MULTI-OBJECTIVE GRASP WITH PATH-RELINKING

RAFAEL MARTÍ, VICENTE CAMPOS, MAURICIO G.C. RESENDE,
AND ABRAHAM DUARTE

Abstract. In this paper we propose an adaptation of the GRASP metaheuristic to solve multi-objective combinatorial optimization problems. In particular we describe several alternatives to specialize the construction and improvement components of GRASP when two or more objectives are considered. GRASP has been successfully coupled with path-relinking for single-objective optimization. In this paper, we propose different hybridizations of GRASP and path-relinking for multi-objective optimization. We apply the proposed GRASP with path-relinking variants to two combinatorial optimization problems, the bi-objective orienteering problem and the bi-objective path dissimilarity problem. We report on empirical tests with 70 instances that show that the proposed heuristics are competitive with the state-of-the-art methods for these problems.

1. Introduction

The GRASP metaheuristic was developed in the late 1980s (Feo and Resende, 1989; 1995). The acronym was coined in Feo et al. (1994).We refer the reader to Resende and Ribeiro (2003; 2010) for recent surveys of this metaheuristic. In short, each GRASP iteration consists in constructing a trial solution with some greedy randomized procedure and then applying local search from the constructed solution. This two-phase process is repeated until some stopping condition is satisfied. A best local optimum found over all local searches is returned as the solution of the heuristic.

The algorithm in Figure 1 shows pseudo-code for a generic GRASP for minimization. The greedy randomized construction seeks to produce a diverse set of good-quality starting solutions from which to start the local search phase. Let x be the partial solution under construction in a given iteration and let C be the candidate set with all the remaining elements that can be added to x. The GRASP construction uses a greedy function $g(c)$ to measure the contribution of each candidate element $c \in C$ to the partial solution x. A restricted candidate list $RCL(C)$ is the subset of candidate elements from C with good evaluations according to g. At each step, the method randomly selects an element c^* from the restricted candidate list and adds this element to the partial solution. The construction is repeated in the inner while loop (steps 4 to 10) until there are no further candidates. If $C = \emptyset$ and x is infeasible, then a repair procedure needs to be applied to make x feasible (steps 11 to 13). Once a feasible solution x is on hand, a local search improvement is

Date: April 18, 2011.

Key words and phrases. Multi-objective optimization, Bi-objective optimization, GRASP, Path-relinking.

AT&T Labs Research Technical Report.

```
begin GRASP
f* ← ∞;
while stopping criterion not satisfied do
    x ← ∅;
    Compute C with the candidate elements that can be added to x;
    while C ≠ ∅ do
        For all c ∈ C compute greedy function value g(c);
        Define RCL(C) ← {c ∈ C | g(c) has a good value};
        Select c* at random from RCL(C);
        Add c* to partial solution: x ← x ∪ {c*};
        Update C with the candidate elements that can be added to x;
    end-while;
    if x is infeasible then
        Apply a repair procedure to make x feasible;
    end
    x ← LocalSearch(x);
    if f(x) < f(x*) then
        x* ← x; f* ← f(x);
    end
end
return x*;
```

FIGURE 1. GRASP algorithm for minimization of $f(x)$.

applied. The resulting solution is a local minimum. The GRASP algorithm terminates when a stopping criterion is met (typically a maximum number of iterations, time limit, or a target solution quality). The best overall solution x^* is returned as the output of the heuristic.

In this paper, we deal with multi-objective optimization problems, in which, without loss of generality, we want to minimize k objective functions: $f_1, f_2, \ldots, f_k$. Specifically, we want to determine the set of efficient points (usually called the efficient Pareto frontier). A point or solution x^* is said to be efficient if there is no other solution x such that $f_i(x) \leq f_i(x^*)$ for all $i = 1, \ldots, k$ and $f_j(x) < f_j(x^*)$ for at least one $j \in \{1, \ldots, k\}$. Essentially, efficiency means that a solution to a multiobjective function is such that no single objective can be improved without deteriorating another objective. Since we propose a heuristic procedure, we obtain an approximation to the set of efficient points.

In Vianna and Arroyo (2004) a GRASP is proposed for the multi-objective knapsack problem. In particular, at each iteration, the construction and the local search are guided by a weighted combination of the objectives,

$$f(x) = \sum_{j=1}^{k} \lambda_j f_j(x), \tag{1}$$

where the preference coefficient λ_j is computed in a particular way in order to obtain a variety of solutions uniformly distributed in the Pareto frontier. Specifically, for a given value of s, they create all the possible vectors $(w_1, w_2, \ldots, w_k)$ where

$\sum_{j=1}^{k} w_j = s$. Then, each GRASP construction and local search is performed with the preference vector set to $\lambda_j = w_j/s$ for $j = 1, \ldots, k$.

The problem of environmental investment decision making is considered in Higgins et al. (2008) to maximize multiple environmental benefits within a budget constraint. In particular, the authors consider travel time of water maximization, biodiversity and carbon sequestration. The environmental investment problem is an extension of the bi-criteria knapsack problem and a GRASP algorithm is proposed to approximate the efficient frontier. The construction phase is guided by a weighed summation objective with random weights in each of the three objective functions. Producing solutions using different sets of weights allow them to be distributed along the Pareto frontier. The method maintains a population of solutions both dominated and non-dominated for the sake of diversity. This set evolves by applying a local search to randomly selected solutions until the time limit is reached.

The multicriteria minimum spanning tree is faced in Arroyo et al. (2008) and a GRASP is proposed for its solution. The construction phase uses Kruskal's algorithm and the local search is based on a *drop-and-add* neighborhood. The construction is guided by a weighted combination of the objectives where the preference coefficient λ_j is computed as in Vianna and Arroyo (2004). Given a constructed solution, the local search generates a new spanning tree by dropping and adding edges. The method is compared with a multi-objective version of the Kruskal algorithm.

In (Ishida et al., 2009) a GRASP with Path Relinking for learning classification rules is proposed, where the goal is to create rules that together have good performance for classification. A frequent measure used to evaluate the performance of a classifier is the AUC, the area under the ROC curve (the curve that relates the false and the true positive proportions). The authors propose an alternative approach based on two criteria: *sensitivity* and *specificity*. In this way, they face a bi-objective problem. They show that their GRASP with Path Relinking algorithm obtains an approximation of the Pareto front that gives a good AUC value.

Li and Landa-Silva (2009) proposed a GRASP for the multi-objective quadratic assignment problem. In this version of the well-known QAP, multiple types of flows are considered between any two facilities. The proposed algorithm, called mGRASP, is characterized by three features: elite greedy randomized construction, adaptation of search directions and cooperation between solutions. To find a diverse set of Pareto optimal solutions, mGRASP uses multiple distinct weight vectors evenly spread in the construction and local search phases. Unlike the classical GRASP algorithm, their method constructs each solution by adding some elements from the previous local optima found.

