

FORMULATION AND A HEURISTIC APPROACH FOR THE ORIENTEERING LOCATION-ROUTING PROBLEM

ALI NADIZADEH*

Abstract. In this paper, a new version of the location-routing problem (LRP), named orienteering location-routing problem (OLRP) is investigated. The problem is composed of two-well known problems: team orienteering problem (TOP) and LRP. There are some challenging practical applications in logistics, tourism, military operations, and other fields, which can be modeled by OLRP. The problem is to consider the location and routing with a special objective function. In the OLRP, a set of nodes with specific scores is given, and some stations among candidate stations should be established. Moreover, there are some routes limited in length, which start from a station, visit some nodes and then return to the same station. Maximizing the sum of the collected scores is the goal of OLRP. To model the problem, an integer linear programming model is proposed. Against a commercial solver, a heuristic GRASP is developed for solving the standard test problems. Most test problems are found difficult to solve optimally with commercial software while the GRASP can find the best known or close to the best-known solution in a short time.

Mathematics Subject Classification. 90B06, 90C27, 90B80 .

Received February 26, 2018. Accepted May 31, 2020.

1. INTRODUCTION

Location routing problem (LRP) is one of the most important problems in the logistics of supply chain management (SCM) [39]. The LRP is a combination of two well-known problems, facility location problem (FLP) [3, 19] and vehicle routing problem (VRP) [14, 33, 36, 48]. Since both problems are related to the class of NP-hard problems, LRP is also an NP-hard problem [8, 9, 41]. In the purview of earlier studies, Salhi and Rand showed that solving a location problem without regard to the routing problem might lead to an undesirable solution [50]. LRP refers to detect the optimal location of depots and vehicle routes in order to supply the customers under different real-world assumptions [38]. The objective function is to minimize the total cost of the vehicle routing, fixed costs of the depots and vehicles [21, 40, 49]. In last few decades, LRP is intensively studied due to its wide variety of fields such as food and drink distribution, newspaper delivery, waste collection, bill delivery, military applications, used oil management, organization of natural disaster, battery swap stations, parcel delivery and various consumer goods distribution, emergency logistics [35, 35, 47, 49, 54, 60]. Details on comprehensive surveys of variants, solution algorithms and extensions of the standard LRP can be found in [18, 42, 46].

Keywords. Orienteering problem, location routing problem, GRASP.

Department of Industrial Engineering, Faculty of Engineering, Ardakan University, PO Box 184, Ardakan, Iran.

*Corresponding author: nadizadeh@ardakan.ac.ir

The orienteering problem (OP) has also attracted a lot of attention recently. The OP is a routing problem whose goal is to determine a subset of nodes to be visited so that the total collected score is maximized [26, 55]. In this problem, all nodes cannot be visited due to the limited time duration [28]. The OP can be considered as a combination of two well-known problems, the knapsack problem and the traveling salesman problem (TSP) [52]. The OP arises in several applications such as the mobile tourist guide [51], mobile-crowdsourcing problem [32], tourist trip design problem [23, 34], sales representative planning problem [59] and other fields [44].

The team orienteering problem (TOP) is a generalization of OP where the goal is to determine R routes, each limited by time duration, which maximizes the total collected score [5]. It is shown that the OP and TOP are NP-hard problems; no polynomial-time algorithm is designed to solve these problems for optimization. This means that the exact solution algorithms are very time consuming and for practical applications, heuristic or meta-heuristic methods will be necessary. Recently, a comprehensive and complete review of OP types, including suggested solution methods and various benchmarking examples has been done by [26, 52].

In this paper, the orienteering location-routing problem (OLRP), which composed of two problems of LRP and TOP, is considered. In OLRP, some nodes with different scores are given in specific places. Some stations are selected among the candidate stations, which will be opened as the starting point for the routes. According to the maximum allowed time (or travel distance), only a subset of nodes can be visited. In fact, each route can support as many nodes as possible so that the limitation of time duration does not exceed. Moreover, the routes start and end to the same station, and the objective function is to maximize the total score collected from visited nodes.

To the best of our knowledge, this paper is the first work that addresses OLRP. The main motivation of authors to develop a new model is some practical applications of the OLRP in the real world. The OLRP is applicable in many fields such as logistics, tourism, military operations, civilian domains and other fields. In the case of tourism, for instance, to travel a country or city, it is often impossible to visit every place that tourists are interested. Therefore, they must choose what they believe to be the most valuable attractions. Making a feasible and appropriate plan in order to travel these attractions in the available time span is often a difficult task. Also, due to the time travel restrictions, the time required to visit a certain place can play an important role in the selection of places. Moreover, it is important for tourists to select the proper station from candidate stations to start their tour. Station selection can depend on a variety of factors such as ticket price, facilities and transportation, close to the most attractive tourist areas, etc.

Figure 1 illustrates an example of OLRP. In this figure, there are 26 interest nodes (lozenge points), 4 candidate stations (square points) and 5 routes. Scores' values are given next to corresponding interest nodes (values range from 1 to 5, and the value of 5 represents the highest attractive). According to the given solution, two routes are assigned together to open station of 1 and three routes are assigned to another opened station. The value of the objective function is the total sum of scores collected by five routes, which equals 46.

The remainder of this paper is organized as follows. In the next section, the literature review of OLRP is summarized. Section 3 described the Integer linear programming model of the OLRP. Details of the heuristic approach to solve the problem are presented in Section 4. In Section 5, numerical experiments are given to reveal the performance of the proposed method. In the final section, the conclusions and suggestions for future researches are outlined.

2. LITERATURE REVIEW

Some assumptions of the OLRP can be referred to the LRP. Laporte is the first researcher who discusses and classifies the LRP models [30]. Since LRP is an NP-hard problem, most of the papers in the field of LRP are focused on only new solution methods that are often based on heuristic or meta-heuristic approaches [37, 43, 58]. For example, Ferreira and Queiroz presented two heuristics based on the simulated annealing method for the capacitated location-routing problem [22]. In their method, the local search reallocates customers and next improves the routes cost and there is a diversification procedure to explore facility locations. In this research, solutions of nine large instances (with 150 and 200 customers) from standard test problems are improved. Dai

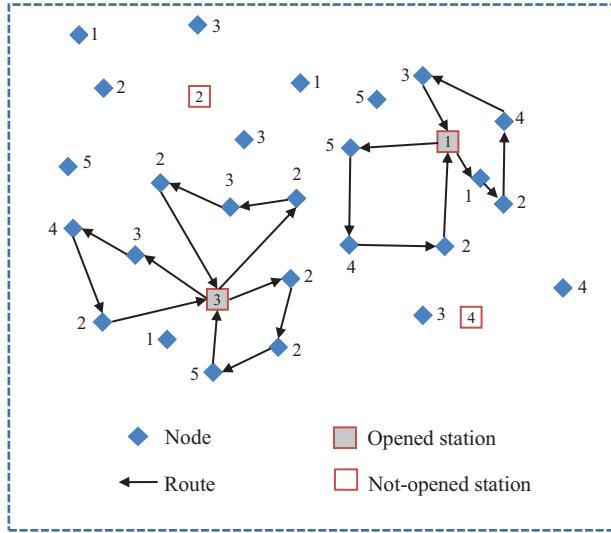


FIGURE 1. Graphical representation of an OLRP example.

