a greedy algorithm with 2-opt and 3-opt local searches is applied. Another example is found in [17] where a bilevel urban transportation network design model is studied. In order to avoid finding the optimal solution for the follower, a local approach was designed for getting the follower's response. Then, in [18], a topological design of Local Area Networks bilevel problem is considered. The lower level must construct a capacitated spanning tree and it is solved by a greedy constructive algorithm similar to Kruskal's algorithm. Also, in [19], a competitive facility location problem is considered; a branch and bound method is applied for solving a nonlinear programming relaxation of the lower level problem. Finally, in [20], an ant colony optimization algorithm for solving a bilevel production-distribution problem was implemented. They illustrated the behavior of the algorithm when the lower level is solved in an exact manner or by a heuristic procedure. In the case when the heuristic procedure was applied to the lower level, it corresponds to a differential evolution algorithm.

As it is mentioned above, in order to obtain bilevel feasible solutions, the lower level problem must be optimally solved by an optimizer or by an exact method. But this is not always possible and the problem needs to be solved somehow. Hence, a heuristic procedure that balances efficiency and computational effort would be desired. Commonly, the utilization of heuristic procedures for solving the bilevel problem results very costly in terms of computational time due to the number of times that the lower level problem is solved. Then, the exploration of methodologies in which the search is guided without the resolution of the lower level turns out to be a matter of interest.

In this paper, we proposed a hybrid algorithm for solving the bilevel version of the uncapacitated facility location problem with preferences of the customers. Also, a discussion is presented about the effects that result from solving the allocation problem with an optimizer, by an alternative exact procedure or by a methodology that avoids solving it at each step. The paper is organized as follows: in Section 3, we present the classical mathematical bilevel formulation of the problem. Section 3 describes the proposed algorithm that hybridizes an evolutionary algorithm with path relinking. The computational experimentation and its corresponding interpretation of the results are shown in Section 4. Finally, Section 5 states the final remarks that arose from the findings derived from the previous section and some ideas for further research.

## 2. Description of the Problem

In this section, the problem and its mathematical formulation are described. The bilevel model considered in this paper is proposed in [12], where the leader aims to minimize the total cost, that is, the locating and distributing costs. On the other hand, the follower aims to minimize the customer's preferences. The sets, parameters, decision variables involved in the mathematical formulation, and the assumptions considered during the research are presented next.

Let  $i \in I$  and  $j \in J$  be the indexes of the facilities and customers, respectively. Let  $c_{ij}$  represent the costs of supplying all the customer's  $j$  demand for the facility  $i$ . Let  $f_i$  denote the cost of locating facility  $i$ . Finally,  $p_{ij}$  represents the preference that customer  $j$  places for being served by facility  $i$ .

The decision variables of the problem are two and are of binary nature, where

$$y_i = \begin{cases} 1 & \text{if the facility } i \text{ is located} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if the facility } i \text{ satisfies the demand of the customer } j \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The following two assumptions are considered in the problem:

- (1) Customers delivered a list of ordered preferences which reflected their desire of being served by each facility; the first position in the list, that is, the 1st, indicates the most preferred facility and therefore  $|I|$ th represents the least preferred facility.
- (2) The facilities have no capacity; therefore, a facility can supply multiple customers but a customer must be served by a single facility.

The mathematical model for the UFLBP is as follows:

$$\min_{y,x} \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i y_i \quad (2)$$

$$\text{subject to: } y_i \in \{0, 1\} \quad \forall i \in I \quad (3)$$

$$x \in \arg \min \sum_{i \in I} \sum_{j \in J} g_{ij} x_{ij} \quad (4)$$

$$\text{subject to: } \sum_{i \in I} x_{ij} = 1 \quad \forall j \in J \quad (5)$$

$$x_{ij} \leq y_i \quad \forall i \in I, j \in J \quad (6)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J. \quad (7)$$

The problem in the upper level is defined by (2)–(4), where (2) represents the leader's objective function who seeks to minimize both location and distribution costs, (3) establish the binary constraints for each variable  $y_i$ , and finally (4) is the constraint that indicates that the variables  $x_{ij}$  are controlled by the follower; these variables are implicitly determined

by the optimal solution of the lower level. This problem is defined by (4)–(7); the follower's objective function is defined in (4) where she/he tries to minimize the ordered preferences of the customers, (5) ensures that each customer is supplied by a single facility, (6) indicates that the customer's allocation can be made only to the located facilities, and, finally, (7) indicates the binary constraints for the decision variables  $x_{ij}$ .

The existence and uniqueness of the lower level's solution are guaranteed due to the way the preferences are given; that is, they are ordered and are different from each other. In other words, it is not allowed to assign the same preference value to more than one facility. The proof which validates the latter is shown in [12]. By assuming this, the bilevel problem is well defined.

## 3. A Hybrid Evolutionary Algorithm with Path Relinking

In this section, the hybrid algorithm and its components are described. Hybrid algorithms combine advantages of two or more heuristics to solve a problem, where the best part of each heuristic is taken in order to obtain either better solutions or a quick convergence. Some applications of hybrid algorithms are shown in [21]. They identify connections and contrasts between heuristics (genetic algorithms and Tabu search) that offer an almost untapped area for empirical research. In [22], a genetic algorithm and particle swarm optimization algorithms are hybridized showing good performance in some applications.

