

Drone-assisted deliveries: new formulations for the Flying Sidekick Traveling Salesman Problem

Mauro Dell’Amico · Roberto Montemanni · Stefano Novellani

Received: 29/03/2019 / Accepted: date

Abstract In this paper we consider a problem related to deliveries assisted by an unmanned aerial vehicle, so-called drone. In particular we consider the Flying Sidekick Traveling Salesman Problem, where a truck and a drone cooperate to delivery parcels to customers minimizing the completion time. In the following we improve the formulation found in the related literature. We propose three-indexed and two-indexed formulations and a set of inequalities that can be implemented in a branch-and-cut fashion. We could find the optimal solutions for most of the literature instances. Moreover, we consider two versions of the problem: one in which the drone is allowed to wait at the customers, as in the literature, and one where waiting is allowed only in flying mode. The solving methodologies are adapted to both versions. A comparison between the two versions is provided.

Keywords aerial drones · routing · branch-and-cut · parcel deliveries · formulations

Mauro Dell’Amico
Dipartimento di Scienze e Metodi dell’Ingegneria (DISMI), Università di Modena e Reggio Emilia (UNIMORE)
via Amendola 2, 42122 Reggio Emilia, Italy

Roberto Montemanni
Dipartimento di Scienze e Metodi dell’Ingegneria (DISMI), Università di Modena e Reggio Emilia (UNIMORE)
via Amendola 2, 42122 Reggio Emilia, Italy

Stefano Novellani
Dipartimento di Scienze e Metodi dell’Ingegneria (DISMI), Università di Modena e Reggio Emilia (UNIMORE)
via Amendola 2, 42122 Reggio Emilia, Italy
Tel.: +39-0522-523537
E-mail: stefano.novellani@unimore.it

1 Introduction

Once restricted to the military domain, *unmanned aerial vehicles* (UAV), also known as (aerial) drones, have received a widespread adoption in civil applications by the humanitarian sector and are currently object of great interest of the commercial sector. This is due to their capability of accomplishing otherwise impossible tasks, performing activities more effectively, or to lead to cost saving solutions. A comprehensive definition of drones that is the one provided by Scott and Scott [47]: “*Drones are devices which are capable of sustained flight, which do not have human on board, and are under sufficient control to perform useful functions*”.

During the last years, drones have interested researchers whose effort was put especially in improving the required technology:

- (a) the hardware, by reducing weight, increasing battery duration, improving the charging system, improving safety, etc.,
- (b) the software, by designing better autonomous operations and guidance systems using improved GPS accuracy, localization techniques, obstacle detection and avoidance techniques, enhanced sensors, image processing, etc.,
- (c) the safety and security elements, by adopting technologies to protect the flight against spoofing and hijacking of UAVs, etc.

A smaller effort has been done for the solution of optimization and operational problems related to the use of drones, although the number of paper in this direction has strongly increased in the very last years.

The paper is organized as follows: we start with Section 2 where we summarize the main real-life applications that use drones. In Section 3 we discuss upon the most relevant related literature, while in Section 4 we formally describe the problem. Mathematical formulations and their implementations are described in Sections 5 and 6. Extensive computational results comparing the models performances are presented in Section 7. The last Section 8 concludes the paper.

2 Applications

Early drones application arose in the military sector, where most of the optimization works are focused on the *path planning*, normally to avoid obstacles or radars, minimizing travel length or altitude, or to improve the usage of batteries. In this paper we are not interested in listing the military applications, but we address the interested reader to the following works: Bortoff [21], Richards et al. [42], Zheng et al. [55], and Roberge et al. [43].

Besides military applications, drones can be used in both humanitarian and commercial sectors. The three sectors make use of drones in such a way that can be divided in two main applications: to collect and deliver information and to collect and deliver goods. On one hand we have the surveillance, monitoring, or covering, and on the other hand deliveries. In the following we describe a large set of applications adhering to this dichotomy.

2.1 Gathering information: Surveillance, monitoring and covering

The information gathering applications include surveillance, monitoring, and covering activities, where displacement of goods is not necessary, in which drones fly autonomously and monitor the environment with different sensors or cameras and communicate and exchange data and information with other drones or with a central station.

Drones can be used to optimize the coverage of an area (see, e.g., Shang et al. [48]) or following specific targets that can also be moving ones (see, e.g., Zorbas et al. [56] and Di Puglia Pugliese et al. [41]). This is done by defining the position of drones, their path (see, e.g., Kashuba et al. [33]), their number, etc., while maximizing the coverage or minimizing the time between an appearance of an event and its covering, the total length, the service costs in energy, etc. In particular, typical applications regard the following domains: traffic management, environmental monitoring, catastrophic events, remote locations surveillance, precision agriculture, building inspection, security surveillance, etc.

2.2 Moving goods: Deliveries

This second set of applications is characterized by the fact that real goods need to be moved from one place to another. The main application is to provide faster, more cost efficient goods or parcel delivery, especially in the last-mile, but also mail delivery and quickly required medical items delivery have been object of interest. Indeed, in emergency and events management drones can be used when infrastructure are damaged or unreliable to access isolated regions to deliver needed goods in addition to surveillance. This point can match the medical items delivery: delivery of blood, medications, vaccines, defibrillators, insulin, oxygen, or other needed health-care items in location with difficult access due to poor infrastructure, remote areas, traffic congestion, inaccessible roads due to weather or disasters, or simply urgently needed. Drone can transport water, food, medical supplies during a crisis or and event.

Commercial applications arise when one needs to provide faster, more cost efficient goods or parcel delivery, especially in the last-mile. Several companies are investigating the use of drones for parcel delivery for e-commerce, such as Amazon [4], Alibaba [3], and Alphabet [17]. Amazon CEO announced Amazon's Prime Air, that uses a fleet of UAVs to deliver parcels from warehouses to customers [4]. Australian textbook distributor Zookal started testing drone parcel deliveries in Australia [18]. DHL Parcel operates an autonomous drone delivery system, to deliver medications and other urgent goods to one of the Germany's North Sea islands of Juist [8]. In 2016 they also tested a delivery system in the Bavaria alpine region typically carrying either sporting goods or urgently needed medicines [13]. Chinese JD.com deploys drones to extend their delivery and logistics network. They consider to use drones in areas with complex terrain and poor infrastructure for last-mile delivery. They started in

2016 with four rural locations of China in the outskirts of Beijing and in the provinces of Jiangsu, Shaanxi, and Sichuan. They announced an agreement with the Shaanxi provincial government to build China’s largest low altitude drone network, serving a 300 km area with drone stations and routes to delivery e-commerce parcels [11,12]. They will build 150 drone launch facilities in China by 2020 [10]. Also Alphabet (Google) announced to enter the field with Project Wing [17]. UPS and DPDgroup are also testing parcel delivery with drones [15,7]. UPS and Zipline are working on a drone network to deliver vaccines and blood to 20 clinics in remote locations in Rwanda [9]. Only 20% of Africans live within 2 km of a road that functions year-round, thus malaria medications, rabies vaccines, etc. can be carried and left with a parachute (see Scott and Scott [47]). UPS and Workhorse are also testing drone and truck combo delivery [14]. Flirtey completed the first Federal Aviation Administration approved drone delivery in July 2015, when it delivered medical supplies to a health clinic in Wise, Virginia. The company has also started a partnership with 7-Eleven for home delivery [1]. In the Netherlands and Sweden a prototype ambulance drone has also been tested for delivering defibrillators [5,6]. Matternet provides an on-demand delivery platform, an end-to-end solution integrating the Matternet’s drones and stations. They provide their platform as a service to healthcare, e-commerce and logistics organizations. In particular, they transport medical items between health-care facilities in Switzerland. They perform drone deliver in Zurich. In this case Matternet and Mercedes-Benz have joined forces to create delivery solutions integrating vans and drones for siroop online shop [16]. The United Arab Emirates say they plan to use unmanned aerial drones to deliver official documents and packages to its citizens as part of efforts to upgrade government services [2].

This is just an incomplete list of many of the cases that arose in the field in the last years. The large number make us suspect an increasing interest in the field in the future.

3 Related literature

The FSTSP is a generalization of the (TSP) and the *vehicle routing problem* (VRP). A large body of literature has been dedicated to these problems; however, we are interested only in those problems that are highly related to the FSTSP, especially where drones and trucks are coupled and synchronized.

A problem that is conceptually related to the FSTSP is the *Close enough traveling salesman problem* (see, e.g., Shuttleworth et al. [50]), that aims at finding the cheapest route for the truck without visiting every customer on his route, but only getting within a certain radius of each customer. A similar problem and even more conceptually related is the *covering salesman problem* (see, e.g., Current and Schilling [26]), that aims at finding the cheapest tour such that all nodes that are not part of the route lie within a specified radius

from a node on the truck route. This reminds the maximum radius that the drone can travel.

In the FSTSP, the vehicle that arrives first at the meeting point has to wait for the other, thus we can state that it lies in the class of problems that require synchronization between vehicles, in particular, among the categories defined by Drexel [30], the FSTSP can be considered under *movement synchronization en route*, where vehicles may join and separate multiple times along a route. The author claims that this class received little attention in literature. One of these problems is also one of the most related to our problem: the *Truck and Trailer Routing Problem* (TRRP) (see e.g., Chao [25]), where two different types of vehicle can serve customers: trucks and trailers. Due to practical constraints (e.g. street size), some customers can only be served by a truck, while other customers can be served by a truck or a truck pulling a trailer. A truck is autonomous, while the trailers always need to be pulled by a truck. Parking places are used to decouple trucks from trailer when convenient to serve some subset of customers that can only be served by single truck. This resembles the idea of the drone truck decoupling. A similar problem to the FSTSP has been studied by Lin [34], where two types of delivery resources are used, vans and foot couriers, which allows for coordination. A heavy resource (a van) may carry both delivery items and one or more units of the lighter resource (foot couriers). Foot couriers can pick up and deliver items independently or travel with a van on its outbound and/or return leg, they can serve more than one customers, in contrary of what a drone can do. Moreover, foot couriers do not need to return to the same van.