et al. are proposed four MIP formulations for the location-routing problem with a tour length constraint of vehicles [16]. A two-phase method based on improved Clarke and Wright savings algorithm is put forward for the problem. The authors showed that the speed of the proposed method is faster than that of other methods and is easy to be used and can be extended. Few exact methods are also developed in the literature of the problem [1, 6, 9, 15]. Recently, Capelle *et al.* proposed an exact method for the location-routing problem with pickup and delivery. They implemented a method based on column generation and Branch and Price and solved the sub-problem using an elementary shortest path procedure. Finally, the authors thoroughly analyzed the impact of fixed costs for depots and routes on solutions [12]. Another new work is related to Boccia *et al.* They proposed a new ILP formulation for the multi-commodity LRP. Their proposed formulation is discussed for the design of a single-tier freight distribution system. Authors developed a branch-and-cut algorithm and experienced test networks with different topologies [11]. Some reviews on solution approaches of LRP exist in the literature that can be found in [17, 20, 57].

The characteristic of the objective function of OLRP is similar to OP. Orienteering problem (OP) is first introduced by Golden *et al.* [25]. OP can be seen as a combination of node selection, determining the shortest Hamiltonian path of the selected nodes and maximizing the total score collected from selected nodes [52]. Recently, a generalization of the orienteering problem, named set orienteering problem (SOP) is developed [4]. In SOP, customers are grouped in clusters and a profit is associated with each cluster. The profit of a cluster is collected only if at least one customer from the cluster is visited. A single vehicle is available to collect the profit and the objective is to find the vehicle route that maximizes the profit collected and such that the route duration does not exceed a given threshold.

Another extending of the OP with multiple routes (paths) is initially introduced by Chao *et al.* [13], which is known to team orienteering problem (TOP). To solve the problem, both exact methods and heuristic approaches with the hybrid of meta-heuristic algorithms are widely carried out in the literature of OP and TOP [10, 26]. Team orienteering problem with time windows and time-dependent scores (TOPTW-TDS) is a variant of the TOP and first introduced by Yu *et al.* [56]. TOPTW-TDS builds a set of paths to maximize the collected scores. The score of visiting an attraction is different depending on the time of visit. The authors proposed a mathematical formulation for this new problem. A hybrid artificial bee colony (HABC) algorithm is developed to find the optimal solutions for small instances and near-optimal solutions for larger instances.

TABLE 1. Comparison of OLRP characteristics with some related problems.

Problem type	Multiple location	Multiple routing	Selection of all nodes (or a group)	Maximize collected scores	Minimize costs	Return route to the starting point	Fixed fleet
VRP	–	+	+	–	+	+	+ or –
LRP	+	+	+	–	+	+	+ or –
GLRPP	+	+	–	+	–	+	–
PCLR P	+	+	–	+	–	+	+
OP	–	–	–	+	–	+	–
SOP	–	–	+	+	–	+	–
TOP	–	+	–	+	–	+	–
TOPTW-TDS	–	+	–	+	–	+	–
OLRP	+	+	–	+	–	+	–

The most relevant attempt in the literature of both LRP and OP is studied by Yakici [53]. This work is expressed in the context of unmanned aerial vehicles (UAVs) as a practical application. The research addressed the problem of locating and routing of small UAVs at the tactical level and defined the problem as a prize collecting location and routing problem (PCLR P). The problem assumes a homogeneous fleet of a fixed number of UAVs, which is required to be distributed to candidate bases in order to maximize the total score of covered interest points. Each UAV is limited by its constrained flight time, which is the sum of time spent in transition between the base and interest points and time elapsed in performing the tasks given at interest points. The problem is formulated as an integer linear program, and tailored ant colony optimization method is developed to provide better solutions in a very short time for the instances that have a relatively large size.

Another relevant work to the OLRP is studied by Ahn *et al.* [2]. Their work is carried out in the context of planetary surface exploration missions. The authors expressed the problem as the generalized location routing problem with profits (GLRPP). The depots and customers correspond to potential bases and exploration sites with profit values. GLRPP maximizes the total profit obtained over a campaign. The GLRPP is just close to both OP and LRP problems, and some of their concepts lie in it. This problem maximizes the total profit obtained over a campaign and problem decisions are about the selection of bases, selection of strategies for the missions assigned to bases, selection of visited sites and their order of being visited and selection of routing tactics. In the mentioned research, a mathematical formulation and two solution approaches for the problem are developed. It is important to note that the problem features described in this work are similar to the assumptions of the GLRPP model and the problem is modeled as OLRP, which can be considered as a new version of GLRPP and based on the LRP model.

To display the characteristic of OLRP and its difference with similar problems in the literature, a comparison is presented in Table 1. It is noted that Table 1 expresses only the attributes of the base model of the mentioned problems. In this table, if an attribute is related to the type of problem, it is indicated by the “+” sign, otherwise it is represented by the “–” sign.

3. MATHEMATICAL FORMULATION

The OLRP can be defined with the aid of a graph $G = (V, A)$, where $V = N_0 \cup N_n$ is a set of vertices in which N_0 and N_n represent the candidate stations and attractive nodes, respectively, and $A = \{(i, j) : i, j \in A\}$ is the set of arcs. The travel time t_{ij} is associated with each arc $a_{ij} \in A$. Each node $j \in N_n$ has a score S_j and all nodes cannot be visited since there is the travel time allowed of T_{\max} for each route. The maximum number of stations that can be opened is limited to Y_{\max} and the start and end station of each route $r \in R$ are identical. The goal of the OLRP is to determine proper station(s) including R routes limited by T_{\max} , that visit some of

the nodes, in order to maximize the total collected score. The scores are completely additive, and each node cannot be viewed more than once.

The following notations are defined to represent the integer linear programming (ILP) formulation for the OLRP.

Sets and parameters:

N_0 : Set of candidate stations and $N_0 = \{1, 2, \dots, M\}$ and M is the number of stations.

N_n : Set of nodes and $N_n = \{M + 1, M + 2, \dots, M + N\}$ and N is the number of nodes.

V : Set of all points; $V = N_0 \cup N_n$ and $V = \{1, 2, \dots, M, M + 1, M + 2, \dots, M + N\}$.

A : Set of arcs (i, j) connecting every pair of nodes $i, j \in V$.

R : Set of routes indexed by r and $R = \{1, 2, \dots, \bar{R}\}$ that \bar{R} is the number of routes.

S_j : Non-negative score of node j .