Considering the discussed improvements for hybrid algorithms, in this paper, hybridization between an evolutionary algorithm and path relinking is proposed. The main idea for including the path relinking scheme is basically in order to substitute the common random crossover. On the other hand, the motivations for developing an evolutionary algorithm (EA) are as follows: first, a population based algorithm gives more information about the interaction between leader and follower due to the large number of solutions being explored; second, it is well known that EAs can handle *not-easy-to-solve* problems in an efficient manner; and, finally, the EAs perform random movements allowing a wider exploration of the solution space while the quality of solutions is intended to be improved.

The EA consists in three phases: in the first one, the construction of the initial population containing feasible solutions is made; the second phase concerns the genetics operators, crossover and mutation; and the last one is the selection mechanism of survival solutions, which depends on two features, quality and diversity. Commonly, the criteria selected in the genetic operators are chosen in a random fashion. The latter might affect the algorithm's performance due to the lack of exploration within the current neighborhood between both solutions. Hence, we decided to implement a combinatorial method called path relinking (PR) for reaching the local optimum in the current neighborhood. Path relinking begins with a pair of good quality solutions, parts from the first one, and changes its components once a time to convert this initial solution into the second one. Furthermore,

```

Procedure Evolutionary algorithm
 $P_t \leftarrow \emptyset$ 
 $P_t \leftarrow \text{Generate Initial Population}(P_t)$ 
 $f \leftarrow \text{Fitness}(P_t)$ 
while  $t < \text{Generations}$ 
     $P_{t+1} \leftarrow \text{Selection}(P_t, P_{t+1}, f)$ 
     $P_{t+1} \leftarrow \text{Crossover}(P_t)$ 
     $P_{t+1} \leftarrow \text{Mutation}(P_t)$ 
     $P_t \leftarrow P_{t+1}$ 
     $t \leftarrow t + 1$ 
end while
return  $P_t$ 
end Procedure

```

PSEUDOCODE 1: Pseudocode of evolutionary algorithm.

```

Procedure Path relinking ( $y_1, y_2$ )
 $y_{\text{new}} \leftarrow \emptyset$ 
 $D \leftarrow d(y_1, y_2)$ 
 $\min \leftarrow \infty$ 
for  $i = 1: \text{number\_difference}$ 
     $y' \leftarrow \text{change}(y_1, D, i)$ 
     $x \leftarrow \text{Solve\_lower\_level}(y')$ 
     $c(y') \leftarrow \text{fitness}(y', x)$ 
    if  $c(y') < \min$ 
         $\min \leftarrow c(y')$ 
         $y_{\text{new}} \leftarrow y'$ 
    end if
end for
return  $y_{\text{new}}$ 
end Procedure

```

PSEUDOCODE 2: Pseudocode of path relinking.

PR explores all the trajectory between leader's solutions. If the second solution is identical to the first one, then the method stops and returns the best solution found in the trajectory. It is important to highlight the notion that since the aim of the proposed algorithm is to solve a bilevel problem, when computing the corresponding objective function value, the follower's problem is solved for each movement. This issue clearly affects the algorithm's efficiency and it is discussed in the next section.

The addition of PR in other algorithm's frameworks has shown a good performance and has attracted the attention of researchers. For example, in [23], there is a description of the elements and methods that conform Scatter Search (SS), including the most recent elements incorporated in successful applications in both global and combinatorial optimization. They also described the hybridization of PR and genetic algorithms (GA) and displayed the notion that the use of PR almost always improves the performance of the heuristic. Also, in [24], a project scheduling problem is considered, where hybridization of PR and a GA is developed to solve it. The performance of the hybrid algorithm is examined on an available test set, and it is shown that PR is more efficient than GA in this specific problem. Moreover, the combination of PR with SS is analyzed in [25]; they solved the capacitated  $p$ -median problem (CpMP). In the CpMP, the intention is to make a partition over a set of costumers, each of them has known demands, in exactly  $p$  facilities. They conducted a series of experiments on different sets, and the results were satisfactory in terms of the quality of the solutions found. Furthermore, they declared that the combination of PR with SS gave the best results. Recently, in [26, 27], path relinking is also hybridized with other heuristics showing good results. Based on these results, we were motivated for hybridizing this procedure within the evolutionary framework.

Next, the principal components and the phases involved in the hybrid algorithm are detailed and depicted in Pseudocode 1.

**3.1. Solution Encoding.** Besides the fact that we are dealing with two subsets of decision variables, solutions that are of

principal interest will be those referring to leader's solutions, that is, the facilities that will be opened. The bilevel nature of the problem allows for considering the solutions within the algorithm in that way. Hence, leader's solutions are represented as a binary string of size  $|I|$ , in which it is indicated if the facility is opened or not. It is clear that the follower's problem also must be solved and its corresponding solution is stored for computing the leader's objective function value. However, the encoding of the allocation of customers to facilities does not affect the hybrid algorithm. In next section, it is pointed out that the method selected for solving the follower's problem affects the performance of the algorithm, not its representation.

**3.2. Initial Population.** In this phase, a predetermined number of solutions are generated. We decided to create feasible solutions in a random manner. For constructing a solution, we generate a vector of random numbers from the same size of the total number of facilities and revise component by component as follows: if the respective number is less (greater) than 0.5, then the corresponding facility remains closed (opened).

**3.3. Selection Mechanism.** In this part, a number of solutions are chosen for being mated. We selected the solutions according to their fitness; while the solutions are better, they have more opportunity of being selected. The strategy that is used in this algorithm for achieving the latter is the tournament selection. This procedure consists in randomly matching each solution with another one from the population, and the best of both solutions is selected.

