Computers and Operations Research 88 (2017) 306–315

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

# A multi-objective green UAV routing problem

Bruno N. Coelho[a,b,c,*], Vitor N. Coelho[a,c,d,*], Igor M. Coelho[a,c,e], Luiz S. Ochi[d], Roozbeh Haghnazar K.[f], Demetrius Zuidema[h], Milton S.F. Lima[g], Adilson R. da Costa[b]

[a] Instituto de Pesquisa e Desenvolvimento de Tecnologias, Ouro Preto, Brazil
[b] REDEMAT – Rede Temática em Engenharia de Materiais, Universidade Federal de Ouro Preto, Brazil
[c] Grupo da Causa Humana, Ouro Preto, Brazil
[d] Institute of Computer Science, Universidade Federal Fluminense, Niterói, Brazil
[e] Computer Science Department, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brazil
[f] Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
[g] IEAv – Institute for Advanced Studies, Department of Aerospace Science and Technology, São José dos Campos, Brazil
[h] Jeppesen Manager Government and Military Aviation, Miami, USA

ARTICLE INFO

*Article history:*
Received 1 July 2016
Revised 24 April 2017
Accepted 24 April 2017
Available online 29 April 2017

*Keywords:*
Unmanned aerial vehicle
Vehicle routing
Microgrids
Multi-objective optimization
Matheuristic

ABSTRACT

This paper introduces an Unmanned Aerial Vehicle (UAV) heterogeneous fleet routing problem, dealing with vehicles limited autonomy by considering multiple charging stations and respecting operational requirements. A green routing problem is designed for overcoming difficulties that exist as a result of limited vehicle driving range. Due to the large amount of drones emerging in the society, UAVs use and efficiency should be optimized. In particular, these kinds of vehicles have been recently used for delivering and collecting products. Here, we design a new real-time routing problem, in which different types of drones can collect and deliver packages. These aerial vehicles are able to collect more than one deliverable at the same time if it fits their maximum capacity. Inspired by a multi-criteria view of real systems, seven different objective functions are considered and sought to be minimized using a Mixed-Integer Linear Programming (MILP) model solved by a matheuristic algorithm. The latter filters the non-dominated solutions from the pool of solutions found in the branch-and-bound optimization tree, using a black-box dynamic search algorithm. A case of study, considering a bi-layer scenario, is presented in order to validate the proposal, which showed to be able to provide good quality solutions for supporting decision making.



## 1. Introduction

A technological boom has taken place in the development of Unmanned Aerial Vehicles (UAV), also known as Drones, or Unmanned Aerial Systems (UAS) (van Blyenburgh, 1999; Gupta et al., 2013; Sundar and Rathinam, 2014). This development dates back the last decade facing applications in the military sector (Enright et al., 2005; Gertler, 2012; Ground et al., 2002), mainly because the large amount of investment that used to be required for UAVs construction and control. Only recently the drones have become more popular due to the miniaturization of electronic control systems and the availability of low-cost sensors in the market. Sensors like accelerometers, gyroscopes, digital magnetometers, barometers and even antennas of Global Positioning System (GPS) (Nex and Remondino, 2014) were, initially, popularized by the smartphone industry, providing a new way of measuring the world we live in. This advance allowed the great popular mass to access these new technologies. Drones emerging in the society is also related to the development of novel materials (Newcome, 2004; Spinka et al., 2011), lighter and more resistant, such as: advanced aluminum alloys, carbon fiber reinforced polymer and other composites. New cutting, drilling and laser welding, techniques also greatly facilitate this advance.

This aforementioned eminent boom came on the scene mainly by hobbyists and the entertainment and photographic industries (providing the acquisition of aerial images) (Colomina and Molina, 2014). Quickly, new applications such as geocoding and mapping areas for industries, tracking systems, signals intelligence operational platforms, border control, vegetation control under electrical transmission lines and surveillance systems are emerging and becoming reality in different areas. These applications are becoming increasingly complex, involving other factors beyond simple images acquisition (Frew and Brown, 2009; Pohl and Lamont, 2008; Turner et al., 2016). In this sense, the ability of UAV in collecting and de-

* Corresponding authors.
*E-mail addresses:* brunonazario@gmail.com (B.N. Coelho), vncoelho@gmail.com (V.N. Coelho).

http://dx.doi.org/10.1016/j.cor.2017.04.011
0305-0548/

**Fig. 1.** Huge UAVs charging station in New York – Extracted from eVolo Magazine 2016 Skyscraper Competition, Mohammad et al. (2016).

livering materials should be highlighted (Alighanbari et al., 2003; Enright et al., 2015; MEMP, 2013; Sundar and Rathinam, 2014; Weinstein and Schumacher, 2007), as well as measurement of air pollution in industrial areas, building painting, skyscrapers windows cleaning, city lights control and maintenance, among countless other applications that may arise from human creativity.

In particular, given the considerable growth of online sales and clients requirements, huge companies are investing in delivery polices that take advantage of the use of UAVs. Patents are being registered (Berg et al., 2016), some of them mention the use of a package delivery apparatus that uses an UAV (Lisso, 2017). The huge potential of using drones as delivery tools has been noticed in the last few years. Currently, in the beginning of this new era, thinking in that way, can lead to a better quality of life. Fig. 1 illustrates an awarded-winning design of a huge drone tower in New York. The idea of the illustration was to show a Smart Cities (SC) (Mohammed et al., 2014) hub for commercial and personal deliveries that might be done by UAV.

In several countries, laws are already quite strict and drone flights are only allowed in private area's. Discussions surrounding UAVs traffic control are still being discussed and implemented. For instance, electronic barriers may limit or block vehicles communication in restricted areas. In particular, considering that they will be like a swarm over our heads, if not controlled, this development could result in accidents and other harmful effects to society. In this sense, paths safety is also a topic under investigation, which may guide vehicles through areas with less risk of accidents. In this new world, UAVs airspace management should be done in real-time, online, allowing and requiring human intervention only in extreme cases. Thus, a time-dependent problem needs to be handled and new smart solutions are being requested.

In particular, time-dependent routing problems can involve a huge amount of stochastic information, in such a way that UAVs should be able to change, adapt, modify and optimize their routes in real-time. Several individual objective functions can be used for guiding each UAV preference, such as reducing individual costs, enhancing its profit, increasing safety, reducing lead time and increasing the load capacity of the entire system. Enright et al. (2015) surveyed algorithms on task assignment/allocation and UAVs scheduling, discussing dynamic time-dependent scenarios where targets arrive at random locations at random times.

Nowadays, one of the major constraints to the use of drones is the power source, which is usually limited to one hour of autonomy. A possible strategy for dealing with this limitation is to consider charging points in order to enhance vehicles covering range. This problem can fit the new class of Green Routing Problems (Erdogan and Miller-Hooks, 2012), which is being investigated for alternative fuel-powered vehicle fleets (Lin et al., 2014). Sundar and Rathinam (2014) introduced a Mixed-Integer Linear Programing (MILP) for a so-called Fuel Constrained UAV Routing Problem (FCURP), considering a single UAV that should visit all targets at least once. On the other hand, SC are now being able to take advantage of the new mesh of microgrid systems (Fathima and Palanisamy, 2015), which are composing the new electric grid paradigm, known as Smart Grids (SG). This new bidirectional system, mainly composed of intermittent Renewable Energy Resources (RER) (Basak et al., 2012), can share energy with a hybrid environment composed of UAVs (Coelho et al., 2016b), which will be able to directly charge from collecting and delivering clients, as well as other independent charging stations. The growth of the SC (Karnouskos et al., 2012) is surely aligned with energy development. The fact that RER are mostly intermittent allows the exceeding amount of energy to be used by those flying and fast moving devices. In this sense, the transition to the SG will allow a greater range to the drones, increasing system capacity and generating profits for both sides, drones owners and energy suppliers.