Let us now consider works that solve problems related to the use of drones. As said, the first applications of UAVs have been performed in the military sector, and the same can be said for the use of optimization techniques when optimizing operations for drones. In Sisson's thesis [51] it is studied a tabu search coupled with Monte Carlo simulation to determine the minimum number of UAVs to cover a pre-selected target set based on stochastic survival probabilities that also incorporate the wind effects. In Ryan's thesis [44] a tabu search within a discrete event simulation is applied to solve a multi TSP with time windows for UAVs, where it is required to attain a level of target coverage using a minimum number of vehicles. Weather and threat are considered. A direct extension of this work is proposed in Ryan et al. [45]. O'Rourke et al. [37] consider dynamic routing of UAVs in operational use with the US Air Force. Dynamic components are wind and emerging targets. They model the problem as a VRP with time windows and use a tabu search to solve it.

Boone et al. [20] solve a multi TSP for drones, they firstly divide customers in clusters by using K-mean clustering, then they solve a TSP for each cluster with a nearest neighbor improved with 2-opt. Dorling et al. [29] study a multi trip VRP for drone delivery where the effect of batteries and weight on energy consumption is considered. They solve one version of the problem in which costs are minimized under a delivery time limit and a second version where delivery time is minimized subject to a budget constraint. The authors

present MILP formulations and a simulated annealing algorithm. Tseng et al. [52] solve a modified TSP where an autonomous drone has to serve delivery points and may use charging stations for charging the battery. They consider wind uncertainty and variable speeds. The used algorithm is based on the Christofides one to solve the shortest spanning tree and then the minimum matching problem to obtain an Eulerian tour.

Location problems linked to drones are studied in the two following works. Scott and Scott [47] consider drone delivery models for healthcare. They consider delivery with trucks that leave a central depot to drone nests, and then from drone nests to delivery points. They use two objective functions: one minimizes the total time and the other minimizes the maximum weighted time for truck/drone delivery. Both depot and nests need to be located. No routing is considered but only direct trips. Shavarani et al. [49] studied the facility location problem for the optimization of drone delivery system evaluating the Amazon prime air case study in San Francisco. They locate both launching stations and recharge stations. No routing is considered but direct trips. The authors solve the problem with a genetic algorithm.

Campbell et al. [24] present some drone arc routing problems, that are continuous optimization problems. They discretized them by approximating the curves with polygonal chains. The drone rural postman problem, in which drones have no capacity, is solved with iterative algorithms for the rural postman problem. They discuss also other problems where drones are capacitated or more than one.

The following papers consider the coupled interventions of drones and trucks. The first paper we consider is slightly different from the others, indeed Savuran and Karakaya [46] study a problem where a truck follows a linear path, in the meanwhile a drone is launched from the truck and must return to the truck after performing a, let's say, open TSP to visit all the targets. They solve the problem with a genetic algorithm. Ferrandez et al. [31] propose a work that couples drones and a truck. They have a set of delivery customers that are clusterized by using K-means clusterization. They thus solve the TSP among the centroid of each cluster that are the points where the truck stops to launch one or more drones. The TSP part is solved with a genetic algorithm. Boysen et al. [23] consider a fixed truck route where the truck represents a loading platform for the drones. The truck and the drone can wait each other. They use at most 2 drones and three restriction of the problem: one where the drone must return to the launching point, one in which the drone can return up to the next vertex, and the last one where the drones can return in one of the following nodes.

In the following we consider problems where drones and vehicles work together to complete operations, in which both can accomplish tasks. In this first part we evaluate what Otto et al. [38] classify as *Drones and vehicles performing independent tasks*. Murray and Chu [36] propose the *parallel drone scheduling TSP* (PDSTSP), where one truck and a fleet of drones can serve

customers only departing from the depot. In this case only customers within a certain range from the depot can be served via drones. The others are served by the truck. Solving this problem can provide good results when many customers lie close to the depot. They propose MILP formulations and simple greedy heuristics for both problems. Saleu et al. [35] propose a two step iterative heuristic based on dynamic programming for the same problem. Ulmer and Thomas [53] study a dynamic variant of the PDSTSP called the *Same-day delivery with heterogeneous fleets of drones and vehicles*, where requests arrive dynamically and they need to be allocated to drones or truck maximizing the number of served customers. They solve the problem with an approximate dynamic programming known as parametric policy function approximation.

We consider, now, routing problems where trucks are equipped with drones and both vehicles can be used to deliver packages to customers. Otto et al. [38] classify these problems under the name *Drones and vehicles as synchronized working units*. In the case of the drone, the flight is called *sortie*: the drone start from its vehicle in a vertex of the network (launch), performs a delivery to a customer, and returns to the truck (rendezvous) in a vertex of the network. Some problems consider the launching and rendezvous times negligible, some other account for these; however, truck and drone must be synchronized and thus wait for each other. The objective function is normally to minimize the completion time. Murray and Chu [36] define and study the *Flying side-kick traveling salesman problem* (FSTSP). In the FSTSP truck and drone can cooperate to serve customers: one drone can leave the truck at a vertex and return to the truck at another vertex after completing a delivery. Customers can be visited only once, but some customers can be visited only by the truck because their request cannot be fulfilled by the drone (for various reasons, such as capacity limitations, requirement of a signature, drone cannot safely land, etc.). Drones cannot return at the launching point. Agatz et al. [19] solve the *TSP with Drone* (TSP-D), where one truck cooperates with one drone to make deliveries. The aim is to find the fastest method to serve customers with a truck or a drone that can leave the truck, serve a customer, and return to the truck in one of the vertices. Each customer has to be visited at least once by one of the vehicles, but they can be visited more than once by the truck if it is convenient for drone launching and return. Launching and rendezvous of the same sortie can coincide. Some vertices cannot be visited by drones. Endurance is unlimited and launching and rendezvous times are considered negligible. The authors preset an ILP and propose route first-cluster second heuristics based on local search and dynamic programming. Bouman et al. [22] solve the TSP-D with with dynamic programming. The drone is not slower than the truck. Instances with up to 20 customers are solved. Ha et al. [32] also use the name TSP-D, albeit in this case nodes cannot be visited multiple times and launch and rendezvous of a sortie are not allowed to happen at the same vertex. The authors declare that the considered problem shared the FSTSP characteristics, but they called it TSP-D. To highlight the difference between the two problems we call this problem TSP-D*. The authors propose two heuristic algorithms: a route first-cluster second one and a cluster first-

route second one. Ponza’s thesis [40] tackles the FSTSP proposing a modified MILP formulation with respect to the Murray and Chu’s [36] one and solve it with a simulated annealing algorithm. The new formulation, with respect to the FSTPS one, among the other few differences, does not allow the drone to wait at customers nodes.

Wang et al. [54] define the *vehicle routing problem with drones* (VRPD), where a homogeneous fleet of trucks equipped with a not necessary unitary number of drones delivers parcels to customers. Drones can be launched from trucks at depot or at any customer vertex. Each drone must return to the same truck also if this happens at the same node where it has been launched. One drone can carry only one parcel. The authors imagine that the drone must travel along the street network. They derive a number of worst case results based on the number of drones per truck and the difference between the drones speed and the trucks speed. Poikonen et al. [39] extend the worst-case results considering different metrics for trucks and drones, considering limited drone batteries, and evaluating different objective functions. They also evaluate connections between the VRPD and other VRP variants. Daknama and Kraus [27] present and solve the *Vehicle routing with drones* where multiple vehicles and drones can be used for deliveries. They minimize the average delivery times instead of the completion time. No mathematical model is presented, but they solve the problem by first solving a multiple TSP heuristically and then introducing drones. Local search procedures are thus applied to improve the solution.

In Table 1 we classify the truck and drone problems. In *Paper* column we report the authors and the reference to the paper, in the second column we report the name of the problem as by the papers (but we include the renaming of TSP-D* to avoid confusion: we recall that Agatz et al. [19] and Ha et al. [32] use the same name for different problems). In $\#t$ and $\#d$ we report the considered number of trucks and drones, respectively. If trucks and drones are multiple we identify it with the letter m . In following columns we consider some features of the problems: the fact that launch and return points may coincide ($L = R$); the fact that trucks may visit vertices multiple times ($m - v$); the fact that some vertices can be visited only by trucks ($t - v$). In last column we provide a rough description of the method used to solve the proposed problems, if any. Table 1 is inspired by the tables proposed by Ponza [40] that we adjusted referring only to the problems where drones and vehicles are considered as synchronized working units and by adding the fact that the studied problems allow the following points: (i) the possibility of having multiple visits; (ii) the possibility of having nodes to be visited only by the truck.

A recent survey by Otto et al. [38] provides a wide overview on civil applications of drones, it gives an insight into optimization approaches used to solve operational problems, in particular where both drones and drones combined with other vehicles are considered, that is the class of problems addressed in this work.

Paper	Name	Formul.	#t	#d	L=R	m-v	t-v	Solving method
Murray and Chu [36]	FSTSP	MILP 2-index	1	1			x	MILP, heuristic (nearest neighbour, savings, sweep + savings for UAV routes definition)
Agatz et al. [19]	TSP-D	ILP set covering	1	1	x	x	x	Heuristic (route first cluster second)
Bouman et al. [22]	TSP-D	none	1	1	x	x	x	Dynamic programming
Ha et al. [32]	TSP-D*	MILP for clustering, set packing	1	1				Heuristic (route first cluster second, cluster first route second)
Ponza [40]	FSTSP	MILP, 2-index	1	1			x	MILP, simulated annealing
Wang et al. [54]	VRPD	none	m	m	x			none
Poikonen et al. [39]	VRPD	none	m	m	x			none
Daknama and Kraus [27]	VRD	none	m	m				Metaheuristic, Local search

Table 1: Classification of problems where drones and vehicles are synchronized working units.

