



UNIVERSITÀ
DEGLI STUDI
FIRENZE

FLORE

Repository istituzionale dell'Università degli Studi di Firenze

How the Wind Can Be Leveraged for Saving Energy in a Truck-Drone Delivery System

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:

How the Wind Can Be Leveraged for Saving Energy in a Truck-Drone Delivery System / Sorbelli, FB; Corò, F; Palazzetti, L; Pinotti, CM; Rigoni, G. - In: IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. - ISSN 1524-9050. - ELETTRONICO. - 24:(2023), pp. 4038-4049. [10.1109/TITS.2023.3234627]

Availability:

The webpage <https://hdl.handle.net/2158/1347486> of the repository was last updated on 2023-12-30T15:58:54Z

Published version:

DOI: 10.1109/TITS.2023.3234627

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

Publisher copyright claim:

La data sopra indicata si riferisce all'ultimo aggiornamento della scheda del Repository FloRe - The above-mentioned date refers to the last update of the record in the Institutional Repository FloRe

(Article begins on next page)

How the Wind Can Be Leveraged for Saving Energy in a Truck-Drone Delivery System

Francesco Betti Sorbelli[✉], *Member, IEEE*, Federico Corò[✉], *Member, IEEE*, Lorenzo Palazzetti[✉], *Member, IEEE*, Cristina M. Pinotti[✉], *Senior Member, IEEE*, and Giulio Rigoni[✉]

Abstract—In this work, we investigate the impact of the wind in a drone-based delivery system. For the first time, to the best of our knowledge, we adapt the trajectory of the drone to the wind. We consider a truck-drone tandem delivery system. The drone actively reacts to the wind adopting the “most tailwind” trajectory available between the truck’s path and the delivery. The truck moves on a predefined route and carries the drone close to the delivery point. We propose the Minimum-energy Drone-trajectory Problem (MDP) which aims, when the wind affects the delivery area, at planning minimum-energy trajectories for the drone to serve the customers starting from and returning to the truck. We then propose two algorithms that optimally solve MDP under two different routes of the truck. We also analytically study the feasibility of sending drones with limited battery to deliver packages. Finally, we first numerically compare our algorithms on randomly generated synthetic and real data, and then we evaluate our model simulating the drone’s flight in the BlueSky simulator.

Index Terms—Drone, Delivery, Wind model, Energy model.

I. INTRODUCTION

Recently, drones or Unmanned Aerial Vehicles (UAVs) have been widely investigated in civil applications, such as agriculture [1], environmental protection [2], task offloading [3], localization [4], [5], last-mile package delivery [6], [7], and so on. In fact, drone-based solutions for the delivery of small packages have been announced by big companies in e-commerce. Initially, it did not seem safe or practical to have robotic objects flying in the sky, but now there are several reasons to believe that package delivery by drones may be coming soon. In fact, new regulation laws have been just released in 2021 by the Federal Aviation Administration (FAA), which allow operators of small drones to fly over people and at night under certain conditions [8].

Drones are considered to be much faster than traditional trucks in some circumstances and can also lead to a reduction in pollution and greenhouse gas emissions. For example, drones are not delayed by rush hours in congested areas and can significantly shorten the paths to reach customers by flying over rivers, green parks, and sea bays. So, it is highly expected that pretty soon, if not already, companies can further extend

their business by relying on drones that cross the last mile to their customers. However, drone-based delivery systems still have to face exciting challenges. First, due to the small size of the battery, drones (i.e., copters) have limited flight autonomy, and such a limitation is more accentuated by their carried payload. Also, due to payload constraints, drones cannot fulfill more than a single delivery at a time, so they must go back to a depot for grabbing other packages for other customers. Finally, weather conditions, in particular *wind*, must be considered during the drones’ deliveries.

In this work, we concentrate on the wind challenge. Surprisingly, in the literature, the wind has been only considered as a limiting factor for drones, to the best of our knowledge. All the manufacturers set the maximum allowed wind speed that can be safely tolerated by their drones.

The wind as a factor that influences the aircraft’s path has been previously considered in [9]. Similarly to us, the authors are interested in finding the “global optimal path” in presence of wind. But, they aim to optimize the duration of the flight. They specifically aim at determining the aircraft’s minimum-time path while also avoiding obstacles, not considered in our work. The aircraft’s energy is not taken into account. Moreover, the kind of the aircraft and its routes are not comparable with ours. Namely, the geographical scale is completely different. The authors in [9] analyze the wind effect on commercial planes, powered by fuel, whose routes’ lengths are much longer than ours (i.e., $\geq 800\text{km}$).

This work provides proof-of-concept of how wind can help drones during deliveries. We consider the following scenario, which is realistic in the current state of technology. We assume drones engaged in last-mile deliveries with lightweight payloads. In a delivery, a drone can afford to carry 1–6kg of payload in a range of 8–12km [10]. To reduce the length of the round trips to/from the depot, inspired by big companies’ drone delivery programs, we assume that a drone is assisted by a truck. The drone picks up the packages from the truck and flies towards the customer, and then back to the truck, while the latter continues its journey near a series of delivery points [11], [12], [13]. In the literature, several scenarios have been addressed using truck-drone tandem delivery systems, but at the best of our knowledge, no one considers adapting drones’ trajectories to wind conditions.

