λ-GRASP with bi-directional path relinking for the bi-objective orienteering problem

Hasnaa Rezki and Brahim Aghezzaf*

Lab. Informatique,
Modélisation des Systèmes et Aide à la Décision (LIMSAD),
Faculté des Sciences Ain Chock,
Université Hassan II de Casablanca,
BP 5366 Maarif, 20100 Casablanca, Morocco
Email: b.aghezzaf@fsac.ac.ma
Email: hasnarezki@hotmail.fr
*Corresponding author

Abstract: This paper presents a new approach to solve the bi-objective orienteering problem (BOOP). The BOOP is a multi-objective extension of the well-known orienteering problem (OP). The multi-objective aspect stems from the personalised tourist routes planning problem, in which each point of interest in a city provides different profits associated with different categories. The aim of the BOOP is to find routes satisfying a given travel cost restriction, and visiting some points of interest that maximise the total collected of different profits. To generate a good approximation of Pareto-optimal solutions, we develop a new metaheuristic method based on hybridisation of λ-GRASP and a new variant of the path relinking procedure called bi-directional path relinking (BDPR). The latter is used as an intensification phase, with the goal to obtain new solutions that can eventually be part of the set of the Pareto-optimal solutions. The proposed approach is tested on benchmark instances taken from the literature. It is compared with the Pareto ant colony optimisation algorithm (P-ACO) and the variable neighbourhood search method (VNS). Computational results show that, compared to the P-ACO and the VNS procedures, the proposed method provide a good approximation of the Pareto front for the bi-objective orienteering problem.

Keywords: bi-objective orienteering problem; BOOP; greedy randomised adaptive search procedure; GRASP; Pareto-optimal solutions; bi-directional path relinking; BDPR; hybridisation.

Reference to this paper should be made as follows: Rezki, H. and Aghezzaf, B. (2018) 'λ-GRASP with bi-directional path relinking for the bi-objective orienteering problem', *Int. J. Logistics Systems and Management*, Vol. 29, No. 4, pp.455–475.

Biographical notes: Hasnaa Rezki is a PhD student in Lab. Informatique, Modélisation des Systèmes et Aide à la Décision (LIMSAD). She received her Licenciate degree in Computer Science, her MSc degree in Optimisation of Transportation Systems and Logistics Engineering from Hassan II University of Casablanca. Her research interests include multiple objective programming, applications in transportation and logistics.

Brahim Aghezzaf is a Full Professor in the Department of Mathematics and Computer Science of the Faculty of Sciences of Hassan II, University of Casablanca, Morocco. He is the Director of the Lab. Informatique,

Modélisation des Systèmes et Aide à la Décision (LIMSAD). He received his PhD degree in Operations Research from the University of Brussels, Brussels, Belgium. His research interests include multiple objective programming, operations research and applications in transportation and logistics. He has published among other journals in *Journal of Optimization Theory and Applications*, *Journal of Convex Analysis*, *Journal of Global Optimization*, *Control and Cybernetics*, *Operations Research Letters*, and *Computers & Operations Research*.

1 Introduction

A number of web and mobile applications have recently integrated personalised electronic tourist guides that lead to improved quality of tourist's vacation. Millions of tourists visit large cities with a rich cultural heritage for several days. With a very large variety of tourist attractions the so-called points of interest (POIs), tourists cannot visit every POI during their stay. They are limited in time. So, they are facing the problem of deciding which POI can be more interesting to visit and how to determine a route for each trip-day. Hence, a complex decision situation arises. It is therefore important to have a tourist trip design system that presents suggestions of efficient routes, which can be interesting for each day of the tourist's visit.

The tourist trip design problem (TTDP) (Vansteenwegen and van Oudheusden, 2007; Souffriau et al., 2008; Vansteenwegen et al. 2009; Gavalas et al., 2015) can be viewed as an orienteering problem (OP) (Tsiligirides, 1984) also known as the selective travelling salesman problem (STSP). In the OP, each vertex of a given directed graph G = (V, A) provides a certain profit. The goal of this problem is to determine a path, subject to a given length restriction, that visits some of the vertices, in order to maximise the total collected profit. The applications of OP arise in various real-world situations such as fuel delivery problems and tourist tour planning problems. In the literature, different variants of OP such as the orienteering problem with stochastic profits (OPSP) introduced by Ilhan et al. (2008) can be found. In this stochastic case, normally distributed profits are associated to the vertices. The goal of the OPSP is to design a route visiting a subset of vertices that maximises, within a time limit, the probability of collecting more than a pre-specified target profit level. Another variant of the OP is the orienteering problem with stochastic travel and service times (OPSTS) introduced by Campbell et al. (2011) in which both travel and service times are stochastic. In this problem, a specific reward is received if a vertex is visited, and if it is not, a certain penalty is incurred. This situation illustrates the challenge of a company who, on a given day, may have more customers than it can serve. Recently, Palomo-Martínez et al. (2017) introduced the orienteering problem with mandatory visits and exclusionary constraints in which the objective is to find a route that visits all mandatory vertices and some optional, without conflicts among them, while a given time restriction is respected and the total collected profit is maximised. The OP is also extended to a version with multiple routes known as the team orienteering problem (TOP) introduced by Chao et al. (1996), in which a set of routes is designed with the aim of maximising the total collected profit without exceeding a given length limit. The time-dependent orienteering problem (TD-OP), in which the travel time between two vertices depends on the departure time at the first vertex is also investigated.

This problem formulation allows to model several real life situations like issues related to the traffic congestion such as morning and evening peaks on the highways or crowded cities (Fomin and Lingas, 2002; Garcia et al., 2013; Verbeeck et al., 2014, 2016). Recently, Mei et al. (2016) have introduced a new multi objective variant of the TD-OP called multi-objective TD-OP in which time-dependent travel time and multiple preferences are considered together. The interested reader in different variants and extensions of OP can be referred to the survey of Vansteenwegen et al. (2011), Gavalas et al. (2014) and Gunawan et al. (2016).

In this paper, we are interested in the bi-objective variant of OP called bi-objective orienteering problem (BOOP). In this variant, two categories for each POI exit (e.g., category culture, leisure) and each POI provides different benefits for each category. The BOOP is a NP-hard optimisation problem that was introduced by Schilde et al. (2009). In the BOOP, a directed graph G = (V, A) is given where V = $\{v_0, v_1, v_2, \ldots v_{n+1}\}$ is the vertices set and A is the arcs set. The vertex v_0 and v_{n+1} are the mandatory starting and ending points. Two different benefits are associated to each vertex $v_i \in V \backslash \{v_0, v_{n+1}\}$ and a cost C_{ij} is associated with each arc $(v_i, v_j) \in A$, representing time or distance needed to travel from vertex v_i to vertex v_j. The main goal is to find feasible routes, which visit some vertices and maximise the sum of both benefits without violating a given travel cost restriction. Generally, in multi-objective problems, no single optimal solution exists that simultaneously optimises all the objective functions. So, the concept of optimality in the multi-objective optimisation is replaced with the Pareto optimality (non-dominated solutions). The non-dominated solutions are the solutions that cannot be improved in one objective function without deteriorating their performance in at least one of the others.