T_{\max} : Maximum allowed time travel of each route.

t_{ij} : Time traveling for each arc $(i, j) \in A$ (note that the t_{ij} can be equivalent to the distance between two nodes).

t_j : Required time for the visit of node j .

Y_{\max} : Maximum number of stations allowed to be opened.

Decision variables:

$$Y_i = \begin{cases} 1 & \text{if a station at candidate site } i \text{ is opened} \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{ij} = \begin{cases} 1 & \text{if node } j \text{ is assigned to the station at candidate site } i \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ijr} = \begin{cases} 1 & \text{if a visit to node } i \text{ is followed by a visit to node } j \text{ in route } r \\ 0 & \text{otherwise} \end{cases}$$

U_{jr} = Auxiliary variables for sub-tour elimination constraints in route r .

$$\text{Maximize} \sum_{i \in V} \sum_{j \in N_n} \sum_{r \in R} S_j x_{ijr} \quad (3.1)$$

Subject to

$$\sum_{i \in N_0} Y_i \leq Y_{\max} \quad (3.2)$$

$$\sum_{j \in N_n} Z_{ij} \leq Y_i \times N \quad \forall i \in N_0 \quad (3.3)$$

$$\sum_{j \in V} x_{ijr} - \sum_{j \in V} x_{jir} = 0 \quad \forall i \in V; \quad \forall r \in R \quad (3.4)$$

$$\sum_{i \in V} \sum_{k \in K} x_{ijk} \leq 1 \quad \forall j \in N_n \quad (3.5)$$

$$\sum_{i \in V} \sum_{j \in V} (t_{ij} + t_j) x_{ijr} \leq T_{\max} \quad \forall r \in R \quad (3.6)$$

$$U_{lr} - U_{jr} + N x_{ljr} \leq N - 1 \quad \forall l, j \in J; \quad \forall r \in R \quad (3.7)$$

$$\sum_{i \in N_0} \sum_{j \in N_n} x_{ijr} \leq 1 \quad \forall r \in R \quad (3.8)$$

$$\sum_{u \in V} (x_{iur} + x_{u jr}) \leq 1 + Z_{ij} \quad \forall i \in N_0; \quad \forall j \in N_n; \quad \forall r \in R \quad (3.9)$$

$$Y_i \in \{0, 1\} \quad \forall i \in N_0 \quad (3.10)$$

$$Z_{ij} \in \{0, 1\} \quad \forall i \in N_0; \quad \forall j \in N_n \quad (3.11)$$

$$x_{ijr} \in \{0, 1\} \quad \forall i \in V; \quad \forall j \in V; \quad \forall r \in R \quad (3.12)$$

$$U_{jr} \in \{N \cup 0\} \quad \forall j \in N_n; \quad \forall r \in R. \quad (3.13)$$

The objective function (3.1), to be maximized, gives the total scores collected at the visited nodes. The maximum number of stations that can be opened is expressed by constraints (3.2), and the maximum number of nodes that can be assigned to each station is shown in constraints (3.3). Constraints (3.4) guarantee the continuity of the routes and return to the original station. Constraints (3.5) ensure that a node should only be served at most within one route and should have only one node connected to its predecessor if visited. Constraints (3.6) state the limited travel time for each route. The sub-tour elimination constraints are assured in constraints (3.7). Constraints (3.8) guarantee that each route can be assigned to only one station. Constraints (3.9) ensure that a node can be assigned to a station if there is a route connected to it. Constraints (3.10)–(3.12) specify the binary variables used in the formulation and finally, the auxiliary variables taking positive integer values are stated in (3.13).

4. HEURISTIC ALGORITHM FOR THE OLRP

A heuristic method based on the greedy randomized adaptive search procedure (GRASP) is presented in this section to solve the OLRP. In general, the proposed GRASP is composed of two major phases. In the first phase or constrictive phase, an initial solution based on a density metric to order the nodes is generated within some steps, which is explained in Section 4.1. In the second phase of the GRASP, a local search method is used to improve the initial solution generated through the first phase. In this phase, three neighborhood operators are presented to construct new solutions. When a better solution is obtained, the new best solution will be replaced by the past best-known solution. This improvement procedure can be continued until the termination condition (*i.e.*, a predetermined number of iterations or *Iter*) is matched. This phase is expressed in more detail in Section 4.2.

4.1. Creating the initial solution

As pointed out earlier, the initial solution is created through the first phase of the GRASP. This phase is inspired by the density ordered heuristic of the knapsack problem [29]. The method is based on the “Route first-cluster second” strategy, which involves constructing a giant route followed by the split of this route into some feasible sub-routes. Firstly, this phase constructs a giant route that contains all nodes beginning with one station. In this route, each node is visited once and the route is finished at the starting station like in the vehicle routing problem (VRP) without considering any constraints. Then, this route is divided into sub-routes where any of them represents a feasible route.

In more detail, let an OLRP test instance, with M candidate stations, N nodes where each has a non-negative score and required time for visiting. The method starts by calculating the density vector between each candidate station and all nodes, named by DV_{M-N} . The density vector of DV_{M-N} is calculated by the following heuristic formula:

$$DV_{M-N}(i) = \sum_{j=1}^N \frac{S_j}{\text{dist}(i, j)} \quad \forall i \in I, \quad (4.1)$$

where S_j is the score of the node j and $\text{dist}(i, j)$ is the distance between candidate station i and node j according to the Euclidian distance. To create the giant route, the station i_1 which has the maximum value of DV_{M-N} is selected to be opened (*e.g.*, $i_1 = \max_i \{DV_{M-N}(i)\}$). The opened station i_1 can be considered as a started point of the giant route. In the next step, first node j_1 should be included to the giant route based on the highest

Input: Test instance of OLRP
1- Calculate the density vector between stations and all nodes (DV_{M-N}).
2- Calculate the density matrix of the nodes (DM_N).
3- Sort the candidate stations based on DV_{M-N} in descending order, and select the first station to be opened.
4- Select the first node that has the highest DM_N value with the opened station and insert to the first sub-route.
5- Insert the next node by calculating the $DM_N(j_{next})$ and repeat it for all remaining nodes.
6- Create a feasible sub-routes base on the splitting process.
Output: Feasible sub-routes with only one station

FIGURE 2. Steps of the first phase of GRASP.

density value with the opened station (e.g., $j_1 = \max_j \left\{ \frac{S_j}{\text{dist}(i_1, j)} \right\} \forall j \in J$). Then, the next customer j_{next} is selected to visit, who has the highest density value with the previously included node j_p in the density matrix of nodes, named by DM_N , according to the following relation:

$$DM_N(j_{\text{next}}) = \max_j \left\{ \frac{S_{j_p} + S_j}{\text{dist}(j_p, j)} \right\} \quad \forall j \in J / \{j_1, \dots, j_p\}. \quad (4.2)$$

The process continues until all the nodes are inserted into the giant route. Consequently, the final step of “Route first-cluster second” strategy is applied to build the feasible sub-routes based on the ordering nodes in the giant route and travel time limit of T_{\max} . To do this, the splitting process begins to add the nodes one by one to the first sub-route until the travel time limit does not exceed. Then, the second sub-route starts to add the first not visited node in the giant route and continue adding nodes through the mentioned constraint. The splitting process is repeated for the last sub-route \bar{R} , until the status of all the nodes added to the sub-routes is checked. It is important to note that, it is possible that some nodes are not visited at all due to the travel time limit and the fixed number of sub-routes \bar{R} . The output of the first phase of the GRASP is a set of sub-routes with the same station representing a visit planning of the nodes based on their order entered the sub-routes. The scheme of the first phase of the GRASP is illustrated in Figure 2.