**3.4. Crossover (Path Relinking).** Crossover is executed after the selection. The path relinking procedure is used in the crossover phase. A pseudocode is shown in Pseudocode 2. In [28], a brief description of the PR is given; its main objective is to explore the search path between a set of (two) solutions. Additionally, PR generates a set of new solutions from the

First solution									
1	0	1	1	1	0	...	1	0	0
Second solution									
1	0	1	0	0	1	...	1	1	0
First movement									
1	0	1	1	0	0	...	1	0	0
Second movement									
1	0	1	1	0	1	...	1	0	0
...									

FIGURE 1: Path relinking.

good solutions in the set. Also, in [23], PR is described as an intensification strategy to explore trajectories connecting elite solutions obtained by heuristic methods. In this case, pairs of solutions are selected in a random way from the population and a crossover point is randomly selected. Here, the first solution suddenly becomes into the second one, so the trajectory is created. After the trajectory is obtained, the two best solutions are chosen. The crossover point is used in order to have a starting point for changing the elements of the first solution. This means that the first component which is different in both solutions after the crossover point will be changed in the first solution; that is, if the component is 0, then it will be changed into 1, or vice versa. This procedure is repeated for all subsequent elements until all components are explored (see Figure 1).

**3.5. Mutation.** After the selection and crossover of individuals were performed and if a random number is greater than a threshold value, the mutation takes place. The mutation involves changing some components of the string from 0 to 1 or vice versa. The associated probability for switching a component is given by an independent number for each one of them. Then, the components of the solutions are mutated in an independent manner; that is, the mutation of a component does not affect the mutation probability of another one.

Finally, the update of the population is given in the elitist way by using their corresponding fitness value. After the proposed algorithm has been described, the computational experimentation for showing its good performance will be presented in the next section.

#### 4. Computational Experimentation

In this section, the methodologies used to solve the UFLBP are described. Also, the computational results obtained from applying those methodologies are presented. First, we considered an evolutionary algorithm (EA) proposed in [15] and conducted the experimentation. Then, we hybridize the EA with a path relinking (PR) procedure, included in the crossover phase, as we mentioned in the previous section. Furthermore, within this hybridization, we propose three alternative ways to solve the lower level problem, which are as follows: (i) solving to optimality by using CPLEX, (ii) solving

to optimality by using an ordered matrix of preferences, and (iii) not solving the lower level at each step in the PR.

The latter alternative is based on the fact that, for having feasible solutions for the bilevel problem, the lower level needs to be optimally solved for each leader's solution. In the local procedure (PR), a new leader's solution is obtained in each move. Therefore, in order to measure the contribution of that specific move, the lower level problem is solved. Hence, the computational cost for following that scheme is high. Thus, it is interesting to investigate the possibility of not solving the lower level for the new leader's solution during all the PR procedure. By doing this, semiflexible solutions are being considered but after the methodology is done the lower level is optimally solved for the incumbent solution. However, due to the structure of the problem considered here, after a PR's move has been conducted, the customers must be reallocated. In other words, in the PR, a facility could be opened, closed, or interchanged and some of the customers allocated to that specific facility will need to be associated with other opened or more preferred facilities.

The experimentation was conducted on 3.60 GHz Intel Core i7-4790 with 32.00 GB RAM running under Windows 8.1 Pro operative system. Those algorithms were implemented on Visual Studio Express 2012 with C++ and, for the first alternative, the lower level was solved with CPLEX 12.6.1. The set of benchmark instances is the same as the one used in [15]. The set consists in 36 instances from three different sizes: 12 small, 12 medium, and 12 large size instances. The small ones contain 50 facilities and 50 customers. On the other hand, the medium size instances contain 50 facilities and 75 customers. Finally, 75 facilities and 100 customers are considered in the large size instances.

Initial testing indicates that the required time to solve the small size instances with the proposed hybrid algorithm (EA+PR) with CPLEX is 4410.455 seconds. As a consequence of the excessive computational time required by CPLEX for solving the lower level, we decided to discard this approach. Hence, we used the ordered matrix of customers' preferences for solving the lower level. In the case where we decide to avoid solving the lower level (this approach will be referred to as EA+PRw) at each step in the PR, two approaches for allocating the customers were proposed. For example, consider the situation presented in Figure 2(a) where an ordered pair associated with each customer represents the nearest open facility and the most preferred open facility, respectively. Consider the case when facility [3] is opened and facility [4] is closed; if we are under the approach when the lower level is not solved at each move of the PR, then the customers that were allocated to facility [4] are now unassigned (see Figure 2(b)). Hence, the first approach considered is when customers will be assigned to their nearest facility (see Figure 2(c)). The second approach is when customers will be assigned to their most preferred facility (see Figure 2(d)). Both approaches do not guarantee the customer's optimal allocation and semiflexible solutions are being considered at that time. However, once the PR procedure has finished, the optimal resolution of the lower level is done for obtaining bilevel feasible solutions.