4 Problem Description and Basic Mathematical Model

In this work we study the FSTSP defined by Murray and Chu [36], that is to serve a set of customers $C = \{1, \dots, c\}$ with either a truck or a drone. The truck starts from depot 0 and returns to the final depot $c+1$, and is equipped with a flying drone that can be used in parallel to serve one customer at a time. The drone can perform a *sortie*, defined by a launching node (where the drone leaves the truck), a served customer, and a rendezvous node (where the drone returns to the truck), that must be different from the launching one. All customers of C can be served by the truck, but only a subset $C' \subseteq C$ can be served by the drone with a sortie. The problem is built on digraph $G = (N, A)$, where the set $N = \{0, 1, \dots, c+1\}$ represents all the nodes, while we define $N_0 = \{0, 1, \dots, c\}$ and $N_+ = \{1, \dots, c+1\}$. The set A is the set of all the arcs $(i, j), i \in N_0, j \in N_+, i \neq j$. Each arc (i, j) is associated with two non-negatives traveling times: τ_{ij}^T and τ_{ij}^D , that represent the time for traveling that arc by the truck and by the drone, respectively. The travel time matrix of the drone and the truck may be different ($\tau_{ij}^T \leq \tau_{ij}^D, (i, j) \in A$). Nodes 0 and $c+1$ represent the same physical point, the depot, and the traveling time between them is set to 0 to account for the case in which there is only one customer served by the drone directly from the depot.

We assume that the capacity of the truck is large enough to serve all customers and that the drone performs only one delivery at the time, i.e., it leaves the truck, serves a customer and returns to the truck before possibly serving a new customer. A sortie is formally defined by a triplet $\langle i, j, k \rangle$, ($i \neq j \neq k$) where $i \in N_0$ is the launching node, $j \in C'$ the customer to serve, and $k \in N_+$ the rendezvous node.

Serving times at customers for both drone and truck are negligible. Service times for preparing the drone at launch and rendezvous are given by σ^L and σ^R . Drones have a battery limit (endurance) of E time units, that limits drone

use. Rendezvous time σ^R contributes to the endurance computation while σ^L does not, since the drone lies on the truck when it is prepared for launch.

Let F be the set of all sorties that can be performed within the endurance E , that means $\tau_{ij}^D + \tau_{jk}^D + \sigma_R \leq E$, $i \in N_0, j \in C', k \in N_+$.

Drones can be launched from trucks only when the truck is stopped at customers points or at the depot, however, they cannot leave the depot before the truck starts its route. In the meanwhile a truck can keep serving customers and the drone can return only at another point, that is not the launching one (if not the depot). This requires a certain synchronization: the vehicle (drone or truck) that arrives first at the meeting point has to wait for the other. The objective is to minimize the completion time, that is the moment when the last vehicle arrives at the depot.

4.1 Murray-Chu Formulation

We report in this section the mathematical formulation proposed by Murray and Chu [36] (MC in the following) that is the basis for our work. A binary variable x_{ij} is created for each arc $(i, j) \in A : i \neq j$ and is set to 1 if the truck uses the corresponding arc, 0 otherwise. For each sortie of the drone is used a binary variable $y_{ijk}, \langle i, j, k \rangle \in F$ that takes value one if the sortie is performed, 0 otherwise.

The non-negative variables $t_i^T, i \in N$ and $t_i^D, i \in N$ represent the time of availability at node i for the truck and the drone, respectively. For the starting depot we fix $t_0^T = t_0^D = 0$. Note that the drone and the truck may leave uncoupled at the starting depot. The waiting of the truck (resp. of the drone) at a node i is modeled as a delayed availability and included in t_i^T (resp. t_i^D).

The MC formulation uses two sets of auxiliary variables. A variable $u_i, i \in N, 1 \leq u_i \leq c + 2$ is used to model subtour elimination constraints for the truck route, as in Miller-Tucker-Zemlin (see, e.g., Desrochers and Laporte [28]). Finally, binary variable p_{ij} is set to 1 if customer $i \in C$ is visited before customer $j \in C$ ($j \neq i$), thus defining a total ordering among all the pair of customers. We set $p_{0j} = 1, j \in C$ to impose the starting of the route from the depot.

The resulting model for the MC formulation is described in the following, presenting one group of logically related constraints at a time.

The objective function (1) minimizes the arrival time at the depot of the truck, but due to next constraints (12) and (13) this is equivalent to minimize $\max\{t_{c+1}^T, t_{c+1}^D\}$.

$$\min t_{c+1}^T \tag{1}$$

Customer covering

The first constraints impose that all customers must be served either by the

truck or by the drone:

$$\sum_{(i,j) \in A} x_{ij} + \sum_{\langle i,j,k \rangle \in F} y_{ijk} = 1 \quad j \in C \quad (2)$$

Truck routing constraints

Constraints (3) and (4) force the truck to depart from depot 0 and to return to depot $c + 1$ at the end of the trip, while constraints (5) guarantee the flow conservation at customers.

$$\sum_{j \in N_+} x_{0j} = 1 \quad (3)$$

$$\sum_{i \in N_0} x_{i,c+1} = 1 \quad (4)$$

$$\sum_{(i,j) \in A} x_{ij} = \sum_{(j,k) \in A} x_{jk} \quad j \in C \quad (5)$$

Single sortie leave/return

Constraints (6)–(7) impose that at most one launch of drone and one rendezvous with the truck is done in any node. The drone cannot start its sorties in $c + 1$ or return to 0.

$$\sum_{\langle i,j,k \rangle \in F} y_{ijk} \leq 1 \quad i \in N_0 \quad (6)$$

$$\sum_{\langle i,j,k \rangle \in F} y_{ijk} \leq 1 \quad k \in N_+ \quad (7)$$

x-y coupling constraints

In constraints (8) we impose that, if the triplet $\langle i, j, k \rangle$ is selected, then the truck must enter in node i and in node k to launch and collect the drone. Constraints (9) is the equivalent of (8) when the drone is launched from the depot.

$$2y_{ijk} \leq \sum_{(h,i) \in A} x_{hi} + \sum_{(l,k) \in A} x_{lk} \quad \langle i, j, k \rangle \in F, i \in N_+ \quad (8)$$

$$y_{0jk} \leq \sum_{(h,k) \in A} x_{hk} \quad \langle 0, j, k \rangle \in F \quad (9)$$

Truck-drone timing constraints

Constraints from (10) to (13) ensure time synchronization between the truck and the drone at launching and return node. Constraints (14) and (15) set the starting time at depot of both drone and truck to zero.

$$t_i^D \geq t_i^T - M(1 - \sum_{\langle i,j,k \rangle \in F} y_{ijk}) \quad i \in C \quad (10)$$

$$t_i^D \leq t_i^T + M(1 - \sum_{\langle i,j,k \rangle \in F} y_{ijk}) \quad i \in C \quad (11)$$

$$t_k^D \geq t_k^T - M(1 - \sum_{\langle i,j,k \rangle \in F} y_{ijk}) \quad k \in N_+ \quad (12)$$

$$t_k^D \leq t_k^T + M(1 - \sum_{\langle i,j,k \rangle \in F} y_{ijk}) \quad k \in N_+ \quad (13)$$

$$t_0^T = 0 \quad (14)$$

$$t_0^D = 0 \quad (15)$$

Truck timing constraints

Constraints (16) state that if the arc $(h, k) \in A$ is used by the truck, then the timing variables must be consistent with the travelling times and the launch and rendezvous times, if drone is used. We will discuss these constraints in detail in Section 4.2.

$$t_k^T \geq t_h^T + \tau_{hk}^T + \sigma_R \sum_{\langle i,j,k \rangle \in F} y_{ijk} + \sigma_L \sum_{\substack{\langle k,l,m \rangle \in F \\ l \neq h}} y_{klm} - M(1 - x_{hk}) \quad (h, k) \in A \quad (16)$$

Drone timing constraints

Constraints (17)-(18) impose consistency on the drone timing variables when a sortie $\langle i, j, k \rangle$ is selected.

$$t_j^D \geq t_i^D + \tau_{ij}^D - M(1 - \sum_{\langle i,j,k \rangle \in F} y_{ijk}) \quad (i, j) \in A, j \in C' \quad (17)$$

$$t_k^D \geq t_j^D + \tau_{jk}^D + \sigma_R - M(1 - \sum_{\langle i,j,k \rangle \in F} y_{ijk}) \quad (j, k) \in A, j \in C' \quad (18)$$

Note that these constraints allow t_j^D to be greater than the arrival time at customer j , and consequently to insert a waiting time at j .

Drone battery endurance constraints

Constraints (19) are the battery endurance constraints, for each sortie.

$$t_k^D - (t_j^D - \tau_{ij}^D) \leq E + M(1 - y_{ijk}) \quad \langle i, j, k \rangle \in F \quad (19)$$

We observe that inequalities (19) do not consider the waiting at j in the computation of the energy consumption, i.e., we can therefore suppose that the drone is allowed to wait on the ground.

