Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 47 (2020) 680–687

Transportation Research Procedia

www.elsevier.com/locate/procedia

22nd EURO Working Group on Transportation Meeting, EWGT 2019, 18-20 September 2019, Barcelona, Spain

A Biased-Randomized Learnheuristic for Solving the Team Orienteering Problem with Dynamic Rewards

L. Reyes-Rubiano[a], A. A. Juan[b,*], C. Bayliss[b], J. Panadero[b], J. Faulin[a], P. Copado[b]

[a] *Institute of Smart Cities, Public Unversity of Navarre, Campus Arrosadia, Pamplona 31006, Spain*
[b] *IN3 – Computer Science Dept., Universitat Oberta de Catalunya, Castelldefels 08860, Spain*

Abstract

In this paper we discuss the team orienteering problem (TOP) with dynamic inputs. In the static version of the TOP, a fixed reward is obtained after visiting each node. Hence, given a limited fleet of vehicles and a threshold time, the goal is to design the set of routes that maximize the total reward collected. While this static version can be efficiently tackled using a biased-randomized heuristic (BR-H), dealing with the dynamic version requires extending the BR-H into a learnheuristic (BR-LH). With that purpose, a 'learning' (white-box) mechanism is incorporated to the heuristic in order to consider the variations in the observed rewards, which follow an unknown (black-box) pattern. In particular, we assume that: *(i)* each node in the network has a 'base' or standard reward value; and *(ii)* depending on the node's position inside its route, the actual reward value might differ from the base one according to the aforementioned unknown pattern. As new observations of this black-box pattern are obtained, the white-box mechanism generates better estimates for the actual rewards after each new decision. Accordingly, better solutions can be generated by using this predictive mechanism. Some numerical experiments contribute to illustrate these concepts.

Keywords: Transportation; team orienteering problem; learnheuristics; dynamic inputs; biased randomization.

1. Introduction

Using a limited fleet of m vehicles, the team orienteering problem or TOP (Chao et al., 1996) consists in determining the optimal set of m routes, each of them connecting an origin depot with a destiny depot, such in a way that: *(i)* the total reward collected after visiting the nodes included in these m routes is maximized; and *(ii)* the length of each route is restricted by a threshold limit, $t_{max} > 0$. While most existing work refers to the static version of the problem –in which the reward associated with each node is a fixed value–, some degree of dynamism in the behaviour of these rewards might appear in real-life applications. Hence, for instance, the actual reward associated with a node might differ from its 'base' (standard) value depending on the order in which the node is visited, e.g.: nodes located at the

* Corresponding author. Tel.: +34-933-263-839; fax: +34-934-176-495.
E-mail address: ajuanp@uoc.edu

2352-1465
10.1016/j.trpro.2020.03.147

very beginning of a route might increase their associated reward levels, while the opposite effect might be observed for those located at the end of a route (Figure 1).

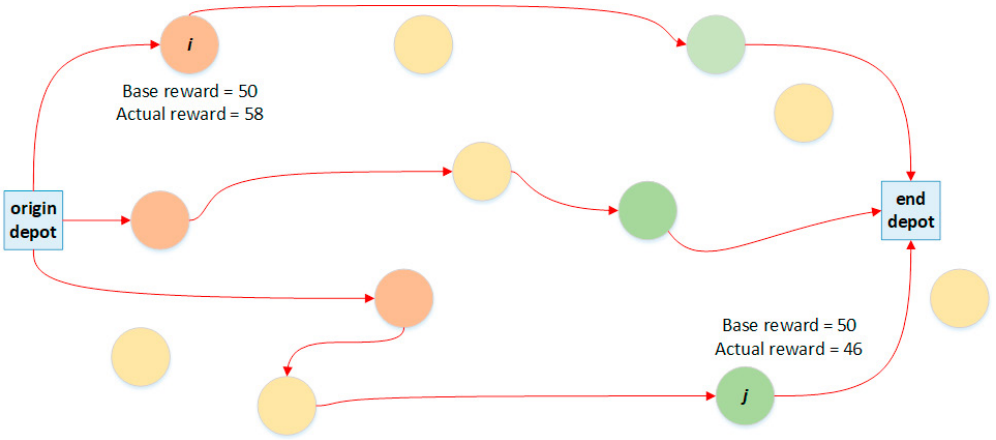

Fig. 1: An illustrative example of the TOP with dynamic rewards.