4.2. Improvement of the solution

In the second phase of the proposed GRASP, a local search method is applied to improve the solutions that come out of the first phase. Local search methods are used to solve many hard combinatorial optimization problems [24, 27]. It also is successfully used wildly in LRP models [17, 20, 31, 43, 45]. In this phase, three neighborhood operators are provided in the local search method that two of them are related to sub-routes and one remaining is related to the stations. These operators are described in detail as follows.

- Add operator: In this operator, node(s) that does not belong to any sub-routes can be added to existing sub-routes. Unvisited nodes that have a higher score and shorter distance with the gravity center of a sub-route have a higher chance to include to the sub-route. On the other hand, each sub-route that has a greater difference between total travel time and T_{\max} has a higher chance of accepting unvisited node(s). Finally, to add an unvisited node to a sub-route, the travel time limit should not be violated. Figure 3 shows the algorithm of the add operator for more details.
- Swap operator: In this operator, an unvisited node can be swapped with a node of sub-route respect to some considerations. The node inside the sub-route that has a lower score is swapped by an unvisited node with a higher score and close to the sub-route. This swap is possible when the travel time limit is not exceeded.

Input: Set of unvisited nodes $J_U = \{1, 2, \dots, j_u\}$ Set of feasible sub-routes are sorted in ascending order based on total travel time $R_s = \{R_1, R_2, \dots, R_{\bar{R}}\}$
1- Calculate the vector of $AO(j) = \frac{S_j}{dist(R_i^*, j)} \quad \forall j \in \{1, 2, \dots, j_u\}$ for unvisited nodes. R_i^* represents the gravity center of sub-route R_i .
2- Sort unvisited nodes based on $AO(j)$ in descending order, and add the node with the highest priority to current sub-route if the travel time limit is not violated. Otherwise, check this condition for the next sorted sub-routes in R_s .
3- Repeat step 1 for the remaining of the unvisited nodes.

Output: Sub-routes with a better solution or equal to the previous solutions

FIGURE 3. Steps of the add operator.

- Assign operator: In this operator, based on Y_{\max} value, other sorted stations in density vector DV_{M-N} can be opened to improve the best-known solution. All sub-routes or some of them can be assigned to the newly opened stations. This assignment can reduce the total travel time of the routes and the two above operators can be implemented again to improve the solution.

Figure 4 presents an illustrative example of the OLRP with a solution by the proposed GRASP. The example consists of 16 nodes and 4 candidate stations, all distributed in random coordinates. Score values are also given next to corresponding nodes. Figures 4b and 4c show the first phase or constructive phase of the GRASP, while the second phase of the heuristic method along with the neighborhood operators are indicated in Figures 4d–4f.

Figure 4a shows the geographic distribution of the nodes and candidate stations. In Figure 4b, the giant route is constructed according to the steps of Figure 2. This giant route starts with station 3 who has the highest density vector DV_{M-N} calculated in relation (4.1) and the first node inserted the route has the highest DV_{M-N} value with station 3. The next nodes are included to the giant route, based on the value of $DM_N(j_{\text{next}})$. In Figure 4c, the giant route is divided into three feasible sub-routes without exceeding the travel time limit. The objective function value of this solution is the sum of the scores collected by three routes, which is equal to 26.

As seen in Figure 4d, one node has been able to add to the sub-route of R_1 according to the steps of Figure 3. This node has increased the total collected scores of the example to 29, without exceeding the allowed travel time of the sub-route R_1 . In Figure 4e, the unvisited node with the score of 3 swapped with the node inside sub-route R_2 with the lower score of 1 respect to the travel time limit. With this exchange, the total collected scores are reached to 32. Finally, assign operator is shown in Figure 4f by opening the station 1 and assigning sub-route R_2 to this station. By this change, two unvisited nodes are also added to sub-route R_2 , since its travel time is reduced. As illustrated in Figure 4f, it is obvious that the second phase of the GRASP is able to improve the total score from a value of 26 to 37.

5. COMPUTATIONAL RESULTS

5.1. Test instances of OLRP and solve them with GRASP

To evaluate the efficiency of the proposed GRASP, computational experiments are carried out. Since no benchmark instances were publicly available for the OLRP, the standard test instances of the CLRP generated by Barreto [7] are used. To apply these standard test instances, some changes are required. The following heuristic steps show the process of changing a CLRP instance to an OLRP one:

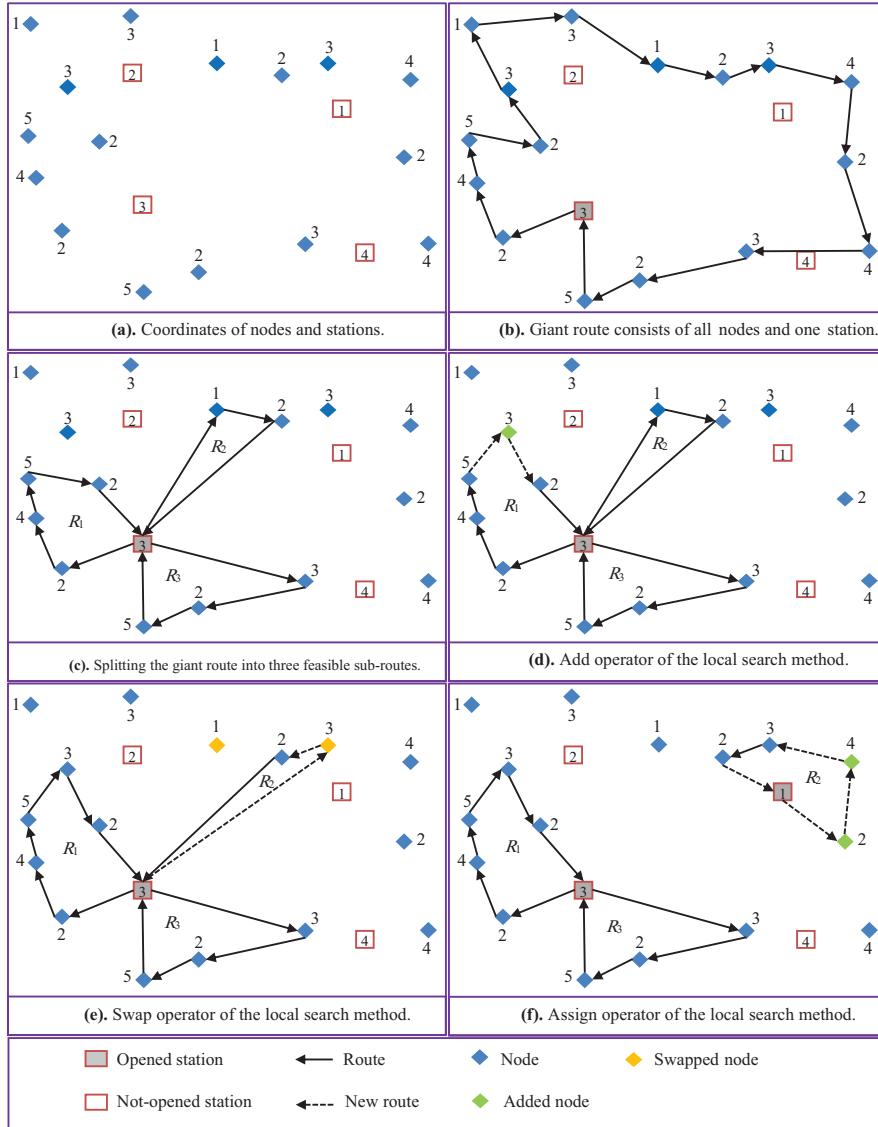


FIGURE 4. An illustrative example of the proposed GRASP.

Step 1. The demand of each customer in the CLRP instance (D_j) is considered as the score of each node in the OLRP instance (S_j).

Step 2. Euclidean distance between all nodes and stations is given for the travel time of the OLRP instance (e.g., $t_{ij} = \text{distance}(i, j)$) and the service time of each node j is obtained by the ratio $t_j = \frac{x_j}{y_j}$, where x_j and y_j are the coordinates of customer j in CLRP instance.

Step 3. Maximum allowed time travel of each sub-route in OLRP instance is equal to three times the number of customers in CLRP instance (e.g., $T_{\max} = 3 \times N$).

Step 4. The number of routes is calculated as $\bar{R} = \left\lceil \frac{\sum D_j}{Q} \right\rceil$, where Q is the vehicle capacity of the CLRP instance and $\left\lceil \frac{\sum D_j}{Q} \right\rceil$ is the smallest integer number which is higher than $\frac{\sum D_j}{Q}$.

Step 5. The maximum number of stations allowed to be opened in the OLRP instance is achieved by the relation of $Y_{\max} = \left\lceil \frac{\sum D_j}{\bar{F}} \right\rceil$, where \bar{F} is the mean of depots capacity of the CLRP instance.

By executing the above heuristic process, each instance of the CLRP becomes OLRP one. It is important to note that various other methods may also be used to generate OLRP instances, such as using random data, but the authors are interested using the standard test instances so that other researchers who like to continue in the case of OLRP are able to compare their results with the results presented in this work.

5.2. Implementation of GRASP on OLRP test instances

The first computational experiment of this section is related to the computation of the proper number of iterations or *Iter* explained in Section 4. Figure 5 depicts the objective function values of two selected large-size test problems against the different range of iterations. This figure indicates that the objective function is approximately constant for the two test problems after 500 runs. Then this value is considered as the termination condition of the improvement phase of the GRASP. It is noted that using values higher than 500 causes that the running time of the GRASP increased without any significant change in the final solutions. The proposed

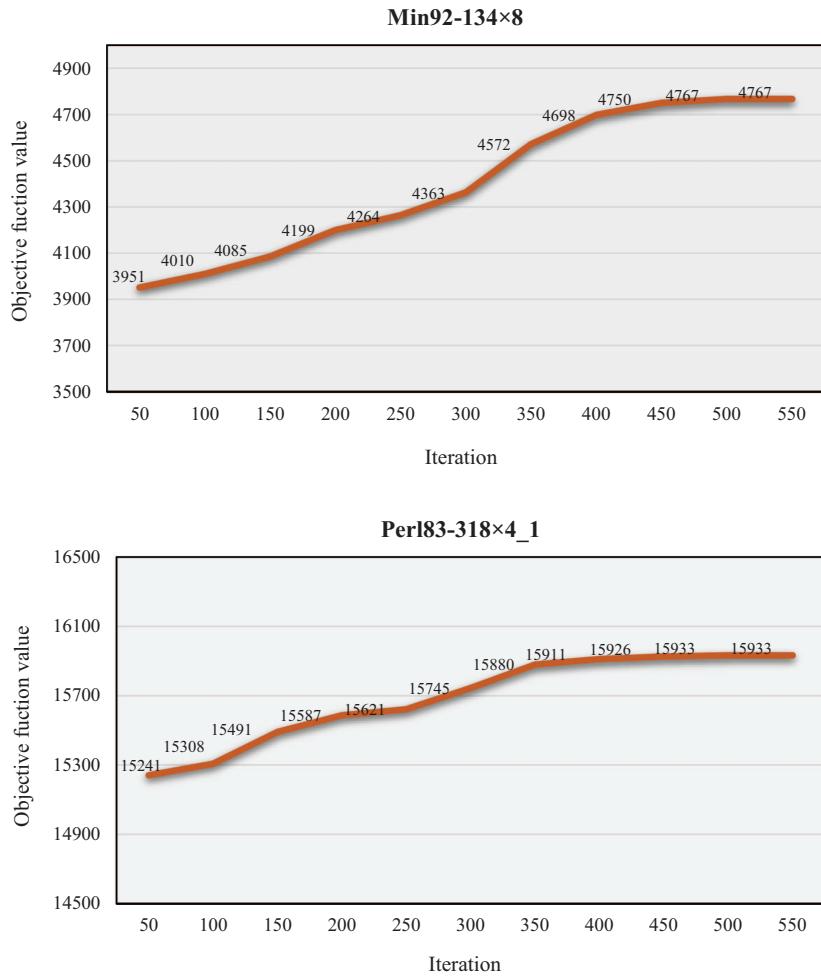


FIGURE 5. Deviation of objective function value for different iterations of runs.

TABLE 2. Computational results of the GRASP on Barreto's instances.

