Agile Optimization for Routing Unmanned Aerial Vehicles under Uncertainty

Javier Panadero*, Alfons Freixes*†,Jose M. Mozos*, and Angel A. Juan*†
*IN3 - Computer Science Dept.
Universitat Oberta de Catalunya
Barcelona ,08018, Spain
{jpanaderom, jmozosr, ajuanp}@uoc.edu
†Euncet Business School
08225 Terrassa, Spain
afreixes@euncet.es

***Abstract*—The use of unmanned aerial vehicles (UAVs) is gaining popularity in contexts such as smart cities, city logistics, humanitarian logistics, natural disasters, or military operations. One optimization challenge directly related to the use of UAVs is the so-called team orienteering problem (TOP). In a TOP, each customer can either be visited only once by a single vehicle or not visited at all. Visiting each customer has associated a predefined reward, and the driving ranges of vehicles are typically limited by the duration of electric batteries. Due to the latter constraint, it is usual that not all customers can be visited. The main goal is then to find a set of open tours that maximizes the total collected reward without exceeding the fleet capacity nor the driving range limitation. In this paper, we consider the stochastic version of the problem, in which travel times are modeled as random variables following theoretical probability distributions. To solve this stochastic version of the TOP, a simheuristic algorithm combining a biased-randomized heuristic with simulation techniques is proposed. One of the main goals of our approach is to provide an 'agile' optimization methodology, i.e., one lightweight algorithm that can be easily implemented in real-life scenarios under uncertainty and, at the same time, can provide solutions in real-time (just a few seconds or even less).**

***Index Terms*—Metaheuristics, simulation, team orienteering problem, unmanned aerial vehicles, agile optimization.**

I. Introduction

The term 'smart city' refers to a series of urban systems and domains that are interconnected, via information technologies (ITs), with the purpose of optimizing their operations and management [1]. Smart cities represent a multidisciplinary research field that is under a continuous updating process driven by urban, social, and technology evolution [2]. These advances are generating new services and products for citizens, which also arises new challenges in data gathering, data analytics, and efficient decision making.

Unmanned aerial vehicles (UAVs) are known for autonomous operation and mobility. Though there are first studies available [3] [4], the usage of UAVs in smart cities is not fully explored yet. Until today they were mainly used for and effectively integrated in military activities, surveillance, security, precision agriculture and goods and services deliveries [5] [6], while there are still concerns on their effective and reliable implementation in smart cities. With the use of a reliable and intelligent transportation system (ITS), it would be possible to replace human road support teams by a set of UAVs overflying highways to monitor possible traffic violations and accidents, or for providing specific information to other transport users. Technological support for interacting with other transport users is given by the ITS, dedicating a short-range communication interface to UAV, which transmit information either vehicle-to-vehicle or vehicle-to-infrastructure. Using wireless links when being close to each other, connected UAV impose improved road safety and traffic efficiency. The execution of these specific roles requires coordination of and collaboration within a group of UAVs. In particular, there is a need to design effective routing plans for a group of UAVs that need to visit a series of locations in order to gather some information (e.g., aerial pictures or videos, etc.). Thus, technological progress is required in the fields of advanced algorithms and other IT-based support tools to ensure: *(i)* a safe and effective navigation of UAV within the transportation infrastructure [7]; *(ii)* predictive analytics fed by critical data, which are needed for an efficient use of energy; and *(iii)* the use of computer vision techniques and remote sensing information for processing aerial real-time video footage.

Initially proposed by [8], the team orienteering problem (TOP) is one realistic variation of the well-known vehicle routing problem [9]. The TOP is gaining interest both in the scientific community and the industry due to the increasing use of electric vehicles and unmanned aerial vehicles, where driving range limitations need to be taken into account [10]. Consider the following elements: *(i)* a set of customer nodes, each of them with an associated reward score that can be collected the first time a customer is visited by any UAV; and *(ii)* a team of m UAVs with limited driving-range capabilities. Then, the goal is to determine a set of m open routes (each of them connecting an $origin$ depot with a $destination$ depot), which maximizes the total collected reward by visiting a subset of available customers without violating the driving range constraint.