Miller-Tucker-Zemlin subtour elimination

Constraints (20) are the classical Miller-Tucker-Zemlin constraints for the truck route, while (21) are the extension to sorties, and impose that node

i precedes node k if sortie $\langle i, j, k \rangle$ is selected.

$$u_i - u_j + 1 \leq (c + 2)(1 - x_{ij}) \quad (i, j) \in A, i \in N_+ \quad (20)$$

$$u_k - u_i \geq 1 - (c + 2)\left(1 - \sum_{\langle i, j, k \rangle \in F} y_{ijk}\right) \quad i \in C, k \in N_+, k \neq i \quad (21)$$

Node total ordering

Constraints (22)–(23) induce variables p_{ij} to define a total ordering of the nodes.

$$p_{ij} + p_{ji} = 1 \quad (i, j) \in A, i < j \quad (22)$$

$$p_{0j} = 1 \quad j \in C \quad (23)$$

u - p congruence

These constraints impose the consistency between the ordering given by u and p variables.

$$u_i - u_j \geq 1 - (c + 2)p_{ij} \quad i, j \in C, j \neq i \quad (24)$$

$$u_i - u_j \leq -1 + (c + 2)(1 - p_{ij}) \quad i, j \in C, j \neq i \quad (25)$$

Simple crossing sorties elimination

Constraints (26) avoid that a sortie starts before a previous sortie is terminated. More specifically, if node i is visited by the truck before node l (i.e., $p_{il} = 1$) and there are two sorties starting from i and l , respectively, than the rendezvous node k of the sortie from i , must be visited before node l .

$$t_l^D \geq t_k^D - M\left(3 - \sum_{\substack{\langle i, j, k \rangle \in F \\ j \neq l}} y_{ijk} - \sum_{\substack{\langle l, m, n \rangle \in F \\ m, n \notin \{i, k\}}} y_{lmn} - p_{il}\right) \quad (26)$$

$$i \in N_0, l \in C, k \in N_+, i \neq l \neq k$$

Variable bounds

$$t_i^T \geq 0 \quad i \in N \quad (27)$$

$$t_i^D \geq 0 \quad i \in N \quad (28)$$

$$p_{ij} \in \{0, 1\} \quad (i, j) \in A \quad (29)$$

$$x_{ij} \in \{0, 1\} \quad (i, j) \in A \quad (30)$$

$$y_{ijk} \in \{0, 1\} \quad \langle i, j, k \rangle \in F. \quad (31)$$

$$1 \leq u_i \leq c + 2 \quad i \in N_+ \quad (32)$$

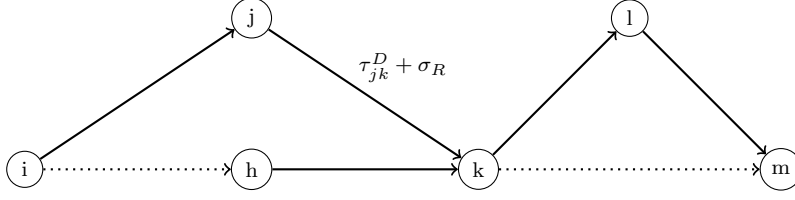


Fig. 1: Improving constraint.

4.2 Formulation $\overline{\text{MC}}$

We propose a formulation called $\overline{\text{MC}}$ built upon MC with just a set of modifications inserted to provide a more realistic interpretation of the problem description given by Murray and Chu.

Let us consider the *truck timing constraints* (16), which applies to a truck running along arc $(h, k) \in A$. Let us refer to Figure 1, showing a solution where the drone performs a sortie which starts from i , a node that precedes h , terminates in k , and immediately starts a second sortie from k . If this happens, constraint (16) imposes $t_k^T \geq t_h^T + \tau_{hk}^T + \sigma_R + \sigma_L$, where σ_R and σ_L refers, respectively, to the rendezvous and launch time arriving in k and restarting from k . Allocating the rendezvous time to t_k^T is correct, but we believe that the launch time must be allocated to the time of the node visited after k in the truck route, otherwise the launching time needed for sortie $\langle k, l, m \rangle$ would be included in the flying time of sortie $\langle i, j, k \rangle$, and that could erroneously make exceed its endurance. Moreover, we believe that the launch time must not be allocated to the truck when the drone starts from the depot, since depot operations are considered to be done offline. Therefore, we substituted (16) with (33)-(34).

$$t_k^T \geq t_h^T + \tau_{hk}^T + \sigma_R \sum_{\langle i, j, k \rangle \in F} y_{ijk} + \sigma_L \sum_{\substack{\langle h, r, s \rangle \in F \\ r \neq k}} y_{hrs} - M(1 - x_{hk}) \quad (33)$$

$$(h, k) \in A, h \in N_+$$

$$t_k^T \geq t_0^T + \tau_{0k}^T + \sigma_R \left(\sum_{\langle i, j, k \rangle \in F} y_{ijk} \right) - M(1 - x_{0k}) \quad (0, k) \in A, k \in N_+ \quad (34)$$

Moreover, we believe that σ_L must be included in the drone timing, when the drone is launched from a node different from the depot, being the time when the drone is on the truck and the operator prepares it. We thus substitute the first *drone timing constraint* (17) with (35)-(36).

$$t_j^D \geq t_i^D + \tau_{ij}^D + \sigma_L - M(1 - \sum_{\langle i, j, k \rangle \in F} y_{ijk}) \quad (i, j) \in A, j \in C', i \in N_+ \quad (35)$$

$$t_j^D \geq t_0^D + \tau_{0j}^D - M(1 - \sum_{\langle 0, j, k \rangle \in F} y_{ijk}) \quad j \in C' \quad (36)$$

4.2.1 Wait and no-wait models

As for the original MC formulation constraints (35)–(36) allow t_j^D to be greater than the arrival time at customer j , without incurring in its computation while calculating energy consumption (19). We refer to this model as the *wait* model. From our experience the waiting is not always technically possible. In this case the *no-wait* model is obtained by substituting the *Drone battery endurance constraints* (19) with (37)–(38).

$$t_k^D - t_i^D \leq E + \sigma_L + M(1 - y_{ijk}) \quad \langle i, j, k \rangle \in F, i \neq 0 \quad (37)$$

$$t_k^D - t_0^D \leq E + M(1 - y_{0jk}) \quad \langle 0, j, k \rangle \in F \quad (38)$$

Note that our reformulation is similar, but slightly different from the one proposed by Ponza [40]. Indeed, in [40] the launching time is allocated to both truck and drone, so it enters in the computation of the energy consumption, while we suppose that launching time does not consume energy, being the time needed by the truck to make the drone ready, while the drone does not fly. Moreover, Ponza [40] includes the launching time in the computation also when the sortie starts from the depot, while we use the original MC model that assumes depot operations are done off-line.

5 Improvements on DMN

In this section we present a first enhanced formulation, built upon formulation $\overline{\text{MC}}$ of Section 4.2. The general idea is to substitute some explicit constraints with exponentially many constraints to be added in a cutting plane fashion.

5.1 Crossing Sorties Elimination

The basic MC model uses the explicit *simple crossing sorties elimination* (26) which are based on timing and use of a ‘big M’ constant. We propose to substitute them with the following structural constraints that directly address and prevent the unfeasible topologies. Let $i \in N_0, l \in C$ be two vertices visited by the truck while running along a path P from i to l , and assume that two sorties $\langle i, j, k \rangle$ and $\langle l, m, n \rangle$ with $k \notin P$ exist. In this case, the second sortie starts before the first one is returned and the overall solution is therefore infeasible. Let us define \mathcal{P} as the set of all the paths with these characteristics. The following *Crossing Sorties Elimination Constraints* (CSEC) (39) can be used to eliminate this type of infeasibility.

$$\sum_{h=1}^{|P|-1} x_{v(h), v(h+1)} + \sum_{\substack{\langle i, j, k \rangle \in F \\ k \notin P}} y_{i, j, k} + \sum_{\langle l, m, n \rangle \in F} y_{l, m, n} \leq |P| \quad P \in \mathcal{P} \quad (39)$$

where $P = \{v(1), v(2), \dots, v(|P|)\}$, $v(1) = i$, $v(|P|) = l$. These constraints impose that one of the arcs of the path or one of the sorties should be set to zero to make the solution feasible.

We can strengthen (39) using the idea of “tournament constraint”. Since the path enters at most once in each vertex, we can include in the constraint also the arcs $(h, j) \in A$ such that $h, j \in P$ and h precedes j in P . We obtain the *Tournament Crossing Constraints* (TCS), given by (39).

$$\sum_{h=1}^{|P|-1} \sum_{j=h+1}^{|P|} x_{v(h)v(j)} + \sum_{\substack{\langle i,j,k \rangle \in F \\ k \notin P}} y_{i,j,k} + \sum_{\langle l,m,n \rangle \in F} y_{l,m,n} \leq |P| \quad P \in \mathcal{P} \quad (40)$$

5.2 Backward Sorties Elimination

A next improvement is the avoidance of backward sorties, that are sorties with a rendezvous that happens before the departure. Let \mathcal{B} denote the set of all truck paths $P = \{v(1), v(2), \dots, v(q)\}$ with $v(1) = 0$ and $v(q) \in C$. Given one of such paths P suppose that exists a sorties $\langle i, j, v(q) \rangle$ with $i \notin P$. The *Backward Sortie Elimination Constraints* (BSEC) are given by (41), and impose that at least one of the arcs of the path or the sortie is eliminated.

$$\sum_{i=1}^{|P|-1} x_{v(i)v(i+1)} + \sum_{\langle i,j,v(q) \rangle \in F} y_{i,j,v(q)} \leq |P| - 1 \quad P \in \mathcal{B} \quad (41)$$

Constraint (41) can be strengthened by considering a tournament type constraint on path P , that imposes that at most one arc enters a node of the path, and adding all sorties that terminate in P , but start at nodes outside P . We obtain the *Tournament Backward Constraints* (TBS), given by (42).

$$\text{(TBS)} \quad \sum_{i=1}^{|P|-1} \sum_{j=i+1}^{|P|} x_{v(i)v(j)} + \sum_{\substack{\langle i,j,k \rangle \in F \\ i \notin P, k \in P}} y_{i,j,k} \leq |P| - 1 \quad P \in \mathcal{B} \quad (42)$$

Finally note that backwards sorties happen only when the time constraint are not respected, in our case in fractional solutions.