In order to take into account this dynamic behaviour in the rewards, this paper proposes extending an effective biased-randomized algorithm into a learnheuristic one. Biased randomization techniques can be applied on a constructive heuristic to enhance their performance (Grasas et al., 2017). In short, these techniques introduce some degree of randomness into the greedy behaviour of the heuristic without destroying its logic. To attain this goal, they use skewed (non-symmetric) probability distributions. Also, as discussed in Calvet et al. (2017) and Arnau et al. (2018), a learnheuristic algorithm combines a heuristic-based approach with some 'learning' (white-box) mechanism in order to solve combinatorial optimization problems with dynamic inputs. In our case, these dynamic inputs follow an unknown (black-box) pattern, and only sample observations of this pattern are obtained as new decisions are made regarding the route design. Using these observations, the learning mechanism is used to estimate the actual rewards as new decisions are made.

The rest of the paper is structured as follows: Section 2 briefly reviews related work; Section 3 provides a detailed description of the dynamic TOP version considered; Section 4 introduces a biased-randomized heuristic, which is then extended into a learnheuristic in Section 5; Section 6 reports the results of the computational experiments; finally, the main findings and future research lines are provided in Section 7.

2. Literature Review

When only one vehicle is considered, the orienteering problem (OP) consists in designing the route which maximizes the collected reward by visiting as many nodes as possible without violating the threshold value for the route length. Typically, each node can be visited at most once –or, alternatively, it only offers rewards the first time it is visited. Royset (2009) models an OP as a set covering problem, with the goal of maximizing profits related to location visits. Campbell et al. (2011) studied an OP version considering deterministic benefits and penalties to estimate the impact of stochastic travel and service times. The problem is solved using a dynamic programming framework, that maximizes the expected total reward. More recently, Dolinskaya et al. (2018) extended the previous work by considering an adaptive approach, where the route could be redefined on-the-fly according to unexpected delays or waiting times.

The multi-route TOP was introduced by Chao et al. (1996) as an extension of the OP. Poggi et al. (2010) developed a robust branch-and-cut-and-price algorithm to solve the traditional version of the problem, setting new benchmarks for it. Vansteenwegen et al. (2011) developed an iterated local search metaheuristic to solve the TOP with time windows. The approach is aimed at maximizing the rewards, assuming that the service should start within a defined schedule. In this sense, an early arrival leads to waiting times, causing a delay. Dang et al. (2013) proposed a particle swarm optimization to solve the classical version of the TOP. Their main contribution was the fast exploration of a large

number of neighborhoods. Souffriau et al. (2013) presented a multi-constraint and time-dependent approach. These authors assume that nodes could consider more than one time window. Their approach is managed as a deterministic problem and solved by means of a hybrid algorithm. Gunawan et al. (2016) presented a survey focusing on the most recent papers and surveys that are related to the TOP and its variants. These authors also studied a number of recent applications and an overview of future trends. They mentioned uncertainties or stochastic aspects that have been studied, especially related to the rewards as well as travel and service times. Simheuristic algorithms (Juan et al., 2018) combining simulation with metaheuristics to solve stochastic versions of the TOP have been proposed by Panadero et al. (2017) and Reyes-Rubiano et al. (2018). However, to the best of our knowledge, this is the first paper proposing the use of learnheuristic algorithms (Calvet et al., 2016; Arnau et al., 2018) to deal with the dynamic version of the TOP.