In Reynolds and de la Iglesia (2009) a GRASP is proposed for the partial classification of a database. This problem basically consists of finding simple classification rules that represent strong descriptions for a particular class of database. Association rules can be evaluated according to a number of conflicting criteria, which has lead to the application to multiobjective metaheuristics. In their GRASP implementation, the authors first apply construction phase, based on a random weighted combination of the objectives, and create a set of non-dominated solutions. The local search phase is then applied to the non-dominated solutions. A comparison with previous evolutionary methods for multiobjective optimization, favors the proposed GRASP. A refinement of this method is proposed in Reynolds et al. (2009) to select a small subset of the rules previously identified.

We have also found multi-objective GRASP applications in the context of flow shop scheduling problems. In particular, Davoudpour and Ashrafi (2009) considered the hybrid flow shop scheduling problem with sequence dependent setup times. The measure of performance of a given solution is computed as a function of the assigned due date of each job, in terms of the earliness, tardiness and completion time. They are combined in a single objective and the proposed GRASP solves the associated mono objective problem.

In this paper we propose different adaptations of the single-objective GRASP outlined in Figure 1 to the multi-objective case. Specifically, we consider the extension of the construction and improvement phases. Moreover, we also include a post-processing phase based on path-relinking for multi-objective optimization. Path Relinking has been successfully hybridize with GRASP in many mono-objective problems, as documented in Resende and Ribeiro (2010).

In Section 2, we propose an adaptation of the GRASP construction phase for multi-objective optimization. Similarly, Sections 3 and 4, respectively, describe the local search phase of GRASP and the path-relinking post-processing phase. Computational experiments with the bi-objective orienteering problem and the bi-objective path dissimilarity problem are described in Section 6. Concluding remarks are made in Section 7.

2. Multi-Objective Construction

In single-objective GRASP, each construction is guided by a greedy function $g(c)$ which measures the contribution of each candidate element $c \in C$ to the partial solution x under construction. In multi-objective GRASP, we have k greedy functions, $g_1, g_2, \ldots, g_k$, where $g_i(c)$ evaluates candidate element c with respect to objective f_i. We distinguish two types of constructions which we call *pure* and *combined*.

In **pure construction** a single objective is considered during a single construction, while in combined construction different objectives guide a single construction. Within the pure construction category we differentiate between two types of methods: Those in which the objective to be considered in a construction is randomly selected (*pure-random*) and those in which the objective is selected in an ordered fashion (*pure-ordered*).

In **pure-random construction**, we randomly select a greedy function g_i for each construction. In each step of a construction we evaluate $g_i(c)$ for each candidate element $c \in C$ to compute the restricted candidate list $RCL(C)$. It should be noted that we only select a greedy function once in each construction and use it in all the steps of the construction. In other words, each entire construction is guided by a single objective function (the one associated with the greedy function selected).

In **pure-ordered construction**, each construction is guided by a different objective, selected one at a time in an ordered fashion. Specifically, in the first construction we evaluate the candidate elements with g_1, in the second construction with g_2, and so on, until we reach the $k+1$-th construction, in which we resort again to g_1. In short, we follow the order of the objectives across different constructions.

- Pure
  - Random (between constructions)
  - Ordered (between constructions)
- Combined
  - Sequential
    - Random (within a construction)
    - Ordered (within a construction)
  - Weighted

FIGURE 2. A classification.

In this way, it is expected that each construction will produce a solution of good quality with respect to the objective that is evaluated.

The **combined construction** considers more than one objective in each construction. We can distinguish between two types of methods. We call *sequential* those in which each construction step is guided by a different objective, and *weighted* those in which each step is guided by a weighted function of the evaluations.

In the **sequential combined** methods, a greedy function is selected at each step of a given construction. Let x be the partial solution under construction in iteration i, and let C_i be the candidate set with all the remaining elements that can be added to x. We evaluate the elements in C_i with a function g_j, for some $j = 1, \ldots, n_g$, thus computing $RCL(C_i) = \{c \in C_i \,|\, g_j(c) \text{ has a good value}\}$. If the function g_j is randomly selected in each iteration we call the combined method *random-sequential*. Alternatively, when the function g_j is selected in order, where we use g_1 in iteration 1, g_2 in iteration 2, and so on, we call the combined method *ordered-sequential*.

Finally, in the **weighted combined** methods, we consider a weighted combination

$$g(c) = \sum_{j=1}^{k} w_j g_j(c) \tag{2}$$

of the evaluation functions in step i of the construction, where w_j is the weight of the evaluation function g_j. We then compute $RCL(C_i) = \{c \in C_i \,|\, g(c) \text{ has a good value}\}$. We can either keep the same weights across different constructions or change them in each construction step. Note that the evaluations (objectives) can have magnitudes that vary significantly and in this case the weights help us scale them into similar (and comparable) magnitudes. Moreover, in multi-objective optimization, some objectives can be minimized while others maximized and therefore the weights can take positive and negative values to reflect this fact.

Figure 2 summarizes this classification in which we have identified five different schemes to design a constructive method: pure-random, pure-ordered, random-sequential combined, ordered-sequential combined, and weighted combined.

3. Multi-Objective Local Search

Local search, also known as neighborhood search, proceeds iteratively from one solution to another until no further improvement is possible. Each solution x has an associated neighborhood $N(x)$, and each solution $y \in N(x)$ is reached from x by an operation called *move*.

In the local search we can define the same two main strategies as in the construction methods, *pure* and *combined*, according to the way in which we select the objective functions.

If we obtain a solution x with a pure constructive method guided by objective function f_i, we attempt to improve it with a local search also guided by the same function. Since x was constructed only considering f_i and ignoring the rest of the objectives, we permit the deterioration of these other objectives in the **pure local search** while improving f_i. The method stops when objective f_i cannot be further improved.

If we obtain a solution x with a combined constructive method, we distinguish whether it is a sequential or a weighted combined method. In the sequential construction, different objectives are applied. We therefore do not allow the objective functions to deteriorate during the local search phase. At each step of the **sequential-combined local search**, we consider a different objective function when selecting the best solution in the neighborhood. The method stops when no objective can be further improved (without deteriorating any of the others).

Finally, if we obtain a solution x with a weighted-combined construction method using (2), we consider in the **weighted-combined local search**, the weighted objective function

$$f(x) = \sum_{j=1}^{k} w_j f_j(x).$$

In this way, this local search is guided by the same objective function as its associated construction method. We can either keep the same weights across different moves or change them in each one. The local search stops when $f(x)$ cannot be further improved.

We apply a post-processing phase within the improvement method to certify the local optimality with respect to all the objectives. In particular, before terminating the local search we attempt to improve, one-by-one, each of the objectives without deteriorating any of the others. We select them in order, from 1 to k, performing at each iteration the best associated move with respect to the corresponding objective. This post-processing finishes when no objective can be improved. Note that in the sequential-combined local search this process is not necessary because it already applies it by definition.

