Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

A SIMHEURISTIC-LEARNHEURISTIC ALGORITHM FOR THE STOCHASTIC TEAM ORIENTEERING PROBLEM WITH DYNAMIC REWARDS

Christopher Bayliss
Pedro J. Copado-Mendez
Javier Panadero
Angel A. Juan
Leandro do C. Martins

IN3 – Computer Science Department
Universitat Oberta de Catalunya
Av. Carl Friedrich Gauss, 5
Castelldefels, 08860, SPAIN

ABSTRACT

In this paper, we consider the stochastic team orienteering problem with dynamic rewards (STOPDR) and stochastic travel times. In the STOPDR, the goal is to generate routes for a fixed set of vehicles such that the sum of the rewards collected is maximized while ensuring that nodes are visited before a fixed time limit expires. The rewards associated with each node are dependent upon the times at which they are visited. Also, the dynamic reward values have to be learnt from simulation experiments during the search process. To solve this problem, we propose a biased-randomized learnheuristic (BRLH), which integrates a learning module and a simulation model. Randomization is important for generating a wide variety of solutions that capture the trade-off between reward and reliability. A series of computational experiments are carried out in order to analyze the performance of our BRLH approach.

1 INTRODUCTION

The team orienteering problem (TOP) was first studied by Chao et al. (1996). It consists of generating time-constrained routes through a graph, for a fixed-size fleet of m vehicles, such that the rewards collected from node visits is maximized. Each of the m vehicles' tours begins and ends at a depot node. The stochastic TOP (STOP) is an extension of the TOP where travel times between nodes or node rewards are uncertain. In this work, only the rewards collected within a limited amount of time count. The STOP introduces the need to consider not only reward maximization, but also solution reliability. In this work, a solution is deemed infeasible if any of the tours is not completed within the specified time limit. The reliability is defined as the probability that a solution can be completed without failures. This work considers a STOP with dynamic rewards (STOPDR), in which the rewards associated with nodes have a static component and also a dynamic component. The dynamic component accounts for: *(i)* bonuses for visiting nodes early in a route; and *(ii)* penalties for nodes visited late in routes. The process through which bonus values and penalty values are generated is a hidden and unknown process. Hence, it has to be learnt from simulation observations. We consider the case in which bonuses are achieved for nodes visited as the first node in a route, while penalties are applied for nodes visited last in routes. The bonus and penalty values for nodes, when applicable, are assumed to follow an unknown statistical distribution. When solving the STOPDR, the challenge is to learn the bonuses and penalties associated with each node from the simulation testing of candidate solutions, while – at the same time – trying to maximize the total reward and also ensure that the solutions have a good level of reliability under stochastic travels times. Figure 1 provides an illustrative diagram depicting some of the main features of a STOPDR, including: stochastic edge traversal times;

bonuses for first nodes in routes; penalties for last nodes in routes; and no rewards at all for nodes visited after the time limit.

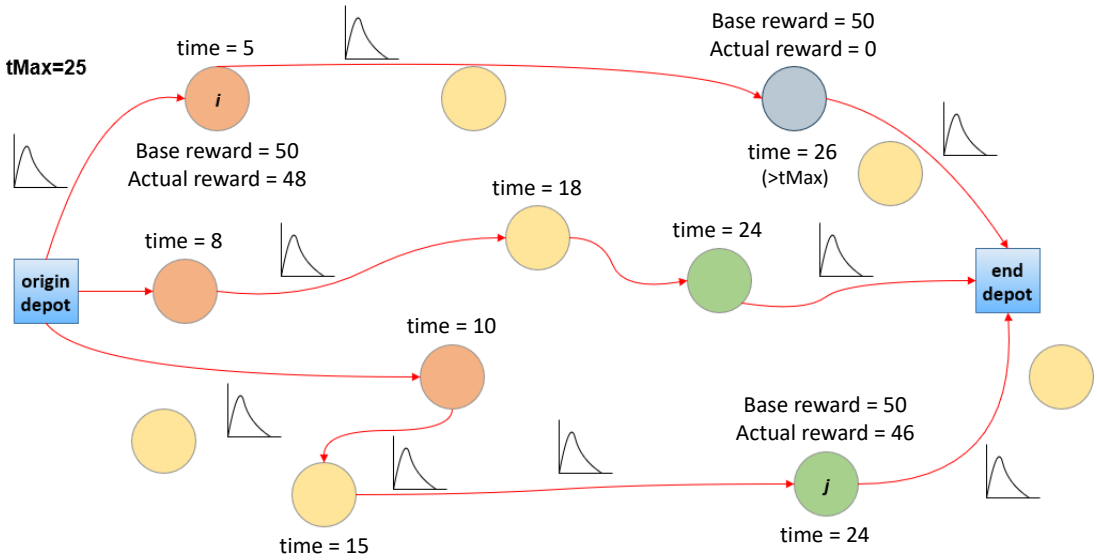

Figure 1: An illustrative example of the STOP with dynamic rewards.