3. A Formal Description of the Dynamic TOP with Position-Dependant Rewards

In the TOP, the fleet is composed of $m \geq 2$ vehicles, and there is a time threshold, $t_{max} > 0$, for completing each route. The set of possible nodes to visit can be described by an undirected graph $G = (N, E)$, where $N = \{1, 2, \ldots, n\}$ is the set of n nodes, and E is the set of edges connecting these nodes. Node 1 represents the origin depot, while node n is the end depot. Visiting a node $i \in N$ for the first time has a reward $u_i \geq 0$, being $u_1 = u_n = 0$ (additional visits to a node are not considered since no more rewards can be collected). In general, $u_i = f(v_i, p)$, where v_i is a 'base' or standard reward associated with each node $i \in N$ and p is the position of node i in its route. Here, $f(v_i, p)$ is an unknown (black-box) function, which values need to be estimated from existing observations and using some learning (white-box) mechanism. Traversing an edge $e = (i, j) \in E$ has a travel time, $t_{ij} = t_{ji} > 0$. As illustrated in Figure 1, the final solution to the problem is a set M of m routes, where each route is defined by an array of nodes starting from node 1 (origin depot) and arriving at node n (end depot). The objective function is the sum of collected rewards, which needs to be maximized without exceeding threshold value for each route.

In a more formal way, the objective function can be written as:

$$\max \sum_{m \in M} \sum_{(i,j) \in E} u_i \cdot x_{ij}^m \tag{1}$$

where: $E = \{(i, j)/i, j \in N, i \neq j\}$ is the set of edges, and x_{ij}^m is a binary decision variable which equals 1 if the edge $(i, j) \in E$ is in the route m and 0 otherwise. Thus, the objective function sums the scores collected at all of the visited nodes. The constraints are given below, with some explanations provided. First, there is a threshold time to complete each route, i.e.:

$$\sum_{(i,j) \in E} x_{ij}^m \cdot t_{ij} \leq t_{max} \qquad \forall m \in M \tag{2}$$

Each point is visited at most once during the route:

$$\sum_{m \in M} \sum_{(i,j) \in E} x_{ij}^m \leq 1 \tag{3}$$

Each route starts at the origin depot (node 1) and arrives at the end depot (node n):

$$\sum_{j \in N} x_{1j}^m = 1 \qquad \forall m \in M \tag{4}$$

$$\sum_{i \in N} x_{in}^{m} = 1 \qquad \forall m \in M \tag{5}$$

Finally, the vehicle leaves each node it visits, except for the end depot:

$$\sum_{i \in N} x_{ih}^{m} - \sum_{j \in N} x_{hj}^{m} = 0 \qquad \forall h \in N \sim \{1, n\}, \forall m \in M \tag{6}$$

4. A Biased-Randomized Heuritic (BR-H) for the Static TOP

A biased-randomized heuristic (BR-H) can be used to efficiently solve the static version of the TOP. As discussed in Grasas et al. (2017), biased-randomization techniques make use of skewed probability distributions to introduce a non-uniform randomization process into a heuristic procedure, thus transforming it into a probabilistic algorithm that can be run multiple times. These techniques have been successfully applied in the past to improve the performance of classical heuristics, both in scheduling applications (Juan et al., 2014) as well as in vehicle routing ones (Faulin and Juan, 2008; Juan et al., 2015; Dominguez et al., 2016a,b). The constructive heuristic used in this paper encompasses the following stages:

- First, an initial 'dummy' solution is built by constructing a route connecting each customer node with the origin and end depots. In order to merge some of these routes –so that a single vehicle can visit more than one customer–, the concept of 'savings' is introduced as follows: the time-based savings of merging any two routes is given by the savings in time associated with completing the merged route instead of the two original ones. This concept is extended to the concept of 'preference', which is a linear combination of time-based savings and accumulated rewards (thus, if we face two potential merges with similar time-based savings, the one generating a greater accumulated reward will be prioritized). The concept of preference is used to generate a sorted list of potential merges, and these are completed following the corresponding order, from higher to lower preference. Of course, a merge can be completed only if the total expected time after the operation does not exceed the maximum time threshold, t_{max}. Notice that the previously described process constitutes a simple but effective heuristic that provides, by construction, a 'good' solution for the deterministic version of the problem.
- Secondly, we employ biased-randomization techniques (Grasas et al., 2017) to transform the previously described heuristic into a probabilistic algorithm. In particular, the selection of the next element from the savings list is driven according to a Geometric distribution. Hence, merging operations with a larger preference are more likely to be selected, but the selection process is not greedy any more.
- By encapsulating the previous steps into a multi-start procedure, high-quality solutions can be obtained very fast. Actually, Table 1 shows that for all the selected instances our BR-H algorithm is able to match the best-known solution (BKS) for the static version of the problem. It is also the case that this is achieved in just a few seconds, which proofs the effectiveness of our BR-H algorithm for solving the static TOP.