Instances	Parameters of the model			Initial phase of GRASP		Improvement phase of GRASP		GAP (%)	
	T_{\max}	R	Y_{\max}	Average solution	Best solution	Average solution	Best solution	Average solution	Best solution
Christofides69-50×5	150	5	1	643	690	712	742	10.7	7.5
Christofides69-75×10	225	4	1	1279.4	1340	1295	1344	1.2	0.3
Christofides69-100×10	300	3	1	1267	1367	1351	1420	6.6	3.9
Daskin95-88×8	264	2	2	43 190 116.4	44 652 630	44 630 242	44 827 665	3.3	0.4
Daskin95-150×10	450	5	3	10 362 092	10 432 329	10 558 787	10 622 016	1.9	1.8
Gaskell67-21×5	63	4	2	12 200	12 200	12 200	12 200	0.0	0.0
Gaskell67-22×5	66	3	1	6400	6475	6475	6475	1.2	0.0
Gaskell67-29×5	87	3	1	6395	6700	6595	6700	3.1	0.0
Gaskell67-32×5_1	96	4	1	21 103	22 230	22 241	23 830	5.4	7.2
Gaskell67-32×5_2	96	3	1	19 163	20 680	20 736	20 780	8.2	0.5
Gaskell67-36×5	108	4	1	646	725	682	750	5.6	3.4
Min92-27×5	540	4	1	5188	5270	5276	5400	1.7	2.5
Min92-134×8	402	10	3	4264	4363	4572	4767	7.2	9.3
Perl83-12×2	36	2	1	101	120	105	120	4.0	0.0
Perl83-55×15	165	2	2	853	920	933	960	9.4	4.3
Perl83-85×7	85	8	2	1352	1440	1523	1560	12.6	8.3
Perl83-318×4.1	1272	5	1	15 745	15 745	15 933	15 933	1.2	1.2
Perl83-318×4.2	1272	4	1	15 745	15 745	15 933	15 933	1.2	1.2
Or76-117×14	351	5	1	259	262	347	355	34.0	35.5
Average				2 824 464	2 905 328	2 910 839	2 924 682	6.2	4.6

GRASP is encoded in MATLAB R2008a on a computer, holding Intel core i7-HQ 6700 with 2.6 GHz of CPU and 16 GB of RAM. Table 2 shows the average computational results of 10 times running the program. Barreto's instances used in Table 2 include 19 test instances where the number of the node varies from 12 to 318, and the number of candidate stations is between 2 and 15. The name of each instance contains information about the name of the author that generated the instance, year of publication, number of nodes $|N_n|$, and the number of candidate stations $|N_0|$ (*i.e.*, Author-Year- $|N_n| \times |N_0|$). The first column of Table 2 gives the names of OLRP instances as explained above. The next three columns show the parameters of each instance calculated based on the mentioned heuristic steps. It should be noted that in some cases, some of the generated parameters cause the problem to be infeasible, so the corresponding parameter is changed to the nearest appropriate value. Bold numbers in columns 2–4 reflect the adjustment. To do so, the nearest number closest to the original value is selected to make the test instance feasible. The four next columns show the computational results of the proposed GRASP in terms of the average and best solutions obtained for initial and improvement phases, respectively. Consequently, the last two columns indicate the performance of the improvement phase of the GRASP by calculating the gap percentage. The gap percentage of each instance is computed as $100 \times [(\text{Improvement solution value} - \text{Initial solution value}) / \text{Improvement solution value}]$. As is shown in Table 2, it is observed that the improvement phase is more powerful than the initial phase and obtained better solutions with average gaps of 6.2% and 4.6% for the average and best solutions, respectively. To show the efficiency of the GRASP, it is essential that the results are compared with other methods or with the lower bound of solutions or with commercial software. In this paper, the quality of solutions is evaluated with the commercial software of GAMS 24.8.5 that is provided in the next section.

TABLE 3. Performance evaluation of GRASP in comparison of CPLEX solver of GAMS.

Instance	Quality of solutions			Gap (%)	CPU time (s)	
	Upper bound	CPLEX solver	GRASP		CPLEX solver	GRASP
Christofides69-50×5	777	756	742	-1.9	28 800	5
Christofides69-75×10	1364	1286	1344	4.3	28 800	12
Christofides69-100×10	1458	1312	1420	7.6	28 800	19
Daskin95-88×8	44 840 571	44 544 254	44 827 665	0.6	28 800	17
Daskin95-150×10	38 796 795	10 622 016	10 622 016	0.0	28 800	1
Gaskell67-21×5	-	12 200*	12 200	0.0	872	1
Gaskell67-22×5	-	6475	6475	0.0	25	1
Gaskell67-29×5	11 342	7175	6700	-7.1	28 800	1
Gaskell67-32×5_1	29 248	24 130	23 830	-1.3	28 800	5
Gaskell67-32×5_2	27 040	21 680	20 780	-4.3	28 800	4
Gaskell67-36×5	900	725	750	3.3	28 800	10
Min92-27×5	6608	5860	5400	-8.5	28 800	5
Min92-134×8	7911	4525	4767	5.1	28 800	22
Perl83-12×2	-	120	120	0.0	2	1
Perl83-55×15	1060	960	960	0.0	28 800	20
Perl83-85×7	1700	980	1560	37.2	28 800	14
Perl83-318×4_1	89 474	15 933	15 933	0.0	28 800	25
Perl83-318×4_2	84 986	15 933	15 933	0.0	28 800	25
Or76-117×14	462	265	355	25.4	28 800	20
Average				3.2	24 300	11

Notes. *Bold numbers represent the solution optimized by CPLEX solver.

5.3. Evaluating the performance of the GRASP

To evaluate the efficiency of the GRASP, shown in Table 2, the results are compared with a commercial solver. Hence, a straightforward way to determine the quality of the solutions is through the comparison of results with a commercial solver. To do this, each of the test instances is solved by using with CPLEX solver of GAMS 24.8.5 with the formulation given in Section 3. Table 3 provides the detailed results where the first column gives the size of test instances explained in the previous section. Two next columns are upper bound and the best CPLEX solution found in eight hours of running time. It is noted that the upper bound solutions are not reported for test instances reached the optimal solutions. The next column gives the best objective function value of the GRASP obtained from Table 2. The fifth column of Table 3 indicates the gap percentage between two solutions of CPLEX and GRASP that is computed as $100 \times [(GRASP - CPLEX)/GRASP]$ for each test instance. Finally, two last columns indicate the time solution of CPLEX and GRASP, respectively.

As shown in Table 3, both CPLEX and GRASP have been able to find the optimal solution for three test instances, which marked with bold numbers in the third column, and in four test instances, GRASP solutions are equal with CPLEX solutions. According to the results of Table 3, it is observed that the proposed GRASP is obtained better solution than CPLEX for seven test instances with an average gap of 11.9% and CPLEX is obtained better solutions than the GRASP for 5 test instances with an average gap of -4.9%. Consequently, as seen in Table 3, the proposed GRASP is more efficient in comparison to CPLEX solver in terms of not only CPU times of solution (*i.e.*, average of 11 for GRASP versus average of 24 300 for CPLEX), but also being able to solve all test instances with better quality (*i.e.*, GRASP solutions are better at an average gap of 3.2%). This result supports the fact that the proposed GRASP can provide far better solutions, especially when the problem size is large.