In the literature, several approaches were introduced to determine an approximation of the Pareto optimal solutions for the BOOP. The first approaches were proposed by Schilde et al. (2009). They have developed two metaheuristic procedures. The first is an adaptation of the Pareto-ant colony optimisation (P-ACO) metaheuristic, and the second is based on the variable neighbourhood search (VNS) metaheuristic. Both methods were hybridised with the path relinking procedure. Additionally, Tricoire (2012) have introduced a multi-directional local search (MDLS) and applied it to the BOOP. Martí et al. (2015) have proposed different adaptations of the greedy randomised adaptive search procedure (GRASP) combined with the path relinking procedure that have been used as a post-processing strategy.

This paper presents a new hybrid GRASP for generating an efficient approximation of the Pareto optimal solutions for the BOOP. The proposed approach is based on a new adaptation of the λ-GRASP (2010), combined with a new variant of the path relinking procedure called bi-directional path relinking (BDPR) which is used as an intensification phase in order to improve the effectiveness of the λ-GRASP. This is established by generating new solutions that can eventually be Pareto optimal. Three essential phases characterise each iteration of our method: construction, improvement, and intensification.

Experimental results on the test instances show that the proposed method is very competitive with the P-ACO and VNS algorithms proposed in the literature in terms of the solutions quality.

The remainder of this paper is organised as follows: Section 2 presents the mathematical formulation of the BOOP. The proposed approach used to solve the BOOP is described in the third section. Experimental results are reported and discussed in Section 4. Finally, Section 5 presents some concluding remarks.

2 Mathematical formulation

The BOOP can be defined on a directed graph G = (V, A) with a set of vertices V = {v_0, v_1, v_2,…v_{n+1}} where v_0 and v_{n+1} are the mandatory starting and ending points, and a set of arcs A = {(v_i, v_j): v_i, v_j ∈ V, $v_i \neq v_j$, $v_i \neq v_{n+1}$, $v_j \neq v_0$}. For every arc (v_i, v_j) ∈ A is associated a cost value c_{ij} which can represent the time or the distance required to travel from vertex v_i to vertex v_j. Two profits are associated to each vertex v_i ∈ V\{v_0, v_{n+1}} p_{i1} and p_{i2}. The aim is to determine routes visiting some of the vertices, subject to a given cost restriction *T*max, that maximise the total collected corresponding to each of the two profits.

The decision variables are defined as follows:

$$x_{ij} = \begin{cases} 1 & \text{if vertex } v_j \text{ is visited immediatelyafter vertex } v_i \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if the vertex } v_i \text{ is visited by thecurrent route} \\ 0 & \text{otherwise} \end{cases}$$

The problem is then formulated as follows:

$$Max \sum\nolimits_{v_i \in \mathrm{V} \setminus \{v_0, v_{n+1}\}} p_{ik} \cdot y_i \quad (k = 1, 2) \tag{1}$$

$$\sum\nolimits_{v_j \in \mathrm{V} \setminus \{v_i\}} x_{ij} = y_i \quad \left(v_i \in \mathrm{V} \setminus \{v_{n+1}\}\right) \tag{2}$$

$$\sum\nolimits_{v_i \in \mathrm{V} \setminus \{v_j\}} x_{ij} = y_j \quad \left(v_j \in \mathrm{V} \setminus \{v_0\}\right) \tag{3}$$

$$\sum\nolimits_{\{v_i, v_j\} \in S} x_{ij} \leq |S| - 1 \quad (S \subseteq \mathrm{V} \wedge S \neq \varnothing) \tag{4}$$

$$y_0 = y_{n+1} = 1 \tag{5}$$

$$\sum\nolimits_{\{v_i, v_j\} \in \mathrm{A}} C_{ij} x_{ij} \leq T_{\max} \tag{6}$$

$$x_{ij} \in \{0, 1\} \quad \left((v_i, v_j) \in \mathrm{A}\right) \tag{7}$$

$$y_i \in \{0, 1\} \quad (v_i \in \mathrm{V}) \tag{8}$$

In (1), two objective functions are to be maximised: $\sum\nolimits_{v_i \in \mathrm{V} \setminus \{v_0, v_{n+1}\}} p_{ik} \cdot y_i$ (k = 1, 2). Constraints (2) and (3) ensure the connectivity of the route and guarantee that every vertex is visited at most once; constraints (4) eliminate sub-tours; constraints (5) make the route start and end at the correct points; constraint (6) ensure that the cost limit *T*max is respected in the route. Finally, constraints (7) and (8) define that the decision variables are binary.

3 Approach description

Before presenting the proposed approach, it is important to give some definitions that characterise the solutions of a bi-objective optimisation problem. In the multi-objective optimisation problems, a solution x_1 is more efficient than another solution x_2, if x_1 is better than x_2 for at least one objective and not worse for all the other ones. However, usually there is no single optimal solution that simultaneously optimises all the objective functions. The concept of optimality is replaced with that of Pareto optimality. The aim in a multi-objective optimisation problem in the Pareto sense is to find all Pareto-optimal solutions among which the decision maker can select the preferred one (Mavrotas, 2009).

Let us consider a bi-objective optimisation problem, where both objective functions are to be maximised:

Definition 1 (dominance): a solution x_1 dominates a solution x_2 ($x_1 \succ x_2$) if and only if $f_1(x_1) \geq f_1(x_2)$ and $f_2(x_1) > f_2(x_2)$ or $f_1(x_1) > f_1(x_2)$ and $f_2(x_1) \geq f_2(x_2)$.

Definition 2 (strict dominance): a solution x_1 strictly dominates a solution x_2 ($x_1 \succ\succ x_2$) if and only if $f_1(x_1) > f_1(x_2)$ and $f_2(x_1) > f_2(x_2)$.

Definition 3 (Pareto-optimal solution): let X a set of all feasible solutions, a solution x_1 is Pareto-optimal or efficient solution if and only if there is no other solution $x_2 \in X$ such that x_2 dominates x_1. Such solutions are also called non-dominated solutions.

Definition 4 (Pareto front): let X a set of all feasible solutions, if Y is the set that contains only all Pareto-optimal solutions of X, then the Pareto front is the set of objective vectors of the solutions in Y.

3.1 Overview of used methods

3.1.1 GRASP

GRASP (Feo and Resende, 1989, 1995) is a multi-start metaheuristic, in which each iteration consists of two phases: construction and local search. The construction phase is guided by a greedy randomised algorithm; it consists in constructing a feasible solution. In the second phase, local search is applied to this solution, in order to obtain a local optimum in its neighbourhood. The best solution over all GRASP iterations is returned as result. The GRASP was successfully adapted and applied to several multi-objective problems. In Delorme et al. (2010) a new approach called λ-GRASP was proposed to solve the bi-objective set packing problem, in which the GRASP is iterated independently with various weighted sum objective functions. Arroyo et al. (2008) introduced the GRASP to solve the multi-criteria minimum spanning tree wherein the construction phase is guided by the Kruskal's algorithm using a weighted combination of the objectives, and a drop-and-add neighbourhood is used in the local search. Arroyo et al. (2010) proposed a new bi-objective GRASP to solve the bi-objective p-median problem. An elitist GRASP metaheuristic algorithm called mGRASP was developed by Li and Landa-Silva (2009) to solve the multi-objective quadratic assignment problem (mQAP).

This version of GRASP is characterised by three features: elite greedy randomised construction adaptation of search directions and cooperation between solutions. The GRASP was also used as a multi-objective approach by Salazar-Aguilar et al. (2013) for solving the bi-objective commercial territory design problem. The pseudo code of general GRASP for maximisation problem is given in Algorithm 1.