5. Extending our BR-H to a Learnheuristic (BR-LH) for the Dynamic TOP

Despite our BR-H approach can be used to solve the static version of the TOP, it does not account for the possible variations in the rewards described in the description of the dynamic TOP. Hence, using the BR-H approach as it is –or any other method that does not consider the dynamic inputs– will generate sub-optimal solutions. For this reason, in this section the BR-H method is extended into a simple but illustrative learnheuristic (BR-LH). As discussed in Arnau et al. (2018) for the vehicle routing problem (Caceres-cruz et al., 2015), our BR-LH makes use of sample observations generated by the unknown (black-box) pattern to estimate the rewards associated with future route-merging decisions. This learnheuristic approach can be seen as a white-box mechanism that tries to predict the black-box pattern, which represents the dynamism generated in a 'real-life' application. In our case, the learning mechanism works as follows (Figure 2):

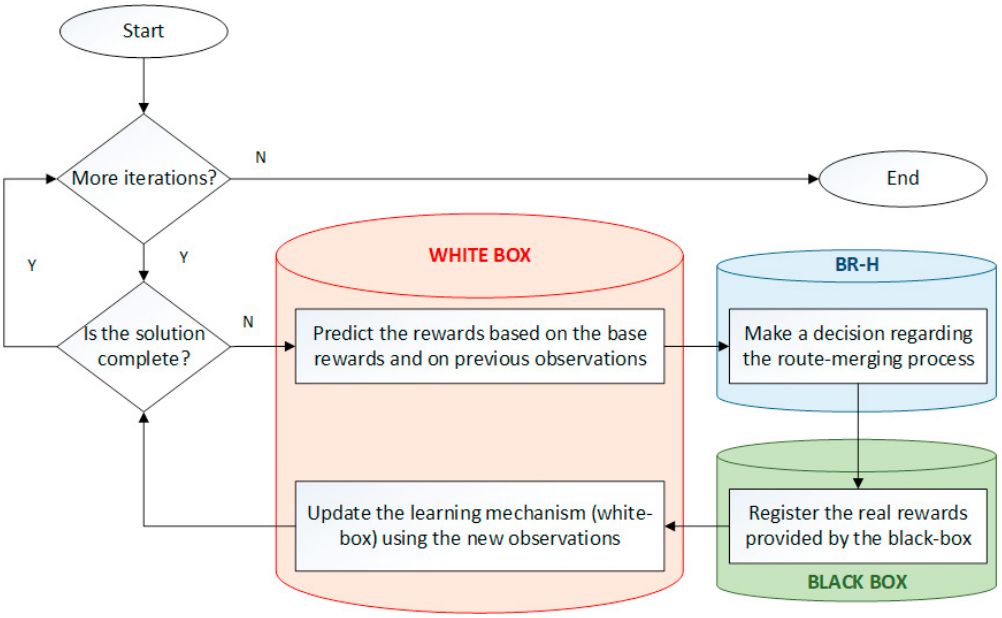

Fig. 2: Extending our BR-H into a learnheuristic (BR-LH) by integrating a simple learning mechanism.