6. CONCLUSION AND FUTURE RESEARCH

Routing problems with profits in recent years have been paying great attention. The Orienteering Problem (OP) is the most studied case of TSP with the benefit of an outdoor sport in the mountains. Tourist tourism problems are the most attractive applications of the OP. In OP, the traveler gets a profit from the visited node, and the objective is to maximize the total profit that the traveler collects during the limited time. In OP, the journey begins in the depot and ends in a given terminal node, and all the nodes may not be visited because of the time or distance limitations. In this paper, a variant of LRP, named orienteering LRP (OLRP) is first introduced, which had two main differences with the classical LRP. First, the goal of the OLRP, which is used in a variety of applications in the real-world, is maximizing the sum of the collected scores of nodes. Moreover, against LRP, only a subset of nodes can be visited due to travel time limit of routes. The problem was modeled by an integer linear programming model. A heuristic GRASP with two main phases was proposed to solve the problem due to it belongs to the class of Np-hard problems. With a strategy of “Route first-cluster second” and some novel relations, an initial solution was created, and a local search method with some operators was used for further improvement of the solutions at the second phase. To show the performance of the model and evaluate the proposed GRASP, numerical experiments were carried out on modified standard test problems. Against the CPLEX solver of GAMS 24.8.5, the proposed GRASP was able to solve all test instances with a short time and high quality with an average of 3.2% for all instances. This paper has some suggestions that are capable for future researches: (a) investigating the fuzzy version of the problem (*i.e.*, scores of nodes or travel time between nodes are considered as fuzzy variables), (b) considering the OLRP with multi-objective function (maximizing the total collected scores and minimizing the total traveling), and (c) developing other solution algorithms.

Acknowledgements. The author would like to thank the referees, together with the Editor-in-Chief, Professor A. Ridha MAHJOUB, for their many helpful comments.

REFERENCES

- [1] A. Ahmadi-Javid, E. Amiri and M. Meskar, A profit-maximization location-routing-pricing problem: a branch-and-price algorithm. *Eur. J. Oper. Res.* **271** (2018) 866–881.
- [2] J. Ahn, O. de Weck, Y. Geng and D. Klabjan, Column generation based heuristics for a generalized location routing problem with profits arising in space exploration. *Eur. J. Oper. Res.* **223** (2012) 47–59.
- [3] L. Alfandari, Improved approximation of the general soft-capacitated facility location problem. *RAIRO:OR* **41** (2007) 83–93.
- [4] C. Archetti, F. Carrabs and R. Cerulli, The set orienteering problem. *Eur. J. Oper. Res.* **267** (2018) 264–272.
- [5] D.G. Baklagis, G. Dikas and I. Minis, The team orienteering pick-up and delivery problem with time windows and its applications in fleet sizing. *RAIRO:OR* **50** (2016) 503–517.
- [6] R. Baldacci, A. Mingozzi and R. Wolfler Calvo, An exact method for the capacitated location-routing problem. *Oper. Res.* **59** (2011) 1284–1296.
- [7] S. Barreto, Available at: http://sweet.us.pt/_iscf143 (2003).
- [8] S. Barreto, C. Ferreira, J. Paixao and B. Sousa Santos, Using clustering analysis in a capacitated location-routing problem. *Eur. J. Oper. Res.* **179** (2007) 968–977.
- [9] J.M. Belenguer, E. Benavent, C. Prins, C. Prodhon and R. Wolfler-Calvo, A branch-and-cut method for the capacitated location-routing problem. *Comput. Oper. Res.* **38** (2011) 931–941.
- [10] N. Bianchessi, R. Mansini and M.G. Speranza, A branch-and-cut algorithm for the Team Orienteering Problem. *Int. Trans. Oper. Res.* **25** (2018) 627–635.
- [11] M. Boccia, T.G. Crainic, A. Sforza and C. Sterle, Multi-commodity location-routing: flow intercepting formulation and branch-and-cut algorithm. *Comput. Oper. Res.* **89** (2018) 94–112.
- [12] T. Capelle, C.E. Cortés, M. Gendreau, P.A. Rey and L.-M. Rousseau, A column generation approach for location-routing problems with pickup and delivery. *Eur. J. Oper. Res.* **272** (2019) 121–131.
- [13] I.M. Chao, B.L. Golden and E.A. Wasil, A fast and effective heuristic for the orienteering problem. *Eur. J. Oper. Res.* **88** (1996) 475–489.
- [14] W.R. Cherif-Khettaf, M.H. Rachid, C. Bloch and P. Chatonnay, New notation and classification scheme for vehicle routing problems. *RAIRO:OR* **49** (2015) 161–194.
- [15] C. Contardo, V. Hemmelmayr and T.G. Crainic, Lower and upper bounds for the two-echelon capacitated location-routing problem. *Comput. Oper. Res.* **39** (2012) 3185–3199.