In order to meet these requirements, we propose a biased randomization learnheuristic algorithm (BRLH) (Grasas et al. 2017). In general, learnheuristics integrate metaheuristics with machine learning algorithms for the purpose of solving combinatorial optimization problems with inherent learning problems, such as those within dynamic inputs generated through an unknown process (Calvet et al. 2017; Arnau et al. 2018). Apart from proposing a learnheuristic, we propose an algorithm that integrates biased randomization with simulation and computational learning. The biased randomization algorithm plays the role of the optimization algorithm. Simulation deals with the stochastic parameters providing observations which, through a simple averaging learning mechanism, play the role of the machine learning algorithm. By combining simulation methods with biased-randomized heuristics or metaheuristics, the resulting simheuristic algorithm aims at extending the capabilities of classical biased-randomization algorithms in the modeling and uncertainty dimensions, and as a component of a learnheuristic. The reasons for this are as follows: *(i)* the parameterized randomization of a biased-randomized constructive solution procedure provides an ideal mechanism for addressing simulation, learning, and optimization requirements of the learnheuristic, since the level of randomness can be increased to increase the diversity of the input data used for learning to predict the values of the dynamic rewards; and *(ii)* it is a simple matter to incorporate and use the predictions of the learning mechanism within a constructive solution approach. The level of randomness introduced into the greedy constructive solution procedure has the additional role of generating a wide range of solutions with different average reward and reliability characteristics. We identify the set of non-dominated solutions as candidate solutions to be considered by the decision-maker.

The rest of the paper is structured as follows: Section 2 reviews related work; Section 3 provides a formal description of the STOPDR; Section 4 presents the biased-randomized heuristic, which is then extended into a simheuristic in Section 5 and finally to a learnheuristic in Section 6; Section 7 provides experimental results based on a set of benchmark instances. Section 8 reports the main conclusions and future research.

2 LITERATURE REVIEW

The single-vehicle orienteering problem (OP) was introduced by Golden et al. (1987). Here, the authors considered the deterministic version of the problem, in which a vehicle chooses the set of nodes to visit as well as the visiting order during a specified time interval. This is a very general problem with many applications, including the tourist trip design problem (Souffriau et al. 2008). Gunawan et al. (2016) provide an excellent review of the OP and its variants.

In this paper, we focus on the TOP, which is a multi-vehicle OP. The TOP was first introduced in Chao et al. (1996), who extended their methodology from the single-vehicle problem to consider multiple vehicles aiming to maximize their combined reward from visiting a selection of points within a given time interval. Other varieties include the time-dependent OP (Verbeeck et al. 2014), the TOP with time windows (Lin and Yu 2012), and the multi-modal TOP with time windows (Yu et al. 2017), just to name a few. Recent methods used to solve the deterministic version of the problem have included particle swarm optimization (Dang et al. 2013), simulated annealing (Lin 2013), genetic algorithms (Ferreira et al. 2014), a Pareto mimic algorithm (Ke et al. 2015), or branch-and-cut-and-price algorithms (Keshtkaran et al. 2015).

Allowing for uncertainty in rewards, travel and service times widen the applicability of the problem. Hence, Ilhan et al. (2008) were the first to incorporate stochasticity into the OP. Studies on the stochastic TOP are much more recent. 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). Panadero et al. (2017) tackle a very similar problem to that considered in this work with the exception that stochastic rewards are not learnt from observations as the algorithm progresses. However, to the best of our knowledge, this is the first paper proposing the use of learnheuristic algorithms (Calvet et al. 2016) to deal with the dynamic version of the STOP.

3 A FORMAL DESCRIPTION OF THE STOP WITH POSITION-DEPENDENT REWARDS

In a team orienteering problem, 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 nodes that can be visited is denoted $N = \{1, 2, \ldots, n\}$. Vehicles travel along the edges, denoted A, connecting all nodes with each other. The STOP is, therefore, set within an undirected graph $G = (N, A)$. All vehicle tours begin at node 1, the origin depot, and end at node n, the end depot. Each non-depot node can be visited at most once and by only one vehicle. In the STOPDR, a reward $u_i \geq 0$ is received for visiting node i. In our dynamic version, if node i is the first non-depot node in a vehicle's tour, then a bonus B_i is added to the reward u_i for visiting that node. It is assumed that B_i follows an unknown statistical distribution. If node i is the last non-depot node in a vehicle's tour, then a penalty $P_i <= u_i$ is subtracted from the reward u_i, with P_i also following an unknown statistical distribution. The values of B_i and P_i $\forall i \in N$ have to be learnt from observations of the real process or from a detailed simulation of that process. Traversing an edge $e = (i, j) \in A$ has a stochastic travel time, $T_{ij} > 0$. In this work, edge traversal times follow log-normal probability distributions

$$T_{ij} \in lognormal\left(\mu_{ij}, \sigma_{ij}^2\right) \forall e = (i, j) \in A,$$

where μ_{ij} and σ_{ij}^2 are the mean and variance of the log transformed edge traversal times, respectively. T_{ij} can be expressed as $e^{\mu_{ij} + \sigma_{ij} Z}$,

where Z is a standard normal variable.

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 0 (origin depot) and arriving at node n (end depot). The objective function is the expected value of the sum of collected rewards, which needs to be maximized without exceeding the threshold value for each route. In a more formal way, the objective function can be written as:

$$\max E\left[\sum_{m \in M} \sum_{(i,j) \in A} I_{mi}\left(u_i + F\left(p_{mi}, l_m\right) B_i + L\left(p_{mi}, l_m\right) P_i\right) \cdot x_{ij}^m\right] \tag{1}$$

where: $A = \{(i,j)/i,j \in N, i \neq j\}$ 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. I_{mi} takes the value 1 if node i is visited before the time t_{max}, since rewards can only be collected from nodes visited before time runs out. It takes the value 0 otherwise. The function $F(p_{mi}, l_m)$ is an indicator function that returns 1 if node i is the first non-depot node in a vehicle's tour. Similarly, the function $L(p_{mi}, l_m)$ indicates whether or not node i is the last non-depot node in a vehicle's tour. The indicator function input argument p_{mi} gives the position number of node i in vehicle m's tour, while l_m gives the length of vehicle m's tour.