5.3 Improving the Objective Function

Preliminary computational results showed that formulation DMN, and thus F1, provides very bad lower bounds at the root node of the decision-tree (see Section 7 for more details). Therefore, we reformulate the objective function $\min t_{c+1}^T$ by explicating its components: the traveling time of the truck, the launching and rendezvous times, and the time truck waits for the drone. The first two components can be modeled using the variables we dispose already,

but the waiting times must be expressed by new variables. We add variables $w_i, i \in N$, to model the truck waiting at node i , thus giving the new objective function (43).

$$\min \sum_{(i,j) \in A} \tau_{ij}^T x_{ij} + \sigma_R \sum_{\langle 0,j,k \rangle \in F} y_{0jk} + (\sigma_L + \sigma_R) \sum_{\substack{\langle i,j,k \rangle \in F \\ i \neq 0}} y_{ijk} + \sum_{i \in N} w_i \quad (43)$$

The waiting variables must be included in the truck timing constraints as in (44)–(47). Note that we use a pair of constraint with opposite versus to impose equality when an arc (h, k) is selected. Non-negativity of the w_i must also be imposed.

$$t_k^T \geq t_h^T + \tau_{hk}^T + \sigma_R \sum_{\langle i,j,k \rangle \in F} y_{ijk} + \sigma_L \sum_{\substack{\langle h,r,s \rangle \in F \\ r \neq k}} y_{hrs} \quad (44)$$

$$t_k^T \leq t_h^T + \tau_{hk}^T + \sigma_R \sum_{\langle i,j,k \rangle \in F} y_{ijk} + \sigma_L \sum_{\substack{\langle h,r,s \rangle \in F \\ r \neq k}} y_{hrs} \quad (45)$$

$$t_k^T \geq t_0^T + \tau_{0k}^T + \sigma_R \left(\sum_{\langle i,j,k \rangle \in F} y_{ijk} \right) - M(1 - x_{hk}) + w_k \quad (h, k) \in A, h \in N_+ \quad (46)$$

$$t_k^T \leq t_0^T + \tau_{0k}^T + \sigma_R \left(\sum_{\langle i,j,k \rangle \in F} y_{ijk} \right) + M(1 - x_{0k}) + w_k \quad (0, k) \in A, k \in N_+ \quad (47)$$

$$w_i \geq 0 \quad i \in N \quad (48)$$

5.4 Subtour Elimination Constraints

When avoiding crossing sorties by using the CSEC or TCS, Miller-Tucker-Zemlin constraint are not needed; indeed, total node ordering variables are not needed and timing constraints are sufficient to prevent subtours; however, the introduction of subtour elimination constraints (49) and the two nodes subtours elimination (50) is profitable. Preliminary computational results confirmed us the reduction of the number of visited nodes by the branch-and-bound and the computing time even if applied to the $\overline{\text{MD}}$ formulation.

$$(\text{SEC}) \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad S \subseteq N, S \neq \emptyset, |S| > 2 \quad (49)$$

$$x_{ij} + x_{ji} \leq 1 \quad i, j \in N, i \neq j \quad (50)$$

The overall formulation, called **DMN** in the following, uses all the improvements of this section with the “tournament” version for crossing sorties (40) and backward sorties (42).

6 Two-indexed Formulations: DMN2

In the previous models we represented the sorties with three-indexed boolean variables, hereby we propose a new formulation which uses two-indexed arc variables. In particular, let us consider the binary variables \vec{g}_{ij} and \overleftarrow{g}_{jk} which take value 1 if the drone enters, respectively leaves, customer node j . In this way the number of variables representing the sorties reduces from n^3 to $2n^2$. To further reduce the variables we preliminary fix to zero all the variables corresponding to arcs with flying time exceeding the battery limit, i.e., we set $\vec{g}_{ij} = 0$ for all $(i, j) \in A : \tau_{ij}^D > E$ and $\overleftarrow{g}_{jk} = 0$ for all $(j, k) \in A : \tau_{jk}^D + \sigma_R > E$. We also fix to zero variables that do not allow to complete a feasible drone fly: $\vec{g}_{ij} = 0, (i, j) \in A, j \notin C'$; $\overleftarrow{g}_{jk} = 0, (j, k) \in A, j \notin C'$; $\vec{g}_{ic+1} = 0 \ i \in N$ and $\overleftarrow{g}_{j0} = 0 \ j \in N$.

Starting from our model DMN, we build the new formulation DMN2 using the above two-indexed variables. The objective function is

$$\min \sum_{(i,j) \in A} \tau_{ij}^T x_{ij} + \sigma^L \sum_{\substack{(i,j) \in A \\ i \neq 0}} \vec{g}_{ij} + \sigma^R \sum_{(j,k) \in A} \overleftarrow{g}_{jk} + \sum_{i \in N} w_i \quad (51)$$

All the constraints using y variables must be rewritten using the \vec{g} and \overleftarrow{g} variables, as follows:

Customer covering constraints (2) become

$$\sum_{(i,j) \in A} x_{ij} + \sum_{(i,j) \in A} \vec{g}_{ij} = 1 \quad j \in C \quad (52)$$

$$\sum_{(i,j) \in A} x_{ij} + \sum_{(i,j) \in A} \overleftarrow{g}_{ij} = 1 \quad i \in C \quad (53)$$

Single sortie leave/return constraints (6) and (7) are no longer necessary, since are induced by the truck routing constraints (3)–(5) within the x - y coupling constraints, now called *x - g coupling constraints*:

$$\sum_{(i,j) \in A} \vec{g}_{ij} \leq \sum_{(i,h) \in A} x_{ih} \quad i \in N_0 \quad (54)$$

$$\sum_{(i,j) \in A} \overleftarrow{g}_{ij} \leq \sum_{(h,j) \in A} x_{hj} \quad j \in N_+ \quad (55)$$

$$\sum_{(i,j) \in A} \vec{g}_{ij} = \sum_{(j,k) \in A} \overleftarrow{g}_{jk} \quad j \in C \quad (56)$$

Drone-truck timing constraints (10)–(13) are substituted by

$$t_i^D \geq t_i^T - M(1 - \sum_{(i,j) \in A} \vec{g}_{ij}) \quad i \in C \quad (57)$$

$$t_i^D \leq t_i^T + M(1 - \sum_{(i,j) \in A} \vec{g}_{ij}) \quad i \in C \quad (58)$$

$$t_k^D \geq t_k^T - M(1 - \sum_{(j,k) \in A} \overleftarrow{g}_{jk}) \quad k \in N_+ \quad (59)$$

$$t_k^D \leq t_k^T + M(1 - \sum_{(j,k) \in A} \overleftarrow{g}_{jk}) \quad k \in N_+ \quad (60)$$

Truck timing constraints (44)–(47) become

$$t_k^T \geq t_h^T + \tau_{hk}^T + \sigma_R \sum_{(i,k) \in A} \overleftarrow{g}_{jk} + \sigma_L \sum_{(h,l) \in A} \vec{g}_{hl} - M(1 - x_{hk}) + w_k \quad (h,k) \in A, h \in N_+ \quad (61)$$

$$t_k^T \leq t_h^T + \tau_{hk}^T + \sigma_R \sum_{(j,k) \in A} \overleftarrow{g}_{jk} + \sigma_L \sum_{(h,l) \in A} \vec{g}_{hl} + M(1 - x_{hk}) + w_k \quad (h,k) \in A, h \in N_+ \quad (62)$$

$$t_k^T \geq t_0^T + \tau_{0k}^T + \sigma_R \sum_{(j,k) \in A} \overleftarrow{g}_{jk} - M(1 - x_{0k}) + w_k \quad k \in N_+ \quad (63)$$

$$t_k^T \leq t_0^T + \tau_{0k}^T + \sigma_R \sum_{(j,k) \in A} \overleftarrow{g}_{jk} + M(1 - x_{0k}) + w_k \quad k \in N_+ \quad (64)$$

$$(65)$$

Drone timing constraints (35), (36) and (18) are now

$$t_j^D \geq t_i^D + \tau_{ij}^D + \sigma_L - M(1 - \vec{g}_{ij}) \quad j \in C', i \in C, i \neq j \quad (66)$$

$$t_j^D \geq t_0^D + \tau_{0j}^D - M(1 - \vec{g}_{0j}) \quad j \in C' \quad (67)$$

$$t_k^D \geq t_j^D + \tau_{jk}^D + \sigma_R - M(1 - \sum_{(j,k) \in A} \overleftarrow{g}_{jk}) \quad j \in C', k \in N_+, j \neq k \quad (68)$$

Drone battery endurance constraints (37)–(38) become:

$$t_k^D - t_i^D \leq E + \sigma_L + M(2 - \vec{g}_{ij} - \overleftarrow{g}_{jk}) \quad (i,j) \in A, (j,k) \in A, j \in C', i \in N_+, i \neq k \quad (69)$$

$$t_k^D - t_0^D \leq E + M(2 - \vec{g}_{0j} - \overleftarrow{g}_{jk}) \quad (j,k) \in A, j \in C' \quad (70)$$

Crossing sorties elimination constraints (40) must be rewritten as follows. Let $i \in N_0, l \in C$ be two vertices encountered by the truck while running along a path P from i to l , and assume that exist two sorties defined by $\vec{g}_{ij} > 0$ and $\vec{g}_{lm} > 0$ and such that there is no node $k \in P \setminus \{i\}$ with $\overleftarrow{g}_{hk} > 0$. In

this case the second sortie starts before the first sortie is terminated and the following “tournament” crossing sorties elimination holds:

$$(TCS2) \sum_{h=1}^{|P|-1} \sum_{j=h+1}^{|P|} x_{v(h)v(j)} + \sum_{\substack{(i,j) \in A, \\ j \notin P}} \vec{g}_{ij} + \sum_{\substack{(l,j) \in A, \\ j \notin P}} \vec{g}_{lj} \leq |P| \quad P \in \mathcal{P} \quad (71)$$

where $P = \{v(1), v(2), \dots, v(q)\}$ with $v(1) = i, v(q) = l$, and \mathcal{P} now defines the set of all the paths with the described characteristics.