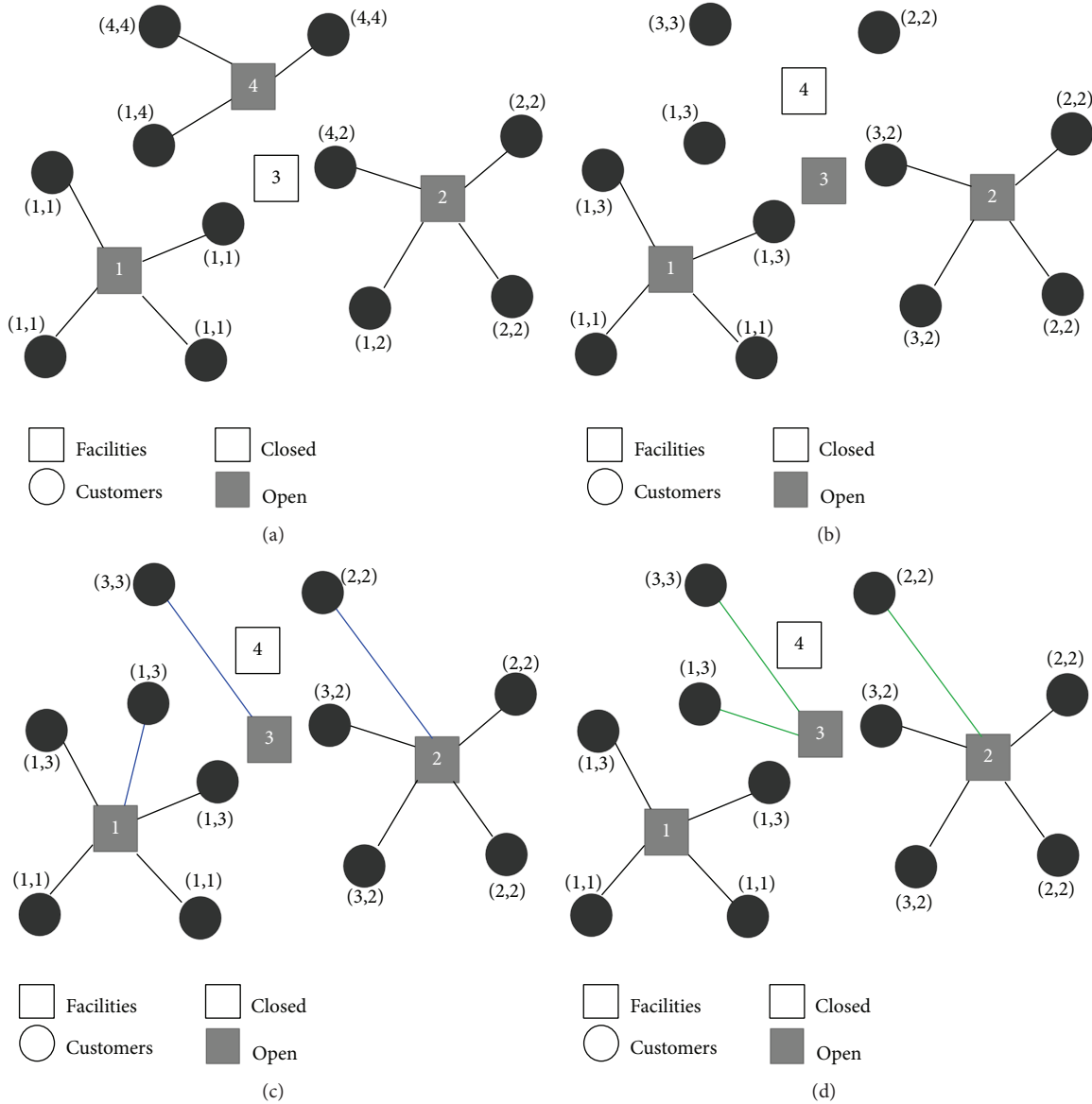


FIGURE 2: Exemplification for reallocating the customers.

Regarding the EA+PRw, the two considered approaches were analyzed and we can observe that both have good quality solutions (see Figures 3 and 4); we can observe that within the first 20 generations the “most preferred” approach has a better development, but in the next generations, the performance is almost the same for the “nearest” and “most preferred” approaches. Since this behavior remained for all the instances, half of the computational experimentation was done with the “nearest” approach and the other half under the “most preferred” approach.

In order to assess the performance of the hybrid algorithm, proper parameter identification has to be done. The parameters considered in the EA+PR are the size of the population ( $n$ ), the number of generations ( $G$ ), and the number of tournaments ( $T$ ). The selected parameters for the EA and the ones chosen for the probability in the genetic

operators (crossover and mutation) were selected exactly as in [15]. The parameter configuration selected for the EA+PR was obtained after preliminary tests; the corresponding values are  $n = 100$ ,  $G = 150$ , and  $T = 5$ . The main reason for reducing the number of generations in EA+PR is because its convergence to the optimal solution is faster than the convergence in the EA (see Figures 5 and 6).

In Tables 1, 2, and 3, the results obtained from applying 10 times the EA, EA+PR, and EA+PRw for each instance are shown. Table 1 corresponds to the results for the small size instances; this set is composed of three subsets of instances, 132, 133, and 134, that differ from each other on the values of distribution and fixed costs. All those instances consider four different matrices of preferences. The same structure is used in the medium and large size instances, which are presented in Tables 2 and 3, respectively. Each table contains the best