Thus, the objective function calculates the total expected score – including bonuses and penalties – taking stochastic travel times into account. The constraints are given below, with some explanations provided. First, there is a threshold time to complete each route, i.e.:

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

Additionally, each node is visited at most once during the route, each route starts at the origin depot (node 1) and arrives at the end depot (node n), and, finally, each vehicle leaves each node it visits, except for the end depot.

4 A BIASED-RANDOMIZED HEURISTIC FOR THE STATIC TOP

In this section, a biased randomization heuristic (BRH) is presented for the non-dynamic and non-stochastic TOP. Biased-randomization algorithms (Grasas et al. 2017) are based on introducing randomness into a greedy constructive algorithm. Constructive algorithms are based on generating solutions by greedily and sequentially selecting the elements that are to be included in a solution. In a biased-randomization, candidate decisions are sorted in a list descending in the objective value of each decision. Decreasing probabilities of selection are assigned to the elements in this list. Using such a skewed probability distribution for the selection of solution elements introduces randomness into a greedy constructive algorithm in a way that preserves the logic underlying the greedy constructive algorithm while allowing for the generation of a vast array of alternative solutions as well as solutions which improve upon the solution generated by the base greedy constructive procedure alone. Such techniques have enjoyed great success improving the performance of classical heuristics, both in scheduling applications (Gonzalez-Neira et al. 2017) in addition to vehicle routing ones (Faulin and Juan 2008; Dominguez et al. 2016). The base constructive heuristic used in this paper is composed of the following phases:

1. Firstly, an initial dummy solution is generated. This is composed of a route for each non-depot node starting at the start depot and ending at the end depot (Figure 2a).
2. A pair of routes is selected for merging. A merge operation is based on trying to add one route onto the end of another (Figure 2b), introducing the possibility of avoiding unnecessary trips to and from the end and start depot, respectively. Possible merge operations are sorted according to a weighted sum of the saving in the travel cost, i.e., travel distance, and rewards associated with nodes either end of the merging edge. The saving associated with edge (i,j) is as follows:

$$saving_{ij} = \alpha(d_{in} + d_{0j} - d_{ij}) + (1-\alpha)(u_i + u_j) \tag{3}$$

where d_{ij} represents the distance to traverse edge (i,j). Equation (3) gives a weight of α to the travel cost saving, and a weight of $(1-\alpha)$ to the rewards associated with the newly added route edge. This reflects the main goals when solving a TOP, i.e., those of generating efficient and high-value vehicle routes. A skewed probability distribution is applied to the sorted list of merge operations, thus introducing "biased randomness" into the base greedy constructive algorithm. In particular, a geometric distribution is used in this work. The equation $randI = Mod\left(\left\lfloor \frac{log(uniRand(0,1))}{log(1-\beta)} \right\rfloor, |C|\right)$ is used to randomly select a decision index ($randI$) from the list, where $|C|$ is the candidate decision

list length, *uniRand* is a uniform random input between 0 and 1 and β is the parameter which controls the level of greediness and randomness in a biased-randomized algorithm. Whenever $\beta = 1$, we have full greediness. On the contrary, whenever $\beta = 0$, we obtain full randomness. When $0 < \beta < 1$, we get trade-offs between the two.

3. Repeat *Step 2*. until no more merges are possible.
4. From those generated by following *Steps 1–3*, select the m routes that maximize the total reward.

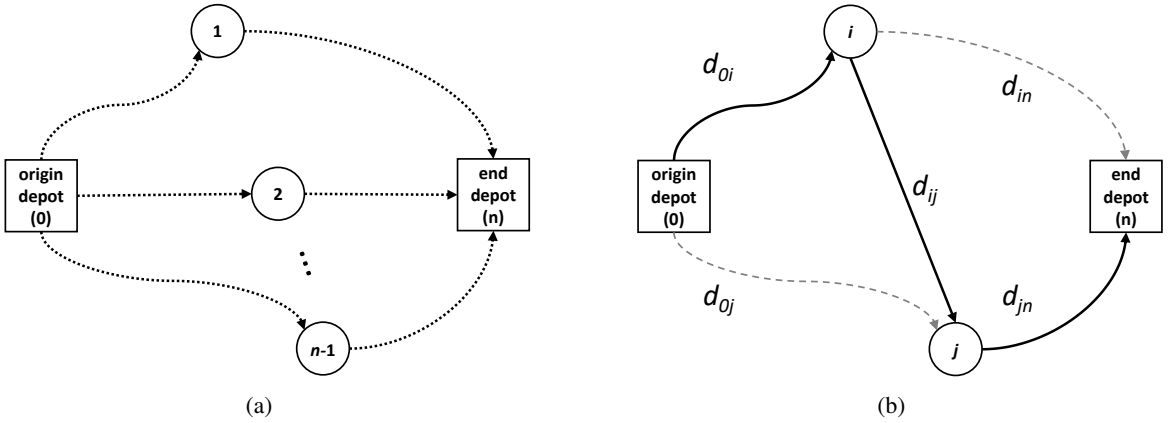

Figure 2: Merging of routes procedure; (a) dummy solution and (b) distance-based savings.

By selecting $0 < \beta < 1$ repeating steps 1–4 for a specified number of iterations or amount of time, the biased randomization algorithm can generate good quality solutions, which achieve different average reward and reliability trade offs.