Notice that each customer can either be visited once or not visited at all. Also, due to the driving range limitation, it is possible that not all customers can be visited. Being an

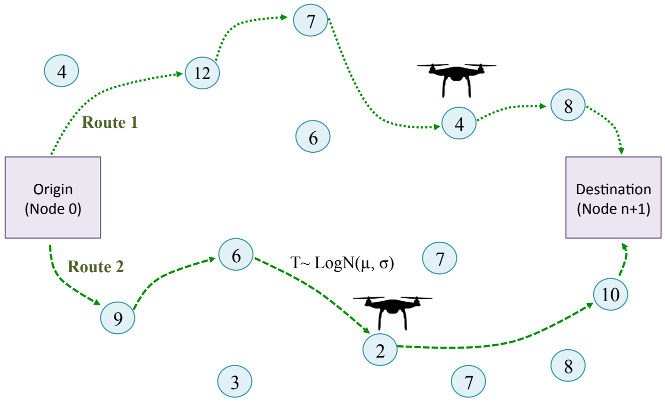


Fig. 1: Routing UAVs and the team orienteering problem

extension of the vehicle routing problem in which a subset of customers have to be selected and a set of routes covering them constructed, the TOP is also a *NP-hard* problem. Accordingly, different metaheuristic approaches have been proposed in recent years to deal with large-scale instances of the deterministic version of the problem. However, the stochastic counterpart, which considers real-life uncertainty in the form of random service and travel times, has received much less attention. This paper analyzes a stochastic TOP variant in which travel times are modeled as random variables (Figure 1). In particular, we consider the problem of recollecting as much reward as possible from visiting customers using a fleet of m UAVs with driving ranges limited by the time-duration of their batteries. An example of practical application could be the use of UAVs to take pictures of different locations after a natural disaster, a terrorist attack, or a humanitarian crisis. Notice that each of these pictures can provide valuable information that can help to improve the conditions of the people affected by the event or even to save their lives by making informed decisions on the more reliable evacuation paths.

Finding a solution (set of open routes) that maximizes the total expected reward is usually the main goal of the stochastic team orienteering problem (STOP). However, since solutions to the STOP are applied in a stochastic environment, other statistical properties should be considered too. Thus, for instance, one could be interested in solutions offering a high reliability level, i.e., routing plans with a low probability of violating the driving-range threshold.

This paper proposes a simulation-optimization algorithm to efficiently cope with the STOP. First, a biased-randomized heuristic for solving the deterministic TOP is introduced. This heuristic is then extended into a simheuristic algorithm [11] to solve the stochastic TOP. Due to their effectiveness, simheuristic algorithms are being increasingly used in solving different stochastic variants of the vehicle routing problem, like the stochastic inventory routing problem [12], the stochastic waste collection problem [13], or the stochastic arc routing problem [14].

The remaining sections of this paper are structured as follows: Section II reviews related work on the TOP. Section III describes our biased-randomized heuristic for solving the deterministic TOP. Section IV describes our extension to a simheuristic to solve the STOP. A round of computational experiments for the STOP are described in Section V. Finally, Section VI summarizes the highlights of this paper and proposes some future research lines.

II. Related Work

The team orienteering problem was first introduced in [8]. To solve the TOP, they propose a heuristic approach where the stops that are farthest from the start and the finish nodes are selected as seeds for the team members, and all possible remaining points are inserted into the routes using the cheapest insertion rule. If unassigned points remain, new team routes are constructed. Additional approaches used to solve the deterministic TOP have been proposed in the literature. Although we can find some exact methods, such as branch-and-cut [15] or branch-and-cut-and-price algorithms [16] to solve the TOP, only small-scale instances can be solved with these methods.

[15] propose a particle swarm optimization (PSO) method to solve the TOP. Similarly, [17] present a multi-start simulated annealing (SA) algorithm to address the TOP. It integrates an SA stage inside a multi-start procedure to reduce the possibility of getting trapped in a local optima. Genetic algorithms (GA) have been also proposed in this area. For instance, [18] introduce a GA which imitates the natural process of evolution to solve the TOP. Other approach to solve the TOP is proposed by [19]. These authors present a Pareto mimic algorithm, which uses a mimic operator to generate a new solution by imitating an incumbent solution.