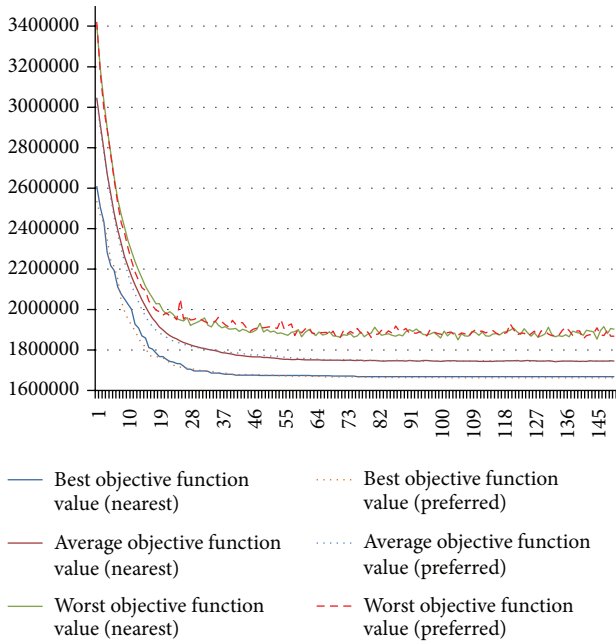


FIGURE 3: Illustrating the convergence for both approaches of EA+PRw in instance a-1 medium.

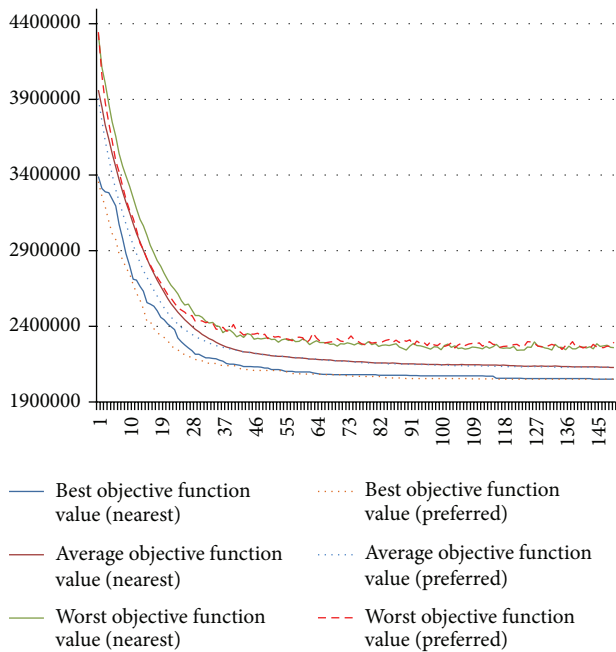


FIGURE 4: Illustrating the convergence for both approaches of EA+PRw in instance b-2 large.

gap found among the 10 runs, that is, the gap between the optimal value and the best value obtained for each approach (EA, EA+PR, and EA+PRw). Also, the average gap, which is the gap between the optimal value and the average of the last population value, is presented in the tables. And, finally, the average time, in seconds, consumed for solving each instance is shown. The gaps described above are computed as follows:

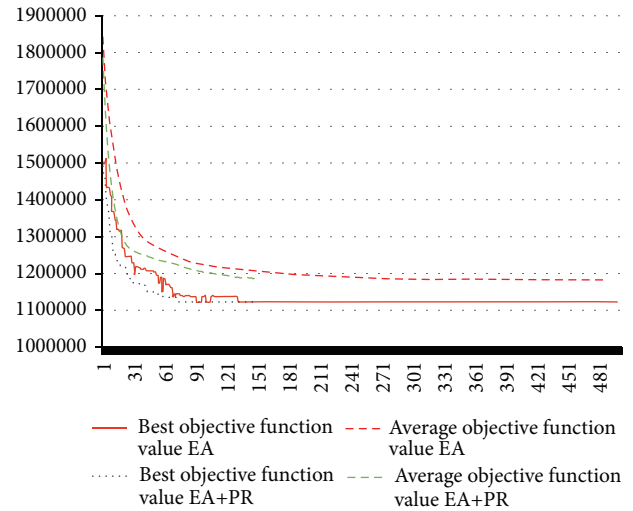


FIGURE 5: Illustrating the convergence for EA and EA+PR in instance 132-1.

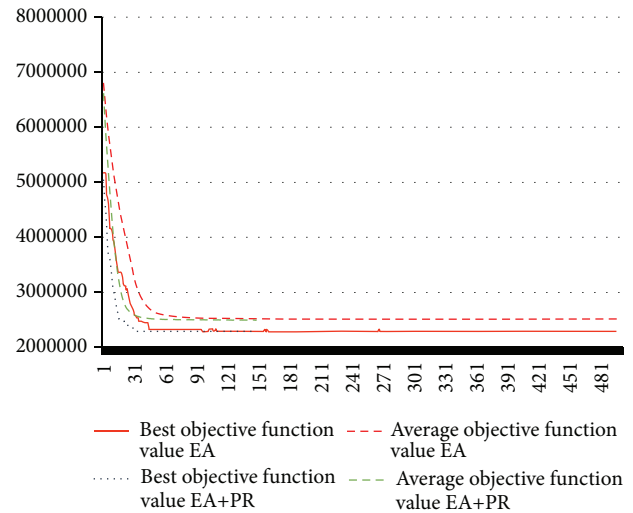


FIGURE 6: Illustrating the convergence for EA and EA+PR in instance b-1 large.

gap =  $100 \times ((x - x^*)/x^*)$ , where  $x$  represents the obtained solution and  $x^*$  the value. It is convenient to remark that optimal values for these instances are known in the literature.

It can be observed from Table 1 that there is no difference in the best gap between EA and EA+PR. In other words, both algorithms showed the same effectiveness but in the computational time the EA+PR is down to almost half of the EA. Regarding the EA+PRw, we can observe that the computational time is shorter than the other ones but for the best gap it is shown that the reached objective function value is worse in most of the instances with the selected parameter configuration than that for EA+PR.