5 EXTENDING OUR BRH TO A SIMHEURISTIC FOR THE STATIC STOP

Despite of being able to provide competitive solutions in short computing times, the described approach, in its original formulation, is not properly able to deal with stochasticity and uncertainty, as required by the STOP. Therefore, the BRH is extended to a simheuristic algorithm (BRSH) for solving the STOP, where solutions offering a good trade-off between total expected reward and variability or risk might be desirable.

As mentioned, simheuristic algorithms combine simulation with metaheuristics to solve stochastic variants of combinatorial optimization problems (Juan et al. 2018). In our case, we combined a multi-start framework with Monte Carlo simulation in order to cope with the STOP, which differs from TOP by considering that the travel time between any pair of nodes is stochastic (Section 3). In order to simulate the stochastic value of a solution, Monte Carlo sampling is performed to estimate the bonuses for first nodes and penalties for last nodes in all routes. Section 7 provides details of the bonus and penalty distributions used for each node. Note that the proposed approach works with any node bonus and penalty distributions.

Our simheuristic approach follows the conventional structure of this procedure (Juan et al. 2015). This simheuristic is controlled by one parameter, which refers to the number of simulations on solutions. Firstly, an initial solution is generated by our BRH, and simulation is conducted at every solution, where the deterministic travel times are replaced by the stochastic ones in order to generate its average stochastic cost and its reliability rate. This reliability rate corresponds to the success of the deterministic solution under uncertainty, i.e., whether this solution is still feasible under such scenarios or not.

6 EXTENDING OUR BRSH TO A LEARNHEURISTIC FOR THE DYNAMIC STOP

In this section, we incorporate the BRSH introduced in Section 5 within a learnheuristic framework for solving the STOPDR. We begin by describing the modifications to the savings calculations, which are required for taking dynamic rewards into account. Performing a merge operation based on joining a route whose last non-depot node is node i with a route whose first non-depot edge is node j (Figure 2) has the effect that the penalty from visiting node i last in a route is avoided, while the bonus from visiting node j first in a route is lost. Given that we denote the current best estimate for the bonus associated with node j as $\hat{b}_j$ and the current best estimate for the penalty associated with node i as $\hat{p}_i$, the savings calculation of Equation (3) is replaced with the following one for the case of dynamic rewards:

$$saving_{ij} = \alpha\left(d_{in} + d_{0j} - d_{ij}\right) + (1-\alpha)\left(u_i + u_j + \hat{p}_i - \hat{b}_j\right) \tag{4}$$

Equation (4) acknowledges that merging the route whose last non-depot node is i with the route whose first non-depot node is j results not only in the avoidance of the penalty associated with node i but also the loss of the bonus associated with node j.

Although learnheuristics can be viewed as an extension of heuristics, our learnheuristic integrates heuristics with simulation for solving stochastic combinatorial optimization problems (De Armas et al. 2017) and incorporates a learning component into the frameworks to deal with dynamic inputs.

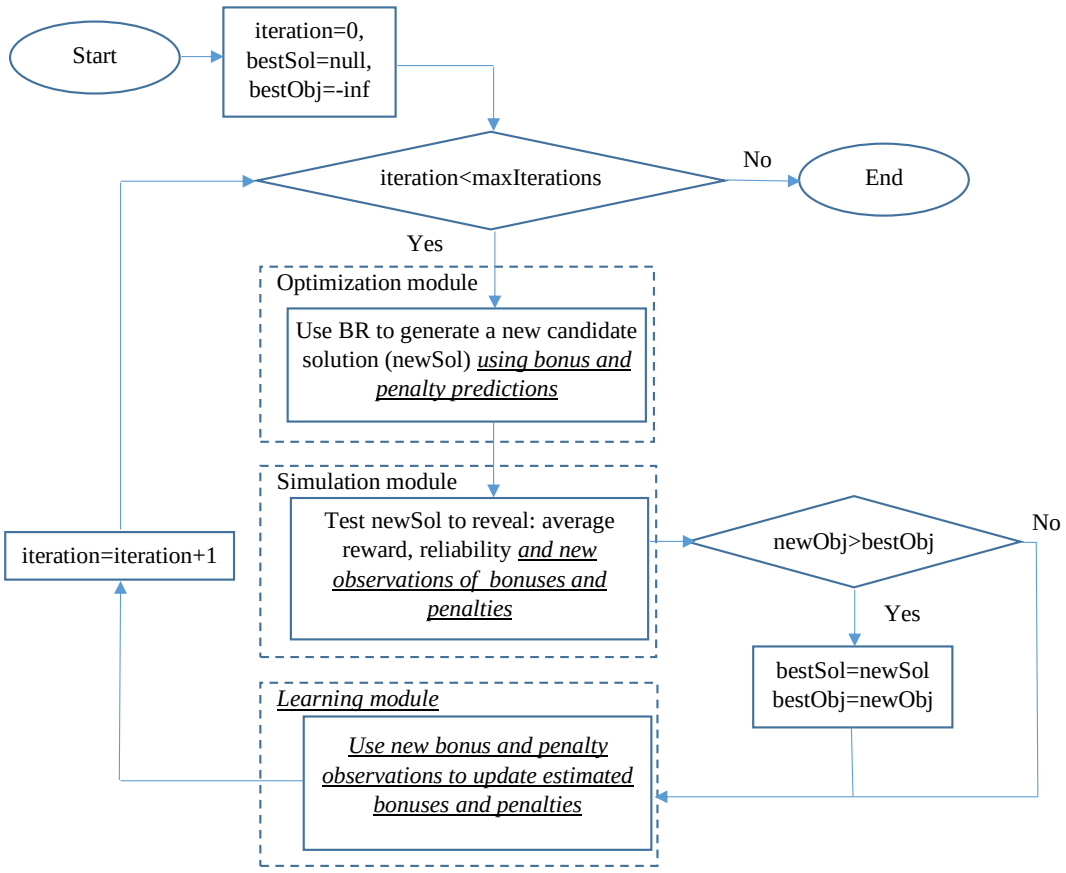

Figure 3: Flowchart representation of the proposed BRLH solution approach.