- At each step of the constructive heuristic encapsulated into the BR-H algorithm, the estimated reward associated with a given node is computed as the average of the observed rewards for the same node whenever it was located at the first position of the route, at the last position, or at an intermediate position.
- The previous estimate is gaining in accuracy as new merging decisions are made and new values for the associated rewards are provided by the black-box. Hence, the simple white-box mechanism based on computing averages of observed values for similar positions is iteratively 'trained' at each new iteration of the multi-start BR-H algorithm.
- Eventually, it is expected that this white-box integrated inside the BR-H algorithm captures part of the dynamism of the unknown (black-box) patter. As a consequence, it should be able to generate better solutions that the ones provided by a static solving approach. Of course, more sophisticated white-box mechanisms could be designed and better solutions could be obtained, but the essential idea behind the learnheuristic approach would be the same.

6. Computational Experiments

Our BR-H and BR-LH algorithms were implemented in Java code and run on a personal computer with 8 GB of RAM and an Intel Core *i*7 at 1.8 GHz. The set of instances that we employ to test our approach is a natural extension of the classical benchmark instances for the static TOP proposed by Chao et al. (1996), which are available at `https://www.mech.kuleuven.be/en/cib/op/instances`. Each instance involves a fleet size, number of nodes, customer profits, and maximum route duration t_{max}. The experimentation relies on a comparison between the solution for the static and dynamic versions of the problem. For each version (static and dynamic) the value of the objective function is estimated to assess the solution performance.

In our experiments, the behaviour of the black-box (which is always non-visible for the algorithm) was defined as follows: $\forall i \in N$, $u_i = v_i$ whenever i is not an external node in its route (i.e., i is not directly connected to a depot). On the contrary, if i is the first node in its route, then $u_i = v_i + \lambda$, where λ is a random variable representing a percentage increase (with respect to the base reward), which follows a Triangular distribution with parameters 0%, 7%, and 10%. Analogously, if i is the last node in its route, then $u_i = v_i - \beta$, with β another random variable following a Triangular distribution with parameters 0%, 5%, and 7%.

Table 1 reports the rewards obtained by both the BR-H and the BR-LH for the dynamic TOP. Notice that the solutions provided by the BR-LH outperform those provided by the BR-H method (average gap about 4.9%), since the latter does not account for the dynamism in the rewards. In other words, using a static method for the dynamic version would result in sub-optimal solutions.

Figure 3 shows a comparison between different scenarios and algorithms. Taking as a reference the best-known solutions for the static version of the problem, the multiple box-plot shows the percentage gaps associated with: *(i)* our BR-H when it is applied to the static version (which is 0% on the average since our biased-randomized approach reaches all the BKS values); *(ii)* our BR-H when it is applied to the the dynamic version (which is about 3% on the average, meaning that higher rewards are collected due to the different parametrization of the random variables λ and β); and *(iii)* our BR-H approach, which is also applied to the dynamic scenario (notice the average gap of about 8%, which again is due to the parametrization of the random variables and to the fact that this method is generating better estimates of the unknown values of the dynamic rewards).

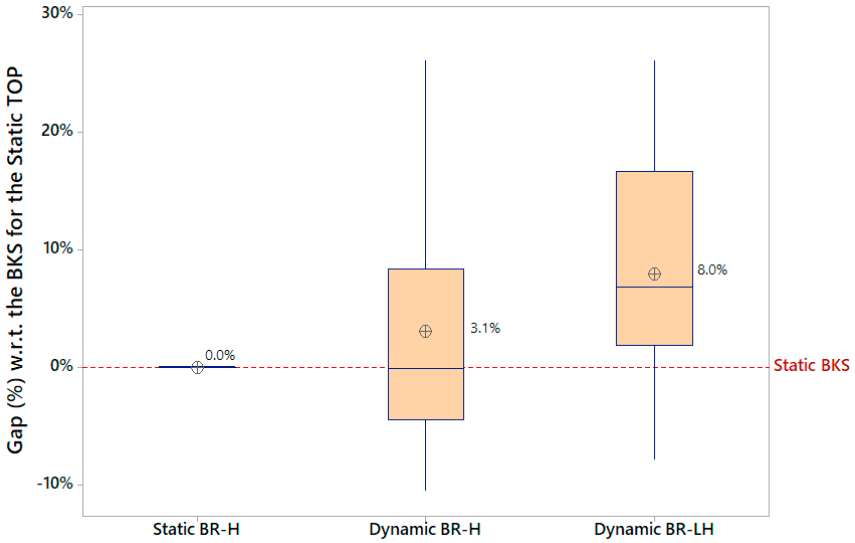

Fig. 3: Comparison of different algorithms and scenarios w.r.t. the BKS for the static TOP.