In light of this complex time-dependent scenario, in this paper, we design a new multi-objective Green UAV Routing Problem (GUAVRP), which minimizes seven objective functions: total traveled distance; uavs maximum speed; number of used vehicles; makespans of the last collected and delivered package; average time spent with each package; and maximize batteries load at the end of the schedule. In the proposed approach, a dynamic scenario is considered, in which new orders may arrive at any moment, in this sense, drones en route can be considered after finishing their current trip. Furthermore, the model respects drones operational requirements, such as: maximum weight they are able to carry, battery minimum Depth-of-Discharge (DoD); UAV maximum speed, among other constraints set by the traffic controller. As mentioned, when UAVs are requested to visit several clients, vehicles might need to come to depot for refueling (Sundar and Rathinam, 2014). In this sense, the case of study introduced here also allows the vehicles to refuel/charge at the charging stations, since their autonomy does not allow them to fly over long periods.

UAV's can be limited to fly only at predetermined levels of altitude, related to their load capacity and size. A simple environment can be idealized with the biggest and fastest ones flying at higher levels. These levels or layers can be interconnected by vertical displacement points, where UAV will be allowed to exchange layer and products. Thus, loads can be redistributed by smaller drones using supporting spots that allow vertical displacement between layers. In our case of study, this delivery and collect flow of goods (and sharing points that promote exchange between the drones) is organized by integrated control centers, which provide different non-dominated solutions for managing this complex air traffic.

The main contributions of this current work are:

- Develop a mathematical programming model for a time-dependent UAV heterogeneous fleet routing problem, in particular:
  - respecting UAVs operational requirements;
  - tackling the micro-airspace considering a multi-layer scenario with package exchanging points;
  - integrating UAVs into the new concepts of mini/microgrid systems, in which vehicles can be charged at different points of the future smart cities.
- Consider a multi-objective optimization framework in order to provide alternative solutions with different possible routes and schedules.

The remainder of this paper is organized as follows. Sections 2.1 and 2.2.1 describe the proposed model and the range of real parameters considered in our analyses, while

Section 2.3 details the method used for solving the problem, providing sets of non-dominated solutions. Section 3 introduces the case of study used for validating the proposed Multi-Objective MILP model. Section 4 presents the computational experiments, and, finally, Section 5 draws the final considerations and discuss possible future research directions.

## 2. Formulation for proposed GUAVRP

The proposed Mixed-Integer Linear Programming (MILP) model is detailed using two Sections 2.1 and 2.2. The first section describes the formulation for the routing problem itself, establishing constraints related to the action of collect and deliver a package, respecting UAV maximum speed and capacity.

The second one considers the inclusion of charging stations, in which UAVs might charge in order to respect operational requirements and extend its coverage area. Finally, Section 2.3 describes the Matheuristic used for tackling the proposed multi-objective routing problem.

### 2.1. MILP for the multi-objective UAV routing problem

#### 2.1.1. Sets, parameters, auxiliary and decision variables

The following parameters were defined and considered by the model:

$T$: Set of discretization schedules $\{1, \ldots, t^{\max}\}$;
*TimeUnit*: fixed time unit (hour) between any instant $t \in T$ to $t+1 \in T$. Thus, $t^{\max} * TimeUnit$ is the maximum amount of time in which the vehicles can finish their delivering tasks;
$Z$: size (in km) of the square map to be considered by the model;
$P$: Set of demands (delivery/collection), containing the following information for each product $p$:

- $C_p^x$: collecting point $x$ coordinate, for $p \in P$;
- $C_p^y$: collecting point $y$ coordinate, for $p \in P$;
- $D_p^x$: delivering point $x$ coordinate, for $p \in P$;
- $D_p^y$: delivering point $y$ coordinate, for $p \in P$;
- $W_p$: package weight for product $p \in P$;

$U$: Set of available UAVs that can be used, providing operational information regarding each vehicle:

- $I_u^x$: UAV initial coordinate $x$ at the beginning of the schedule (t=1);
- $I_u^y$: UAV initial coordinate $y$ at the beginning of the schedule (t=1);
- $Occ_{u,p}$: time-dependent binary parameter that indicates if UAV $u$ is occupied and, consequently, already delivering a product $p$ when the problem is sought to be solved;
- $V_u^{\max}$: UAV maximum speed (km/h);
- $Q_u^{\max}$: UAV maximum capacity (kg);
- *MFC*: minimum amount of energy that each vehicle should be charged when the routing schedule finishes;

The following decision variables used to provide a complete solution to the proposed model.

$uC_{u,p,t}$: binary variable that indicates if package $p$ was collected by UAV $u$ at time $t$.
$uD_{u,p,t}$: binary variable that indicates if package $p$ was delivered by UAV $u$ at time $t$.
$uID_{u,p,t}$: binary variable indicates if UAV $u$ is on-route at the instant $t$, in particular, delivering the item $p$.
$pos_{u,t}^x$: position of UAV coordinate $x$ at time $t$;
$pos_{u,t}^y$: position UAV coordinate $y$ at time $t$;

Some auxiliary variables were made necessary for calculating for checking operational requirements, constraints or assist the calculus of a desired metric:

$q_{u,t}$: UAV load at time $t$ (amount of weight being carried by the vehicle);
$v_{u,t}$: UAV speed at time $t$ (starting at interval $t-1$ to $t \in T$);
$used_u$: UAV was used during the routing schedule, binary;

#### 2.1.2. Goals to be optimized

Objective function values are calculated in each of the following variables:

*dist*: total distance traveled by the UAVs;
*time*: total delivering time taken by the UAVs;
*drones*: number of used drones;
*maxvel*: maximum speed reached by the UAVs;
*makespanC*: indicates the last $t \in T$ that a package was collected;
*makespanD*: analogous to the previous one, but considering the last delivered package;
*toFull*: indicates the total amount of energy necessary for completing the battery of the UAVs.

$$\begin{aligned}\text{minimize } & \lambda_1 dist + \lambda_2 time + \lambda_3 drones \\ & + \lambda_4 maxvel + \lambda_5 toFull \\ & + \lambda_6 makespanC + \lambda_7 makespanD \end{aligned} \quad (1)$$

The routing part of mathematical model proposed in this study can be seen from Eqs. (2) to (25). Traveled distance calculus, Eqs. (2) and (24), absolute values of the displacement in each coordinate, was linearized using four additional constraints and variables.

$$dist = \sum_{u \in U} \sum_{t \in t : t \geq 2} \left( |pos_{u,t}^x - pos_{u,t-1}^x| + |pos_{u,t}^y - pos_{u,t-1}^y| \right) \quad (2)$$

Eq. (3) checks the drones that were used (counting with the constraint stated in Eq. (4)) and calculates the total number of UAVs that are assisting routing scheduling.

$$drones = \sum_{u \in U} used_u \quad (3)$$

$$used_u * M = \sum_{p \in P} \sum_{t \in T} uID_{u,p,t} \qquad \forall u \in U \quad (4)$$

The total amount of time UAVs are holding packages is summed in Eq. (5).

$$time = \sum_{p \in P} \sum_{u \in U} \sum_{t \in T} uID_{u,p,t} \quad (5)$$

Three makespan values are also calculated. Eqs. (6) and (7) indicate the last time a package is collected and delivered, respectively. On the other hand, the maximum speed reached by the UAVs is determined at Eq. (8).