Algorithm 1: pseudo code of general GRASP

```
GRASP(MaxIter)
$x^* \leftarrow Ø$
$f(x^*) \leftarrow 0$
For $i = 1$ to MaxIter
    $x \leftarrow$ ConstructionProcedure(~)
    x ← LocalSearch(x)
    If $f(x) > f(x^*)$
        $x^* \leftarrow x$
    End if
End For
Return $x$
End
```

3.1.2 Path-relinking

The path-relinking (PR) was originally introduced by Glover and Laguna (1997) as a technique to integrate intensification and diversification strategies in the context of tabu search, where the objective is to generate new solutions by exploring the search space or path between two elite solutions (high-quality solutions). Starting from one of these solutions, called an initiating or origin solution, a path is generated in the neighbourhood 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. The PR was successfully integrated as an intensification phase of the GRASP.

This combination was first proposed by Laguna and Marti (1999). The relinking in the intensification context consists in generating a path between a solution found with GRASP and a chosen elite solution. Different variants of PR were introduced in the literature, such as the greedy randomised path relinking (GRPR) introduced by Faria et al. (2005) where the moves between the initiating and the guiding solutions are done in a greedy randomised way, and the evolutionary path relinking (EvPR) proposed by Resende and Werneck (2004) as a post-processing phase for GRASP with PR. Marti et al. (2015) proposed different adaptations of the multi-objective path relinking used as a post-processing phase in the multi-objective GRASP; they have introduced the pure path relinking in which one objective function is considered to select the intermediate solutions in the entire path between two solutions; the sequential path relinking in which the objective function, used to generate the intermediate solutions, is alternated in the same path, and the weighted path relinking in which the weighted objective function is used to select all the intermediate solutions.

This paper proposes a new adaptation of the path relinking in a bi-objective context, called BDPR, used as an intensification phase in the λ-GRASP. An outline of the proposed path relinking is given in Algorithm 5.

3.2 Proposed approach

In order to solve the BOOP, we propose an hybrid procedure called GRASP+BDPR. This approach is based on a new adaptation of the λ-GRASP (2010), a version derived from the GRASP, combined with the new variant of the path relinking procedure called BDPR, employed as an intensification phase. The main goal in the use of BDPR is to improve the performance of the λ-GRASP procedure by providing new solutions that can eventually be part of the set of non-dominated solutions. Each iteration of GRASP + BDPR consists of three phases: construction, local search, and intensification. The pseudo-code of the proposed approach is presented in Algorithm 4.

In the λ-GRASP, the standard GRASP procedure is performed independently on a certain number of search directions with different weighted sum objective functions. So, for each weight *λ*, a construction procedure and a local search phase are applied and repeated until a certain stopping condition is met. the latter can be represented by the maximum number of iterations. The general process of λ-GRASP is given in Algorithm 2.

Algorithm 2: pseudo code of λ-GRASP

```
λ-GRASP($N$, MaxIter)
Input: $N$          //number of search direction
    MaxIter         //number of iterations of GRASP
    ES ← Ø
    For $\lambda \in \left\{0, \frac{1}{N-1}, \frac{2}{N-1}, \dots 1\right\}$
        $x$ ← GRASP ($\lambda$, MaxIter)
        if ($\nexists s \in$ ES: $s \succ x$) then:
            ES ← ES $\cup\, x$
            ES ← ES\{$s$: $\exists\, s' \in$ ES: $s' \succ s$}
        End if
    End For
    Return ES
End
```

In the proposed approach, each local search phase of the λ-GRASP is guided by the two objectives to be maximised.

The GRASP+BDPR is based on two input parameters: *N* which represents the number of search directions, and MaxIter, the maximum number of iterations which represents the stopping criterion of each GRASP application. The proposed solution procedure returns a set of non-dominated solutions called ES. The different steps of GRASP + BDPR are described in the next sub-sections.

3.2.1 Constructive phase

In each iteration of GRASP, the construction phase is guided by a greedy randomised procedure with a certain parameter ~ that allows controlling the amounts of greediness and randomness in the algorithm. This step consists in building a solution while trying to maximise the value of the weighted sum objective function $\lambda f_1 + (1 - \lambda) f_2$ with $\lambda \in [0, 1]$. The constructed solution is checked for its possible inclusion in the set of non-dominated solutions ES. The pseudo code of the GRASP constructive procedure is shown in Algorithm 3.

Algorithm 3: pseudo code of construction procedure

```
Greedy randomised procedure ($\lambda$, ~)
$X \leftarrow \emptyset$
Initialise the candidate list: $CL$
Evaluate the incremental increase of the evaluation function $P(v_i)$ for all $v_i \in CL$
While $CL \neq \emptyset$
$P_{\min} \leftarrow \min\{P(v_i)/v_i \in CL\}$
$P_{\max} \leftarrow \max\{P(v_i)/v_i \in CL\}$
$RCL \leftarrow \{v_i \in CL/P(v_i) \geq P_{\min} + \sim(P_{\max} - P_{\min})\}$
Select at random an element e from RCL
$X \leftarrow X \cup \{e\}$
Update the candidate list $CL$
Reevaluate the incremental increase $P(v_i)$ for all $v_i \in CL$
End while
Return $X$
End
```

In the BOOP, a solution *x* is a route including the start and the end vertex, in addition to some of the vertices where the total cost satisfies a given cost restriction *T*max. The greedy randomised procedure starts with an initial partial solution x formed with the start and the end vertex $\{v_0, v_{n+1}\}$ and a total cost of route C_{0n+1}. At each insertion, a candidate list *CL* is initialised with all vertices not present in the partial solution that can be added without exceeding the cost limit *T*max. In the first step:

$$CL = \left\{v_i \in \mathrm{V} \big/ \left\{v_0, v_{n+1}\right\} :\ C_{0i} + C_{in+1} \leq T\,\mathrm{max}\right\}$$

All elements of *CL* are evaluated with a greedy evaluation function *P* that takes into account the incremental increase in the cost function produced by selecting a certain vertex v_i. Specifically, for each candidate v_i in *CL*, we define $P(v_i)$ as follows:

$$P(v_i) = \lambda\left(\sum\nolimits_{v_j \in x} p_{j1} + p_{i1}\right) + (1 - \lambda)\left(\sum\nolimits_{v_j \in x} p_{j2} + p_{i2}\right)$$

Algorithm 4: pseudo-code of the proposed GRASP + BDPR

```
GRASP + BDPR ($N$, MaxIter)
ES ← Ø
F ← Ø
$F_p$ ← Ø
For $\lambda \in \left\{0, \frac{1}{N-1}, \frac{2}{N-1}, \ldots 1\right\}$
        For iter = 1 to MaxIter
            $\sim = \frac{\text{iter} - 1}{\text{MaxIter} - 1}$
            $L_1 = L_2$ = Ø
            Constructive phase
                $x$ ← greedy randomised procedure ($\lambda$, ~) ~ ∈ [0, 1]
                if ($\nexists$ s ∈ ES: s ≻ $x$) then:
                    ES ← ES ∪ {$x$}
                    ES ← ES\{$s$: ∃ $s'$ ∈ ES: $s'$ ≻ s}
                End if
            Local search phase
                $L_1$ ← $LS_1(x)$                  // improve $x$ maximising $f_1$
                $L_2$ ← $LS_2(x)$                  // improve $x$ maximising $f_2$
                ES ← non-dominated solutions of (ES ∪ $L_1$ ∪ $L_2$)
            Intensification phase
                LP = Ø
                Select at random $x_0$ from ES
                if (∃ $x_1$ ∈ L1 ∪ $L_2$: ($x_1$, $x_0$) ∉ $F_p$) then:
                    LP ← BDPR($x_1$, $x_0$)
                    $F_p$ ← $F_p$ ∪ {( $x_1$, $x_0$)}
                    if($x_0$ ∉ F) then:
                        F ← F ∪ {$x_0$}
                    End if
                else
                    Select at random $x_0$ from ES\F, $x_1$ from $L_1$, and $x_2$ from $L_2$
                    $LP_{1,0}$ ← BDPR($x_1$, $x_0$) and $LP_{2,0}$ ← BDPR($x_2$, $x_0$)
                    LP ← $LP_{1,0}$ ∪ $LP_{2,0}$
                    F ← F ∪ {$x_0$}
                    $F_p$ ← $F_p$ ∪ {($x_1$, $x_0$), ($x_2$, $x_0$)}
                End if
                ES ← non-dominated solutions of (ES ∪ LP)
        End For
End For
Return ES
End
```