The stochastic version of the orienteering problem has only received attention in recent years. To the best of our knowledge, previous work has only considered the single-route problem rather than the STOP that we analyze here. There is also some variation in which aspects of the problem are stochastic. For example, the original stochastic single-vehicle orienteering problem (OP) [20] defines the OP with stochastic profits, which assumes that only the scores associated with each node are stochastic – in particular, it is assumed they follow a Normal probability distribution. There are also other works that study the OP with stochastic travel times [21]–[24]. This version can be classified as the orienteering problem with stochastic weights. Notice that our work extends these previous ones by considering multiple vehicles or routes.

In developing solutions to the STOP, one critical question is how to deal with open routes which exceed the designated time limit imposed by the driving-range constraint. In [25], exceeding the time limit incurs in a penalty cost that is proportional to the amount exceeding it. A similar approach is used in [26]. An alternative concept is presented in [27], where the probability of exceeding the time limit must be lower than a threshold value. The problem presented by [21] is partially different, since they do not force the vehicle to return to a set of depots but, instead, it can stop at any location once the time limit is reached. Also, penalties are incurred if a vehicle does not manage to visit a scheduled node within the time limit. In contrast, [24] keep the hard constraint on the tour length that is used in the deterministic version of the problem and abort the route if the expected arrival time to the destination depot is equal to the remaining time. In the previous works, solving methodologies such as VNS metaheuristics and two-stage stochastic optimization were employed.

III. A Biased-Randomized Heuristic for Routing UAVs

A novel constructive heuristic for the TOP has been designed as a first step in our solving approach. One of the main goals of our proposed heuristic is to provide an 'agile' optimization methodology. The term 'agile' referring to software development methodologies ('agile' programming) was introduced by Beck et. al. [28], and it refers to any rapid and easy software development of high-quality. Inspired in this definition, we propose an 'agile' optimization methodology to develop optimization lightweight algorithms, which can be easily implemented in a short period of time, and they can be used in a efficient way in real-life scenarios under uncertainty, providing solutions in real-time (just a few seconds).

The proposed heuristic, which has been designed following this 'agile' optimization methodology, it is inspired on the well-known savings heuristic for the vehicle routing problem [29]. It has to be adapted to consider the particular characteristics of the TOP, i.e.: *(i)* the origin depot could be different from the destination one; *(ii)* not all the customers have to be visited; and *(iii)* the reward collected by visiting nodes must be considered during the construction of the routing plan. The goal was to design a new savings-based heuristic able to outperform the traditional one employed for solving the TOP [30].

Algorithm 1 provides a high-level description of the constructive heuristic. It starts by generating an initial dummy solution (line 1), in which one route per customer is considered –i.e., for each customer $i \in A$, a vehicle departs from the origin depot (node 0), visits i, and then resumes its trip towards the destination depot (node $n+1$) (Figure 2a). If any route in this dummy solution does not satisfy the driving-range constraint, the associated customer is discarded from the problem, since it cannot be reached with the current fleet of vehicles. Next, we compute the 'savings' associated with each edge connecting two different customers (line 2), i.e.: the benefits obtained by visiting both customers in the same route instead of using distinct routes.

In order to compute the savings associated with an edge, one has to consider both the travel time required to traverse that edge as well as the aggregated reward generated by visiting both customers. Thus, we define the concept of savings, s'_{ij} as described in Equation 1. Notice that it takes into account the trade-off between the classical time-based savings, s_{ij}, and the aggregated reward, $u_i + u_j$, i.e.:

$$s'_{ij} = \alpha \cdot s_{ij} + (1-\alpha) \cdot (u_i + u_j) \quad (1)$$

Algorithm 1 Savings-based heuristic for the TOP

```
1: sol ← generateDummySolution(Inputs)
2: savingsList ← computeSortedSanvingsList(Inputs, α)
3: while (savingsList is not empty) do
4:     arc ← selectNextArcAtRandom(savingsList, β )
5:     iRoute ← getStartingRoute(arc)
6:     jRoute ← getClosingRoute(arc)
7:     newRoute ← mergeRoutes(iRoute, jRoute)
8:     travelTimeNewRoute ← calcRouteTravelTime(newRoute)
9:     isMergeValid ← validateMergeDrivingConstraints(travelTimeNewRoute, drivingRange)
10:    if (isMergeValid) then
11:        sol ← updateSolution(newRoute, iRoute, jRoute, sol)
12:    end if
13:    deleteEdgeFromSavingList(arc)
14: end while
15: sortRoutesByProfit(sol)
16: deleteRoutesByProfit(sol, maxVehicles)
17: return sol
```

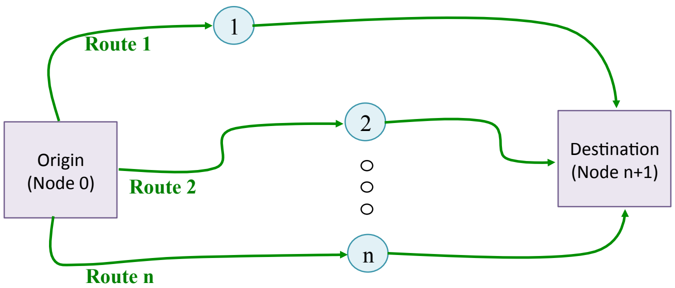
where $s_{ij} = t_{i(n+1)} + t_{0j} - t_{ij}$ (Figure 2b), and $\alpha \in (0, 1)$. The specific value of α needs to be empirically tuned, since it will depend on the heterogeneity of the customers in terms of rewards. Thus, in a scenario with high heterogeneity, α will be close to zero. On the contrary, α will be close to one for homogeneous scenarios. Notice that for each edge there are two associated savings, depending on the actual direction in which the edge is traversed. Thus, each edge generates two different arcs.

After computing all the savings, the list of arcs can be sorted from higher to lower savings. Then, a route-merging process, based on the savings list, is started. In each iteration, the savings list of arcs is randomized using a biased probability distribution, and an arc is selected (line 4). As discussed in detail in Juan et al [31], the biased randomization of the savings list allows arcs to be selected in a different order in each iteration, where arcs with higher savings are more likely to be selected than those with lower savings, while at the same time, the logic behind the savings heuristic is maintained. In our case, a skewed Geometric Distribution is employed to induce this biased randomization behaviour. The Geometric Distribution uses one single parameter, β, which is relatively easy to set since $0 < \beta < 1$. After completing some preliminary tests with different values for and analysing the corresponding outcomes, we decided to set $\beta = 0.3$ in our computational experiments. The selected arc connects two routes, which are merged into a new route as far as this new route does not violate the driving-range constraint (line 9). Finally, the list of routes are sorted according to the total reward provided (line 15) to select as many routes from this list as possible taking into account the restricted number of vehicles in the fleet.

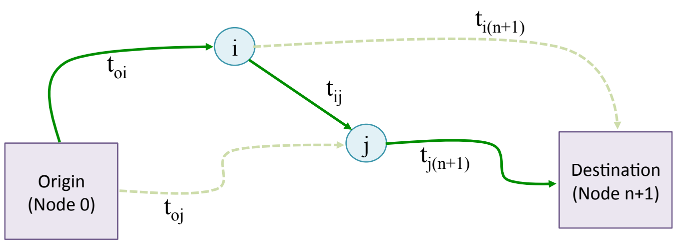
This heuristic is encapsulated within a multi-start process. This allows to run the biased-randomised heuristic several times, thus increasing our chances of finding a better solution.