It is worth mention that an important difference between mono-objective local search and multi-objective local search is that in the former we only need to check if the final solution obtained (the local optima) improves upon the best known solution. On the contrary, in multi-objective local search, every solution visited have to be checked for its possible inclusion in the set of non-dominated solutions.

4. Multi-Objective Path-relinking

Path-relinking (PR) was suggested as an approach to integrate intensification and diversification strategies in the context of tabu search (Glover and Laguna (1997)). This approach generates new solutions by exploring trajectories that connect high-quality solutions by starting from one of these solutions, called an *initiating solution*, and generating a path in the neighborhood space that leads toward the other solutions, called *guiding solutions*. This is accomplished by selecting moves that introduce attributes contained in the guiding solutions, and incorporating them in an *intermediate solution* initially originated in the initiating solution.

Laguna and Martí (1999) adapted PR in the context of GRASP as a form of intensification. The relinking in this context consists in finding a path between a solution found with GRASP and a chosen elite solution. Therefore, the relinking concept has a different interpretation within GRASP since the solutions found in one GRASP iteration to the next are not linked by a sequence of moves (as in the case of tabu search). Resende and Ribeiro (2003) present numerous examples of GRASP with PR.

Let x and y be two solutions of the multi-objective problem. The path relinking procedure $\mathrm{PR}(x, y)$ starts with the first solution x, and gradually transforms it into the second one y, by swapping out elements in x with elements in y. The elements in both solutions x and y, remain in the intermediate solutions generated in the path between them. Note that an element can be a node in a graph, a value of a variable, an edge or a path in a network, or any other attribute depending on the particular problem that we are solving.

Let El_{x-y} be the set of elements in x and not present in y and symmetrically, let El_{y-x} be the set of elements in y and not present in x. Let $p_0(x, y) = x$ be the initiating solution in the path $\mathrm{P}(x, y)$ from x to y. To obtain the solution $p_1(x, y)$ in this path, we can remove from x a single element $i \in El_{x-y}$, or add an element $j \in El_{y-x}$, or both (add i and remove j) thus obtaining

$$El_{p_1(x,y)} = El_{p_0(x,y)} \setminus \{i\},$$
or
$$El_{p_1(x,y)} = El_{p_0(x,y)} \cup \{j\},$$
or
$$El_{p_1(x,y)} = El_{p_0(x,y)} \setminus \{i\} \cup \{j\}.$$

Following the classification proposed in the previous sections, we consider here three implementations of multi-objective path relinking. In the **pure path relinking** algorithm, the selection of the elements i and j is made according to one objective function. To obtain $p_{k+1}(x, y)$ from $p_k(x, y)$, we evaluate all the possibilities for $i \in El_{p_k(x,y)-y}$ to be removed and $j \in El_{y-p_k(x,y)}$ to be added, and perform the best swap in terms of one objective function. In each application of PR, we consider one objective function, say f_i and use it to select the intermediate solutions in the entire path $\mathrm{P}(x, y)$. On the contrary, in the **sequential path relinking** we alternate the objective function used to select the intermediate solutions. In this way, if we use f_i to select $p_k(x, y)$, we then use f_{i+1} to select $p_{k+1}(x, y)$. Finally in the **weighted path relinking** the weighted objective function

$$f(x) = \sum_{j=1}^{k} w_j f_j(x).$$

is used to select all the intermediate solutions in every application of $\mathrm{PR}(x, y)$.

The PR algorithm operates on a set of solutions, called *elite set* (ES), constructed with the application of a previous method. In this paper, we apply GRASP to build the elite set. In a multi-objective problem we can identify this set with the set of non-dominated solutions. Initially *ES* is empty, and we apply GRASP for a certain number of iterations to populate it with the non-dominated solutions obtained. Then, in the following iterations, we apply PR to all the pairs of solutions in *ES*. Specifically, for each pair x and y we apply PR(x, y) and PR(y, x). The GRASP with PR algorithm terminates when all the pairs in *ES* have been submitted to the PR method. As previously documented (Laguna and Martí (1999)) we can apply a local search to some of the intermediate solutions in every PR path to obtain improved outcomes.

All the intermediate solutions from x and y, found in the P(x, y) path (i.e., $p_1(x, y), p_2(x, y), \ldots, p_{r-1}(x, y)$ where $r = |El_{x-y}| = |El_{y-x}|$) have to be checked for their possible inclusion in the set of non-dominated solutions found by the algorithm. To simplify the design, we store these intermediate solutions in a pool, *Intermediate Pool*(*IP*), and do not check whether they are non-dominated or not until the Path Relinking method finishes. At that point, we merge the elite set, *ES*, which initially contained the non-dominated solutions, with *IP* and return the non-dominated solutions, considering both sets, as the output of the method. In *Evolutionary Path Relinking*, instead of stopping the search at this point, we would apply Path Relinking again to the new non-dominated set of solutions (identifying it as the new elite set, *ES*). This entire process is applied as long as PR is able to generate solutions dominating solutions in *ES* (i.e., while intermediate solutions qualify to enter in the new *ES*).

5. Optimization problems used for Testing

We have used two bi-objective combinatorial optimization problems to test the different GRASP and Path Relinking variants proposed in the previous sections.

- The path dissimilarity problem
- The bi-orienteering problem

We target these problems because they are well-known, they are different in nature and high quality solutions to several problem instances are available. We now provide a brief description of each problem class.

The **path dissimilarity problem** (PDP) (Dell'Ollmo et al., 2005) is a bi-objective routing problem in which a set of p paths from an origin to a destination must be generated with minimum length and maximum dissimilarity. Finding different paths in a graph is a classical optimization problem. The best known is the s-shortest path problem in which the shortest, second shortest, s-th shortest paths from an origin o to a destination d are obtained in a graph. However, many of these alternative paths are likely to share a large number of edges. This is why in some applications we need to consider an alternative approach. For example, in the context of hazmat transportation we want to obtain spatially dissimilar paths that minimize the risk (distributing the risk over all regional zones to be crossed uniformly).

Given an undirected graph $G = (V, E)$ with V the set of vertices and E the set of edges with associated cost c_{ij} for $(i, j) \in E$, and a pair of origin-destination vertices, $o - d$, we define $P(o, d)$ as the set of all paths in G from o to d. Note that in most applications the cost c_{ij} of edge (i, j) is its Euclidean distance. Given an integer number $p > 1$, a feasible solution to the path dissimilarity problem, PDP, is a set $S \subseteq P(o, d)$ such that $\mid S \mid= p$. Given a solution $S = \{P_1, P_2, \ldots, P_p\}$, we define its value $f_1(S)$ as the average of the costs of the paths in S:

$$f_1(S) = \frac{\sum_{t=1}^{p} c(P_t)}{p} \quad \text{where} \quad c(P_t) = \sum_{(i,j)\in P_t} c_{ij}$$

We also define its dissimilarity value $f_2(S)$ as the average of the dissimilarity between the $\binom{p}{2}$ distinct pairs of paths in S:

$$f_2(S) = \frac{\sum_{i=1}^{p-1} \sum_{j=i+1}^{p} dis(P_i, P_j)}{\binom{p}{2}}$$

As applied in Martí et al. (2009) the dissimilarity $dis(P_i, P_j)$ between two paths P_i and P_j is computed as the average of the distances between each vertex in P_i to the path P_j plus the average of the distances between each vertex in P_j to the path P_i.