To select the element to be inserted in the partial solution, a restricted candidate list (*RCL*) is formed with the best elements whose quality is superior to a certain threshold value in which the parameter ~ allows to control the amounts of greediness and randomness in the choice of the element to be inserted in the solution under construction. Note that the case ~ = 0 corresponds to a completely random choice, while ~ = 1 corresponds to a pure greedy selection:

$$RCL = \left\{ v_i \in CL : P(v_i) \geq P\min + \sim (P\max - P\min) \right\}$$

where $P\max = \max_{v_i \in CL} \left(P(v_i) \right); P\min = \min_{v_i \in CL} \left(P(v_i) \right);$ and $\sim \in [0,1]$.

One vertex is selected at random from *RCL* and it is inserted in the best position into the current solution. The candidate list *CL* is updated after each insertion. All of this process is performed as long as *CL* is not empty.

3.2.2 Local search

In this stage, a local search is performed in order to improve the initial solution generated by the constructive phase. The local search phase of our approach explores two search directions by performing a local search with respect to each objective (i.e., the constructed solution is improved by maximising the objective functions f_1 and f_2, respectively).

For each objective, two different neighbourhoods are used such as it was employed by Campos et al. (2014) to solve the single objective OP. Two types of moves are considered: 1-1 exchange and insertion.

Let *Y* the solution generated in the constructive phase: in the 1-1 exchange, we examine each vertex *u* in the solution *Y*, trying to exchange it with another vertex *v* not included in *Y* that leads to the best improvement of the objective $k(k = 1,2)$ considered in the local search, without exceeding the limit cost *T*max. Specifically, for each vertex *u* in *Y*, we construct the different combinations between the vertex *u* and each vertex *v* not in *Y*, then we compute the difference Δ between the profits associated to both vertices such that $\Delta = p_{vk} - p_{uk}$, and we perform the exchange that provides a maximum improvement. If for a vertex *u* in *Y*, no improving exchange exists, we try to exchange it with a vertex that provides the same profit value p_{uk} but that allows to reduce the route total cost $\left(\sum_{(v_i, v_j) \in Y} C_{ij} x_{ij} \right)$. If the exchange is not performed, we resort to the next vertex *u* in *Y*.

After 1-1 exchange, we try to insert as possible the vertices not included in the solution *Y* without exceeding *T*max. Note that the vertices are always inserted in the best position of the solution.

The examination of vertices is done in their order in the solution. When a certain move is performed for any vertex in the solution, the visited solution is inserted in a set L_k. The exploration of vertices restart from the beginning until no further improvement is possible.

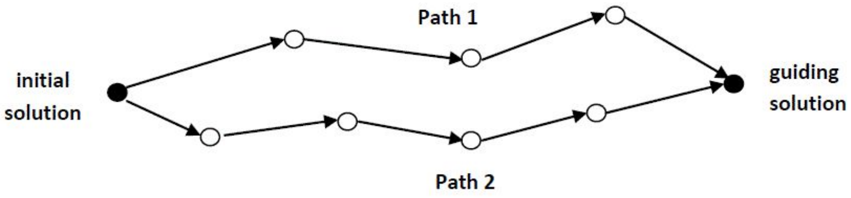
In the proposed algorithm, the local search phase returns two sets: L_1 associated to the local search for the objective 1 denoted LS_1 in algorithm 4, and L_2 associated to the local search for the objective 2 denoted LS_2. Each solution in L_1 and L_2 is checked for its possible inclusion in the set ES of non-dominated solutions.

3.2.3 Intensification phase with BDPR

In the intensification phase of the proposed approach, we introduce a new variant of the path relinking procedure in a bi-objective context that we call BDPR in order to generate new solutions that can eventually be Pareto-optimal.

The key idea of BDPR is so to use two path relinking procedures for each pair of solutions (the initiating solution and the guiding solution), each of them working on a certain search direction. In other words, for a certain optimisation problem with two objectives, a path relinking procedure PR$_k$ is performed for each objective $k(k = 1,2)$ that allows to produce a sequence of successive intermediate solutions in each direction. In each path relinking application PR$_k$$(k = 1,2)$, the intermediate solutions in the entire path are selected with respect to the objective function f_k. Figure 1 shows an illustration of two paths generated by a BDPR procedure where the path 1 and the path 2 are composed of all intermediate solutions according, respectively, to the objective function f_1 and the objective function f_2.

Figure 1 Paths generated by exploring two search directions



The BDPR method has two parameters: the initiating solution, and the guiding solution. The general algorithm of the proposed path relinking is presented in Algorithm 5.

Let x_0 and x_g two solutions of BOOP (two routes); x_0 denotes the initiating solution, and x_g the guiding solution. Let $x_{0\text{–g}}$ be the set of vertices present in x_0 and not in x_g, and $x_{g–0}$ denotes the set of vertices present in x_g and not in x_0. The path relinking procedure PR$_k$$(x_0, x_g)$ starts with the solution x_0 and gradually transforms it into the other solution x_g.

Algorithm 5: the BDPR algorithm

```
BDPR($x_0$, $x_g$)
        $LP_{0,g} = LP_1 = LP_2 = Ø$
        $LP_1 \leftarrow PR_1(x_0, x_g)$
        $LP_2 \leftarrow PR_2(x_0, x_g)$
        $LP_{0,g} \leftarrow LP_1 \cup LP_2$
Return $LP_{0,g}$
```

This is accomplished at first by inserting an element from $x_{g–0}$ in x_0. The choice of the element to be inserted is done in a greedy way. Specifically, the best element according to the objective function f_k is selected. If the resulting solution is feasible, it is considered as the next intermediate solution in the path. Otherwise, the worst vertices according to the objective function f_k in $x_{0–g}$ are removed until the current solution becomes feasible. This whole process is repeated for as long as Tmax is not exceeded, and the set $x_{g–0}$ is not empty.