IV. A Simheuristic for Routing UAVs under Uncertainty

Algorithm 2 provides an overview of our multi-stage simheuristic approach, which extends the biased-randomized



(a) Dummy Solution



(b) Time-based savings

Fig. 2: Dummy solution (top) and time-based savings (down).

heuristic in order to deal with the STOP. In the first stage, a feasible initial solution *(initSolution)* is constructed using the savings-based heuristic described in Section III (line 1). During the second stage, an adaptive heuristic enhances the initial feasible solution by iteratively exploring the search space and conducting a 'reduced' number of simulation runs that allow to: *(i)* obtain observations on the total time employed by the current solution (from which the expected time and other statistics can be estimated); and *(ii)* provide feedback that can be used by the heuristic to better guide the search (e.g., by updating the base solution according to the estimated statistics). From this stage, a reduced set of 'elite' solutions is obtained.

Notice that during the second stage, whenever a *newSol* is 'promising', it is sent through a fast Monte Carlo simulation process (line 8) to estimate the following values: *(i)* the expected return; and *(ii)* its reliability, measured in terms of the percentage of routes that are effectively completed without violating the driving range constraint. Also, whenever the stochastic value of the *newSol* outperforms that of the *baseSol* and / or that of some elite solution *(eliteSols)*, these solutions are updated to *newSol*.

Algorithm 2 Simheuristic approach

```
1: initSolution ← ComputeInitSolution(Inputs)
2: fastSimulation(solution)                      ▷ Monte Carlo Simulation
3: baseSol ← initSolution
4: while (ending condition is not met) do
5:    newSol ← savingBasedHeuristic(Inputs, α, β)   ▷ biased-randomized heuristic
6:    if (detProfit(newSol) improves detProfit(baseSol) then
7:       fastSimulation(newSol)                   ▷ Monte Carlo Simulation
8:       if (stochProfit(newSol) improves stochProfit(baseSol) then
9:          baseSol ← newSol
10:         if (stochProfit(baseSol) improves worstStochProfit(eliteSols) then
11:            eliteSols ← update(eliteSols, baseSol)
12:         end if
13:      end if
14:   end if
15: end while
16: return eliteSols
```

V. Computational Experiments: Stochastic Case

There are not STOP instances to compare with in the literature. For that reason, we have extended the deterministic instances proposed in [8] into stochastic ones.

In our computational experiments, we have modeled the travel times T_{ij} using Log-Normal probability distributions. As discussed in [32], the Log-Normal distribution is a more natural choice than the Normal distribution when modeling non-negative random variables, such as the elapsed time until an event occurs (e.g., the time it takes the vehicle to traverse a given edge). In a real-world application, historical data could be used to model each T_{ij} by a different probability distribution. The Log-Normal has two parameters, namely: the location parameter, μ, and the scale parameter, σ. According to the properties of the Log-Normal distribution, these parameters will be given by the following expressions considering stochastic travel times between nodes i and j:

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

$$\sigma_{ij} = \left|\sqrt{\ln\left(1 + \frac{Var[T_{ij}]}{E[T_{ij}]^2}\right)}\right| \tag{3}$$

In our experiments, which extend classical deterministic instances into stochastic ones, it is assumed that $E[T_{ij}] = t_{ij}$ ($\forall i, j \in N$), being t_{ij} the travel time provided in the deterministic instance. Similarly, it is considered that $Var[T_{ij}] = c \cdot t_{ij}$, being $c \geq 0$ a design parameter. Notice that the deterministic instances are a particular case of the stochastic ones, which are obtained for $c = 0$. In our experiments, we have used the value $c = 0.05$.

The classic deterministic benchmarks consist of 7 different classes, Table I refers to class 1, and it shows: *(i)* the best-known solution for the deterministic variant of the problem (BKS), obtained from the existing TOP literature; *(ii)* our best solution for the deterministic variant of the problem (OBS-D); *(iii)* the computational time in seconds to obtain the OBS-D; *(iv)* the gap between the BKS and the OBS-D; *(v)* the reward of OBS-D when it is applied as a solution of the stochastic variant of the problem (OBS-D-S); *(vi)* our best solution for the stochastic variant of the problem (OBS-S); and *(vii)* the computational time in seconds to obtain the OBS-S.