Figure 3 outlines the proposed learnheuristic framework for solving the STOPDR. The parts both underlined and in italics are the steps that our learnheuristic solution approach has in addition to the steps

involved in a typical simheuristic approach for solving the stochastic but static version of the problem. Notice that a learnheuristic solution approach is based upon an iterative framework that takes advantage of the observations provided by the simulation module. In each iteration of a learnheuristic, the optimization module (BRH, in this case) is used to generate a new candidate solution. Following this, a simulation model is used to test that solution in order to reveal the average reward attained, its level of reliability, and also to provide new bonus and policy observations. Afterwards, the best overall solution is updated if a new best solution has been identified. Then, the new bonus and penalty observations are used to update the estimates of the bonuses and penalties associated with each node. These steps are repeated for a specified amount of time or number of iterations. Given new observations of the unknown and stochastic bonus and penalty values for a node i, denoted $\bar{b}_i$ and $\bar{p}_i$, the best estimates for the bonus and penalty associated with node i are updated according to $\hat{b}_i \leftarrow \frac{\hat{b}_i N_i^b + \bar{b}_i}{N_i^b+1}$, $N_i^b \leftarrow N_i^b + 1$, $\hat{p}_i \leftarrow \frac{\hat{p}_i N_i^p + \bar{p}_i}{N_i^p+1}$ and $N_i^p \leftarrow N_i^p + 1$. Here, N_i^b is the number of observations of bonuses attained from visits to node i, N_i^p is the number of observations of penalties attained from visiting node i last in a route. In Section 7, we demonstrate that this simple learning mechanism is effective and allows our BRLH algorithm to generate good solutions that take advantage of the dynamic rewards, thus proving the concept of learnheuristics.

7 COMPUTATIONAL EXPERIMENTS

Our BRLH algorithm was implemented in Java code and run sequentially on a personal computer with 16 GB of RAM and an Intel Core i7-8750H at 2.2 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 is available online (KU Leuven 2020). Each instance involves a fleet size, number of nodes, customer profits, and maximum route duration t_{max}. The stochastic travel times for each edge of these deterministic instances follow a log-normal distribution with the deterministic travel time as the mean, and a variance of 0.05 times of this mean travel time. The mean and standard deviation parameters of the log transformed edge traversal times are recovered from the following two equations:

$$\mu_{ij} = \ln(E[T_{ij}]) - \frac{1}{2}\ln\left(1 + \frac{Var[T_{ij}]}{E[T_{ij}]^2}\right)$$

$$\sigma_{ij} = \left|\sqrt{\ln\left(1 + \frac{Var[T_{ij}]}{E[T_{ij}]^2}\right)}\right|$$

The bonus values gained from the nodes visited first in routes and penalties applied for nodes visited last in routes both follow a triangular distribution with minimum, mode, and maximum parameters generated by: *(i)* selecting and sorting three uniformly distributed random numbers in the interval $[0,1]$; and *(ii)* multiplying them by the base reward of the given node. We begin by analyzing the impact of the choice of the β parameter of the geometric distribution used in the BRLH algorithm and its effect on the average reward of the solutions generated by that algorithm. Figure 4 shows that low values of β have the effect of increasing the diversity of the solutions generated by the BRLH algorithm, while higher values decrease diversity. Intermediate values of β, such as 0.3, achieve an ideal trade-off between diversity and exploration, achieving the highest average reward solution as well as a diverse array of other solutions.

We now consider the trade-off between average reward and reliability. A solution is considered to be *non-dominated* if no other solutions have both a higher reliability or higher average reward. Then, for instance *p.2.2.b*, the set of non-dominated solutions generated from the BRLH algorithm has been computed using $\beta = 0.1$ and $\beta = 1$. Figure 5 shows that the β randomization parameter of the proposed BRLH algorithm is useful for generating solutions with a wide variety of characteristics. It was found that for a low β value it was possible to generate solutions capturing all aspects of the trade-off between reward and reliability, including solutions with very high reliability but low average reward and solutions with high average reward but low reliability. Using a value of $\beta = 1$ leads to a non-dominated set of