Backward sorties elimination are modified as follows. Let $i \in N_0, j \in N_+$ be two nodes encountered by the truck while traveling on a path P , suppose that exist a sortie identified by $\overleftarrow{g}_{k,i} > 0$ and such that $k \notin P$ and a sortie identified by $\vec{g}_{v(q),k} > 0$ and such that $k \notin P$, that, together determine an infeasible solution. Let \mathcal{B} now denote the set of all truck paths $\{v(1), \dots, v(m), \dots, v(q)\}$ with $v(1) = 0, v(m) = i$, and $v(q) = j$. The tournament version of the backward sorties elimination constraints is:

$$(TBS2) \sum_{h=1}^{|P|-1} \sum_{l=h+1}^{|P|} x_{v(h)v(l)} + \vec{g}_{jk} + \sum_{\substack{(l,k) \in A, \\ l \notin P}} \vec{g}_{lk} + \sum_{\substack{(k,l) \in A, \\ l \notin P}} \overleftarrow{g}_{kl} \leq |P| \quad P \in \mathcal{B} \quad (72)$$

We finally need to impose the following constraints to avoid infeasibilities:

$$\vec{g}_{ij} + \overleftarrow{g}_{ij} \leq 1 \quad (i, j) \in A \quad (73)$$

$$\vec{g}_{ij} + \overleftarrow{g}_{ji} \leq 1 \quad (i, j) \in A \quad (74)$$

7 Computational Experiments

To test the above models we have implemented them to run on an Intel Core i3-2100 CPU, with 3.10 GHz and 8.00 GB of RAM, running Windows 7 operating system. CPLEX 12.71 was used as MILP solver, and only a single thread was utilized during the testing.

Formulation \overline{MC} was solved directly by CPLEX, while DMN and DMN2 required a branch-and-cut implementation, since they include exponentially many constraints. To separate these constraints we considered the residual graph $G' = (N, A')$ obtained from G (see Section 4) by selecting the only arcs associated with a non zero variable (x, y, \vec{g} and \overleftarrow{g}) in the continuous relaxation of the model. For the *crossing sorties elimination* constraints and for the *backward sorties elimination* constraints we simply explore the graph starting from depot 0, until a truck path violating one of the constraints is identified, if any. The overall procedure has a time complexity $O(|A'|)$. To separate the subtour elimination constraints we use the standard approach which requires to solve at most $O(n^2)$ max flow problems on the residual graph.

To facilitate the solution of the problem, we give all the algorithms an initial heuristic solution computed as follow. We first find a heuristic (TSP) solution in which all the customers are served by the truck. We build this solution with a greedy constructive algorithm followed by a local search improvement using as moves to define neighbouring solution the relocate of one customer and the swap of two customers. Next we examine all the possible sorties and we select the one, if any, that improves the solution. We repeat the search for improving sorties (and update of the solution) until no one is found. We based our tests on the 72 benchmark instances provided by Murray and Chu [36]. In each of these, ten customers are randomly distributed across an eight-mile square region, while the depot is located in four different positions. The endurance of the drone was chosen to be either 20 or 40 minutes, while the speed of the drone was selected to be 15, 25, 35 miles/h based on Euclidean distances. The truck speed was assumed to be 25 miles/h and based on Manhattan distances. In the first 24 instances it has been set a ratio $|C'|/|C| = 90\%$, while for the remaining ones they set the same ratio to 80%. For each instance we tested the formulations on both the ‘no-wait’ case (in which the drone is not allowed to wait at a customer), and the ‘wait’ case, adopted in [36], where wait is allowed, but not considered in the computation of the battery endurance (see Section 1). Overall 144 instances are available in our test bed.

The interested reader can find all the results and solutions in our web site www.or.unimore.it following the *online resources* link.

7.1 Tests

Table 2 presents the performances of the lower bounds for different formulations. The entries of the table are the average gap over the instances referred in each row, computed as $100 \cdot (opt - LB)/opt$, where ‘opt’ is the value of the optimal solution obtained giving to formulation DMN2 enough time to close all instances. The first column, labeled ‘E’, reports the battery endurance value, while column labeled ‘speed’ refers to the drone speed. Column ‘ \overline{MC} ’ reports the lower bound gap obtained by the Murray-Chu formulation, modified as in Section 4.2, after one hour of computing time. All the other columns give the lower bound obtained at the root node. Columns labeled ‘newF’ (new objective function) refer to formulation \overline{MC} modified by adopting the objective function introduced in Section 5. Columns labeled ‘newF2’ refer to formulation newF modified by substituting the three index variables y with the two-indexed variables \overrightarrow{y} and \overleftarrow{y} as in Section 6. Finally ‘DMN’ and ‘DMN2’ report on the lower bound gaps of our complete formulations. Formulation \overline{MC} clearly gives bad results, in terms of lower bound gap, providing a gap close to 80% for all class of instances. The adoption of the new objective function greatly improves the behaviour, drastically reducing the gap to about 22%. The use of two-indexed variables and complete formulations further improve the gap. Concerning the effect of the waiting time at the customers we observe that instances where is allowed to wait have slightly smaller gaps.

Fig. 2: Comparison of lower bound gaps at the root node for the different proposed methods.

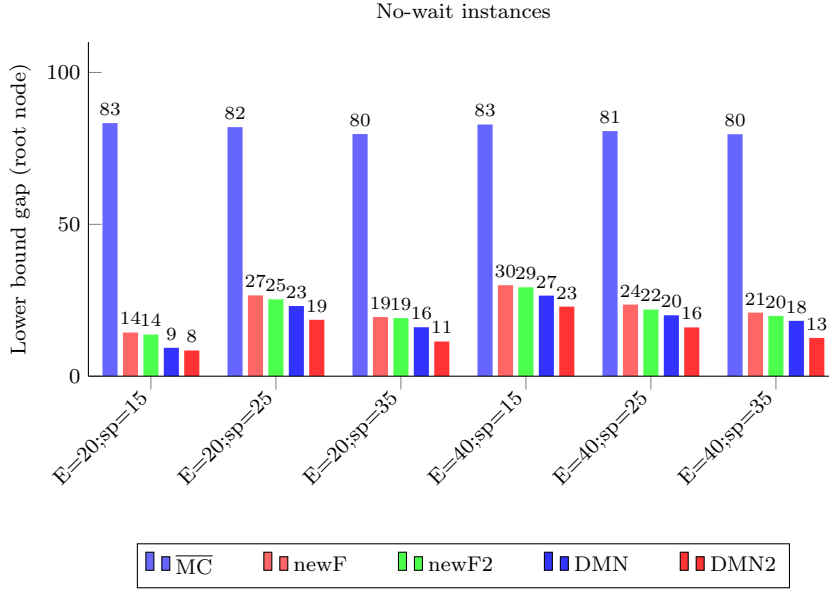


Table 2: Lower bounds at the root node, by speed class

<i>E</i>	speed	No-Wait					Wait				
		\overline{MC}	newF	newF2	DMN	DMN2	\overline{MC}	newF	newF2	DMN	DMN2
20	15	83.31	14.38	13.71	9.35	8.46	83.42	13.83	12.73	8.67	6.83
	25	81.99	26.60	25.30	23.10	18.57	81.39	23.99	21.79	20.25	15.36
	35	79.72	19.47	19.14	16.12	11.41	79.64	19.21	18.56	15.99	11.12
40	15	82.86	29.95	29.27	26.53	22.90	82.78	29.77	29.30	27.16	23.06
	25	80.68	23.58	21.94	20.05	16.06	80.68	23.58	22.55	20.28	15.80
	35	79.66	20.93	19.85	18.22	12.60	79.65	20.79	19.61	17.82	12.49
All		81.37	22.49	21.53	18.89	15.00	81.26	21.86	20.76	18.36	14.11

Table 3 gives the same data as in Table 2, but grouped by depot position. Column labeled dep refers to the four depot positions, with respect to the square in which are generated the customers. In the first position a the depot is almost in the barycentre of the customers, in b, c and d it is close to the right border of the square, with a vertical position which is, respectively, in the barycentre, on the bottom border and below the bottom border at a distance which is equal to the distance of the barycenter from this border. One can see that the worst bound gaps are obtained when the depot is in position ‘b’ and improve when it is far from the barycenter (positions ‘c’ and ‘d’). The best gaps are obtained when the depot is outside the square, that is far from all customers. In this case there is a sort of ‘offset’ distance that must be covered by any route to reach the customers, and this improves the bound.

Table 3: Lower bounds at the root node, by depot location

<i>E</i>	dep	No-Wait					Wait				
		\overline{MC}	newF	newF2	DMN	DMN2	\overline{MC}	newF	newF2	DMN	DMN2
20	a	89.87	20.97	20.24	15.88	13.79	89.67	19.10	17.17	13.80	11.21
	b	88.08	25.19	24.25	21.57	16.28	88.14	24.16	22.81	19.86	14.72
	c	82.06	19.03	17.86	14.53	11.10	81.94	18.46	17.67	14.91	9.95
	d	66.69	15.41	15.18	12.78	10.08	66.18	14.31	13.11	11.31	8.55
40	a	90.08	26.65	25.03	23.00	18.71	90.07	26.46	24.60	23.07	18.84
	b	87.41	29.92	28.72	26.86	21.08	87.41	29.92	28.72	27.55	20.75
	c	81.30	23.49	21.88	20.16	16.01	81.29	23.46	23.42	19.87	15.79
	d	65.46	19.21	19.11	16.38	12.95	65.37	19.01	18.53	16.51	13.08

Figure 2 provides a graphical representation of the lower bound gaps for the ‘no-wait’ instances. One can see that the lower bound gap at the root node is improved largely by the proposed methods with respect to the original model with no difference based on endurance and speed class. The reader can notice that each of the improvement features applied provide a relevant contribution to the results.