$$makespanC \geq \sum_{u \in U} \sum_{t \in T} uC_{u,p,t} \qquad \forall p \in P \quad (6)$$

$$makespanD \geq \sum_{u \in U} \sum_{t \in T} uD_{u,p,t} \qquad \forall p \in P \quad (7)$$

$$maxvel \geq v_{u,t} \qquad \forall u \in U, t \geq 2 \in T \quad (8)$$

Finally, Eq. (9) measures the amount of energy that is required for fulfilling UAVs batteries. The explanation of this objective function and its variables is presented in Section 2.2.

$$toFull \geq \sum_{u \in U} (100 - batRate_{u,t}) \quad (9)$$

*2.1.3. Model constraints and operational requirements*

Eq. (10) initializes en route UAVs that are already delivering goods. These time-dependent constraints updates drones' current capacity with the sum of all packages they are delivering when the optimizer is called. Eq. (11) updates vehicles capacity during the consecutive intervals of the time indexed routing problem.

$$q_{u,1} = \sum_{p \in P} W_p \, Occ_{u,p} \qquad \forall u \in U \tag{10}$$

$$q_{u,t} = q_{u,t-1} + \sum_{p \in P} (uC_{u,p,t} - uD_{u,p,t}) W_p \qquad \forall u \in U, t \in T : t \geq 2 \tag{11}$$

Vehicles maximum capacity should be respected and overloading is not allowed by the constraint stated in Eq. (12).

$$q_{u,t} \leq Q_u^{max} \qquad \forall u \in U, t \in T \tag{12}$$

Eq. (13) defines that each demand $p \in P$ should be collected or initialized being collected by the time-dependent variable $Occ_{u,p}$. Eq. (14) is designed in order to force delivering only by who collected it.

$$\sum_{u \in U} Occ_{u,p} + \sum_{t \in T} \sum_{u \in U} uC_{u,p,t} = 1 \quad \forall p \in P \tag{13}$$

$$\sum_{t \in T} uD_{u,p,t} = Occ_{u,p} + \sum_{t \in T} uC_{u,p,t} \qquad \forall p \in P, u \in U \tag{14}$$

The family of Eqs. (15)–(17) updates vehicles delivering status. The first one is a time-dependent constraint set at the beginning of the scheduling process, setting the value one for those which are already flying and delivering products. Eq. (16) updates variables $uID_{u,p,t}$ when packages are collected and delivered. Eq. (17) ensures that at the end of all intervals UAVs will not be delivering anything.

$$uID_{u,p,1} = Occ_{u,p} \qquad \forall p \in P, u \in U \tag{15}$$

$$uID_{u,p,t} = uID_{u,p,t-1} + uC_{u,p,t} - uD_{u,p,t} \qquad \forall p \in P, u \in U, t \geq 2 \in t \tag{16}$$

$$uID_{u,p,t^{max}} = 0 \qquad \forall p \in P, u \in U \tag{17}$$

UAVs initial position are updated in Eqs. (18) and (19). From Eq. (20) to Eq. (23) it is provided that UAVs can only collect packages at the right position, exactly where its targets are located. Four other analogous constraints were include to ensure that packages were delivered accordingly. In those ones, collecting position $C_p$ was changed to $D_p$ and binary variables $uC_{d,p,t}$ modified to $uD_{d,p,t}$.

$$pos^x_{u,1} = I^x_u \qquad \forall u \in U \tag{18}$$

$$pos^y_{u,1} = I^y_u \qquad \forall u \in U \tag{19}$$

$$pos^x_{u,t} - C^x_p \leq Z(1 - uC_{u,p,t}) \qquad \forall p \in P, u \in U, t \in T \tag{20}$$

$$-pos^x_{u,t} + C^x_p \leq Z(1 - uC_{u,p,t}) \qquad \forall p \in P, u \in U, t \in T \tag{21}$$

$$pos^y_{u,t} - C^y_p \leq Z(1 - uC_{u,p,t}) \qquad \forall p \in P, u \in U, t \in T \tag{22}$$

$$-pos^y_{u,t} + C^y_p \leq Z(1 - uC_{u,p,t}) \qquad \forall p \in P, u \in U, t \in T \tag{23}$$

Finally, (24) and (25) calculate vehicles speed at each interval and limit it to the maximum vehicle set for each UAV, respectively.

$$v_{u,t} = \left(|pos^x_{u,t} - pos^x_{u,t-1}| + |pos^y_{u,t} - pos^y_{u,t-1}|\right) \frac{1}{TimeUnit} \qquad \forall u \in U, t \in T : t \geq 2 \tag{24}$$

$$v_{u,t} \leq V^{max}_u \qquad \forall u \in U, t \in T : t \geq 2 \tag{25}$$

*2.2. Additional constraints for considering limited vehicles' autonomy and charging stations*

*2.2.1. Sets, parameters, auxiliary and decision variables*

The set of charging stations $E$ has only two parameters, which are the charging station coordinates (that may coincide or not with the microgrid interaction points):

$E$: Set of energy stations $e$ for charging:

- $CS^x_e$: energy charging station $e$ coordinate $x$;
- $CS^y_e$: energy charging station $e$ coordinate $y$;

Five additional parameters were included for each drone $u \in U$:

$U$: Set of UAVs available to be used:

- $DOD_u$: UAV depth of discharge (%);
- $VEV_u$: variable energy consumption (%) related to drone speed (per hour);
- $FEV_u$: fixed energy consumption (%) per hour;
- $RC_u$: UAV rate of charge per hour (%);
- $BI_u$: Drone battery rate on arrival (%);

The main decision variable regarding the issue of updating vehicle battery rate and scheduling vehicle charging is done by variable $charge_{u,e,t}$, which is a binary variable that indicates if UAV $u$ is charging at point $e$ at time $t$:

- $charge_{u,e,t}$: UAV $u$ is charging in $e$ at time $t$.

Furthermore, three additional auxiliary variables were necessary for updating UAV battery rate at each interval $t$, as well as checking if the UAVs are online at each $t$. Variable $excess_{u,t}$ is an auxiliary variable that also should be minimized, thus, it is included in Eq. (1). However, these variables are not the focus of the optimization itself and are quoted to be optimized with different weights. These variables allow the battery to be slightly "overcharged". This fact does not mean that the battery will be really overcharged in the real case, but it plays a trick in the mathematical model to allow the rate of battery $batRate_{u,t}$ to match the charging rate when values slightly exceed 100%. In a real case, the charging station would stop charging when full charge was reached.

- $batRate_{u,t}$: UAV battery rate at time $t$, $\leq 0$ and $\geq 100$;
- $on_{u,t}$: UAV is running, turned on, at time $t$, binary;
- $excess_{u,t}$: UAV excess of energy at time $t$;

*2.2.2. Model constraints and batteries operational requirements*

Constraints related to UAVs charging station and amount of battery rate requirements are stated below:

In particular, Eqs. (26), (27) and (30), are inspired by the work of Coelho et al. (2016a), which focused on updating battery rates of Plug-in Electric Vehicles. Eq. (26) defines vehicles battery on arrival, while constrain (27) updates battery rate at each interval, increasing it if it was charged and reducing the amount of energy

used during its current trip (due to its average speed and turned on).

$$batRate_{u,1} = BI_u \qquad \forall u \in U \tag{26}$$

$$batRate_{u,t} = batRate_{u,t-1} - TimeUnit\left(\frac{VEV_u\, v_{u,t}}{V_u^{\max}} - on_{u,t}\, FEV_u + \sum_{e \in E} charge_{u,e,t}\, RC_u\right) + excess_{u,t} \qquad \forall u \in U, t \in T : t \geq 2 \tag{27}$$

$$excess = \sum_{u \in U} \sum_{t \in t : t \geq 2} excess_{u,t} \tag{28}$$

As can be verified in Eq. (29), the UAV is considered switched on whenever it has moved.

$$on_{u,t} * M = v_{u,t} \qquad \forall u \in U, t \in T \tag{29}$$

Eq. (30) is a hard constraint that limits the minimal acceptable discharge. On the other hand, Eq. (31) inserts the requirements that UAV battery rate should be higher than *MFC* at the end of the last interval $t^{\max}$.

$$batRate_{u,t} \geq DOD_u \qquad \forall u \in U, t \in T \tag{30}$$

$$batRate_{u,t^{max}} \geq MFC \qquad \forall u \in U \tag{31}$$

Four additional constraints, analogous to those ones described from Eq. (20) to Eq. (23), were included to ensure charging at the right position. Collecting position $c_p$ was changed to $cs_{cs}$ and binary variables $uC_{d,\,p,\,t}$ modified to $uCS_{d,\,cs,\,t}$. In this sense, those four constraints ensure that drones will charge only if they are physically located at the charging station.