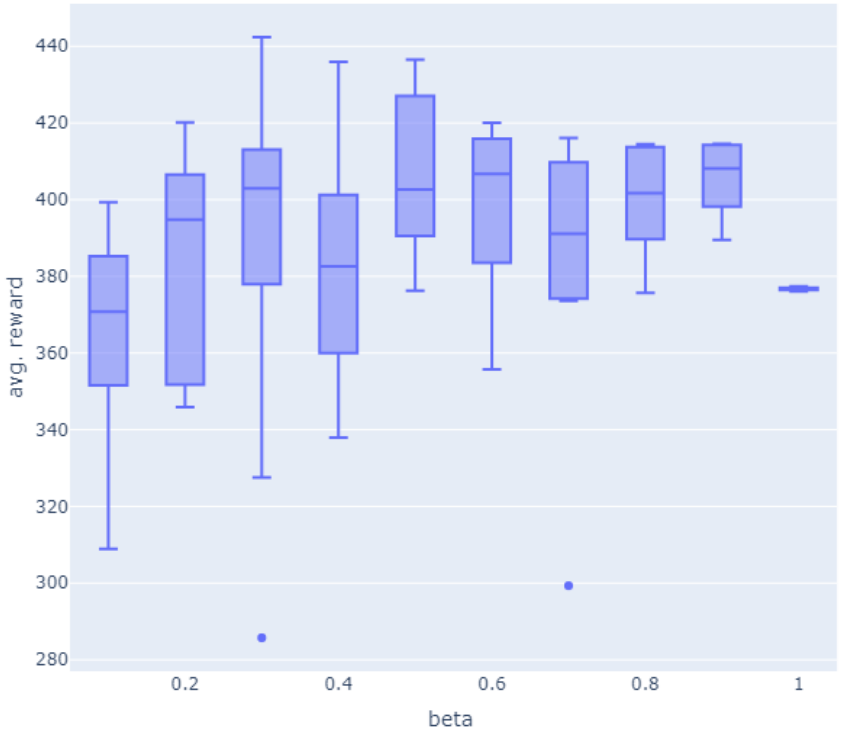

Figure 4: The effect of β on average reward and solution diversity.

solutions that lie within a very narrow range of average reward and reliability values. Table 1 shows the performance of our proposed BRLH algorithm, with and without learning, in terms of average reward, reliability, and solution speed – expressed in terms of the number of iterations of the BRLH that can be performed within 10 seconds of computing time. For a sample of eight instances converted into stochastic

Table 1: Comparison of the BRLH with and without learning for standard TOP instances.

	With learning			Without learning		
Instance	average reward	reliability	Iterations in 10 seconds	average reward	reliability	Iterations in 10 seconds
p1.4.j	98.36	0.19	2550	90.62	0.15	4648
p2.2.c	142.98	0.56	5937	131.11	0.41	6281
p2.2.g	224.63	0.37	3349	210.38	0.88	3961
p2.2.i	252.32	0.47	3368	238.73	0.63	3674
p2.3.e	146.76	0.40	4469	139.24	0.48	4878
p3.4.h	264.36	0.20	1765	236.84	0.32	3490
p3.4.e	170.12	0.33	3184	156.64	0.47	4746
p3.4.f	226.32	0.38	2226	200.66	0.58	3807
average	190.73	0.36	3356.00	175.53	0.49	4435.63

instances with dynamic rewards, our proposed BRLH algorithm is able to successfully learn and exploit information about bonuses and penalties that apply for nodes in routes that are joined by an edge to a depot node. In both cases, the best average reward solutions require a compromise with regard to the reliability of the solutions. The results for the number of iterations that can be performed within 10 seconds reflect the fact that, marginally, more computing time is required when learning is used. This makes sense since

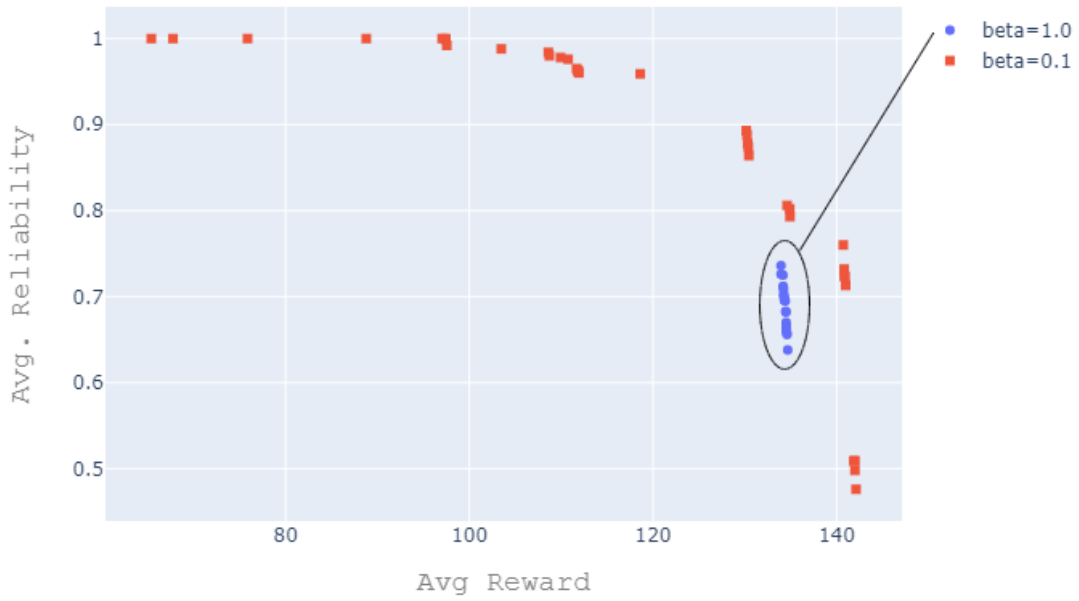

Figure 5: Non-dominated solutions generated by BRLH using $\beta = 0.1$ and $\beta = 1$.

the savings lists have to recalculated after each merge operation, when estimated bonus and penalty values are taken into account within the BRLH. Finally, we analyze the simultaneous learning and optimization