With these elements, we can formulate the PDP as:

$$\begin{aligned} \text{(PDP) min} \quad & f_1(S) \\ \max \quad & f_2(S) \\ \text{subject to} \quad & S \subseteq P(o,d) \\ & \mid S \mid = p \end{aligned}$$

The **bi-orienteering problem** (BOP) considered here is a bi-objective optimization problem that is a generalization of the single-objective version also known as the selective traveling salesman problem, introduced by Tsiligirides (1984). In the OP, each vertex of a given directed graph $G = (V, A)$ has two different profits. The aim of this problem is to select a subset of vertices in order to maximize the sum of both profits. Moreover, the tour visiting the selected vertices cannot exceed a maximum length (or time). The motivation of this problem was the planning of a set of tourist routes in a large city. Each point of interest has different profits associated with different activities (say for instance culture and leisure). Since the maximization of the profits associated with one activity does not imply the maximization of the profits of another activity, this problem is multi-objective in nature.

There are several problems related with the orienteering problem. For instance, in the Prize-Collecting TSP, see Balas (1988), each vertex has a given prize and penalty, and the goal is to minimize the length of the tour plus the total of the penalties of the vertices not in the tour, while collecting a given quota of the prizes. Feillet et al. (2005) classified these problem types as TSP with profits. Archetti et al. (2007) extended the TSP with profits to several tours naming this version the Vehicle Routing Problem (VRP) with profits. Note that all of them are mono-objective approaches to similar problems. Recently however, Schilde et al. (2009) proposed two metaheuristic procedures for solving the bi-objective orienteering problem. The first is based on Ant Colony Optimization (ACO), introduced by Dorigo and Gambardella (1997) and the second is based on Variable Neighborhood Search (VNS) by Mladenović and Hansen (1997). Both algorithms were combined with a Path Relinking procedure.

The bi-objective OP, called BOP, can be stated on a directed graph $G = (V, A)$ with $V = \{0, 1, 2, \ldots, n+1\}$ the set of vertices and $A = \{(i,j) : i,j \in V, i \neq j, i \neq n+1, j \neq 0\}$ the set of arcs. Without loss of generality we suppose that G is a complete graph with associated cost c_{ij} for $(i,j) \in A$. We have two profits f_{i1}, f_{i2}

associated with each vertex $i \in V \setminus \{0, n+1\}$. Both vertices 0 and $n+1$ have no profits and represent the starting and ending vertices, respectively. Sometimes vertices 0 and $n+1$ denote the same physical point. A tour in the original problem is represented by a directed path in G from vertex 0 to vertex $n+1$. Let L be the maximum length allowed to tour τ and consider the set of all feasible tours $\Theta = \{\tau \mid c(\tau) \leq L\}$.

With these elements the BOP can be formulated as:

$$\begin{aligned} \text{(BOP) max } & f_1(\tau) \\ \text{max } & f_2(\tau) \\ \text{subject to } & \tau \in \Theta \end{aligned}$$

A detailed formulation of the general form of BOP when considering multi-objective functions can be found in Schilde et al. (2009). Since both problems, PDP and BOP, have two objective functions, this provides the opportunity of visualizing the pairs of objective values of non-dominated solutions in the objective space and then comparing the quality of the procedures. It is possible to construct an approximation to the *Pareto-efficient frontier* which is formed by the points with the values of the objectives of all the non-dominated solutions.

Our goal is to apply multi-objective optimization GRASP techniques to the PDP and the BOP in order to obtain as many *Pareto-efficient solutions* as possible. In this way the decision maker will have a set of good options to choose from.

6. Computational experiments

This section describes the computational experiments that we performed to test the efficiency of our GRASP with path relinking procedures as well as to compare them with the previous methods identified to be the state-of-the-art for both the path dissimilarity problem (PDP) and the bi-orienteering problem (BOP).

The 30 PDP test instances with approximately 500 vertices in our experimentation are taken from Martí et al. (2009). These instances were generated removing most of the edges in the following 10 original instances from the well known TSP Library: ali535, att532, d493, d657, fl417, gr666, gr431, rat575, u574, and pcb442 (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/). Specifically, the authors only included those edges with a cost (distance value) lower than 10% of the maximum distance value in each instance, resulting in a sparse and connected graph. The farthest points are taken as the origin and destination in the PDP. For each of these 10 TSP instances, we consider the number of paths p =5, 10 and 15; thus obtaining 30 PDP instances.

The 30 BOP test instances with vertices ranging from 21 to 2143 in our experimentation are taken from Schilde et al. (2009). These authors collected them from different sources. Specifically, they are: 2-p21, 2-p32, 2-p33, 2-p64, 2-p66, 2-p97, 2-p292, 2-p559, 2-p562, and 2-p2143. For any of them, we consider three maximum lengths, the smallest, medium and maximum of those considered by these authors, obtaining 30 instances. Additionally, we have generated 10 medium sized instances to calibrate the methods.

All the instances and results of our experiments are available at http://www.optsicom.es. Experiments for the PDP have been performed on an Intel Core 2 Quad CPU and 6 GB RAM, while those for the BOP have have been run on an Intel I5 at 3.2

GHz. The performance measures that we employ to compare the methods are the standard in multi-objective optimization:

- **Number of points**: This refers to the ability of finding efficient points. We assume that the decision maker prefers more rather than fewer efficient points.
- **SSC**: This metric suggested by Zitzler and Thiele (1999) measures the size of the space covered (SSC). In other words, SSC measures the volume of the dominated points. Hence, the larger the SSC value the better.
- **k-distance**: This density estimation technique (Zitzler et al., 2003) is based on the k-th nearest neighbor method of Silverman (1986). The metric is simply the distance to the k-th nearest efficient point. We use $k = 5$ and calculate both the mean and the max of k-distance values. The k-distance value is such that the smaller the better in terms of frontier density.
- **C(A,B)**: This is known as the coverage of two sets measure (Zitzler and Thiele, 1999). C(A,B) represents the proportion of points in the estimated efficient frontier B that are dominated by the efficient points in the estimated frontier A.