### 2.3. Matheuristic algorithm

In order to obtain a set of non-dominated solutions for the proposed routing problem, in restricted computational time, the use of the Multi-Objective Smart Pool Search (MOSPOOLS) Matheuristic, introduced by Coelho et al. (2016a) and extended in Coelho et al. (2016b), is proposed. The main core of the MOSPOOLS is to solve the mathematical model using any Black-Box solver for MILP problems.

A predefined set of objective function weights is defined as $v\Lambda$. This set $v\Lambda = \{v\lambda_1, \ldots, v\lambda_i, \ldots, v\lambda_7\}$ contains seven vectors with the desired weights for each objective function. Several different MILP problems are generated by the linear combination of the weights of each vector $v\lambda_i = \{\lambda_i^1, \ldots, \lambda_i^k\}$, composed of $k$ possible weights. Thus, the Cartesian product of them $\Lambda = \{v\lambda_1 \times \ldots \times v\lambda_7\}$ defines the number of MILP problems to be solved (Eq. (32)). This strategy is capable of providing a good balance between each objective, by ensuring that several problems, created using decision maker specific desired weights, are solved.

$$|\Lambda| = |v\lambda_1| \times \ldots \times |v\lambda_i| \times \ldots \times |v\lambda_7| \tag{32}$$

Algorithm 1 states MOSPOOLS pseudocode, which presents the procedure used for generating and solving the weighted MILP problems, as well as filtering the obtained solutions in order to create a Pareto front. Furthermore, we consider the version introduced by Coelho et al. (2016d), which brings a smart strategy for using MIP start solutions (line 4) in the beginning of the search done by the Black-Box solver.

Line 3 of Algorithm 1 calls the math model, described in Sections 2.1 and 2.2, with a unique combination of weights for each of the seven objective functions. The generated model is solved by a black-box tool (Line 5) and, optionally, returns all feasible solutions obtained during the optimization process. In our case, each solution found in the branches of a Branch-and-Bound (BB) tree is considered by the addSolution procedure (Line 8), extracted from Lust and Teghem (2010) (described in Algorithm 2). This procedure tries to add each solution from the pool of solution $s \in$ *poolSol* to the set of non-dominated solutions *Xe*.

**Algorithm 1.** Multi-Objective Smart Pool Search (MOSPOOLS) Matheuristic.

**Input**: solver time limit $sTLimit$ and vector of MILP weights $\Lambda = \{v\lambda_1 \times ... \times v\lambda_7\}$
**Output**: Set of non-dominated solutions $Xe$

```
1  mipPop ← ∅
2  forall λ_i ∈ Λ | ∀i = {1, ..., 7} do
3  |  model ← MILP model with weights λ_i
4  |  mipStart ← best solution ∈ mipPop
   |  regarding current model weights λ_i
5  |  [poolSol, poolEval] ← MILP-Solver(model,
   |  sTLimit, mipStart)
6  |  mipPop ← mipPop + new solutions inside
   |  poolSol
7  |  forall nS ∈ poolSol do
8  |  |  Xe ←
   |  |  addSolution(Xe, poolSol_nS, poolEval_nS)
9  |  end
10 end
11 return Xe
```

**Algorithm 2.** addSolution.

**Input**: Population $Xe$ potentially efficient; Solution $s$, and its evaluations $z(s)$
**Output**: $Xe$

```
1  Added ← true
2  forall x ∈ Xe do
3  |  if z(x) ⪯ z(s) then
4  |  |  Added ← false
5  |  |  Break
6  |  end
7  |  if z(s) ≺ z(x) then
8  |  |  Xe ← Xe \ x
9  |  end
10 end
11 if Added = true then
12 |  Xe ← Xe ∪ s
13 end
14 return Xe
```

## 3. Case of study considering a bi-level UAV routing

The case of study designed here is composed of an airspace divided into two layers: 1 – lower layer, located at low altitudes, in which smaller UAVs travel with lower speed; 2 – upper layer, located at high altitudes, where the traffic is mainly composed of faster drones with heavier loads. Each layer is subdivided into horizontal and vertical strips, where vehicles are allowed to move following Manhattan distance, also known as taxicab geometry or city block distance. Loads may be exchanged between layers at predefined supporting points. Smaller drones usually collect goods from clients and directly deliver them to the supporting points, and vice versa. Thus, in the upper layer, loads are usually transported over longer distances and, throughout the supporting spots, transferred to the lower layer UAVs.

Energy stations $E$ are spread in the routing area and accessed by drones for refueling/recharging their batteries. This essential component enhances vehicles autonomy, allowing them to reach their

**Table 1**
Layers characteristics.

| | Inf. layer | Sup. layer |
|---|---|---|
| Size (m) | 1000 | 5000 |
| Area (km$^2$) | 1 | 25 |
| Supporting points | 3 | 3 |
| Energy points | 5 | 5 |
| Interaction points | 25 | 10 |
| Average load (Kg) | 0.61 | 0.61 |
| Total Dist. (m) | 18,010 m | 32,900 m |
| Avg load Dist (m) | 720 m | 3290 m |

**Table 2**
UAVs parameters.

| Drone Type | Layer | Units | Max. load (kg) | Max. speed (km/h) |
|---|---|---|---|---|
| D1 | 1 | 12 | 9 | 20 |
| D2 | 1 | 9 | 1 | 35 |
| D3 | 1 | 9 | 0.3 | 50 |
| D4 | 2 | 2 | 5 | 80 |
| D5 | 2 | 3 | 4 | 100 |
| D6 | 2 | 2 | 3 | 120 |

desired targets and respect operational requirements. We consider these points as independent microgrids sources of energy, usually composed of RER.

The lower layer was generated considering an area of 1000 m $\times$ 1000 m (1 km$^2$) and the upper layer covered a square of 5000 m $\times$ 5000 m (25 km$^2$). Three supporting and energy points were considered at the same position (x,y), allowing exchanges between layers and optional recharge: (700,800), (200,600) and (500,300). Two other independent energy points were randomly generated.

The upper layer contains ten different packages, at the end of upper layer routing cycle, all goods should be collected and delivered to those supporting spots. While some drones from the lower layer deliver those packages, other ones collect them at distinct exchange points. In this sense, these packages delivered at supporting points are initialized in the subsequent routing problem of the lower layer. Thus, after routing the upper layer, another routing problem is solved for the lower layer, which contains those 10 packages delivered at the supporting points and 15 additional packages, at their respective interaction points.

Layers characteristics are depicted in Table 1.

Six different classes of drones were considered, three for each layer. UAVs parameters are shown in Table 2. As can be verified, vehicles capacity range from [0.3, 5] kg and maximum speed varying from [20, 120] km/h. The number of available units for each type and layer are also detailed.