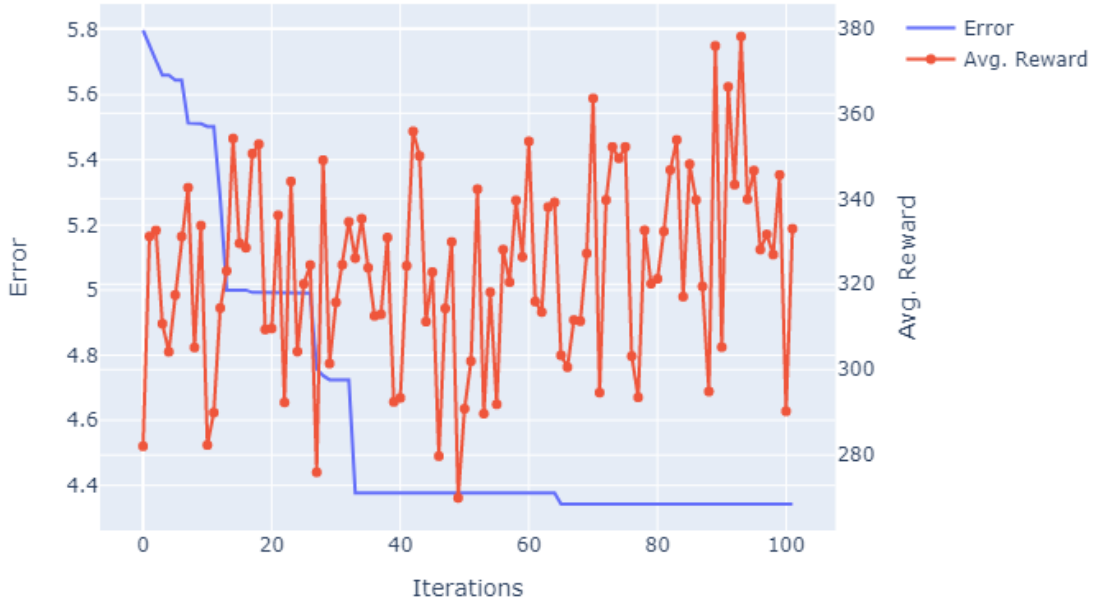

Figure 6: Average reward and error over the course of the BRLH algorithm.

which occurs in our BRLH algorithm. Figure 6 shows how the accuracy of the predicted values for the bonus and penalty values associated with each node improves as more iterations of the proposed BRLH algorithm are performed. The error values reported are root mean squared errors calculated by comparing the estimated parameter values with the actual mean values of the bonuses and penalties, which are never visible to the BRLH algorithm. The BRLH algorithm also displays a weak positive correlation between

iteration number and average reward. This suggests that the information learnt by the simple learning mechanism is being used effectively.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we have considered a stochastic team orienteering problem with dynamic rewards. We have demonstrated how a simple learning mechanism can be incorporated in a biased-randomization algorithm to exploit information previously learnt about the unknown bonus and penalty process from simulation tests of candidate solutions. Our experimental results demonstrated how the randomization parameter of a biased randomization algorithm is useful for ensuring that dynamic reward information is learnt about all nodes as well as being useful for generating a wide variety of solutions that capture the trade-off between average reward and reliability.

In future work, we plan to increase the complexity of the process used to generate dynamic rewards, thus motivating the consideration of more-complex learning algorithms, such as neural networks. For instance, bonuses and rewards can apply for visited nodes be dependent upon the exact time at which that node is visited, or which node was visited immediately prior to the given node.

ACKNOWLEDGMENTS

This work has been partially funded by the IoF2020 project of the European Union (grant agreement no. 731884), the Spanish Ministry of Science, Innovation, and Universities (RED2018-102642-T), and the Erasmus+ Program (2019-I-ES01-KA103-062602).

REFERENCES

Arnau, Q., A. A. Juan, and I. Serra. 2018. "On the Use of Learnheuristics in Vehicle Routing Optimization Problems with Dynamic Inputs". *Algorithms* 11(12):208.

Calvet, L., J. de Armas, D. Masip, and A. A. Juan. 2017. "Learnheuristics: Hybridizing Metaheuristics with Machine Learning for Optimization with Dynamic Inputs". *Open Mathematics* 15(1):261–280.

Calvet, L., A. Ferrer, M. I. Gomes, A. A. Juan, and D. Masip. 2016. "Combining Statistical Learning with Metaheuristics for the Multi-Depot Vehicle Routing Problem with Market Segmentation". *Computers & Industrial Engineering* 94:93–104.

Chao, I.-M., B. L. Golden, and E. A. Wasil. 1996. "The Team Orienteering Problem". *European Journal of Operational Research* 88(3):464–474.

Dang, D.-C., R. N. Guibadj, and A. Moukrim. 2013. "An Effective PSO-Inspired Algorithm for the Team Orienteering Problem". *European Journal of Operational Research* 229(2):332–344.

De Armas, J., A. A. Juan, J. M. Marquès, and J. P. Pedroso. 2017. "Solving the Deterministic and Stochastic Uncapacitated Facility Location Problem: From a Heuristic to a Simheuristic". *Journal of the Operational Research Society* 68(10):1161–1176.

Dominguez, O., D. Guimarans, A. A. Juan, and I. de la Nuez. 2016. "A Biased-Randomised Large Neighbourhood Search for the Two-Dimensional Vehicle Routing Problem with Backhauls". *European Journal of Operational Research* 255(2):442–462.

Faulin, J., and A. A. Juan. 2008. "The ALGACEA-1 Method for the Capacitated Vehicle Routing Problem". *International Transactions in Operational Research* 15(5):599–621.

Ferreira, J., A. Quintas, and J. Oliveira. 2014. "Solving the Team Orienteering Problem: Developing a Solution Tool Using a Genetic Algorithm Approach". In *Soft Computing in Industrial Applications*, edited by V. Snášel, P. Krömer, M. Koeppen, and G. Schaefer, 365–375. Cham: Springer.

Golden, B., L. Levy, and R. Vohra. 1987. "The Orienteering Problem". *Naval Research Logistics* 34:307–318.