In our first experiment we compare the different constructive methods for both the PDP and BOP. We consider the following five methods described in Section 2: pure-random (PR), pure-ordered (PO), random-sequential combined (RSC), ordered-sequential combined (OSC), and weighted combined (WC). Table 1 shows the number of points, k-distance (mean and maximum) and SSC for the 30 PDP instances. Table 2 shows the coverage between all pairs of these five methods. Similarly, Table 3 shows the number of points, k-distance (mean and maximum) and SSC for the 10 medium sized biorienteering instances and Table 4 shows the coverage between all pairs of these five methods on the BOP.

In the PDP a solution is a set with p paths from the origin to the destination. The constructive methods start by creating a set S with a large number of paths (we generate 2000 as recommended in Martí et al. (2009), the first 1000 with the Yen's implementation of the k-shortest path method described in Carotenuto et al. (2007) and the other 1000 with the Iterative Penalty Method described in Johnson et al. (1992)). Then, in each iteration, they remove from S a path with low contribution until it only contains p paths. The contribution is measured in terms of both problem objectives, the distance from the origin to the destination, f_1 and the dissimilarity among paths, f_2. Therefore, for each candidate path $c \in S$ to be removed, its evaluations are g_1 and g_2, which measure respectively its contribution to f_1 and f_2.

In the BOP a solution is a directed tour τ visiting some of the vertices from an origin to a destination in the graph. This tour cannot exceed a maximum length L. Each vertex has two profits and the two objective values of the tour, $f_1(\tau)$ and $f_2(\tau)$, are computed by adding their corresponding profits. In the constructive methods, for each candidate vertex c to be included in the partial tour, $g_1(c)$ and $g_2(c)$ are the profits of c. In the construction process, the selected vertex is inserted in the best position in the partial tour.

Considering the PDP instances, Tables 1 and 2 show that pure methods provide better approximations to the efficient frontier than the sequential combined ones; although the best method seems to be the weighted combined. For example the coverage of the PR with respect to the RSC method, C(PR,RSC), is 0.28 while the

TABLE 1. Constructive methods on PDP instances

	N. Points	k-distance (mean)	k-distance (max.)	SSC
PR	17.27	0.27	0.64	0.63
PO	17.67	0.28	0.59	0.62
RSC	13.17	0.20	0.42	0.65
OSC	13.40	0.21	0.43	0.64
WC	24.67	0.12	0.42	0.77

TABLE 2. Coverage of constructive methods on PDP

	PR	PO	RSC	OSC	WC
PR	0.00	0.30	0.28	0.25	0.09
PO	0.24	0.00	0.28	0.23	0.09
RSC	0.17	0.21	0.00	0.38	0.06
OSC	0.17	0.20	0.31	0.00	0.05
WC	0.39	0.39	0.85	0.87	0.00

TABLE 3. Constructive methods on Biorienteering instances

	N. Points	k-distance (mean)	k-distance (max.)	SSC
PR	6.18	0.08	0.11	0.94
PO	6.36	0.09	0.11	0.95
RSC	5.45	0.13	0.22	0.98
OSC	4.45	0.16	0.21	0.98
WC	4.55	0.12	0.15	0.98

TABLE 4. Coverage of constructive methods on BOP

	PR	PO	RSC	OSC	WC
PR	0.00	0.30	0.33	0.32	0.38
PO	0.23	0.00	0.23	0.20	0.20
RSC	0.30	0.41	0.00	0.16	0.16
OSC	0.35	0.42	0.41	0.00	0.23
WC	0.40	0.31	0.36	0.24	0.00

C(RSC,PR)=0.17. Moreover, PR and PO obtain a larger number of non-dominated points (N. Points) than RSC and OSC. On the other hand, we can see that WC obtains more points than any other method and a better volume value (SSC). WC exhibits a coverage value better than the rest of the methods. In particular, C(WC,PR)=0.39, C(WC,PO)=0.39, C(WC,RSC)=0.85, and C(WC,OSC)=0.87,

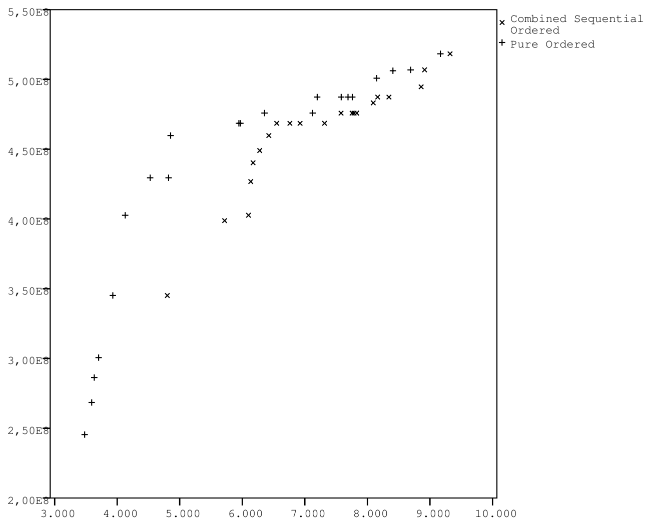

FIGURE 3. Non-dominated points with 2 constructive methods on the PDP

which compare favorably with C(PR,WC)=0.09, C(PO,WC)=0.09, C(RSC,WC)=0.06, and C(OSC,WC)=0.05. We will consider the best method in each category (pure, sequential combined and weighted combined) for the next experiments. Specifically, we select the PO, RSC and WC methods.

Tables 3 and 4 indicate that on the Biorienteering problem the combined methods are able to obtain better approximations to the efficient frontier than the pure methods. This is very interesting since we observed a different behavior on the PDP instances. Specifically, OSC and WC improve upon the rest of the competitors (Table 4 shows, for example, that C(OSC,PO)=0.42 and C(PO,OSC)=0.20). Table 3 shows that the five methods considered obtain a reduced number of non-dominated points, close to 5, which is significantly lower than the number of points obtained in the PDP as shown in Table 1. On the other hand, considering the two pure methods, it seems that PR performs better than PO. To complement this information we consider a representation of the non-dominated points generated with some of the methods. Specifically, Figure 3 shows the non-dominated points obtained with PO and OSC on a medium-sized PDP instance while Figure 4 shows the non-dominated points obtained with PR and OSC on a medium-sized BOP instance. We therefore will consider PR, OSC and WC in our next experiment to be coupled with the respective local search method (pure, sequential-combined and weighted-combined) in order to make a GRASP procedure.