7. Conclusions and Future Work

This paper has discussed a dynamic version of the team orienteering problem, and how a learning mechanism can be combined with a biased-randomized heuristic to obtain solutions that are able to take into account this dynamic and difficult-to-predict behaviour of the rewards. The computational experiments show that the best solutions under a static environment should not be used into a dynamic environment, since they generate sub-optimal values. In contrast, a learnheuristic approach –such as the one considered here– is able to incorporate the dynamism of the problem into the solving approach. Despite the simplicity of the learning mechanism employed in this paper, the main concepts behind the learnheuristic approach are clearly displayed in this paper.

In future work, we plan to: *(i)* enrich both the learning (white-box) mechanism as well as the black-box pattern in order to generate more realistic and sophisticated environments; *(ii)* test our approach with large-sized instances; and *(iii)* extend the computational experiments to analyze how results vary as different dynamic components are used to model rewards and travel or service times.

Table 1: Results for the static and dynamic versions of the TOP – comparing a heuristic-based algorithm (BR-H) with a learnheuristic (BR-LH).

				Static TOP			Dynamic TOP				
				BKS	BR-H		BR-H		BR-LH		
Instance	nodes	vehicles	t_max	Reward [1]	Reward [2]	Gap [1]-[2]	Reward [3]	Time (s)	Reward [4]	Time (s)	Gap [3]-[4]
p1.4.j	32	4	12.5	75	75	0.0%	94.6	1	94.6	4	0.0%
p1.4.k	32	4	13.8	100	100	0.0%	108.4	1	119.2	6	10.0%
p1.4.l	32	4	15.0	120	120	0.0%	141.8	1	142.7	7	0.7%
p1.4.m	32	4	16.2	130	130	0.0%	138.5	1	143.4	9	3.5%
p1.4.n	32	4	17.5	155	155	0.0%	154.9	1	165.6	10	6.9%
p2.2.a	21	2	7.5	90	90	0.0%	86.3	1	91.9	1	6.5%
p2.2.b	21	2	10.0	120	120	0.0%	129.8	1	140.0	1	7.8%
p2.2.c	21	2	11.5	140	140	0.0%	125.4	1	129.2	1	3.1%
p2.2.d	21	2	12.5	160	160	0.0%	152.9	1	163.0	1	6.6%
p2.2.e	21	2	13.5	190	190	0.0%	184.9	1	184.9	1	0.0%
p3.3.g	33	3	15.0	270	270	0.0%	269.3	2	280.8	9	4.2%
p3.3.h	33	3	16.7	300	300	0.0%	338.3	1	344.7	10	1.9%
p3.3.i	33	3	18.3	330	330	0.0%	310.7	1	364.9	11	17.4%
p3.3.j	33	3	20.0	380	380	0.0%	394.3	1	402.3	12	2.1%
p3.3.k	33	3	21.7	440	440	0.0%	398.9	1	409.1	14	2.6%
Averages				*200.0*	*200.0*	*0.0%*	*201.9*	*1*	*211.8*	*6*	*4.9%*

References

Arnau, Q., Juan, A., Serra, I., 2018. On the use of learnheuristics in vehicle routing optimization problems with dynamic inputs. Algorithms 11, 208.

Caceres-cruz, J., Arias, P., Guimarans, D., Riera, D., Juan, A.A., 2015. Rich vehicle routing problem: survey. ACM Computing Surveys 47, 1–32.