Gonzalez-Neira, E. M., D. Ferone, S. Hatami, and A. A. Juan. 2017. "A Biased-Randomized Simheuristic for the Distributed Assembly Permutation Flowshop Problem with Stochastic Processing Times". *Simulation Modelling Practice and Theory* 79:23–36.

Grasas, A., A. A. Juan, J. Faulin, J. de Armas, and H. Ramalhinho. 2017. "Biased Randomization of Heuristics Using Skewed Probability Distributions: A Survey and Some Applications". *Computers & Industrial Engineering* 110:216–228.

Gunawan, A., H. C. Lau, and P. Vansteenwegen. 2016. "Orienteering Problem: A Survey of Recent Variants, Solution Approaches and Applications". *European Journal of Operational Research* 255(2):315–332.

Ilhan, T., S. Iravani, and M. Daskin. 2008. "The Orienteering Problem with Stochastic Profits". *IIE Transactions* 40:406–421.

Juan, A. A., J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. 2015. "A Review of Simheuristics: Extending Metaheuristics to Deal with Stochastic Combinatorial Optimization Problems". *Operations Research Perspectives* 2:62–72.

Juan, A. A., W. D. Kelton, C. S. Currie, and J. Faulin. 2018. "Simheuristics Applications: Dealing with Uncertainty in Logistics, Transportation, and other Supply Chain Areas". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3048–3059. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Ke, L., L. Zhai, J. Li, and F. Chan. 2015. "Pareto Mimic Algorithm: An Approach to the Team Orienteering Problem". *Omega* 61:155–166.

Keshtkaran, M., K. Ziarati, A. Bettinelli, and D. Vigo. 2015. "Enhanced Exact Solution Methods for the Team Orienteering Problem". *International Journal of Production Research* 54:591–601.

KU Leuven 2020. "Orienteering Problem Instances". https://www.mech.kuleuven.be/en/cib/op/instances, accessed 31st July 2020.

Lin, S. 2013. "Solving the Team Orienteering Problem Using Effective Multi-Start Simulated Annealing". *Applied Soft Computing* 13:1064–1073.

Lin, S.-W., and V. F. Yu. 2012. "A Simulated Annealing Heuristic for the Team Orienteering Problem with Time Windows". *European Journal of Operational Research* 217(1):94–107.

Panadero, J., J. de Armas, C. S. Currie, and A. A. Juan. 2017. "A Simheuristic Approach for the Stochastic Team Orienteering Problem". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 3208–3217. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Reyes-Rubiano, L. S., C. F. Ospina-Trujillo, J. Faulin, J. M. Mozos, J. Panadero, and A. A. Juan. 2018. "The Team Orienteering Problem with Stochastic Service Times and Driving-Range Limitations: A Simheuristic Approach". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3025–3035. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Souffriau, W., P. Vansteenwegen, J. Vertommen, G. V. Berghe, and D. V. Oudheusden. 2008. "A Personalized Tourist Trip Design Algorithm For Mobile Tourist Guides". *Applied Artificial Intelligence* 22(10):964–985.

Verbeeck, C., K. Sörensen, E.-H. Aghezzaf, and P. Vansteenwegen. 2014. "A Fast Solution Method for the Time-Dependent Orienteering Problem". *European Journal of Operational Research* 236(2):419–432.

Yu, V. F., P. Jewpanya, C.-J. Ting, and A. P. Redi. 2017. "Two-Level Particle Swarm Optimization for the Multi-Modal Team Orienteering Problem with Time Windows". *Applied Soft Computing* 61:1022–1040.

AUTHOR BIOGRAPHIES

CHRISTOPHER BAYLISS is a post-doctoral researcher in the ICSO group at the IN3 – Universitat Oberta de Catalunya (UOC). His main research interests include metaheuristics, simulation optimization, revenue management, packing problems, airline scheduling, and logistics optimization. His email address is cbayliss@uoc.edu.

PEDRO J. COPADO-MENDEZ is a post-doctoral researcher in the ICSO group at the IN3 – UOC. He completed his PhD at the University Rovira i Virgili (Spain). His main research interests include metaheuristics, hybrid heuristics and their applications. His email address is pcopadom@uoc.edu.

JAVIER PANADERO is an assistant professor of Simulation and High Performance Computing in the Computer Science Department at the UOC (Spain). He holds a PhD in Computer Science. His major research areas are: high-performance computing, modeling and analysis of parallel applications, and simheuristics. His website address is http://www.javierpanadero.com and his email address is jpanaderom@uoc.edu.

ANGEL A. JUAN is a Full Professor of Operations Research & Industrial Engineering in the Computer Science Department at the UOC (Barcelona, Spain). He is also the Director of the ICSO research group at the Internet Interdisciplinary Institute and Lecturer at the Euncet Business School. Dr. Juan holds a PhD in Industrial Engineering and a MSc in Mathematics. He completed a predoctoral internship at Harvard University and postdoctoral internships at Massachusetts Institute of Technology and Georgia Institute of Technology. His main research interests include applications of simheuristics and learnheuristics in computational logistics and finance as well as mathematical e-learning. He has published more than 85 articles in JCR-indexed journals and over 200 papers indexed in Scopus. His website address is http://ajuanp.wordpress.com and his email address is ajuanp@uoc.edu.

LEANDRO DO C. MARTINS is a PhD candidate and researcher in the ICSO group at the IN3 – UOC. He holds a BSc and MSc degrees in Computer Science from the Federal University of Ouro Preto (Brazil). His main research interests include metaheuristics, biased-randomized algorithms, and vehicle routing. His email address is leandrocm@uoc.edu.