In our second experiment we compare three different GRASP variants for both PDP and BOP. In the first one, Pure-GRASP, the construction is a pure method coupled with the pure local search (as described in Section 3); in the second one, Seq-GRASP, the construction implements a sequential combined method and is coupled with the sequential-combined local search, finally in the third one, Weight-GRASP, the construction and the local search consist in the combined weighted methods respectively. According to the first experiment we consider for the PDP

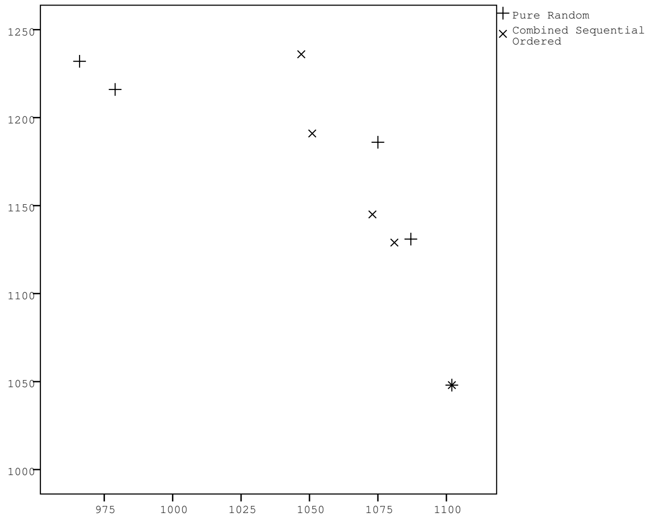

FIGURE 4. Non-dominated points with 2 constructive methods on the BOP

the PO method as the pure construction and the RSC as the sequential combined. Similarly, for the BOP we consider the PR and OSC constructions respectively.

In the PDP the neighborhood of the local search consists of exchanging a path in the solution with another path not included on it taken from the set S defined above with the 2000 initial paths. Therefore, the two associated move values are respectively the change in each objective function. In particular, they are the average of the costs of the paths from the origin to the destination, f_1, and the average of the dissimilarities among paths, f_2. Note that to speed up the process the paths in S are ordered according to their value (f_1, f_2 or a weighted sum). This can be performed off-line as a pre-processing of the method.

In the BOP the neighborhood is based on a exchange between a vertex in the tour and another vertex not in the tour (without exceeding the maximum length L). The difference between the profits of both vertices provides the respective move values. When a one-to-one exchange is performed, we try an insertion move in which a vertex not present in the current tour is considered to be added to the tour. Note that in this problem the insertion of a new point into the tour could not necessarily increment its length (some points are in the same location). It could even reduce the tour length because the matrix distance does not satisfy the triangular inequality in the instances tested, and therefore after we add a vertex to the tour we have to check the addition of more vertices. This is why after an exchange we consider insertions as long as we can add vertices in the tour without exceeding the maximum length L. The added vertices are inserted in the best position.

Table 5 shows the number of points, k-distance (mean and maximum) and SSC for the three GRASP methods on the 30 PDP instances. Table 6 shows the coverage between all pairs of these three methods. Similarly, Tables 7 and 8 show these statistics for the three GRASP methods on the 10 medium sized biorienteering instances.

Table 5. GRASP on PDP instances

	N. Points	k-distance (mean)	k-distance (max.)	SSC
Pure-GRASP	39.70	0.13	0.34	0.78
Seq-GRASP	18.30	0.18	0.42	0.67
Weight-GRASP	44.33	0.07	0.33	0.79

Table 6. Coverage of GRASP methods on PDP instances

	Pure-GRASP	Seq-GRASP	Weight-GRASP
Pure-GRASP	0.00	0.67	0.07
Seq-GRASP	0.06	0.00	0.01
Weight-GRASP	0.27	0.86	0.00

Table 7. GRASP on BOP instances

	N. Points	k-distance (mean)	k-distance (max.)	SSC
Pure-GRASP	9.45	0.58	0.82	0.73
Seq-GRASP	6.27	0.51	0.79	0.74
Weight-GRASP	6.64	0.47	0.64	0.51

Table 8. Coverage of GRASP methods on BOP instances

	Pure-GRASP	Seq-GRASP	Weight-GRASP
Pure-GRASP	0.00	0.13	0.67
Seq-GRASP	0.22	0.00	0.64
Weight-GRASP	0.02	0.01	0.00

Tables 5 and 6 show that the best GRASP variant for the PDP is the Weight-GRASP since it obtains a larger number of non-dominated points and a lower k-distance values than the others. Moreover, the coverage values shown in Table 6 also indicate its superiority. Specifically, C(Weight-GRASP,Pure-GRASP) $= 0.27 > 0.07 =$ C(Pure-GRASP,Weight-GRASP) and C(Weight-GRASP,Seq-GRASP) $= 0.86 > 0.01 =$ C(Seq-GRASP,Weight-GRASP). On the other hand, Table 7 indicates that the differences between the GRASP variants is smaller in the BOP instances. In particular, the number of non-dominated points obtained with these methods ranges from 6.27 to 9.45 and the mean k-distance value from 0.47 to 0.58. According to the coverage values shown in Table 8 the Seq-GRASP seems to perform better than its competitors. Finally, if we compare these values with the results obtained in the previous experiment, we conclude that, as in mono-objective GRASP, the addition of the local search phase significantly improves the results of the constructive method. For example, in Table 1 we can see that the PO method obtains 17.67 non-dominated points on average, while in Table 5 we observe that

TABLE 9. GRASP with PR on PDP instances

	N. Points	k-distance (mean)	k-distance (max.)	SSC	CPU
Pure-PR	72.17	0.07	0.26	0.82	132.30
Seq-PR	29.00	0.12	0.29	0.73	120.43
Weight-PR	34.90	0.12	0.28	0.77	125.60
MSPA	4.90	0.34	0.48	0.49	122.83
GP	34.57	0.19	0.42	0.74	127.80

when this method is coupled with a local search, called Pure-GRASP, it is able to obtain 39.70 non-dominated points on average. Similarly, for the BOP, Table 3 shows that PR obtains 6.18 non-dominated points on average and when this method is coupled with a local search it obtains 9.45 non-dominated points on average (as shown in Table 7).

In our final experiment we undertake to study the hybridization of the path relinking (PR) methodology described in Section 4 with the three GRASP variants tested above in both the PDP and the BOP. Moreover, we compare the resulting hybrid methods with the best algorithms known for both problems.

Given two solutions x and y, we defined (see Section 4) El_{x-y} as the set of elements in x not present in y and El_{y-x} as the set of elements in y not present in x. In the PDP the solutions x and y are sets of p paths connecting the origin o with the destination d and the elements, present or not in these two solutions, are the paths. At each iteration of the PR from x to y, the paths in El_{y-x} are considered to be added to the current intermediate solution (set of p paths), replacing one of the paths in El_{x-y} (to keep the number of paths constant, and equal to p). The method performs the best exchange in terms of the average of the costs of the paths from the origin to the destination, f_1, and the average of the dissimilarities among paths, f_2, depending on the PR variant that we are implementing (pure, sequential or weighted as described in Section 4). Table 9 shows the number of points, k-distance (mean and maximum), SSC and CPU time in seconds for the three PR variants (Pure-PR, Seq-PR and Weight-PR) and the two previous methods identified to be the best (GP (Martí et al., 2009) and MSPA (Dell'Ollmo et al., 2005)) on the 30 PDP instances. Note that each PR variant is applied to the elite set of solutions obtained with the application of the corresponding GRASP method (pure, sequential or weighted respectively). Table 10 shows the coverage between all pairs of these five methods. These five methods have been run for a similar CPU time in the same computer (close to 2 minutes).