Calvet, L., de Armas, J., Masip, D., Juan, A.A., 2017. Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. Open Mathematics 15, 261–280.

Calvet, L., Ferrer, A., Gomes, M.I., Juan, A.A., Masip, D., 2016. Combining statistical learning with metaheuristics for the multi-depot vehicle routing problem with market segmentation. Computers & Industrial Engineering 94, 93–104.

Campbell, A.M., Gendreau, M., Thomas, B.W., 2011. The orienteering problem with stochastic travel and service times. Annals of Operations Research 186, 61–81.

Chao, I.M., Golden, B.L., Wasil, E.A., 1996. The team orienteering problem. European Journal of Operational Research 88, 464–474.

Dang, D.C., Guibadj, R.N., Moukrim, A., 2013. An effective PSO-inspired algorithm for the team orienteering problem. European Journal of Operational Research 229, 332–344.

Dolinskaya, I., Shi, Z.E., Smilowitz, K., 2018. Adaptive orienteering problem with stochastic travel times. Transportation Research Part E: Logistics and Transportation Review 109, 1–19.

Dominguez, O., Guimarans, D., Juan, A.A., de la Nuez, I., 2016a. A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. European Journal of Operational Research 255, 442–462.

Dominguez, O., Juan, A.A., Barrios, B., Faulin, J., Agustin, A., 2016b. Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet. Annals of Operations Research 236, 383–404.

Faulin, J., Juan, A.A., 2008. The algacea-1 method for the capacitated vehicle routing problem. International Transactions in Operational Research 15, 599–621.

Grasas, A., Juan, A.A., Faulin, J., de Armas, J., Ramalhinho, H., 2017. Biased randomization of heuristics using skewed probability distributions: a survey and some applications. Computers & Industrial Engineering 110, 216–228.

Gunawan, A., Lau, H.C., Vansteenwegen, P., 2016. Orienteering problem: a survey of recent variants, solution approaches and applications. European Journal of Operational Research 255, 315–332.

Juan, A.A., Kelton, W.D., Currie, C.S., Faulin, J., 2018. Simheuristics applications: dealing with uncertainty in logistics, transportation, and other supply chain areas, in: Proceedings of the 2018 Winter Simulation Conference, IEEE Press. pp. 3048–3059.

Juan, A.A., Lourenço, H.R., Mateo, M., Luo, R., Castella, Q., 2014. Using iterated local search for solving the flow-shop problem: parallelization, parametrization, and randomization issues. International Transactions in Operational Research 21, 103–126.

Juan, A.A., Pascual, I., Guimarans, D., Barrios, B., 2015. Combining biased randomization with iterated local search for solving the multidepot vehicle routing problem. International Transactions in Operational Research 22, 647–667.

Panadero, J., de Armas, J., Currie, C.S., Juan, A.A., 2017. A simheuristic approach for the stochastic team orienteering problem, in: Chan et al., W.K.V. (Ed.), Proceedings of the 2017 Winter Simulation Conference, IEEE, Piscataway, New Jersey. pp. 3208–3217.

Poggi, M., Henrique, V., Uchoa, E., 2010. The team orienteering problem: formulations and branch-cut and price, in: Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, pp. 464–474.

Reyes-Rubiano, L.S., Ospina-Trujillo, C.F., Faulin, J., Mozos, J.M., Panadero, J., Juan, A.A., 2018. The team orienteering problem with stochastic service times and driving-range limitations: a simheuristic approach, in: 2018 Winter Simulation Conference (WSC), IEEE. pp. 3025–3035.

Royset, J., 2009. Optimized routing of unmanned aerial systems for the interdiction of improvised explosive devices. Military Operations Research 14, 1–32.

Souffriau, W., Vansteenwegen, P., Oudheusden, D.V., 2013. The multi-constraint team orienteering problem with multiple time windows. Seventh Triennial Symposium on Transportation Analysis 47, 717–720.

Vansteenwegen, P., Souffriau, W., Oudheusden, D.V., 2011. The orienteering problem: a survey. European Journal of Operational Research 209, 1–10.