Figure 3 shows, for the analyzed class, the percentage gaps between: *(i)* the best-found deterministic solution when applied into stochastic conditions (OBS-D-S) and itself when applied in a deterministic environment (OBS-D); and *(ii)* the best-found stochastic solution when applied into stochastic conditions (OBS-S) and the best-found solution for the deterministic version (OBS-D). Notice that the OBS-S boxplot is

TABLE I: Results for class 1 benchmark instances.

	Deterministic execution				Stochastic execution		
Instance	BKS Reward [1]	OBS-D Reward [2]	Time (s) [3]	Gap (%) [4](1-2)	OBS-D-S E[Reward] [5]	OBS-S E[Reward] [6]	Time (s) [7]
p1.2.b	15	15	0	0.0	14.3	14.3	4
p1.2.c	20	20	0	0.0	18.2	18.2	4
p1.2.d	30	30	0	0.0	26.0	26.3	4
p1.2.e	45	45	0	0.0	41.7	42.2	4
p1.2.f	80	80	0	0.0	64.6	70.5	5
p1.2.g	90	90	0	0.0	75.8	83.7	4
p1.2.h	110	110	0	0.0	97.5	103.6	17
p1.2.i	135	135	0	0.0	94.5	122.9	4
p1.2.j	155	155	0	0.0	123.1	141.0	5
p1.2.k	175	175	0	0.0	124.2	163.3	7
p1.2.l	195	195	5	0.0	125.8	179.5	9
p1.2.m	215	215	9	0.0	170.2	201.4	16
p1.2.n	235	235	0	0.0	152.9	218.8	5
p1.2.o	240	240	4	0.0	204.4	230.7	7
p1.2.p	250	250	0	0.0	167.3	230.9	5
p1.2.q	265	265	1	0.0	205.4	247.8	7
p1.2.r	280	280	21	0.0	168.9	259.9	4
p1.3.c	15	15	0	0.0	14.3	14.3	6
p1.3.d	15	15	0	0.0	15.0	15.0	6
p1.3.e	30	30	0	0.0	24.9	27.0	6
p1.3.f	40	40	0	0.0	31.8	32.6	23
p1.3.g	50	50	0	0.0	44.2	48.4	6
p1.3.h	70	70	1	0.0	66.3	66.3	11
p1.3.i	105	105	0	0.0	87.2	94.1	6
p1.3.j	115	115	0	0.0	90.7	102.2	6
p1.3.k	135	135	0	0.0	109.2	121.4	17
p1.3.l	155	155	0	0.0	124.0	138.8	13
p1.3.m	175	175	16	0.0	127.7	159.5	9
p1.3.n	190	190	1	0.0	152.2	175.1	15
p1.3.o	205	205	0	0.0	163.5	183.0	11
p1.3.p	220	220	1	0.0	177.5	199.0	32
p1.3.q	230	230	1	0.0	190.0	218.0	6
p1.3.r	250	250	17	0.0	207.2	233.5	27
p1.4.d	15	15	0	0.0	14.3	14.3	8
p1.4.e	15	15	0	0.0	15.0	15.0	8
p1.4.f	25	25	0	0.0	23.2	23.2	8
p1.4.g	35	35	0	0.0	30.1	34.3	8
p1.4.h	45	45	0	0.0	37.9	40.8	9
p1.4.i	60	60	0	0.0	48.5	55.0	8
p1.4.j	75	75	8	0.0	63.1	67.2	8
p1.4.k	100	100	27	0.0	86.3	86.3	8
p1.4.l	120	120	0	0.0	107.1	116.0	9
p1.4.m	130	130	0	0.0	112.0	124.9	10
p1.4.n	155	155	0	0.0	121.4	135.7	8
p1.4.o	165	165	0	0.0	129.6	149.0	28
p1.4.p	175	175	6	0.0	152.2	159.1	16
p1.4.q	190	190	6	0.0	167.1	171.9	8
p1.4.r	210	210	0	0.0	170.8	191.1	9
		Average:	3	0.0			10