The average distance to be traveled is the sum of distances between the collecting and delivering points of each target. In the lower layer, this sum is equal to 18,010 m, providing an average distance of 720 m for each of the 25 packages. The sum is 32,900 in the upper layer, resulting in an average value of 3.3 km for each package. This difference exemplifies the real need of using a multi-layer transportation system, since long distances should be covered by more powerful drones, improving the system overall routing efficiency. Regarding the total weight carried in both layers, the sum of the items is 15.3 kg (25 packages) and 6.1 kg (10 products), providing an average weight of 0.61 kg in both cases.

## 4. Computational experiments

This section is divided into four subsections. Section 4.1 presents the computational resources, some considerations about the code and the implemented algorithm. Section 4.2 describes basic configurations set by us in order to run the first batch of experiments. Section 4.3 analyses the obtained results regarding different characteristics of the proposed model. Finally, Section 4.4 describes some characteristics extracted from an obtained Pareto front, as well as a brief analysis of the conflicts between the objective functions.

**Table 3**
Number of lines in the linear programming file.

| $t^{max}$ | Rows | Columns | Nonzeros | Binaries |
|---|---|---|---|---|
| 30 | 151,191 | 55,044 | 453,971 | 46,430 |
| 15 | 64,966 | 23,314 | 192,981 | 19,200 |
| 10 | 36,211 | 12,804 | 105,819 | 10,192 |
| 7 | 19,260 | 6722 | 54,714 | 5010 |

### 4.1. Software and hardware configurations

The core of the MOSPOOLS algorithm was implemented in C++ in the framework OptFrame 2.2[1] (Coelho et al., 2011), which has already been used for tackling other $\mathcal{NP}$-Hard combinatorial optimization problems (Coelho et al., 2016c; 2016e; Souza et al., 2010). It is important to point out that all code used in this research is, from this moment, available as an example on OptFrame core, as an open-source tool under GNU LGPL 3.

The tests were carried out on a OPTIPLEX 9010 Intel Core i7-3770, 3.40 x 8 GHZ with 32GB of RAM, with operating system Ubuntu 14.04, and compiled by g++ 4.8.4, using the Eclipse Kepler Release.

Model optimization (Line 5 of Algorithm 1) was done by the IBM ILOG CPLEX 12.6.3, as a black box tool, calling its dynamic search algorithm. It solves each MILP problem by exploring a BB tree formed by linear programming relaxation nodes.

### 4.2. Initial considerations about the proposed model

An important parameter of the model is the discretization levels $T = \{1, \ldots, t^{max}\}$ associated with the time unit of each interval. In the following batch of experiments, the time unit was set to 1 min, $TimeUnit = 0.0167$. In this sense, different routing horizons from 5 to 30 min are generated with $t^{max} = \{5, 6, 7, 10, 15, 30\}$.

For the lower layer with 5 indexed time intervals, $t^{max} = 5$, 5 min was not enough time for enabling the drones to collect and deliver all products (respecting operational requirements), resulting in an unfeasible problem. For the upper layer, until $t^{max} = 6$ the problem showed to be unfeasible. Table 3 shows some problem parameters, for the lower layer instance, of the reduced MIP after CPLEX pre-solve phase. As can be verified, the discretization level exponentially increases the number of decision variables and constraints.

Right on the empirical analysis, it was noticed that without the MIP start strategy (described in Section 2.3), it would be quite hard to achieve feasible solutions in short computational time for the cases with $t^{max} \geq 15$. A typical execution performed by the black-box solver, considering a single set of $\lambda$ weights, in the problems with $|T| = t^{max} = \{15, 30\}$, took around 3 to 5 min to find the first feasible solution.

Considering these cases, we decided to proceed with the following strategy: increase the optimization time limit until a first feasible solution is found and, then, continue to solve the following MILP problems with the desired fixed optimization time limit *sTLimit* (input of Algorithm 1). An additional parameter *maxNTries*

[1] Available at http://sourceforge.net/projects/optframe/.

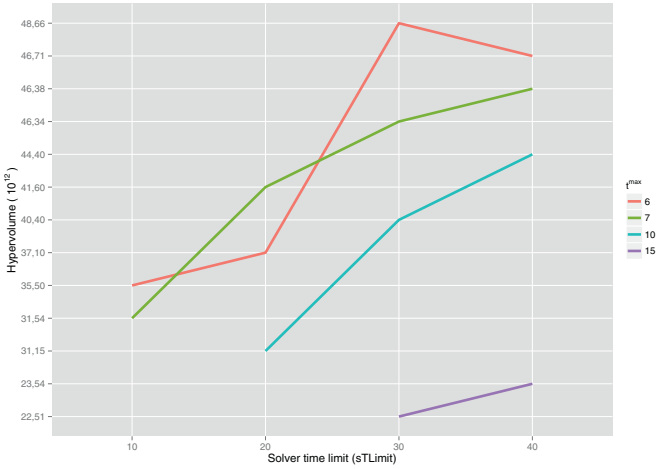


**Fig. 2.** Pareto front quality according to maximum solver time, *sTLimit*, and discretization level, $t^{\max}$.

guides this strategy. In the coming experiments, we set the optimizer to try to find a first feasible solution until *sTLimit*maxNTries* seconds.

The following weights were adopted for each objective function: $\lambda_1 = \{1, 10\}$, $\lambda_2 = \{0.1, 1\}$, $\lambda_3 = \{1, 10\}$, $\lambda_4 = \{1\}$, $\lambda_5 = \{0.01, 0.1\}$, $\lambda_6 = \{1, 10\}$ and $\lambda_7 = \{1, 10\}$. Respectively, objective functions: *dist, time, drones, maxvel, toFull, makespanC* and *makespanD*.

Given all possible combinations of these sets of weights, 64 MILP problems are going to be solved to find solutions for both layers of the case of study.

### 4.3. Lower layer experiments with different parameters

Four different test-problem were generated for the lower layer, considering number of intervals $t^{\max} = \{6, 7, 10, 15\}$. MOSPOOLS algorithm was called 64 times for solving each of these four. As should be noticed, results from any of these cases are able to provide a feasible routing schedule for the lower layer. However, the idea of this batch was to exemplify the effect of this parameter in the optimization process. Furthermore, different time limits were set to the solver, $sTLimit = \{10, 20, 30, 40\}$ s. The maximum number of tries until finding a feasible solution was set to $maxNTries = 200$.

**Table 4**
Coverage and cardinality against the reference set of non-dominated solutions – (cardinality/coverage(%)) .

| $t^{\max}$ | *sTLimit* | | | |
|---|---|---|---|---|
| | 10 | 20 | 30 | 40 |
| 6 | 6/1 | **93/20** | 78/17 | 65/14 |
| 7 | 0/0 | 44/9 | 10/2 | 32/7 |
| 10 | – | 0/0 | 24/5 | 46/10 |
| 15 | – | – | 17/3 | 33/7 |

The quality of the obtained Pareto Fronts was analyzed according to four different indicators: 1 – convergence, analyzed by the Hypervolume (HV) (Zitzler and Thiele, 1998) quality indicator (computed with the tool provided by Beume et al. (2009)); 2 – diversity, measured with the $\Delta$ metric (Deb et al., 2002). 3 – coverage (Zitzler and Thiele, 1998); and 4 – cardinality (Zitzler and Thiele, 1998). Fig. 2 illustrates the obtained results according the HV indicator.

As can be noticed, any feasible solution was found in three combinations of parameters $t^{\max} = 10$ with $sTLimit = 10$ and $t^{\max} = 15$ with $sTLimit = \{10, 20\}$. As already mentioned, the proposed MIP strategy allows the solver to be initialized with different feasible solutions, according to the best one that fits the current model that is being solved. Due to this stochastic search, different quality of Pareto Front may be obtained according to the branches visited in the BB tree.