In the intensification stage of GRASP + BDPR, two sets are incorporated: F_p and F. The first contains all pairs of solutions (x_0, x_g) explored in past BDPR applications, while the second set includes all solutions that were used as guiding solution by the BDPR. In each intensification phase, a first solution x_0 is selected at random from the set of non-dominated solutions ES. Another solution x_1 is selected from the set $L_1 \cup L_2$ such as the pair (x_1, x_0) is different from every pair in F_p. The BDPR is then applied on (x_1, x_0). This pair is stored in the set F_p. In the case where there is no solution in $L_1 \cup L_2$ that satisfies the selection condition, a new solution x_0 is randomly selected from ES\F if it is not empty (ES\F represents the set of solutions in the set of non-dominated solutions ES that have never been used as guiding solutions), and two different solutions x_1 and x_2 are selected at random from L_1 and L_2, respectively. The BDPR is applied on each pair of solutions (x_1, x_0), and (x_2, x_0). The solution x_0 and both pairs are stored in F and F_p, respectively. All solutions in the different sets resulting by different BDPR applications are checked for their possible inclusion in the set ES of the overall non-dominated solutions to be returned by GRASP + BDPR.

4 Computational experiments

The GRASP + BDPR was coded in Java and was run on a computer Intel(R) with 2.1 GHz processor and 2 GB of RAM memory. The performance of the proposed algorithm was tested on 100 test instances taken from Schilde et al. (2009) with number of vertices ranging from 21 to 559. These instances are 2_p21, 2_p32, 2_p33, 2_p64 (dia), 2_p66 (squ), 2_p97 (pad), 2_p273 (wie), and 2_p559 (stm). Each of them is tested with different values of route length restriction (*T*max). The characteristics regarding the number of test instances, the number of vertices and the different route length restriction values *T*max of these problems are given in Table 1. The proposed GRASP + BDPR was executed ten times for each instance in order to obtain the Pareto front. Two parameters were employed:

Table 1 Benchmark BOOP instances

Instance	*Number of test instances*	*Number of vertices*	*Route length restriction Tmax*
2_p21	11	21	15, 20, 23, 25, 27, 30, 32, 35, 38, 40, 45
2_p32	16	32	5, 10, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85
2_p33	13	33	15, 30, 35, 40, 50, 60, 70, 75, 80, 85, 90, 100, 110
2_p64	13	64	15, 20, 25, 30, 35, 45, 50, 55, 60, 65, 70, 75, 80
2_p66	16	66	5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 80, 90
2_p97	19	97	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20
2_p273	6	273	1, 2, 5, 10, 15, 20
2_p559	6	559	10, 20, 30, 40, 50, 80

N: the number of search directions, and MaxIter: the maximum number of iterations considered in each GRASP application. According to the experimental results as well as performance evaluations, we have kept the following values that have yielded the best results: $N = 50$, so $\lambda \in \left\{0, \frac{1}{49}, \frac{2}{49}, \ldots, 1\right\}$; MaxIter = 20.

In this research work, we compare our GRASP + BDPR with the P-ACO and the VNS (Schilde et al., 2009) in order to evaluate the effectiveness of the proposed method.

In the multi-objective optimisation, it is so difficult to measure the quality among approximations of Pareto front obtained by different methods. Consequently, the performance evaluation on the multi-objective optimisation is realised using different metrics that have been introduced in the literature like the following:

- Distance metric (Dist) proposed by Czyzżak and Jaszkiewicz (1998) and Knowles and Corne (2002), this indicator evaluates the distribution of an estimated efficient frontier A as well as the proximity of A to the reference set R; it represents the average distance from each point y ∈ R to its closest point in A. This indicator is calculated as follows:

  $$Dist(A) = \frac{1}{|R|} \sum_{y \in R} \min_{x \in A} d(x, y)$$

  where $d(x, y)$ represents the Euclidean distance between x and y with:

  $$d(x, y) = \sqrt{\left(f_1^*(y) - f_1^*(x)\right)^2 + \left(f_2^*(y) - f_2^*(x)\right)^2}$$

  and $f_i^*(.)$ represents the normalisation of the objective function i to the range [0, 1], where 1 stands for the best result and 0 for the worst.

- Cardinal measure $|A \cap R|$: it represents. The number of solutions of the efficient frontier A that are included in the reference set R.

- Overall non-dominated vector generation ratio (ONVGR) (2006): it measures the ratio of $|A \cap R|$ and the number of solutions in the reference set R. this metric is defined as follows:

  $$ONVGR = \frac{|A \cap R|}{|R|} \times 100$$

- The coverage metric C(A,B) proposed by Zitzler and Thiele (1999): this metric represents the proportion of points in the estimated efficient frontier B that are dominated by the efficient points in the estimated frontier A. Note that C(A, B) = 1 means that all the solutions in B are dominated by solutions in A, while C(A, B) = 0 represents the situation when none of the solutions in B are dominated by solutions in A. This metric is defined as follows:

  $$C(A, B) = \frac{\left|x \in B / \exists y \in A : y \succ x\right|}{|B|}$$

In our experimentation, the reference set R is defined as the set of all non-dominated points obtained by combining all non-dominated solutions generated by the three methods (GRASP + BDPR, P-ACO, and VNS) and finding out the non-dominated solutions from the combination.

Figures 2 to 4 show a graphical representation of the non-dominated solutions obtained by the three methods for the problem set 2_p273 with a total length limit of *T*max = 20, the problem set 2_p66 with *T*max = 30, and the instance 2_p559 with *T*max = 80. It can be observed that the GRASP + BDPR algorithm provides a better approximation of the efficient frontier in these instances.

Figure 2 Pareto front approximation for instance 2_p273 with *T*max = 20 (see online version for colours)

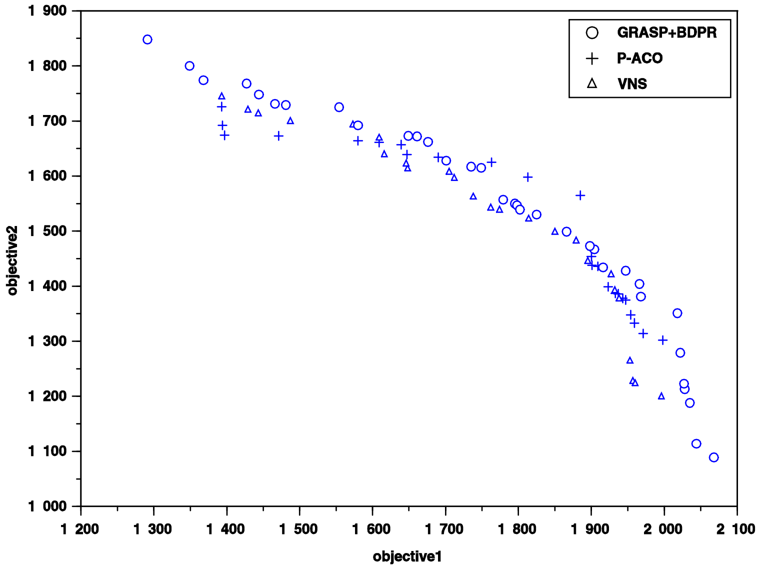


Figure 3 Pareto front approximation for instance 2_p66 with *T*max = 30 (see online version for colours)