From Table 2, it can be seen again that both EA and EA+PR seem to have the same good performance. However, the required computational time from EA+PR remains as low as half of the EA. Moreover, for the average gap, in 11 of the

TABLE 1: Results for the small size instances.

Instance	EA			EA+PR			EA+PRw		
	Best gap	Average gap	Time (sec)	Best gap	Average gap	Time (sec)	Best gap	Average gap	Time (sec)
132-1	0.00%	5.41%	1.51	0.00%	5.73%	0.76	0.00%	7.91%	0.72
132-2	0.00%	4.40%	1.45	0.00%	4.14%	0.74	0.00%	4.46%	0.73
132-3	0.00%	7.81%	1.53	0.00%	6.90%	0.77	0.00%	8.59%	0.73
132-4	0.00%	4.05%	1.49	0.00%	4.12%	0.78	0.00%	4.08%	0.74
133-1	0.00%	5.13%	1.53	0.00%	4.61%	0.76	0.28%	5.03%	0.74
133-2	0.00%	5.99%	1.51	0.00%	6.99%	0.72	0.00%	6.42%	0.71
133-3	0.00%	7.45%	1.54	0.15%	6.98%	0.79	0.15%	7.96%	0.73
133-4	0.00%	2.34%	1.50	0.00%	2.04%	0.75	0.00%	2.47%	0.71
134-1	0.00%	6.72%	1.52	0.00%	6.07%	0.78	0.00%	6.78%	0.74
134-2	0.00%	6.20%	1.54	0.00%	5.67%	0.78	0.00%	6.44%	0.73
134-3	0.00%	5.72%	1.52	0.00%	5.00%	0.78	0.00%	5.69%	0.74
134-4	0.00%	4.95%	1.50	0.00%	5.07%	0.75	0.00%	5.21%	0.71

TABLE 2: Results for the medium size instances.

Instance	EA			EA+PR			EA+PRw		
	Best gap	Average gap	Time (sec)	Best gap	Average gap	Time (sec)	Best gap	Average gap	Time (sec)
a-1	0.09%	4.89%	2.07	0.00%	4.44%	1.04	0.09%	5.04%	0.93
a-2	0.00%	6.51%	2.08	0.00%	5.59%	1.11	0.00%	6.86%	1.00
a-3	0.04%	4.72%	2.08	0.00%	4.22%	1.10	0.00%	4.93%	0.98
a-4	0.00%	5.48%	2.07	0.00%	5.02%	1.07	0.00%	5.57%	1.00
b-1	0.00%	5.68%	2.06	0.00%	5.76%	1.08	0.00%	6.08%	1.00
b-2	0.10%	3.70%	2.07	0.00%	3.10%	1.12	0.00%	4.72%	1.08
b-3	0.00%	5.06%	2.07	0.00%	4.52%	1.10	0.49%	5.57%	1.02
b-4	0.00%	4.80%	2.04	0.02%	4.45%	1.02	0.52%	5.06%	0.98
c-1	0.26%	2.95%	2.03	0.26%	2.54%	1.10	0.26%	2.96%	1.04
c-2	0.25%	3.46%	2.04	0.00%	3.12%	1.10	0.00%	3.92%	1.04
c-3	0.00%	3.71%	2.01	0.00%	3.52%	1.06	0.00%	3.91%	1.00
c-4	0.00%	3.03%	2.04	0.00%	2.72%	1.05	0.00%	3.14%	0.99

TABLE 3: Results for the large size instances.

Instance	EA			EA+PR			EA+PRw		
	Best gap	Average gap	Time (sec)	Best gap	Average gap	Time (sec)	Best gap	Average gap	Time (sec)
a-1	0.00%	9.76%	3.55	0.00%	8.89%	2.07	0.00%	9.89%	1.84
a-2	0.00%	6.94%	3.49	0.00%	6.30%	2.02	0.00%	7.12%	1.77
a-3	0.00%	7.23%	3.48	0.00%	6.25%	2.09	0.00%	7.23%	1.86
a-4	0.00%	5.85%	3.61	0.00%	5.08%	2.11	0.00%	5.72%	1.85
b-1	0.77%	5.39%	3.67	0.77%	4.72%	2.35	0.87%	6.09%	2.09
b-2	1.01%	4.71%	3.65	0.29%	4.00%	2.16	0.63%	5.29%	2.05
b-3	0.00%	4.37%	3.65	0.00%	4.07%	2.13	0.86%	4.48%	2.00
b-4	0.00%	6.47%	3.71	0.00%	5.88%	2.25	1.14%	6.74%	2.08
c-1	0.00%	4.04%	3.69	0.00%	3.65%	2.22	0.00%	4.21%	2.00
c-2	0.00%	3.82%	3.65	0.00%	4.17%	2.28	0.00%	4.62%	2.05
c-3	0.00%	4.53%	3.61	0.00%	4.05%	2.12	0.00%	4.91%	2.00
c-4	0.00%	4.67%	3.69	0.00%	4.35%	2.25	0.00%	5.14%	2.05



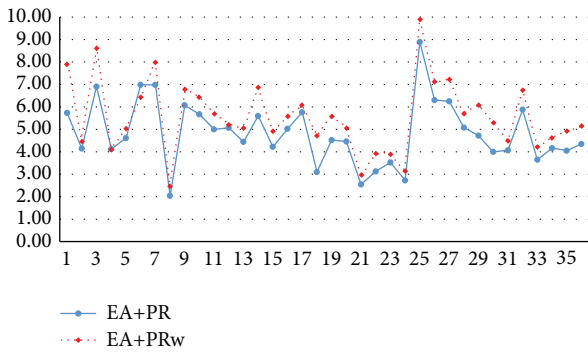


FIGURE 7: Comparing the average gaps.