The problem with less variables, with $t^{\max} = 6$ reported the higher HV and better diversity (according to an analogous analyses with $\Delta$ metric results). A reference Pareto Front, composed of 448 solutions, was created with the union of all sets of non-dominated solutions from each run. Highest coverage values were also achieved with the non-dominated solutions from $t^{\max} = 6$ and $t^{\max} = 10$, as presented in Table 4.

### 4.4. Obtained sets of non-dominated solutions

Due to a high number of objective functions and high dimensionality requested for the trade-off visualization, non-dominated

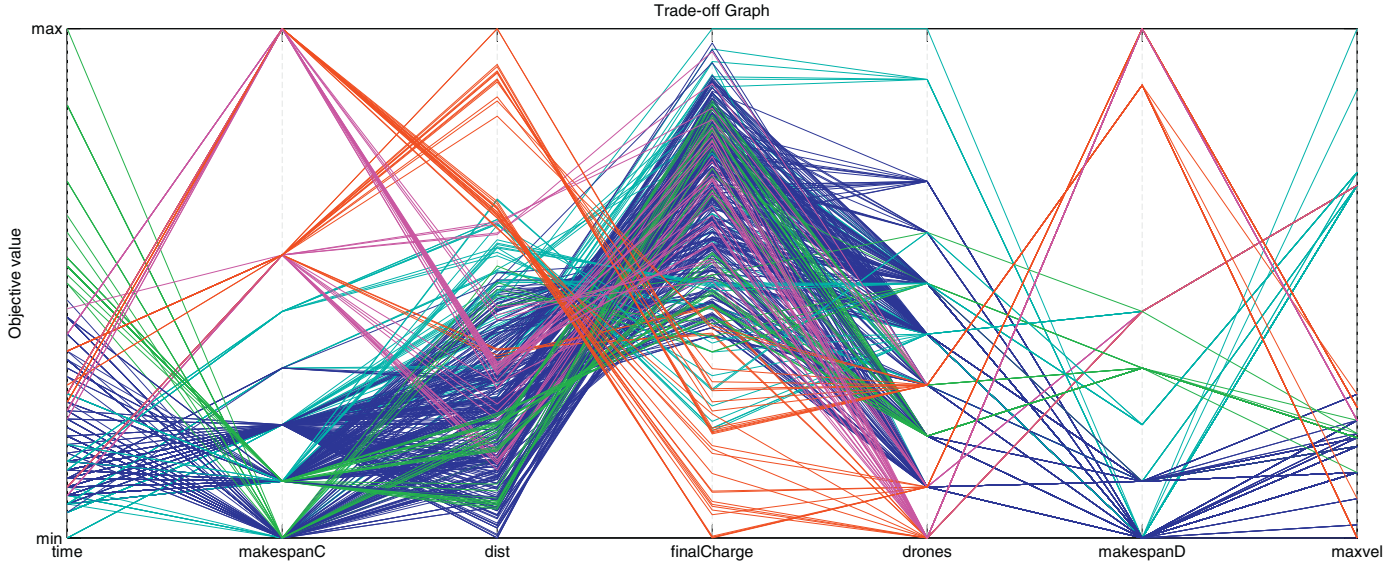


**Fig. 3.** Pareto reference trade-off visualization with a parallel coordinate plot for the lower layer case.

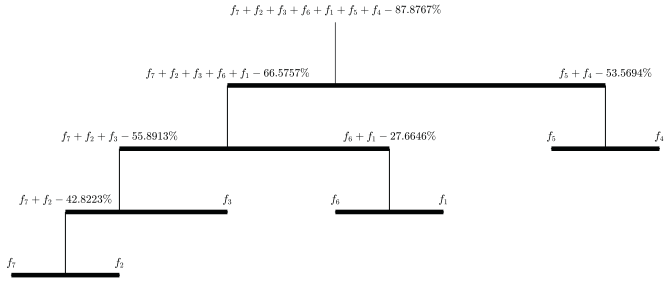


**Fig. 4.** Aggregation tree for the lower layer.

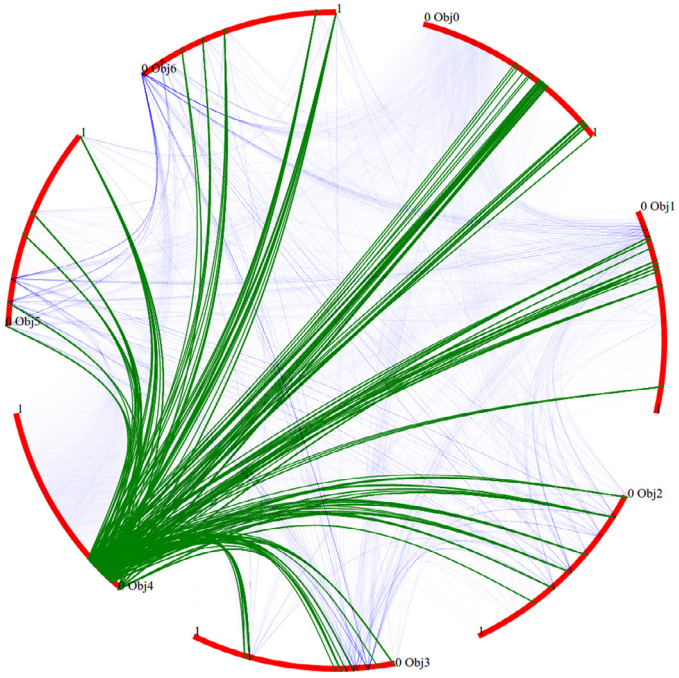


**Fig. 5.** Interactive visualization tool for checking relations between objectives – Pareto reference for the lower layer case.

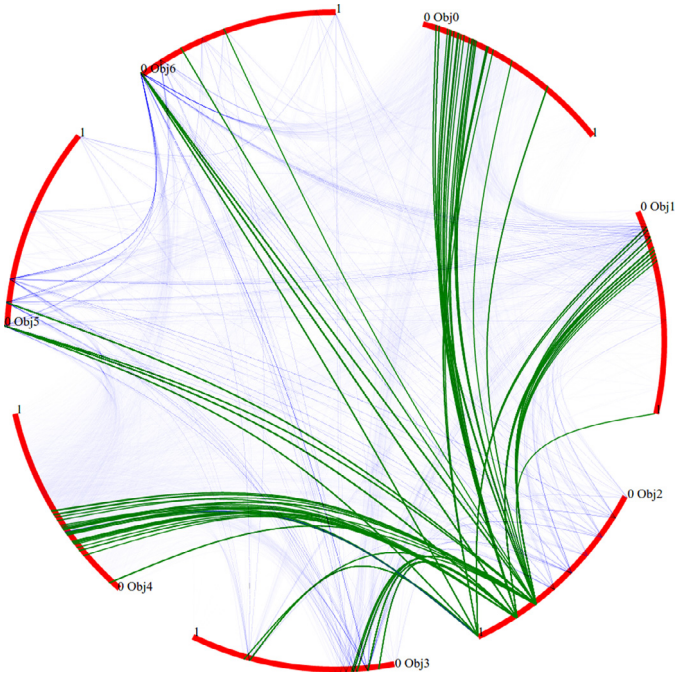


**Fig. 6.** Number of drones (obj2) and last packages collected and delivered (obj5 and obj6) – Pareto reference for the lower layer case.