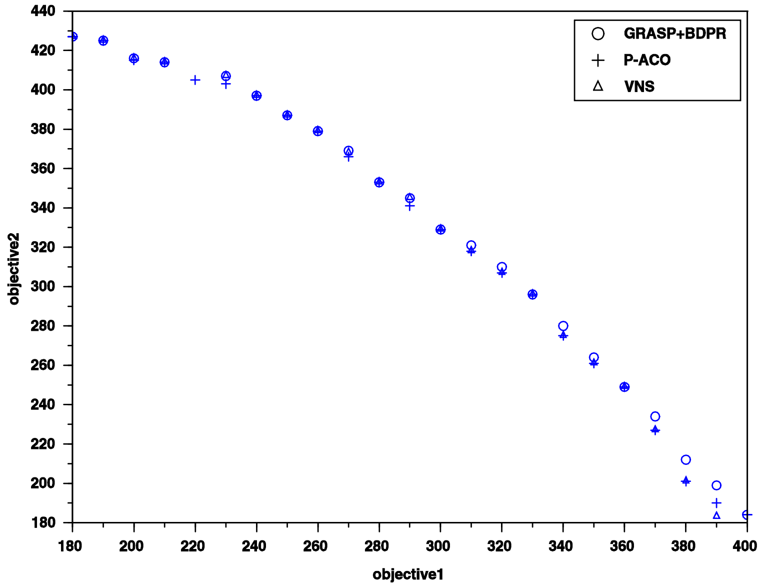


Figure 4 Pareto front approximation for instance 2_p559 with *T*max = 80 (see online version for colours)

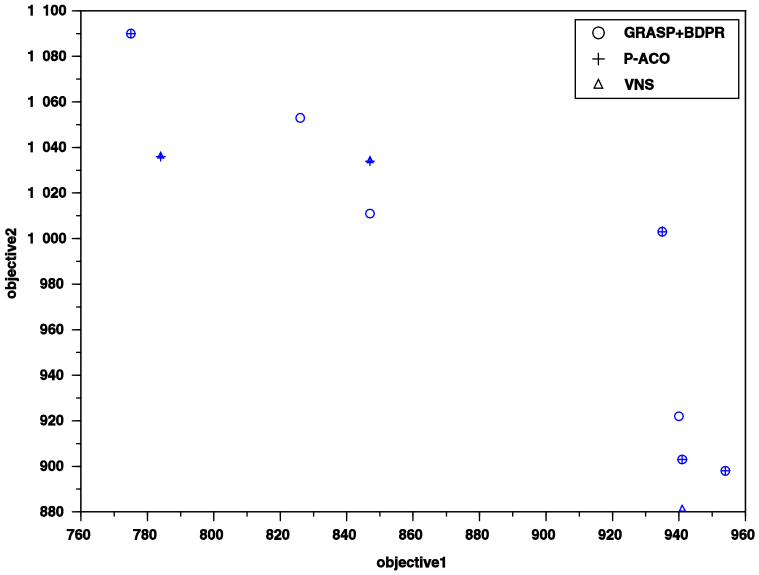


Table 2 Average of distance metric and computational time on different instances

Instances	*GRASP + BDPR*		*P-ACO*		*VNS*	
	Dist	*CPU*	*Dist*	*CPU*	*Dist*	*CPU*
2_p21	0.000	1.60	0.000	1.35	0.000	1.35
2_p32	*0.002*	5.43	0.017	3.25	0.098	3.25
2_p33	0.012	6.37	0.011	5.14	0.069	5.14
2_p64	0.010	20.79	0.018	14.97	0.005	14.97
2_p66	*0.006*	25.05	0.013	16.17	0.007	16.17
2_p97	*0.038*	3.25	0.150	1.17	0.141	1.17
2_p273	*0.009*	50.80	0.682	31.48	0.177	31.48
2_p559	*0.002*	57.12	0.017	36.57	0.032	36.57
Average	*0.009*	21.301	0.113	13.762	0.066	13.762

Table 2 summarises the results of the three methods on the distance measure Dist and the computational time in seconds. Each indicator value represents the average of the indicator values calculated for each problem set with different values of *T*max.

The overall results show that GRASP + BDPR outperforms P-ACO and VNS procedures regarding the distance metric. The GRASP + BDPR gives the best average value for all instances. In particular, for instances 2_p273, the average value of Dist provided by GRASP + BDPR (0.009) is smaller than that of P-ACO (0.682) and VNS (0.177). The superiority of GRASP + BDPR can be also observed on the instances 2_p66, 2_p97, 2_p559. Furthermore, we can conclude that the non-dominated solutions obtained by our algorithm have a closer distance to the reference set than the non-dominated solutions generated by P-ACO and VNS.

Table 3 Cardinal measure and ONVGR on different instances

Instances	|*R*|	*GRASP + BDPR*			*P-ACO*			*VNS*		
		|*A*|	|*A*ą*R*|	*ONVGR (%)*	|*A*|	|*A*ą*R*|	*ONVGR (%)*	|*A*|	|*A*ą*R*|	*ONVGR (%)*
2_p21	21	21	21	100	21	21	100	21	21	100
2_p32	60	60	*59*	*98*	59	56	93	55	54	90
2_p33	76	75	*73*	*96*	76	68	89	73	63	82
2_p64	564	468	248	44	465	85	15	557	349	62
2_p66	401	390	189	47	391	113	28	390	291	72
2_p97	40	38	*38*	*95*	33	33	82	34	34	85
2_p273	113	113	*70*	*62*	77	34	30	89	28	25
2_p559	20	20	*19*	*95*	17	16	80	15	13	65

Table 3 shows the number of Pareto-optimal solutions in the reference set (|R|), the number of non-dominated solutions provided by each method (|A|), the number of reference solutions provided by each algorithm (|A ∩ R|), and the percentage of non-dominated solutions for each algorithm (ONVGR).

GRASP + BDPR is superior to P-ACO regarding the cardinal measure on all problem sets. A larger number of reference solutions are generated by GRASP + BDPR. For example, on the instance 2_p64, 564 reference solutions were generated, from which 248 solutions were found by GRASP + BDPR, and only 85 solutions by P-ACO. The value of the ONVGR indicator obtained by GRASP + BDPR is higher than that of P-ACO for the same instance where 44% of the solutions in the reference set were provided by GRASP + BDPR. This outperformance can be also observed on the instance 2_p97, in which 95% of solutions in the reference set were obtained by our method while 82% were found by P-ACO.

The GRASP + BDPR outperform the VNS algorithm on the problem instances 2_p32, 2_p33, 2_p97, 2_p273, and 2_p559. Specially, on the problem set 2_p273, 113 reference solutions were generated, from which 70 (62%) and 28 (25%) reference solutions were provided, respectively, by GRASP + BDPR and VNS. Additionally, we can observe the superiority of GRASP + BDPR on the problem set 2_p33, 76 reference solutions were generated from which 73 (96%) solutions were obtained by GRASP + BDPR, and 63 (82%) solutions by VNS. Furthermore, on the problem instances 2_p64 and 2_p66, the VNS provides a larger number of reference solutions. For example, on the instance 2_p64, 564 reference solutions were generated, from which 248 (44%) solution were obtained by GRASP + BDPR and 349 (62%) solutions were found by VNS.

Twenty-five instances are randomly chosen for a brief comparison of GRASP + BDPR with P-ACO, and VNS regarding the objective ranges. Table 4 reports the objective ranges of non-dominated solutions generated by the three methods and the reference set on these instances. The objective ranges are expressed in the form of a pair of the minimum and the maximum values obtained for the objective. The objective ranges associated to each method are similar for 14/25 instances while the GRASP + BDPR significantly outperforms the other methods on 10/25 instances. It can be deduced that GRASP + BDPR provides a better spread of solutions from the approximation of efficient frontier in the objective space than P-ACO and VNS.