Results in Table 9 clearly indicate that the Pure-PR is the best method overall in the PDP. It obtains a larger number of efficient points (72.17 on average) than the other PR variants (29 and 34.9 for the Seq-PR and Weight-PR respectively), exhibits a lower k-distance mean value (0.07) than them (0.12 for both Seq-PR and Weight-PR) and a larger SSC (0.82, 0,73 and 0.77 for the Pure-PR, Seq-PR and Weight-PR respectively). Moreover, Pure-PR improves upon the two previous methods identified to be the best for this problem: MSPA and GP. Specifically, these two methods obtain 4.9 and 34.57 efficient points on average (while Pure-PR obtains 72.17), a k-distance mean value of 0.34 and 0.19 (while Pure-PR obtains

Table 10. Coverage of GRASP with PR methods on PDP instances

	Pure-PR	Seq-PR	Weight-PR	MSPA	GP
Pure-PR	0.00	0.83	0.48	0.97	0.84
Seq-PR	0.04	0.00	0.21	0.76	0.33
Weight-PR	0.12	0.59	0.00	0.85	0.51
MSPA	0.00	0.03	0.04	0.00	0.04
GP	0.01	0.30	0.10	0.84	0.00

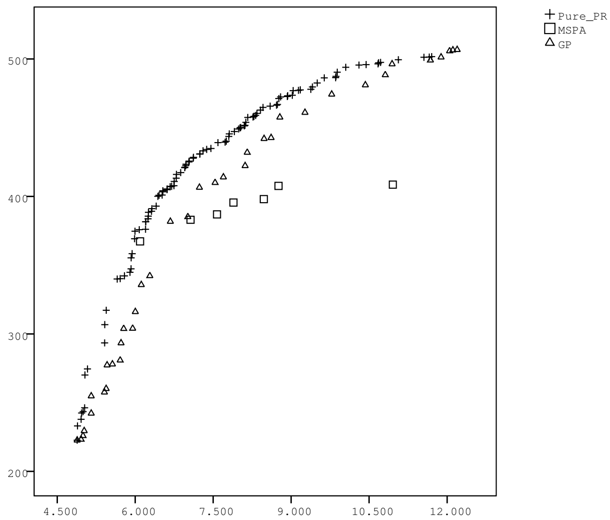

Figure 5. Non-dominated points of best PDP methods on the d657 instance.

0.07) and a SSC of 0.49 and 0.74 respectively (and Pure-PR obtains 0.82). To sum it up, Pure-PR obtains better values in all these statistics than the other four methods under comparison.

The coverage values shown in Table 10 confirm the analysis above. In particular, for any method M considered, C(Pure-PR,M)>C(M,Pure-PR). Moreover, according to these coverage values, the other two PR variants, Seq-PR and Weight-PR, also improve the two previous methods given that C(Seq-PR,MSPA)>C(MSPA,Seq-PR), C(Seq-PR,GP)>C(GP,Seq-PR), C(Weight-PR,MSPA)>C(MSPA,Weight-PR), and C(Weight-PR,GP)>C(GP,Weight-PR). To complement this information, we depict in Figure 5 the approximation of the efficient frontier obtained with Pure-PR and the two previous methods, MSPA and GP on one of the largest instances (d657 with p=10). This figure illustrates the superiority of the Pure-PR w.r.t the previous methods in terms of obtaining a good approximation of the efficient frontier.

In the BOP the solutions x and y are two tours in the graph and the elements in El_{x-y} are the vertices present in tour x and not present in tour y (conversely El_{y-x}). At each iteration of the PR from x to y we add a vertex in El_{y-x} to the

Table 11. GRASP with PR on BOP instances

	N. Points	k-distance (mean)	k-distance (max.)	SSC	CPU
Pure-PR	9.66	0.38	0.57	0.60	177.22
Seq-PR	6.41	0.27	0.44	0.68	109.01
Weight-PR	14.75	0.23	0.44	0.77	175.92
ACO	7.13	0.28	0.45	0.74	367.58
VNS	7.06	0.28	0.42	0.74	367.59

Table 12. Coverage of GRASP with PR methods on BOP instances

	Pure-PR	Seq-PR	Weight-PR	ACO	VNS
Pure-PR	0.00	0.07	0.01	0.07	0.03
Seq-PR	0.22	0.00	0.01	0.09	0.08
Weight-PR	0.35	0.37	0.00	0.32	0.30
ACO	0.30	0.26	0.02	0.00	0.14
VNS	0.31	0.27	0.02	0.13	0.00

current intermediate solution according to its profit values (f_1, f_2 or a weighted sum depending on the PR variant). If the resulting solution is feasible, we consider it as the next intermediate solution in the path; otherwise, we remove from the current intermediate solution the worst vertex in El_{x-y} in terms of the profit values. We keep removing vertices until the current solution becomes feasible. At this point we consider it as the current intermediate solution and resort to the next PR step. Table 11 shows the number of points, k-distance (mean and maximum), SSC and CPU time in seconds for the three PR variants (Pure-PR, Seq-PR and Weight-PR) and the two previous methods identified to be the best (ACO and VNS (Schilde et al., 2009)) on the 30 BOP instances previously reported. Table 12 shows the coverage between all pairs of these five methods.

Results in Table 11 show that the Weight-PR is the best method in the BOP. Specifically, it obtains a larger number of efficient points (14.75 on average) than the other methods (9.66, 6.41, 7.13 and 7.06 for the Pure-PR, Seq-PR, ACO and VNS respectively). Moreover, it is able to achieve a lower k-distance mean value (0.23) than the others (0.38, 0.27, 0.28 and 0.28 for the Pure-PR, Seq-PR, ACO and VNS respectively). Finally, it exhibits the largest SSC value (0.77) among the five methods tested. The running times of the previous methods, ACO and VNS, correspond to an Intel Pentium 4D at 3.2 GHz (Schilde et al., 2009)) while the PR variants have been run on an Intel I5 at 3.2 GHz, which is considered 1.21 times faster than the previous one. We therefore run our PR variants for shorter running times on average as shown in this table. The coverage values shown in Table 12 confirm the superiority of the Weight-PR. In particular, for any method M considered in this table, C(Weight-PR,M)>C(M,Weight-PR). As in the PDP described above, we complement this information with a scatter-plot of an instance (2-p559 with $L = 150$) shown in Figure 6. This figure clearly shows the superiority