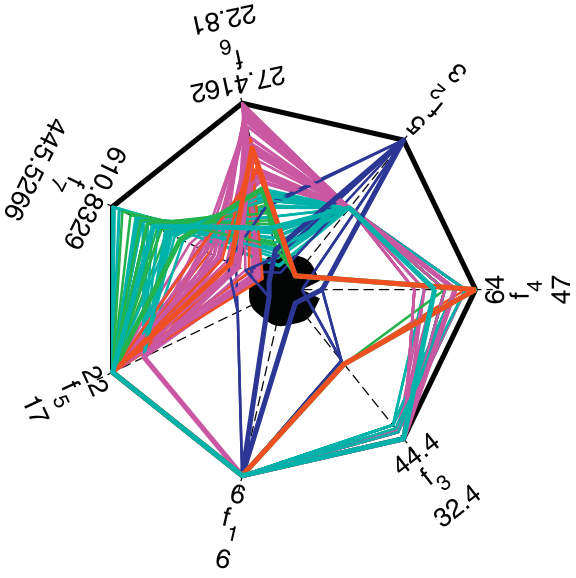


**Fig. 7.** Lower layer trade-off with polar graphs for $t^{\max} = 6$ and $sTLimit = 20$.

solutions are depicted with parallel coordinates, polar graphs and using the tool proposed by Koochaksaraei et al. (2016). For the parallel coordinates and the polar graphs, the order of the objective function in both graphs was chosen with assistance of Aggregation Trees (AT) (de Freitas et al., 2015), defining the order based on the concept of harmony and conflict between objectives.

The obtained Pareto Reference, composed of 448 solutions can be seen in Fig. 3. With assistance of the AT (Fig. 4) and the parallel coordinates graph, we concluded that by minimizing the last collect, *makespanC*, consequently, the model minimized the last delivered, *makespanD*, with high correlation between both objectives.

As can be noticed in Fig. 5[2], finish the schedule with more charged batteries (obj4) implied in solutions that run higher distances (obj0). It was also observed that using more drones resulted in schedules that finish earlier (Fig. 6).

Other sets of non-dominated solutions are shown with one polar graph (Fig. 7) and a parallel coordinate plot (Fig. 8). By solving models with different sizes of the discretization set, $|T| = t_{\max}$, more complete trade-offs were found, specially related to the number of used drones and maximum speed reached during the routing.

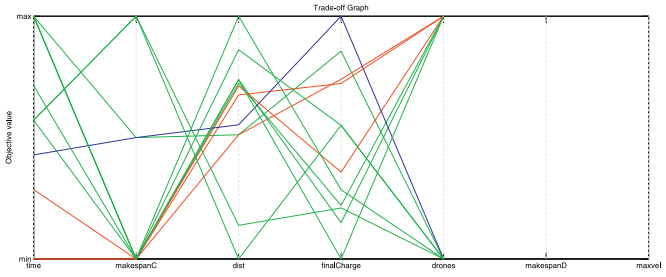


**Fig. 8.** Upper layer trade-off for $t^{\max} = 7$ and $sTLimit = 20$.

**Table 5**
Different non-dominated solutions characteristics .

| Dist. (km) | Sum of Delivering time (min.) | Used Drones | Highest Speed (km/h) | Average energy to Complete batteries (%) | Last package | |
|---|---|---|---|---|---|---|
| | | | | | Collected | Delivered |
| Lower layer | | | | | | |
| 19.8 | 77 | 15 | 33.2 | 37.4 | 4 | 9 |
| 19.9 | 56 | 18 | 30.9 | 32.4 | 5 | 6 |
| 24.5 | 58 | 13 | 33.6 | 36.5 | 12 | 15 |
| 26.3 | 48 | 23 | 41.2 | 27.8 | 3 | 6 |
| 27.2 | 48 | 17 | 39.0 | 26.1 | 4 | 6 |
| 35.4 | 62 | 16 | 34.2 | 22.5 | 8 | 14 |
| Upper layer | | | | | | |
| 30.2 | 33 | 5 | 104.4 | 28.2 | 5 | 7 |
| 32.1 | 29 | 6 | 104.4 | 17.4 | 3 | 7 |
| 32.9 | 36 | 5 | 108.0 | 26.7 | 3 | 7 |

Table 5 shows some characteristics of some of the obtained non-dominated solutions.

As can be verified, an interesting range of distances was found among the obtained solutions, (19.8 km to 35.4 km) and (30.2 km to 32.9 km) for the lower and upper layer, respectively. Different combinations of drones were able to accomplish the requested routing.

## 5. Conclusions and extensions

### 5.1. Summary and final considerations

Inspired by a global trend for using UAVs inside cities, in several daily life applications, we introduced a novel multi-objective UAV routing problem. Bearing in mind the wide range of possible applications, seven different indicators, to be optimized, were considered. A MILP was formulated, considering the possibility of using en route drones. UAVs time-dependent vehicle routing is a problem to be faced when these vehicles are used in our daily life situations. Realistic applications that deal with drones should be able to consider a dynamic environment, with real-time orders being constantly updated. In particular, the proposed model is able to find solutions with UAVs that are already delivering goods. Whenever new environment conditions are detected, the model is executed considering the updated scenario.

Furthermore, the insertion of these drones was contextualized inside in the context of the smart cities, considering microgrid energy systems. In this sense, the model considered vehicles autonomy and the possibility/need of charging them during their routes.

A simple case of study was designed, comprising a bi-layer routing problem composed of different UAVs configurations. Trade-offs were analyzed and conflicts between the objective functions were discussed using visualization tools.

### 5.2. Extensions

An instance generator could be carefully designed for generating a set of benchmark instances. Future works can investigate other UAVs applications that require smart and efficient routing schedules, such as considering paths safety. In terms of computational efficiency, implement a metaheuristic algorithm might be a reasonable approach for achieving good quality solutions in large problems.

[2] Interactive tool available at http://cpdee.ufmg.br/~roozbeh.haghnazar/Harmony/.

## Acknowledgment

Bruno N. Coelho is thankful to the REDEMAT/UFOP and CAPES, for their important support during the development of this project. Vitor N. Coelho would like to thank the Brazilian agency FAPERJ (E-26/202.868/2016). Luiz S. Ochi was supported by FAPERJ and CNPq (301593/2013-2) and Igor M. Coelho by FAPERJ. Finally, we would like to thank the guest editors for their support, opportunity, valuable comments and feedback.

## References

Alighanbari, M., Kuwata, Y., How, J.P., 2003. Coordination and control of multiple uavs with timing constraints and loitering. In: American Control Conference, 2003, 6, pp. 5311–5316. doi:10.1109/ACC.2003.1242572.

Basak, P., Chowdhury, S., nee Dey, S.H., Chowdhury, S.P., 2012. A literature review on integration of distributed energy resources in the perspective of control, protection and stability of microgrid. Renew. Sustain. Energy Rev. 16 (8), 5545–5556.

Berg, P. W., Isaacs, S., Blodgett, K. L., 2016. Airborne fulfillment center utilizing unmanned aerial vehicles for item delivery. US Patent 9305280.

Beume, N., Fonseca, C., Lopez-Ibanez, M., Paquete, L., Vahrenhold, J., 2009. On the complexity of computing the hypervolume indicator. Evol. Comput. IEEE Trans. 13 (5), 1075–1082.

van Blyenburgh, P., 1999. UAVs: an overview. Air Sp. Eur. 1 (5), 43–47. http://dx.doi.org/10.1016/S1290-0958(00)88869-3.

Coelho, I.M., Munhoz, P.L.A., Haddad, M.N., Coelho, V.N., Silva, M.M., Souza, M.J.F., Ochi, L.S., 2011. A computational framework for combinatorial optimization problems. In: VII ALIO/EURO Workshop on Applied Combinatorial Optimization, pp. 51–54.