Table 4 Objective ranges for some instances

Instances	*Tmax*	*min f_1, max f_1*				*min f_2, max f_2*			
		GRASP + BDPR	*P-ACO*	*VNS*	*R*	*GRASP + BDPR*	*P-ACO*	*VNS*	*R*
2_p21	15	120.120	120.120	120.120	120.120	310.310	310.310	310.310	310.310
2_p21	32	290.300	290.300	290.300	290.300	470.500	470.500	470.500	470.500
2_p21	40	370.395	370.395	370.395	370.395	500.570	500.570	500.570	500.570
2_p32	20	45.65	45.65	45.65	45.65	80.95	80.95	80.95	80.95
2_p32	45	155.175	155.175	155.170	155.175	193.238	192.238	218.238	193.238
2_p32	80	270.280	270.280	270.280	270.280	308.319	308.319	308.319	308.319
2_p33	15	150.170	150.170	150.170	150.170	218.236	218.236	218.236	218.236
2_p33	30	230.320	230.320	230.320	230.320	309.447	309.447	309.447	309.447
2_p33	50	410.520	410.520	410.520	410.520	474.628	474.628	474.628	474.628
2_p33	90	750.770	750.770	750.770	750.770	628.742	628.742	620.742	628.742
2_p64	20	*180.294*	192.288	180.288	180.294	*338.455*	346.453	346.455	338.455
2_p64	50	*504.900*	522.894	516.870	504.900	717.1088	709.1084	750.1074	717.1088
2_p64	75	*954.1236*	954.1236	960.1236	954.1236	*1,224.1473*	1,224.1471	1,224.1470	1,224.1473
2_p66	15	75.120	75.120	75.120	75.120	144.255	144.255	144.255	144.255
2_p66	40	*305.575*	305.555	305.545	305.575	*213.519*	239.517	262.517	213.519
2_p66	80	*805.1215*	805.1215	805.1190	805.1215	*381.789*	387.789	373.789	387.789
2_p97	5	86.121	86.121	86.121	86.121	72.145	72.145	72.145	72.145
2_p97	12	*284.325*	284.307	284.307	284.325	*243.335*	294.335	294.335	243.335
2_p97	20	346.466	346.466	346.466	346.466	404.436	404.436	404.436	404.436
2_p273	5	*257.458*	266.452	266.453	257.458	*230.431*	321.428	339.428	230.431
2_p273	15	849.1556	1,030.1520	947.1508	947.1556	*810.1402*	933.1389	869.1402	810.1402
2_p273	20	*1,291.2068*	1,393.1998	1,393.1996	1,291.2068	*1,089.1848*	1,302.1726	1,198.1743	1,089.1848
2_p559	30	68.152	68.152	68.152	68.152	164.183	164.183	164.183	164.183
2_p559	50	*453.503*	453.501	453.501	453.503	*451.534*	459.534	459.534	451.534
2_p559	80	*775.954*	*775.954*	784.941	775.954	*898.1090*	*898.1090*	881.1036	898.1090

Table 5 Coverage metric

Instances	*C(GRASP + BDPR, P-ACO)*	*C(PACO, GRASP + BDPR)*	*C(GRASP + BDPR, VNS)*	*C(VNS, GRASP + BDPR)*
2_p21	0.000	0.000	0.000	0.000
2_p32	*0.045*	0.012	*0.02*	0.012
2_p33	*0.076*	0.014	*0.09*	0.008
2_p64	*0.475*	0.20	0.23	0.41
2_p66	*0.44*	0.16	0.18	0.34
2_p97	0.000	0.000	0.000	0.000
2_p273	*0.43*	0.19	*0.31*	0.14
2_p559	*0.03*	0.02	*0.08*	0.02
Average	*0.187*	0.074	0.114	0.116

The effectiveness of the proposed method can be also viewed in Table 5, where the average of coverage values on GRASP + BDPR outperforms P-ACO. For example, on the instance 2_p66, 44% of the non-dominated solutions obtained by P-ACO are dominated by GRASP + BDPR, while only 16% of those provided by GRASP + BDPR are dominated by P-ACO. The GRASP + BDPR is able to dominate large parts of the non-dominated solutions provided by P-ACO algorithm while a marginal number of non-dominated solutions found by GRASP + BDPR are dominated by P-ACO. The GRASP + BDPR outperforms VNS regarding the coverage metric on the problem sets 2_p32, 2_p33, 2_p273, 2_p559 with the exception of the problem sets 2_p64 and 2_p66. For example, on the instance 2_p273, the GRASP + BDPR dominate 31% of non-dominated solutions generated by VNS while only 14% of those obtained by GRASP + BDPR are dominated by VNS.

In summary, the superiority of GRASP + BDPR in comparison with P-ACO and VNS according to the distance metric, the cardinal measure, the ONVGR and the coverage metric reinforces our conclusion that the GRASP + BDPR is more efficient to generate a good approximation of the Pareto optimal solutions except for the problem instances 2_p64 and 2_p66 where the VNS procedure outperforms the GRASP + BDPR on the cardinal measure and the coverage metric. On average, the proposed algorithm requires a computational time superior to that of P-ACO and VNS (Table 2). This computational effort is very reasonable taking into account the high quality solutions produced by GRASP + BDPR.

5 Conclusions

In this paper, we presented a metaheuristic method called GRASP + BDPR for approximating the Pareto front of the BOOP that arises in the tourist routes design problem. The proposed approach combines the λ-GRASP method with a new variant of the path relinking procedure that we have called BDPR which was used as an intensification phase in the λ-GRASP with the aim to improve the efficiency of the method by generating better solutions that can eventually be Pareto optimal.

To verify the performance of the proposed approach, the GRASP + BDPR was compared with P-ACO and VNS procedures on some benchmark instances taken from the literature by using four measures: distance metric, cardinal measure, ONVGR and the coverage metric. The Computational results show that the proposed approach is competitive with P-ACO and VNS procedures. The GRASP + BDPR outperforms these two methods on all measures except for the problem sets 2_p64 and 2_p66 on which VNS outperforms GRASP + BDPR on the cardinal measure and the coverage metric. New and more efficient solutions on large test instances were achieved within a reasonable CPU effort. This computational effort is an acceptable trade-off for the high quality of solutions obtained by GRASP + BDPR.

The BDPR can be viewed as an efficient approach in a bi-objective context for generating new solutions that can be Pareto optimal. However, we think that the use of the key idea of the BDPR in a multi-objective problems in which the number of objectives exceeds 2, might not be efficient since it could require more computational effort without achieving significant improvements. Additionally, the random character of the constructive phase of the algorithm is very important in terms of solutions quality, but it might produce many identical solutions. It is also important to note that the choice of the parameters used in the GRASP + BDPR is very crucial in terms of the quality of solutions on the one hand and the computational time on the other hand. For example, the use of a high value of the search directions parameter might lead to higher computational effort but could ensure a good coverage of the efficient frontier.

In the future research, we will try to adapt the solution procedure to be reactive to human stimuli by considering more complex real-world constraints such as time windows and budget limitations of tourists for entrance fees corresponding to the POIs in order to illustrate a more realistic TTDP.

References

Arroyo, J., Soares, M. and dos Santos, P. (2010) 'GRASP heuristic with path-relinking for a bi-objective p-median problem', *10th International Conference on Hybrid Intelligent Systems*.

Arroyo, J., Vieira, P. and Vianna, D. (2008) 'A GRASP algorithm for the multi-criteria minimum spanning tree problem', *Annals of Operations Research*, Vol. 159, No. 1, pp.125–133.