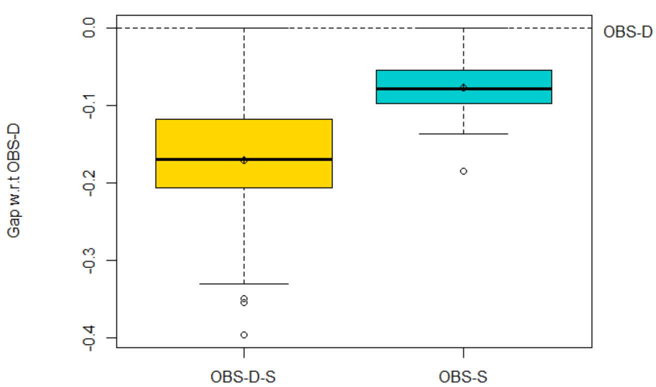


Fig. 3: Boxplot comparison of gaps OBS-D-S and OBS-S w.r.t. OBS-D.

always closer to the OBS-D value than the OBS-D-S boxplot. In other words, employing the best-found deterministic plan into a stochastic environment usually leads to suboptimal solutions. Notice also that the OBS-D value can be seen as an upper bound for the expected reward under stochastic conditions.

VI. Conclusion

The incorporation of unmanned aerial vehicles in urban areas describe promising research fields whose full potential are still to be explored in future. Still, these innovations raise a number of concerns and challenges that complicate decision-making processes for citizens and city managers. New hybrid optimization-simulation and optimization-machine-learning algorithms have to be developed to efficiently face these challenges. Scenarios for dynamic and uncertain real-life features have to be included.

Acknowledgment

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness, FEDER (TRA2015-71883-REDT) and the Erasmus+ program (2016-1-ES01-KA108-023465).

References

[1] S. E. Bibri and J. Krogstie, "On the social shaping dimensions of smart sustainable cities: A study in science, technology, and society," *Sustainable Cities and Society*, vol. 29, pp. 219–246, 2017.

[2] M. Angelidou, "Smart cities: A conjuncture of four forces," *Cities*, vol. 47, pp. 95–106, 2015.

[3] A. G. Foina, R. Sengupta, P. Lerchi, Z. Liu, and C. Krainer, "Drones in smart cities: Overcoming barriers through air traffic control research," in *Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2015 Workshop on*. IEEE, 2015, pp. 351–359.

[4] O. B. Jensen, "Drone city-power, design and aerial mobility in the age of" smart cities"," *Geographica Helvetica*, vol. 71, no. 2, p. 67, 2016.

[5] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "Uav-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.

[6] F. Mohammed, A. Idries, N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "Opportunities and challenges of using uavs for dubai smart city," in *New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on*. IEEE, 2014, pp. 1–4.

[7] E. N. Barmpounakis, E. I. Vlahogianni, and J. C. Golias, "Unmanned aerial aircraft systems for transportation engineering: Current practice and future challenges," *International Journal of Transportation Science and Technology*, vol. 5, no. 3, pp. 111–122, 2016.

[8] I.-M. Chao, B. Golden, and E. Wasil, "The team orienteering problem," *European Journal of Operational Research*, vol. 88, pp. 464–474, 1996.

[9] J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan, "Rich vehicle routing problem: Survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, p. 32, 2015.

[10] A. A. Juan, J. Goentzel, and T. Bektaş, "Routing fleets with multiple driving ranges: Is it possible to use greener fleet configurations?" *Applied Soft Computing*, vol. 21, pp. 84–94, 2014.

[11] A. A. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira, "A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems," *Operations Research Perspectives*, vol. 2, no. 1, pp. 62–72, 2015.

[12] A. Gruler, J. Panadero, J. de Armas, J. A. M. Pérez, and A. A. Juan, "A variable neighborhood search simheuristic for the multiperiod inventory routing problem with stochastic demands," *International Transactions in Operational Research*, 2018.

[13] A. Gruler, C. Fikar, A. A. Juan, P. Hirsch, and C. Contreras-Bolton, "Supporting multi-depot and stochastic waste collection management in clustered urban areas via simulation–optimization," *Journal of simulation*, vol. 11, no. 1, pp. 11–19, 2017.