Coelho, V.N., Coelho, I.M., Coelho, B.N., Cohen, M.W., Reis, A.J., Silva, S.M., Souza, M.J., Fleming, P.J., Guimarães, F.G., 2016a. Multi-objective energy storage power dispatching using plug-in vehicles in a smart-microgrid. Renew. Energy 89, 730–742. http://dx.doi.org/10.1016/j.renene.2015.11.084.

Coelho, V.N., Coelho, I.M., Coelho, B.N., de Oliveira, G.C., Barbosa, A.C., Pereira, L., de Freitas, A., Santos, H.G., Ochi, L.S., Guimarães, F.G., 2016b. A communitarian microgrid storage planning system inside the scope of a smart city. Appl. Energy http://doi.org/10.1016/j.apenergy.2016.12.043.

Coelho, V.N., Coelho, I.M., Coelho, B.N., Reis, A.J., Enayatifar, R., Souza, M.J., Guimarães, F.G., 2016c. A self-adaptive evolutionary fuzzy model for load forecasting problems on smart grid environment. Appl. Energy 169, 567–584. http://dx.doi.org/10.1016/j.apenergy.2016.02.045.

Coelho, V.N., Coelho, I.M., Coelho, B.N., Souza, M.J., Gambini, H., Mladenović, N., Guimarães, F.G., 2016d. A smart pool search matheuristic for solving a multi-objective microgrid storage planning problem. Energy Procedia 103, 292–297. http://dx.doi.org/10.1016/j.egypro.2016.11.288.

Coelho, V.N., Grasas, A., Ramalhinho, H., Coelho, I.M., Souza, M.J.F., Cruz, R.C., 2016e. An ILS-based algorithm to solve a large-scale real heterogeneous fleet VRP with multi-trips and docking constraints. Eur. J. Oper. Res. 250 (2), 367–376. http://dx.doi.org/10.1016/j.ejor.2015.09.047.

Colomina, I., Molina, P., 2014. Unmanned aerial systems for photogrammetry and remote sensing: a review. ISPRS J. Photogram. Remote Sens. 92, 79–97. http://dx.doi.org/10.1016/j.isprsjprs.2014.02.013.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. Evol. Comput. IEEE Trans. 6 (2), 182–197. doi:10.1109/4235.996017.

Enright, J.J., Frazzoli, E., Pavone, M., Savla, K., 2015. Uav routing and coordination in stochastic, dynamic environments. In: Handbook of Unmanned Aerial Vehicles. Springer, pp. 2079–2109.

Enright, J.J., Frazzoli, E., Savla, K., Bullo, F., 2005. On multiple UAV routing with stochastic targets: Performance bounds and algorithms. In: Proceedings of the AIAA Conference on Guidance, Navigation, and Control. American Institute of Aeronautics and Astronautics, pp. 1–15.

Erdogan, S., Miller-Hooks, E., 2012. A green vehicle routing problem. Transport. Res. Part E 48 (1), 100–114. http://dx.doi.org/10.1016/j.tre.2011.08.001.

Fathima, A.H., Palanisamy, K., 2015. Optimization in microgrids with hybrid energy systems – a review. Renew. Sustain. Energy Rev. 45, 431–446. http://dx.doi.org/10.1016/j.rser.2015.01.059.

de Freitas, A.R.R., Fleming, P.J., Guimarães, F.G., 2015. Aggregation trees for visualization and dimension reduction in many-objective optimization. Inf. Sci. (Ny) 298, 288–314. http://dx.doi.org/10.1016/j.ins.2014.11.044.

Frew, E.W., Brown, T.X., 2009. Networking issues for small unmanned aircraft systems. J. Intell. Rob. Syst. 54 (1), 21–37. doi:10.1007/s10846-008-9253-2.

Gertler, J., 2012. US unmanned aerial systems. DTIC Document.

Ground, L., Kott, A., Budd, R., 2002. A Knowledge-Based Tool for Planning of Military Operations: The Coalition Perspective. Technical Report. DTIC Document.

Gupta, S.G., Ghonge, M.M., Jawandhiya, P., 2013. Review of unmanned aircraft system (UAS). Int. J. Adv. Res. Comput. Eng. Technol. 2 (4).

Karnouskos, S., Da Silva, P.G., Ilic, D., 2012. Energy services for the smart grid city. In: 2012 6th IEEE International Conference on Digital Ecosystems and Technologies (DEST). IEEE, pp. 1–6.

Koochaksaraei, H.R., Enayatifar, R., Guimarães, F.G., 2016. A new visualization tool in many-objective optimization problems, pp. 213–224 10.1007/978-3-319-32034-2_18.

Lin, C., Choy, K., Ho, G., Chung, S., Lam, H., 2014. Survey of green vehicle routing problem: past and future trends. Expert Syst. Appl. 41 (4, Part 1), 1118–1138.

Lisso, G. K., 2017. Delivery of packages by unmanned aerial vehicles. Applicant: Amazon Technologies, Inc. US Patent 9536216.

Lust, T., Teghem, J., 2010. Two-phase pareto local search for the biobjective traveling salesman problem. J. Heuristics 16, 475–510.

MEMP, H., 2013, Optimizing Immunization Systems: Delivering Vaccines with Unmanned Aerial Vehicles http://hst.mit.edu/spotlights/optimizing-immunization-systems-delivering-vaccines-unmanned-aerial-vehicles.

Mohammed, F., Idries, A., Mohamed, N., Al-Jaroodi, J., Jawhar, I., 2014. UAVs for smart cities: opportunities and challenges. In: Unmanned Aircraft Systems (ICUAS), 2014 International Conference on. IEEE, pp. 267–273.

Mohammad H.A., Zhao Y., Zhu C., 2016. The future of skyscrapers: a huge drone hive. EVolo Magazine 2016 Skyscraper Competition. Acessado em 10 de julho de 2016.

Newcome, L.R., 2004. Unmanned Aviation: A Brief History of Unmanned Aerial Vehicles. AIAA – American Institute of Aeronautics and Astronomics.

Nex, F., Remondino, F., 2014. Uav for 3d mapping applications: a review. Appl. Geomatics 6 (1), 1–15.

Pohl, A.J., Lamont, G.B., 2008. Multi-objective UAV mission planning using evolutionary computation. In: Proceedings of the 40th Conference on Winter Simulation. Winter Simulation Conference, pp. 1268–1279.

Souza, M.J.F., Coelho, I.M., Ribas, S., Santos, H.G., Merschmann, L.H.C., 2010. A hybrid heuristic algorithm for the open-pit-mining operational planning problem. Eur. J. Oper. Res. 207 (2), 1041–1051.

Spinka, O., Holub, O., Hanzalek, Z., 2011. Low-cost reconfigurable control system for small UAVs. IEEE Trans. Ind. Electron. 58 (3), 880–889.

Sundar, K., Rathinam, S., 2014. Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. IEEE Trans. Autom. Sci. Eng. 11 (1), 287–294. doi:10.1109/TASE.2013.2279544.

Turner, I.L., Harley, M.D., Drummond, C.D., 2016. UAVS for coastal surveying. Coastal Eng. 114, 19–24. http://dx.doi.org/10.1016/j.coastaleng.2016.03.011.

Weinstein, A.L., Schumacher, C., 2007. Uav scheduling via the vehicle routing problem with time windows. In: Proceedings AIAA Infotech@ Aerospace 2007 Conference and Exhibit.

Zitzler, E., Thiele, L., 1998. Multiobjective optimization using evolutionary algorithms - a comparative case study. In: Eiben, A., Back, T., Schoenauer, M., Schwefel, H.-P. (Eds.), Parallel Problem Solving from Nature - PPSN V. In: Lecture Notes in Computer Science, 1498. Springer Berlin / Heidelberg, pp. 292–301.