- [16] Z. Dai, F. Aqlan, K. Gao and Y. Zhou, A two-phase method for multi-echelon location-routing problems in supply chains. *Expert Syst. App.* **115** (2019) 618–634.
- [17] H. Derbel, B. Jarboui, S. Hanafi and H. Chabchoub, Genetic algorithm with iterated local search for solving a location-routing problem. *Expert Syst. App.* **39** (2012) 2865–2871.
- [18] M. Drezl and M. Schneider, A survey of variants and extensions of the location-routing problem. *Eur. J. Oper. Res.* **241** (2015) 283–308.
- [19] T. Drezner, Z. Drezner and D. Zerom, Competitive facility location with random attractiveness. *Oper. Res. Lett.* **46** (2018) 312–317.
- [20] C. Duhamel, P. Lacomme, C. Prins and C. Prodhon, A GRASP×ELS approach for the capacitated location-routing problem. *Comput. Oper. Res.* **37** (2010) 1912–1923.
- [21] J.W. Escobar, R. Linfati and P. Toth, A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. *Comput. Oper. Res.* **40** (2013) 70–79.
- [22] K.M. Ferreira and T.A. de Queiroz, Two effective simulated annealing algorithms for the Location-Routing Problem. *Appl. Soft Comput.* **70** (2018) 389–422.
- [23] D. Gavalas, V. Kasapakis, C. Konstantopoulos, G. Pantziou, N. Vathis and C. Zaroliagis, The eCOMPASS multimodal tourist tour planner. *Expert Syst. App.* **42** (2015) 7303–7316.
- [24] H. Gavranović and M. Buljubašić, A hybrid approach combining local search and constraint programming for a large scale energy management problem. *RAIRO:OR* **47** (2013) 481–500.
- [25] B.L. Golden, L. Levy and R. Vohra, The orienteering problem. *Nav. Res. Logist.* **34** (1987) 307–318.
- [26] A. Gunawan, H.C. Lau and P. Vansteenwegen, Orienteering problem: a survey of recent variants, solution approaches and applications. *Eur. J. Oper. Res.* **255** (2016) 315–332.
- [27] M. Khorbatly, H. Dkhil, H. Alabboud and A. Yassine, A hybrid algorithm Tabu Search – GRASP for wounded evacuation in disaster response. *RAIRO:OR* **54** (2020) 19–36.
- [28] G. Kobeaga, M. Merino and J.A. Lozano, An efficient evolutionary algorithm for the orienteering problem. *Comput. Oper. Res.* **90** (2018) 42–59.
- [29] R. Kohli and R. Krishnamurti, A total-value greedy heuristic for the integer knapsack problem. *Oper. Res. Lett.* **12** (1992) 65–71.
- [30] G. Laporte, Location-routing problems, edited by B.L. Golden and A.A. Assad. In: *Vehicle Routing: Methods and Studies*. North-Holland, Amsterdam (1988) 163–198.
- [31] J. Li, P.M. Pardalos, H. Sun, J. Pei and Y. Zhang, Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. *Expert Syst. App.* **42** (2015) 3551–3561.
- [32] B. Ludwig, B. Zenker and J. Schrader, Recommendation of personalized routes with public transport connections. *Intelligent Interactive Assistance and Mobile Multimedia Computing*, edited by D. Tavangarian, T. Kirste and D. Timmermann. In: *International Conference, IMC 2009, Rostock-Warnemünde, Germany, November 9–11, 2009. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg (2009) 97–107.
- [33] S. Majidi, S.-M. Hosseini-Motlagh, S. Yaghoubi and A. Jokar, Fuzzy green vehicle routing problem with simultaneous pickup – delivery and time windows. *RAIRO:OR* **51** (2017) 1151–1176.
- [34] F. Malucelli, A. Giovannini and M. Nonato, Designing single origin-destination itineraries for several classes of cycle-tourists. *Transp. Res. Proc.* **10** (2015) 413–422.
- [35] S.M.H. Manzour-al-Ajjad, S.A. Torabi and S. Salhi, A hierarchical algorithm for the planar single-facility location routing problem. *Comput. Oper. Res.* **39** (2012) 461–470.
- [36] A. Nadizadeh, The fuzzy multi-depot vehicle routing problem with simultaneous pickup and delivery: formulation and a heuristic algorithm. *Int. J. Ind. Eng. Prod. Res.* **28** (2017) 325–345.
- [37] A. Nadizadeh and H. Hosseini Nasab, Modelling and solving the capacitated location-routing problem with simultaneous pickup and delivery demands. *Int. J. Transp. Eng.* **6** (2019) 217–235.
- [38] A. Nadizadeh and B. Kafash, Fuzzy capacitated location-routing problem with simultaneous pickup and delivery demands. *Transp. Lett.* **11** (2019) 1–19.
- [39] A. Nadizadeh and H.H. Nasab, Solving the dynamic capacitated location-routing problem with fuzzy demands by hybrid heuristic algorithm. *Eur. J. Oper. Res.* **238** (2014) 458–470.
- [40] A. Nadizadeh, R. Sahraeian, A. Sabzevari Zadeh and S.M. Homayouni, Using greedy clustering method to solve capacitated location-routing problem. *Afr. J. Bus. Manage.* **5** (2011) 7499–7506.
- [41] A. Nadizadeh, A. Sadegheih and A. Sabzevari Zadeh, A hybrid heuristic algorithm to solve capacitated location-routing problem with fuzzy demands. *Int. J. Ind. Math.* **9** (2017) 1–20.
- [42] G. Nagy and S. Salhi, Location-routing: issues, models and methods. *Eur. J. Oper. Res.* **177** (2007) 649–672.
- [43] V.P. Nguyen, C. Prins and C. Prodhon, Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. *Eur. J. Oper. Res.* **216** (2012) 113–126.
- [44] J. Pietz and J.O. Royset, Generalized orienteering problem with resource dependent rewards. *Nav. Res. Logist.* **60** (2013) 294–312.
- [45] C. Prins, C. Prodhon and R. Wolfler Calvo, Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *Oper. Res. Q.* **4** (2006) 221–238.
- [46] C. Prodhon and C. Prins, A survey of recent research on location-routing problems. *Eur. J. Oper. Res.* **238** (2014) 1–17.

- [47] S. Rath and W.J. Gutjahr, A math-heuristic for the warehouse location–routing problem in disaster relief. *Comput. Oper. Res.* **42** (2014) 25–39.
- [48] M. Rodrigues de Holanda Maia, A. Plastino and P.H. Vaz Penna, Hybrid data mining heuristics for the heterogeneous fleet vehicle routing problem. *RAIRO:OR* **52** (2018) 661–690.
- [49] R. Sahraeian and A. Nadizadeh, Using greedy clustering method to solve capacitated location-routing problem. In: *XIII Congreso de Ingeniería de Organización. Barcelona* (2009) 1721–1729.
- [50] S. Salhi and G.K. Rand, The effect of ignoring routes when locating depots. *Eur. J. Oper. Res.* **39** (1989) 150–156.
- [51] W. Souffriau, P. Vansteenwegen, J. Vertommen, G.V. Berghe and D.V. Oudheusden, A personalized tourist trip design algorithm for mobile tourist guides. *Appl. Artif. Intell.* **22** (2008) 964–985.
- [52] P. Vansteenwegen, W. Souffriau and D.V. Oudheusden, The orienteering problem: a survey. *Eur. J. Oper. Res.* **209** (2011) 1–10.
- [53] E. Yakici, Solving location and routing problem for UAVs. *Comput. Ind. Eng.* **102** (2016) 294–301.
- [54] J. Yang and H. Sun, Battery swap station location-routing problem with capacitated electric vehicles. *Comput. Oper. Res.* **55** (2015) 217–232.
- [55] Q. Yu, K. Fang, N. Zhu and S. Ma, A matheuristic approach to the orienteering problem with service time dependent profits. *Eur. J. Oper. Res.* **273** (2019) 488–503.
- [56] V.F. Yu, P. Jewpanya, S.-W. Lin and A.A.N.P. Redi, Team orienteering problem with time windows and time-dependent scores. *Comput. Ind. Eng.* **127** (2019) 213–224.
- [57] Y. Zare Mehrjerdi and A. Nadizadeh, Using greedy clustering method to solve capacitated location-routing problem with fuzzy demands. *Eur. J. Oper. Res.* **229** (2013) 75–84.
- [58] Y. Zare Mehrjerdi and A. Nadizadeh, Using greedy clustering method to solve capacitated location-routing problem with fuzzy demands. *Int. J. Ind. Eng. Prod. Res.* **27** (2016) 1–19.
- [59] S. Zhang, J.W. Ohlmann and B.W. Thomas, A priori orienteering with time windows and stochastic wait times at customers. *Eur. J. Oper. Res.* **239** (2014) 70–79.
- [60] J. Zhao and V. Verter, A bi-objective model for the used oil location-routing problem. *Comput. Oper. Res.* **62** (2015) 157–168.