[14] S. Gonzalez-Martin, A. A. Juan, D. Riera, M. G. Elizondo, and J. J. Ramos, "A simheuristic algorithm for solving the arc routing problem with stochastic demands," *Journal of Simulation*, vol. 12, no. 1, pp. 53–66, 2018.

[15] D.-C. Dang, R. N. Guibadj, and A. Moukrim, "An effective pso-inspired algorithm for the team orienteering problem," *European Journal of Operational Research*, vol. 229, no. 2, pp. 332–344, 2013.

[16] M. Keshtkaran, K. Ziarati, A. Bettinelli, and D. Vigo, "Enhanced exact solution methods for the team orienteering problem," *International Journal of Production Research*, vol. 54, pp. 591–601, 2015.

[17] S. Lin, "Solving the team orienteering problem using effective multi-start simulated annealing," *Applied Soft Computing*, vol. 13, pp. 1064–1073, 2013.

[18] J. Ferreira, A. Quintas, and J. Oliveira, "Solving the team orienteering problem: developing a solution tool using a genetic algorithm approach," in *Soft computing in industrial applications. Advances in Intelligent Systems and Computing: 223*. Springer, 2014, pp. 365–375.

[19] L. Ke, L. Zhai, J. Li, and F. Chan, "Pareto mimic algorithm: an approach to the team orienteering problem," *Omega*, vol. 61, pp. 155–166, 2015.

[20] T. Ilhan, S. Iravani, and M. Daskin, "The orienteering problem with stochastic profits," *IIE Transactions*, vol. 40, pp. 406–421, 2008.

[21] A. Campbell, M. Gendreau, and B. Thomas, "The orienteering problem with stochastic travel and service times," *Annals of Operations Research*, vol. 186, pp. 61–81, 2011.

[22] V. Papapanagiotou, R. Montemanni, and L. Gambardella, "Objective function evaluation methods for the orienteering problem with stochastic travel and service times," *Journal of Applied Operations Research*, vol. 6, pp. 16–29, 2014.

[23] C. Verbeeck, P. Vansteenwegen, and E.-H. Aghezzaf, "Solving the stochastic time-dependent orienteering problem with time windows," *European Journal of Operational Research*, vol. 255, pp. 699–718, 2016.

[24] L. Evers, K. Glorie, S. van der Ster, A. Barros, and H. Monsuur, "A two-stage approach to the orienteering problem with stochastic weights," *Computers and Operations Research*, vol. 43, pp. 248–260, 2014.

[25] S. Teng, H. Ong, and H. Huang, "An integer l-shaped algorithm for time-constrained traveling salesman problem with stochastic travel and service times," *Asia-Pacific Journal of Operational Research*, vol. 21, pp. 241–257, 2004.

[26] H. C. Lau, W. Yeoh, P. Varakantham, D. T. Nguyen, and H. Chen, "Dynamic stochastic orienteering problems for risk-aware applications," *CoRR*, vol. abs/1210.4874, 2012. [Online]. Available: http://arxiv.org/abs/1210.4874

[27] H. Tang and E. Miller-Hooks, "A tabu search heuristic for the team orienteering problem," *Computers & Operations Research*, vol. 32, no. 6, pp. 1379–1407, 2005.

[28] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto for agile software development," 2001. [Online]. Available: http://www.agilemanifesto.org/

[29] G. Clarke and J. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, pp. pp. 568–581, 1964.

[30] H. Tang and E. Miller-Hooks, "A tabu search heuristic for the team orienteering problem," *Computers & Operations Research*, vol. 32, no. 6, pp. 1379–1407, 2005.

[31] A. A. Juan, J. Faulin, R. Ruiz, B. Barrios, and S. Caballé, "The sr-gcws hybrid algorithm for solving the capacitated vehicle routing problem," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 215–224, Jan. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2009.07.003

[32] A. A. Juan, J. Faulin, S. E. Grasman, D. Riera, J. Marull, and C. Mendez, "Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 5, pp. 751–765, 2011.