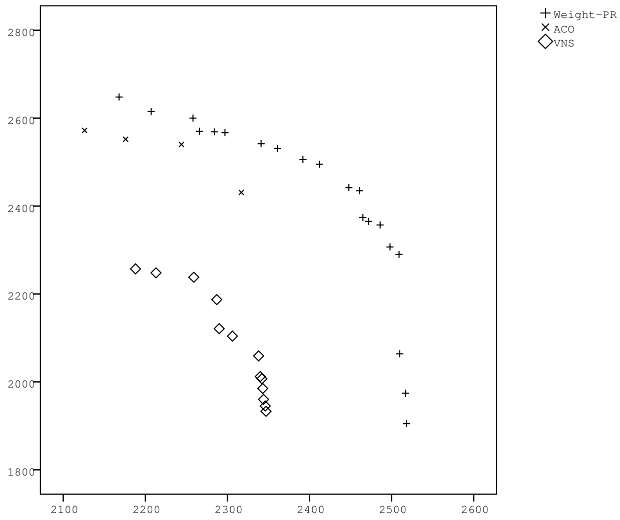

FIGURE 6. Non-dominated points of best BOP methods on the 2-p559 instance.

of the Weight-PR w.r.t VNS and ACO in terms of obtaining a better approximation of the efficient frontier in this instance.

7. Conclusions

The objective of this study has been to advance the current state of knowledge about implementations of GRASP and path relinking procedures in the context of multi-objective optimization. First we have revised previous applications of both methodologies to then establish a classification of the different ways in which they can be applied. Specifically, we have considered three basic ways to implement them: pure, when each objective is optimized in isolation, sequential, when each objective alternates to guide the search, and weighted, when all the objectives are combined in a single master objective.

We have considered two hard bi-objective combinatorial problems to test the different GRASP and PR variants proposed in the paper: the path dissimilarity (PDP) and the bi-orienteering problems (BOP). We compare these variants with the best methods previously reported on 70 instances and the comparison favors some of our GRASP with PR implementations. An interesting conclusion of our study is that in each problem the best results are obtained with a different GRASP with PR variant. Specifically, in the PDP the pure variant achieves the best results, while in the BOP the weighted variant is the winner.

Acknowledgments

This research has been partially supported by the *Ministerio de Ciencia e Innovación* of Spain (Grant Ref. TIN2009-07516). The authors thank Profs. Schilde, Doerner, Hartl and Kiechle for making their results on the BOP available to them.

References

C. Archetti, A. Hertz, and M.G. Speranza. Metaheuristics for the team orienteering problem. *Journal of Heuristics*, 13(1):49–76, 2007.

J. Arroyo, P. Vieira, and D. Vianna. A GRASP algorithm for the multi-criteria minimum spanning tree problem. *Annals of Operations Research*, 159:125–133, 2008.

E. Balas. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636, 1988.

P. Carotenuto, S. Giordani, and S. Ricciardelli. Finding minimum and equitable risk routes for hazmat shipments. *Computers & Operations Research*, 34:1304–1327, 2007.

H. Davoudpour and M. Ashrafi. Solving multi-objective SDST flexible flow shop using GRASP algorithm. *The International Journal of Advanced Manufacturing Technology*, 44:737–747, 2009.

P. Dell'Ollmo, M. Gentilli, and A. Scozzari. On finding dissimilar pareto-optimal paths. *European Journal of Operational Research*, 162:70–82, 2005.

M. Dorigo and L.M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

D. Feillet, P. Dejax, and M. Gendreau. Traveling salesman problems with profits. *Transportation Science*, 39(2):188–205, 2005.

T.A. Feo and M.G.C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71, 1989.

T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

T.A. Feo, M.G.C. Resende, and S.H. Smith. A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, 42:860–878, 1994.

F. Glover and M. Laguna. *Tabu search*. Kluwer Academic Publishers, 1997.

A.J. Higgins, S. Hajkowicz, and E. Bui. A multi-objective model for environmental investment decision making. *Computers & Operations Research*, 35:253–266, 2008.

C. Ishida, A. Pozo, E. Goldbarg, and M. Goldbarg. Multiobjective optimization and rule learning: Subselection algorithm or meta-heuristic algorithm? In N. Nedjah, L.M. Mourelle, and J. Kacprzyk, editors, *Innovative Applications in Data Mining*, pages 47–70. Springer, 2009.

P.E. Johnson, D.S. Joy, and D.B. Clarke. HIGHWAY 3.01, an enhancement routing model: program, description, methodology and revised user's manual. *Technical Report, Oak Ridge National Laboratories*, 1, 1992.

M. Laguna and R. Martí. GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11:44–52, 1999.

H. Li and D. Landa-Silva. An elitist GRASP metaheuristic for the multi-objective quadratic assignment problem. In M. Ehrgott, C.M. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, editors, *Evolutionary Multi-Criterion Optimization*, volume 5467 of *Lecture Notes in Computer Science*, pages 481–494. 2009.

R. Martí, J.L. González-Velarde, and A. Duarte. Heuristics for the bi-objective path dissimilarity problem. *Computers and Operations Research*, 36:2905–2912, 2009.

M. Mladenović and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100, 1997.

M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219–250. Kluwer Academic Publishers, 2003.

M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures. In J.I. Potvin and M. Gendrau, editors, *Handbook of Metaheuristics - 2nd edition*, pages –. Kluwer Academic Publishers, 2010.

A.P. Reynolds and B. de la Iglesia. A multi-objective GRASP for partial classification. *Soft Computing*, 13:227–243, 2009.

A.P. Reynolds, D.W. Corne, and B. de la Iglesia. A multiobjective GRASP for rule selection. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 643–650, Montreal, Quebec, Canada, 2009. ACM.

M. Schilde, K.F.Doerner, R.F.Hartl, and G.Kiechle. Metaheuristics for the bi-objective orienteering problem. *Swarm Intell*, 3:179–201, 2009.

T. Tsiligirides. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9):797–809, 1984.

D.S. Vianna and J.E.C. Arroyo. A GRASP algorithm for the multi-objective knapsack problem. In *Proceedings of the XXIV International Conference of the Chilean Computer Science Society (SCCC04)*. IEEE Computer Society, 2004.

E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

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

(R. Martí) Departamento de Estadística e Investigación Operativa, Universidad de Valencia, Spain
E-mail address: rafael.marti@uv.es

(V. Campos) Departamento de Estadística e Investigación Operativa, Universidad de Valencia, Spain
E-mail address: vicente.campos@uv.es

(M.G.C. Resende) Algorithms and Optimization Research Department, AT&T Labs Research, 180 Park Avenue, Room C241, Florham Park, NJ 07932 USA.
E-mail address: mgcr@research.att.com

(A. Duarte) Departamento de Ciencias de la Computación, Universidad Rey Juan Carlos, Spain.
E-mail address: abraham.duarte@urjc.es