12 instances, the EA+PR has a lower value than the EA. The latter can be interpreted as that the quality, in terms of leader's objective function value, from the entire population is better for the EA+PR. On the other hand, when comparing the EA+PR with EA+PRw, we can observe that the performance gets worse in most of the cases despite the fact that the computational time is lower than the EA+PR and EA. In a reasoned way, this was expected; that is, avoiding solving the lower level problem at each step during the local search leads to a blinded exploration. Furthermore, semiflexible solutions are being considered, which later will have to be repaired for reaching bilevel feasibility.

Finally, the behavior discussed in the above paragraphs is maintained for the large size instances. The "best gap" and "average gap" columns do not show a significant difference between EA and EA+PR. However, as before, the computational time required for the EA+PR is also 1 second less than the EA. In addition, the EA+PR has a better performance in terms of quality than the EA+PRw but the consumed time is almost the same. This is caused because in the EA+PRw when a facility is closed, the customers associated with it need to be reallocated following a predefined criterion. Hence, a quickly reallocation is made; but the required time for solving the lower level problem using the ordered matrix of preferences is very low. Then, when the algorithm reaches its stop criterion, the cumulative time reduction is not as significant as we expected.

The average gap's values obtained by both EA+PR and EA+PRw are plotted in Figure 7 for all the tested instances. The results support the ideas obtained from the discussion presented previously; that is, the performance of the EA+PRw is not as good as the EA+PR in most of the cases due to the semiflexible solutions. It can be seen from Figure 7 that only twice (instances 132-4 and 133-2) the value for the EA+PRw is better than the value that corresponds for the EA+PR, but in all other instances, it is the worst. For example, in the 132-1 and b-2 (medium size) instances, the difference between the gaps is 2.17 and 1.62, respectively.

## 5. Conclusions and Further Research

In this paper, a hybrid evolutionary algorithm with path relinking for solving the uncapacitated facility location

problem with customer's preferences is proposed. A computational analysis which involves three approaches for dealing with the preferences' problem associated with the lower level is presented. First, the EA is implemented in the same manner as it was proposed in [15]. Then, the hybrid EA+PR is developed for exploiting the bilevel structure of the problem. Finally, a variant of the hybrid named EA+PRw was tested. The variant consisted in the fact of not having to solve the lower level in each movement within path relinking.

As a result from the computational experimentation, it seems that the computational time required to solve the lower level is very short. For example, as we showed in Section 4, EA+PR has a better performance than EA+PRw; and regarding the computational time, there is an insignificant difference between both algorithms. However, if solving the lower level requires an excessive computational time, then it will be convenient not to solve the lower level for each leader's solution. Thus, semiflexible solutions will be recommended for guiding the search. As a result of the latter approach, we can observe that the quality of the solutions would not improve significantly; in spite of that, the computational time will decrease noticeably. Hence, we remark the importance of having alternative options for optimally solving the lower level problem instead of using commercial optimization software. Nevertheless, this issue depends on the structure of the problem being studied.

Picking up the abovementioned ideas, future research directions could involve the advantage of some particular properties existing in the lower level that may help to sort this issue. That is to avoid a blind search in the lower level, because of the use of semiflexible solutions due to the fact that the lower level is not solved in several consecutive iterations in the methodology selected.

For example, recently in [29], an attempt for approximating the lower level optimal solutions without explicitly solving it is presented. The approximation consists in creating quadratic functions based on a set of initial bilevel feasible solutions. Then, for an upper level solution, the lower level reaction is approximated by using those functions. After a fixed number of steps following this approach, the lower level is optimally solved for each upper level solution. By doing this, bilevel feasible solutions are obtained replacing the bilevel semiflexible ones. These ideas can be used in the EA+PRw for improving its performance.

Another option is to apply a decomposition approach based on the primal and dual relationships of the problem, for example, as in [30] in which a decomposition approach was applied for obtaining high quality solutions in a short term generation planning. In that problem, the computational effort is increased in an exponential manner as the number of the components involved in the problem also augments. Hence, the decomposition approach showed that it is a good alternative for dealing with high complexity problems.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

The research activity of all the authors was partially supported by the Mexican National Council for Science and Technology (CONACYT) through Grant SEP-CONACYTCB-2014-01-240814 and by the Support Program for Scientific Research and Technology (PAICYT-UANL) with Project CE-2015.