In Tables 4-5 we report on the performances of the formulations in finding the proven optimal solution for the instances where waiting at the customer’s location is allowed. The values in the table are averaged over the 12 (resp. 9) instances of each class of speed (resp. depot location). The overall results are computed over the 72 instances. In the columns labeled ‘gap%’, ‘opt’, ‘nodes’ and ‘time’, we report, respectively: the percentage gap computed as $100 \cdot (UB - LB) / UB$ (average on the only instances not solved to the optimum), the number of proven optimal solutions, the average computing time in CPU seconds, and the average number of branch-decision-trees explored. As already anticipated, formulation \overline{MC} , in one hour of CPU time, was not able to solve any instance given the poor performances of the lower bound, so we do not report the CPU time for this formulation.

Looking at Table 4 we observe that the instances with endurance 20 are much easier than those with endurance 40: in terms of obtained optimal solutions and computing times. This is due to the fact that the number of possible drone sorties are much less in the first case, thus reducing the possible choices of the algorithm. The two-indexed formulation DMN2 has better performances than DMN, since it solves 13 more instances, the unsolved instances have smaller gap and also the computing time is smaller.

For the ‘no-wait’ instances the formulations exhibit a very similar behaviour that is not reported here (the interested reader can find the related results at www.or.unimore.it).

Looking at Table 5 we can observe that instances of class ‘a’, where the depot is located almost in the middle of the customers, are more difficult than the other cases. Class ‘d’, which gave the best results for the root node lower bounds, has, instead, the same difficulty of the classes ‘b’ and ‘c’.

Tables 6 depicts the difference between the ‘wait’ and ‘no-wait’ solutions with respect to the speed class and the depot location. The entries of Table 6-(a) and Table 6-(b), are the endurance, the speed (depot location, respec-

Table 4: Exact solutions for ‘wait’ instances, by speed class

E	sp	MC			DMN				DMN2			
		gap%	opt	nodes	gap%	opt	time	nodes	gap%	opt	time	nodes
20	15	83.48	0	180356.6	0.00	12	10.2	379.9	0.00	12	6.1	340.4
	25	82.02	0	159165.3	8.25	11	797.1	17056.3	1.75	11	503.1	19934.1
	35	80.81	0	142095.8	8.72	9	1342.2	24671.7	4.18	10	920.4	35803.4
40	15	82.96	0	153948.3	10.59	2	3313.3	54078.5	7.67	9	2247.2	91281.3
	25	81.47	0	145158.7	10.14	4	2533.3	46797.5	6.70	7	2033.7	83462.6
	35	80.63	0	140049.8	5.28	8	1433.6	25229.2	8.93	10	1020.6	45452.8
All		81.90	0	153462.4	9.33	46	1571.68	28035.5	6.50	59	1121.89	46045.8

Table 5: Exact solutions for ‘wait’ instances, by depot location

E	dep	MC			DMN				DMN2			
		gap%	opt	nodes	gap%	opt	time	nodes	gap%	opt	time	nodes
20	a	90.14	0	161330.4	9.03	6	1496.6	26211.3	3.37	6	1293.5	41864.8
	b	88.38	0	162479.0	7.34	8	1034.0	22432.4	0.00	9	559.4	29790.4
	c	82.72	0	157777.2	0.00	9	155.0	3771.9	0.00	9	29.2	1703.1
	d	67.18	0	160570.2	0.00	9	180.7	3728.2	0.00	9	24.2	1412.2
40	a	90.64	0	145540.9	11.71	2	2937.7	49395.2	8.23	3	2662.4	86080.9
	b	87.79	0	145526.4	10.68	3	2555.8	46533.9	7.67	6	2081.0	86932.6
	c	82.02	0	151308.9	7.51	5	2033.6	35907.6	0.00	9	1064.3	53117.3
	d	66.30	0	143166.2	6.41	4	2180.0	36303.6	1.96	8	1261.1	67464.7
All		81.90	0	153462.4	9.33	46	1571.7	28035.5	6.50	59	1121.9	46045.8

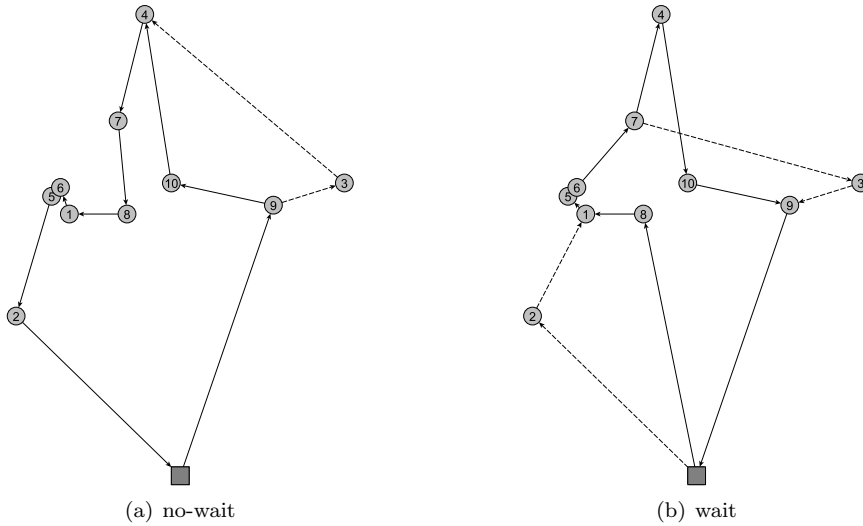
tively), the gap% computed as $100 \cdot (opt^n - opt^w) / opt^n$ averaged only when the two solution differs, being opt^n and opt^w the optimal solution of the no-wait and wait case, respectively. The last column reports the number of occurrences of a difference between the two optimal solutions. One can note that a more restrictive endurance make the allowance of waiting at customer without flying more important. This appears more relevant than the speed class. On the other hand, depot position ‘a’ (centrally positioned) and ‘d’ (positioned out of the square) have a strong impact on the solution when waiting is allowed. Moreover, based on extensive computational tests, we can state that the ‘wait’ model is only slightly easier to be solved than the ‘no-wait’ one.

Table 6: Comparison between ‘wait’ and ‘no-wait’ solutions

(a) By speed class				(b) By depot location			
E	speed	gap%	occur.	E	dep	gap%	occur.
20	15	1.65	5	20	a	4.08	5
	25	3.86	10		b	2.64	5
	35	2.24	2		c	1.22	4
40	15	1.42	4		d	3.89	3
	25	0.00	0	40	a	2.66	1
	35	0.00	0		b	0.00	0
All					c	0.37	1
		2.72	21		d	1.32	2
				All		2.72	21

Figure 3 shows the difference between a 'no-wait' and a 'wait' solution. For this particular instance, the gap between the value of the two optimal solutions is around 9%. Note that one more sortie can be performed in the 'wait' solution. In particular, allowing to wait at customer 2, the drone can save battery while the truck travels path (0,8,1), whose length would exceed battery endurance if the drone had to wait while flying. Similarly, the 'wait' solution can let the drone wait at customer 3, while the truck runs on the path (7,4,10,9), whose time would exceed the battery endurance. In this case the 'no-wait' solution performs a shorter sortie, instead; however, this difference does not improve the solution, but it shows, nevertheless, that the instances of the FSTSP can have more than one optimal solution, with the same truck route, and different sorties. This last consideration is one of the reasons why these problems are hard to solve also for small instances and it further justifies the increasing interest of researchers.

Fig. 3: Comparison between the 'no-wait' (a) and 'wait' (b) solution of an instance with $E = 20$, depot location 'd', and 'speed' = 25 miles/h. The solid, dashed, lines represent the truck route, sorties, respectively. The square indicates the depot while the circles indicate the customers.



8 Conclusions

This paper considered one of the most promising fields in parcel deliveries nowadays, the combined use of traditional vehicles and drones. We focused on

one particular problem called the Flying Sidekick Traveling Salesman Problem, where a drone and a truck are coupled and synchronized, and must serve a set of customers. We provided improvements on the literature by proposing a three and a two-indexed formulations for which we proposed a novel objective function that could increase remarkably the lower bounds, and a set of good inequalities to be separated in a branch-and-cut fashion that provided an important contribution to obtain better solutions in a faster way. Our method outperforms the previously proposed one, being the two-indexed formulation the preferable one: 59 out of the 72 benchmark instances could be solved to optimality with an average percentage gap between the best lower and upper bound of 1.70% and within an average computing time of less than 20 minutes. We evaluated two versions of the problem: one in which the drone is allowed to wait at the customers to save battery and one where this is prohibited. Both are hard to solve even when considering small sized instances. The version in which waiting is allowed is only slightly easier to be solved with respect to the other one: larger sets of feasible solutions must be treated by the algorithm, but feasible solutions are easier to find and possibly with a smaller cost. Among the several features considered, the drone endurance is the one that has the stronger influence on the convergence of the algorithm, a smaller endurance allow the algorithms to have better performances, while reducing the number of feasible sorties. To conclude, our ideas could be easily applied to similar problems: this could suggest a direction of future research where analogous problems, with different constraints, are considered. Another possible direction of research could be the development of metaheuristic algorithms for these sets of problems.