Campbell, A., Gendreau, M. and Thomas, B. (2011) 'The orienteering problem with stochastic travel and service times', *Annals of Operations Research*, Vol. 186, No. 1, pp.61–81.

Campos, V., Martí, R., Sánchez-Oro, J. and Duarte, A. (2014) 'GRASP with path relinking for the orienteering problem', *Journal of the Operational Research Society*, Vol. 65, No. 12, pp.1800–1813.

Chao, I-M., Golden, B.L. and Wasil, E.A. (1996) 'The team orienteering problem', *European Journal of Operational Research*, Vol. 88, No. 3, pp.464–474.

Czyzżak, P. and Jaszkiewicz, A. (1998) 'Pareto-simulated annealing – a metaheuristic technique for multi-objective combinatorial optimization', *J Multi Criteria Decis Anal.*, Vol. 7, No. 1, pp.34–47.

Delorme, X., Gandibleux, X. and Degoutin, F. (2010) 'Evolutionary, constructive and hybrid procedures for the bi-objective set packing problem', *European Journal of Operational Research*, Vol. 204, No. 2, pp.206–217.

Faria Jr., H., Binato, S., Resende, M.G.C. and Falcão, D.J. (2005) 'Transmission network design by a greedy randomized adaptive path relinking approach', *IEEE Transactions on Power Systems*, Vol. 20, No. 1, pp.43–49.

Feo, T.A. and Resende, M.G.C. (1989) 'A probabilistic heuristic for a computationally difficult set covering problem', *Operations Research Letters*, Vol. 8, No. 2, pp.67–71.

Feo, T.A. and Resende, M.G.C. (1995) 'Greedy randomized adaptive search procedures', *Journal of Global Optimization*, Vol. 6, No. 1, pp.109–133.

Fomin, F. and Lingas, A. (2002) 'Approximation algorithms for time-dependent orienteering', *Information Processing Letters*, Vol. 83, No. 2, pp.57–62.

Garcia, A., Vansteenwegen, P., Arbelaitz, O., Souffriau, W. and Linaza, M. (2013) 'Integrating public transportation in personalised electronic tourist guides', *Computers and Operations Research*, Vol. 40, No. 3, pp.758–774.

Gavalas, D., Kasapakis, V., Konstantopoulos, C., Pantziou, G., Vathis, N. and Zaroliagis, C. (2015) 'The eCOMPASS multimodal tourist tour planner', *Expert Systems with Applications*, Vol. 42, No. 21, pp.7303–7316.

Gavalas, D., Konstantopoulos, C., Mastakas, K. and Pantziou, G. (2014) 'A survey on algorithmic approaches for solving tourist trip design problems', *Journal of Heuristics*, Vol. 20, No. 3, pp.291–328.

Glover, F. and Laguna, M. (1997) *Tabu Search*, Kluwer Academic Publishers, Dordrecht.

Gunawan, A., Chuin Lau, H. and Vansteenwegen, P. (2016) 'Orienteering problem: a survey of recent variants, solution approaches and applications', *European Journal of Operational Research*, Vol. 255, No. 2, pp.315–332.

Ilhan, T., Iravani, S.M.R. and Daskin, M.S. (2008) 'The orienteering problem with stochastic profits', *IIE Transactions*, Vol. 40, No. 4, pp.406–421.

Knowles, J.D. and Corne, D.W. (2002) 'On metrics for comparing non-dominated sets', *Proceedings of the 2002 Congress on Evolutionary Computation Conference*, Honolulu, Hawaii, pp.711–716.

Laguna, M. and Martí, R. (1999) 'GRASP and path relinking for 2-layer straight line crossing minimization', *INFORMS Journal on Computing*, Vol. 11, No. 1, pp.44–52.

Li, H. and Landa-Silva, D. (2009) 'An elitist GRASP metaheuristic for the multi-objective quadratic assignment problem', in Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J-K. and Sevaux, M. (Eds.): *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, Vol. 5467, pp.481–494, Springer.

Martí, R., Campos, V., Resende, M.G.C. and Duarte, A. (2015) 'Multiobjective GRASP with path relinking', *European Journal of Operational Research*, Vol. 240, No. 1, pp.54–71.

Mavrotas, G. (2009) 'Effective implementation of the e-constraint method in multi-objective mathematical programming problems', *Applied Mathematics and Computation*, Vol. 213, No. 2, pp.455–465.

Mei, Y., Salim, F.D. and Li, X. (2016) 'Efficient meta-heuristics for the multi-objective time-dependent orienteering problem', *European Journal of Operational Research*, Vol. 254, No. 2, pp.443–457.

Palomo-Martínez, P.J., Salazar-Aguilar, M.A., Laporte, G. and Langevin, A. (2017) 'A hybrid variable neighborhood search for the orienteering problem with mandatory visits and exclusionary constraints', *Computers and Operation Research*February, Vol. 78, pp. 408–419, In Press [online] http://dx.doi.org/10.1016/j.cor.2015.11.007.

Resende, M.G.C. and Werneck, RF. (2004) 'A hybrid heuristic for the p-median problem', *Journal of Heuristics*, Vol. 10, No. 1, pp.59–88.

Salazar-Aguilar, M.A., Ríos-Mercado, R.Z. and González-Velarde, J.L. (2013) 'Grasp strategies for a bi-objective commercial territory design problem', *Journal of Heuristics*, Vol. 19, No. 2, pp.179–200.

Schilde, M., Doerner, K.F., Hartl, R.F. and Kiechle, G. (2009) 'Metaheuristics for the biobjective orienteering problem', *Swarm Intell.*, Vol. 3, No. 3, pp.179–201.

Souffriau, W., Vansteenwegen, P., Vertommen, J., Vanden Berghe, G. and VanOudheusden, D. (2008) 'A personalised tourist trip design algorithm for mobile tourist guides', *Applied Artificial Intelligence*, Vol. 22, No. 10, pp.964–985.

Tricoire, F. (2012) 'Multi-directional local search', *Computers & Operations Research*, Vol. 39, No. 12, pp.3089–3101.

Tsiligirides, T. (1984) 'Heuristic methods applied to orienteering', *The Journal of the Operational Research Society*, Vol. 35, No. 9, pp.797–809.

Vansteenwegen, P. and van Oudheusden, D. (2007) 'The mobile tourist guide: an or opportunity', *OR Insights*, Vol. 20, No. 3, pp.21–27.

Vansteenwegen, P., Souffriau, W. and van Oudheusden, D. (2011) 'The orienteering problem: a survey', *European Journal of Operational Research*, Vol. 209, No. 1, pp.1–10.

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G. and van Oudheusden, D. (2009) 'Metaheuristics for tourist trip planning', in Geiger, M., Habenicht, W., Sevaux, M. and Sorensen, K. (Eds.): *Metaheuristics in the Service Industry, Lecture Notes in Economics and Mathematical Systems*, Vol. 624, pp.15–31, Springer-Verlag.

Verbeeck, C., Sörensen, K., Aghezzaf, E-H. and Vansteenwegen, P. (2014) 'A fast solution method for the time-dependent orienteering problem', *European Journal of Operational Research*, Vol. 236, No. 2, pp.419–432.

Verbeeck, C., Vansteenwegen, P. and Aghezzaf, E-H. (2016) 'Solving the stochastic time-dependent orienteering problem with time windows', *European Journal of Operational Research*, Vol. 255, No. 3, pp.699–718.

Zitzler, E. and Thiele, L. (1999) 'Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach', *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp.257–271.