## References

- [1] A. Weber, *Theory of the Location of Industries*, University of Chicago Press, Chicago, Ill, USA, 1929.
- [2] A. A. Kuehn and M. J. Hamburger, "A heuristic program for locating warehouses," *Management Science*, vol. 9, no. 4, pp. 643–666, 1963.
- [3] M. S. Daskin, *Network and Discrete Location: Models, Algorithms and Applications*, John Wiley & Sons, New York, NY, USA, 1995.
- [4] J. Puerto and F. R. Fernández, "The symmetrical single facility location problem," Tech. Rep., Faculty of Mathematics, University of Seville, Seville, Spain, 1995.
- [5] S. Nickel, "Discrete ordered Weber problems," in *Operations Research Proceedings*, pp. 71–76, Springer, Berlin, Germany, 2001.
- [6] N. Boland, P. Domínguez-Marín, S. Nickel, and J. Puerto, "Exact procedures for solving the discrete ordered median problem," *Computers & Operations Research*, vol. 33, no. 11, pp. 3270–3300, 2006.
- [7] J. Kalcsics, S. Nickel, J. Puerto, and A. M. Rodríguez-Chía, "The ordered capacitated facility location problem," *TOP*, vol. 18, no. 1, pp. 203–222, 2010.
- [8] S. Nickel and J. Puerto, *Location Theory: A Unified Approach*, Springer, Berlin, Germany, 2005.
- [9] P. Hanjoul and D. Peeters, "A facility location problem with clients' preference orderings," *Regional Science and Urban Economics*, vol. 17, no. 3, pp. 451–473, 1987.
- [10] G. Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann, *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*, Springer, Berlin, Germany, 1999.
- [11] P. Hansen, Y. Kochetov, and N. Mladenovic, "Lower bounds for the uncapacitated facility location problem with user preferences," Preprint G 2004-24, GERAD-HEC, Montreal, Canada, 2004.
- [12] I. L. Vasil'ev, K. B. Klimentova, and Y. A. Kochetov, "New lower bounds for the facility location problem with client preferences," *Computational Mathematics and Mathematical Physics*, vol. 49, no. 6, pp. 1055–1066, 2009.
- [13] I. L. Vasil'ev and K. B. Klimentova, "The branch and cut method for the facility location problem with client's preferences," *Journal of Applied and Industrial Mathematics*, vol. 4, no. 3, pp. 441–454, 2010.
- [14] M. Marić, Z. Stanimirović, and N. Milenković, "Metaheuristic methods for solving the bilevel uncapacitated facility location problem with clients' preferences," *Electronic Notes in Discrete Mathematics*, vol. 39, pp. 43–50, 2012.
- [15] J.-F. Camacho-Vallejo, A. E. Cordero-Franco, and R. G. González-Ramírez, "Solving the bilevel facility location problem under preferences by a Stackelberg-evolutionary algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 430243, 14 pages, 2014.
- [16] H. I. Calvete, C. Galé, and J. A. Iranzo, "An efficient evolutionary algorithm for the ring star problem," *European Journal of Operational Research*, vol. 231, no. 1, pp. 22–33, 2013.
- [17] M. Gallo, L. D'Acierno, and B. Montella, "A meta-heuristic approach for solving the urban network design problem," *European Journal of Operational Research*, vol. 201, no. 1, pp. 144–157, 2010.
- [18] J.-F. Camacho-Vallejo, J. Mar-Ortiz, F. López-Ramos, and R. P. Rodríguez, "A genetic algorithm for the bi-level topological design of local area networks," *PLoS ONE*, vol. 10, no. 6, Article ID e0128067, 2015.
- [19] H. Küçükaydin, N. Aras, and I. K. Altinel, "A hybrid tabu search heuristic for a bilevel competitive facility location model," *Lecture Notes in Computer Science*, vol. 6373, pp. 31–45, 2010.
- [20] J. S. Angelo and H. J. C. Barbosa, "A study on the use of heuristics to solve a bilevel programming problem," *International Transactions in Operational Research*, vol. 22, no. 5, pp. 861–882, 2015.
- [21] F. Glover, J. P. Kelly, and M. Laguna, "Genetic algorithms and tabu search: hybrids for optimization," *Computers & Operations Research*, vol. 22, no. 1, pp. 111–134, 1995.
- [22] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and L. M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm," *Information Processing Letters*, vol. 93, no. 5, pp. 255–261, 2005.
- [23] M. G. Resende, C. C. Ribeiro, F. Glover, and R. Martí, "Scatter search and path-relinking: fundamentals, advances, and applications," in *Handbook of Metaheuristics*, pp. 87–107, Springer, New York, NY, USA, 2010.
- [24] M. Ranjbar, F. Kianfar, and S. Shadrokh, "Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm," *Applied Mathematics and Computation*, vol. 196, no. 2, pp. 879–888, 2008.
- [25] J. A. Díaz and E. Fernandez, "Hybrid scatter search and path relinking for the capacitated  $p$ -median problem," *European Journal of Operational Research*, vol. 169, no. 2, pp. 570–585, 2006.
- [26] D. Ferone, P. Festa, and M. G. Resende, "Hybridizations of GRASP with path relinking for the far from most string problem," *International Transactions in Operational Research*, vol. 23, no. 3, pp. 481–506, 2016.
- [27] L. S. Pessoa, M. G. C. Resende, and C. C. Ribeiro, "A hybrid Lagrangean heuristic with GRASP and path-relinking for set  $k$ -covering," *Computers & Operations Research*, vol. 40, no. 12, pp. 3132–3146, 2013.
- [28] E. Vallada and R. Ruiz, "Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem," *Omega*, vol. 38, no. 1-2, pp. 57–67, 2010.
- [29] A. Sinha, P. Malo, and K. Deb, "An improved bilevel evolutionary algorithm based on quadratic approximations," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '14)*, pp. 1870–1877, July 2014.
- [30] J. A. Marmolejo-Saucedo and R. Rodríguez-Aguilar, "Short-term generation planning by primal and dual decomposition techniques," *DYNA*, vol. 82, no. 191, pp. 58–62, 2015.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