References

1. 7-eleven, flirtey make first faa-approved drone delivery to home. <https://www.gpsworld.com/7-eleven-flirtey-make-first-faa-approved-drone-delivery-to-home/>. Accessed: 2019-03-22
2. Aerial id card renewal: Uae to use drones for government services. <https://www.reuters.com/article/us-emirates-drones/aerial-id-card-renewal-uae-to-use-drones-for-government-services-idUSBREA1906E20140210>. Accessed: 2019-03-22
3. Alibaba's drones deliver packages to islands. http://www.chinadaily.com.cn/business/2017-11/07/content_34230012.htm. Accessed: 2019-03-22
4. Amazon primeair. <https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011>. Accessed: 2019-03-22
5. Ambulance drone. <https://www.tudelft.nl/en/ide/research/research-labs/applied-labs/ambulance-drone/>. Accessed: 2019-03-22
6. The defibrillator drone that can beat ambulance times. <https://www.bbc.com/news/av/technology-40360164/the-defibrillator-drone-that-can-beat-ambulance-times>. Accessed: 2019-03-22
7. Dpdgroup drone delivers parcels using regular commercial line. <https://www.geopostgroup.com/en/news/dpdgroup-drone-delivers-parcels-using-regular-commercial-line>. Accessed: 2019-03-22
8. Drone delivery: Dhl 'parcelcopter' flies to german isle. <https://www.reuters.com/article/us-deutsche-post-drones/>

- drone-delivery-dhl-parcelcopter-flies-to-german-isle-idUSKCN0HJ1ED20140924. Accessed: 2019-03-22
9. How ups and zipline are using drones to save lives in remote rwandan clinics. <https://www.freeenterprise.com/how-ups-and-zipline-are-using-drones-to-save-lives/>. Accessed: 2019-03-22
 10. Jd.com to build 150 drone launch facilities in china by 2020. <https://aircargoworld.com/allposts/jd-com-to-build-150-drone-launch-facilities-in-china-by-2020-video/>. Accessed: 2019-03-22
 11. Jd.coms drone delivery program takes flight in rural china. <http://corporate.jd.com/whatIsNewDetail?contentCode=6IhXLeeSAFLjLLlyuZatDA>. Accessed: 2019-03-22
 12. Jd's first large-scale uav debuts in shaanxi. <http://www.chinadaily.com.cn/a/201811/20/WS5bf3af9ca310eff303289e31.html>. Accessed: 2019-03-22
 13. Parcelcopter: Dhl's drone. <https://discover.dhl.com/business/business-ethics/parcelcopter-drone-technology>. Accessed: 2019-03-22
 14. Ups drivers may tag team deliveries with drones. <https://money.cnn.com/2017/02/21/technology/ups-drone-delivery/index.html>. Accessed: 2019-03-22
 15. Ups tests residential delivery via drone. https://www.youtube.com/watch?v=xx9_60yjjrQ. Accessed: 2019-03-22
 16. Vans & drones in zurich. <https://www.mercedes-benz.com/en/mercedes-benz/vehicles/transporter/vans-drones-in-zurich/>. Accessed: 2019-03-22
 17. Wing - transforming the way goods are transported. <https://x.company/projects/wing/>. Accessed: 2019-03-22
 18. Zookal will deliver textbooks using drones in australia next year. <https://edition.cnn.com/2013/10/18/tech/innovation/zookal-will-deliver-textbooks-using-drones/index.html>. Accessed: 2019-03-22
 19. Agatz, N., Bouman, P., Schmidt, M.: Optimization approaches for the traveling salesman problem with drone. *Transportation Science* (2018)
 20. Boone, N., Sathyan, A., Cohen, K.: Enhanced approaches to solving the multiple traveling salesman problem. In: *Proc. of the AIAA Infotech@ Aerospace Conference* (2015)
 21. Bortoff, S.A.: Path planning for uavs. In: *American Control Conference*, 2000. *Proceedings of the 2000*, vol. 1, pp. 364–368. IEEE (2000)
 22. Bouman, P., Agatz, N., Schmidt, M.: Dynamic programming approaches for the traveling salesman problem with drone. *Networks* **72**(4), 528–542 (2018)
 23. Boysen, N., Briskorn, D., Fedtke, S., Schwerdfeger, S.: Drone delivery from trucks: Drone scheduling for given truck routes. *Networks* **72**(4), 506–527 (2018)
 24. Campbell, J.F., Corberán, Á., Plana, I., Sanchis, J.M.: Drone arc routing problems. *Networks* **72**(4), 543–559 (2018)
 25. Chao, I.M.: A tabu search method for the truck and trailer routing problem. *Computers & Operations Research* **29**(1), 33–51 (2002)
 26. Current, J.R., Schilling, D.A.: The covering salesman problem. *Transportation science* **23**(3), 208–213 (1989)
 27. Daknama, R., Kraus, E.: Vehicle routing with drones. *arXiv preprint arXiv:1705.06431* (2017)
 28. Desrochers, M., Laporte, G.: Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters* **10**(1), 27–36 (1991)
 29. Dorling, K., Heinrichs, J., Messier, G.G., Magierowski, S.: Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **47**(1), 70–85 (2017)
 30. Drexler, M.: Synchronization in vehicle routing a survey of vrps with multiple synchronization constraints. *Transportation Science* **46**(3), 297–316 (2012)
 31. Ferrandez, S.M., Harbison, T., Weber, T., Sturges, R., Rich, R.: Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management* **9**(2), 374 (2016)
 32. Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H.: Heuristic methods for the traveling salesman problem with drone. *Comput. Sci* (2015)

33. Kashuba, S., Novikov, V., Lysenko, O., Alekseeva, I.: Optimization of uav path for wireless sensor network data gathering. In: *Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, 2015 IEEE International Conference, pp. 280–283. IEEE (2015)
34. Lin, C.: A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources. *Computers & Operations Research* **38**(11), 1596–1609 (2011)
35. Mbiadou Saleu, R.G., Deroussi, L., Feillet, D., Grangeon, N., Quilliot, A.: An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks* **72**(4), 459–474 (2018)
36. Murray, C.C., Chu, A.G.: The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies* **54**, 86–109 (2015)
37. O'Rourke, K.P., Bailey, T.G., Hill, R., Carlton, W.B.: Dynamic routing of unmanned aerial vehicles using reactive tabu search. Tech. rep., AIR FORCE INST OF TECH WRIGHT-PATTERSONAFB OH (1999)
38. Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E.: Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks* **72**(4), 411–458 (2018)
39. Poikonen, S., Wang, X., Golden, B.: The vehicle routing problem with drones: Extended models and connections. *Networks* **70**(1), 34–43 (2017)
40. Ponza, A.: Optimization of drone-assisted parcel delivery (2016)
41. Pugliese, L.D.P., Guerriero, F., Zorbas, D., Razafindralambo, T.: Modelling the mobile target covering problem using flying drones. *Optimization Letters* **10**(5), 1021–1052 (2016)
42. Richards, A., Bellingham, J., Tillerson, M., How, J.: Coordination and control of multiple uavs. In: *AIAA guidance, navigation, and control conference*, Monterey, CA (2002)
43. Roberge, V., Tarbouchi, M., Labonté, G.: Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning. *IEEE Transactions on Industrial Informatics* **9**(1), 132–141 (2013)
44. Ryan, J.L.: Embedding a reactive tabu search heuristic in unmanned aerial vehicle simulations. Tech. rep., AIR FORCE INST OF TECH WRIGHT-PATTERSONAFB OH SCHOOL OF ENGINEERING (1998)
45. Ryan, J.L., Bailey, T.G., Moore, J.T., Carlton, W.B.: Reactive tabu search in unmanned aerial reconnaissance simulations. In: *Proceedings of the 30th conference on Winter simulation*, pp. 873–880. IEEE Computer Society Press (1998)
46. Savuran, H., Karakaya, M.: Route optimization method for unmanned air vehicle launched from a carrier. *Lecture Notes on Software Engineering* **3**(4), 279 (2015)
47. Scott, J., Scott, C.: Drone delivery models for healthcare. In: *Proceedings of the 50th Hawaii International Conference on System Sciences* (2017)
48. Shang, B., Wu, C., Hu, Y., Yang, J.: An algorithm of visual reconnaissance path planning for uavs in complex spaces. *Journal of Computational Information Systems* **8**(1), 1–8 (2014)
49. Shavarani, S.M., Nejad, M.G., Rismanchian, F., Izbirak, G.: Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of amazon prime air in the city of san francisco. *The International Journal of Advanced Manufacturing Technology* pp. 1–13 (2017)
50. Shuttleworth, R., Golden, B.L., Smith, S., Wasil, E.: Advances in meter reading: Heuristic solution of the close enough traveling salesman problem over a street network. In: *The Vehicle Routing Problem: Latest Advances and New Challenges*, pp. 487–501. Springer (2008)
51. Sisson, M.R.: Applying tabu heuristic to wind influenced, minimum risk and maximum expected coverage routes. Tech. rep., AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING (1997)
52. Tseng, C.M., Chau, C.K., Elbassioni, K., Khonji, M.: Flight tour planning with recharging optimization for battery-operated autonomous drones. *arXiv preprint arXiv:1703.10049* (2017)
53. Ulmer, M.W., Thomas, B.W.: Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks* **72**(4), 475–505 (2018)

-
54. Wang, X., Poikonen, S., Golden, B.: The vehicle routing problem with drones: Several worst-case results. *Optimization Letters* **11**(4), 679–697 (2017)
 55. Zheng, C., Li, L., Xu, F., Sun, F., Ding, M.: Evolutionary route planner for unmanned air vehicles. *IEEE Transactions on Robotics* **21**(4), 609–620 (2005)
 56. Zorbas, D., Razafindralambo, T., Guerriero, F., et al.: Energy efficient mobile target tracking using flying drones. *Procedia Computer Science* **19**, 80–87 (2013)