In this work, instead, motivated by the observation that when the drone operates under “tailwind” conditions (i.e., when the wind blows in the direction of the drone’s movement) it consumes less energy, we explore the possibility that the drone selects the detaching point from the truck to reach the customer following, as much as it can, the “tailwind”

Manuscript received Month xx, yyyy; revised Month xx, yyyy; accepted Month xx, yyyy. Date of publication Month xx, yyyy; date of current version Month xx, yyyy. This work was supported in part by the “GNCS – INDAM”, and by RESIDUAL UNIPG. The Associate Editor for this article was FirstName, Last Name. (*Corresponding author: Lorenzo Palazzetti.*)

F. Betti Sorbelli, F. Corò, and C. M. Pinotti are with Dept. of Computer Science and Mathematics, University of Perugia, Italy. L. Palazzetti is with Dept. of Computer Science and Mathematics, University of Florence, Italy. G. Rigoni is with Dept. of Mathematics, University of Padua, Italy

Digital Object Identifier XXXXXXXX/TITS.YYYYYYYY

trajectory. We do that even at the cost of increasing the length of the trajectory, as long as we do not lose energy. The same approach is followed to return back to the truck (see Figure 1).

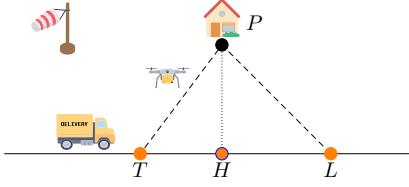


Fig. 1. Abstraction of a delivery performed by the truck-drone tandem in a supply chain scenario. The truck moves carrying the drone; P is the delivery.

In this work, the drone's route consists of a *polygonal chain* that starts from the *take-off point* T on the truck's road and returns to the *landing point* L passing through the delivery point P (see Figure 1). Notice that, in general, the take-off and landing points do not coincide. We only assume that the take-off point precedes the landing point on the truck's route. To give the drone the freedom to arbitrarily select the take-off and landing points, we assume a rural or suburban environment, like a residential area neighborhood, where already at low altitudes, the drone can freely move in any direction having no significant obstacles on its way. We also assume a very mild truck-drone synchronization, and so, the first vehicle that arrives at the landing point, waits for the other.

Our goal is twofold: i) to determine the take-off and landing points on the truck's route that minimize the energy consumed by the drone for the delivery, and ii) to study the delivery feasibility because the drones are energy-constrained. This work extends our prior conference paper [14], and our results are summarized as follows:

- We define the Minimum-energy Drone-trajectory Problem (MDP) whose goal is to find the minimum-energy trajectories when the wind influences the delivery area;
- We devise two algorithms for MDP, i.e., SINGLE and MULTI which assume that the truck's path is a straight line or a polygon, respectively;
- We analyze the feasibility in terms of energy of sending drones for deliveries in a windy area, by presenting the FEASIBILITY algorithm;
- We numerically compare our algorithms on synthetic and real data sets. Moreover, we evaluate the energy spent by our solution simulating the drone's flight in BlueSky.

The rest of the work is organized as follows. Section II surveys the related work. Section III introduces the wind and energy models. Section IV defines the MDP and details the math that supports the SINGLE and the FEASIBILITY solutions. Section V gives the pseudo-code and the flowcharts of the SINGLE, the MULTI, and the FEASIBILITY algorithms. Section VI numerically evaluates our algorithms through synthetic and real winds, and also simulates the drone's flight by using BlueSky. Finally, Section VII offers conclusions.

II. RELATED WORK

In this section, we review works about truck-drone delivery systems with an emphasis to those influenced by the wind.

A. Truck-Drone Tandem Delivery Systems

Recently, the problem of delivering goods with drones has been approached by several papers [12], [15]. The vast majority of them assume that the drone has a limited battery and has to return back to the warehouse after every delivery due to the payload and energy budget constraints. Many solutions consider drones working in tandem with a truck, and there are many ways the two means of transportation can collaborate. Some solutions divide the deliveries between the two, while others make them collaborate in every single mission [16], [17], [18].

The authors in [16] consider symbiotic cooperation between a truck and a drone to accomplish a set of deliveries. Precisely, the goal is to minimize the makespan to serve all the customers by either the truck or the drone. This problem is presented as a variant of the TSP known as the flying sidekicks traveling salesman problem. The authors propose a hybrid heuristic divided into three steps. Initially, the algorithm creates a TSP solution where the truck serves all the customers. Next, the algorithm greedily assigns deliveries to the drone considering the quantity of time saved with respect to the truck without exceeding the battery budget. Each delivery is served with a polygonal chain that starts from and returns to a point of the truck's route passing through the customer. Finally, the algorithm tries to improve the solution by visiting the neighborhood of each customer, i.e., by evaluating a different assignment for each customer. Our work is completely different. Although like [16] we use a polygonal chain to serve a delivery customer, we select the polygonal chain by minimizing the energy for every single delivery and not to minimize the total makespan of all the customers. Moreover, in our solution, any position on the truck's road is a candidate for the take-off/landing, whereas in [16] only a subset of positions, which correspond to points on the truck's route, are candidates.

A similar problem is proposed in [18] where, given a fixed sequence of stops that constitute a truck's route and a set of customers, the goal is to find a scheduling for drones (i.e., a set of trips defined by a drone's take-off, customer service, and landing) so that all the customers are served and the makespan is minimized. The authors provide a mixed-integer programming formulation to solve the single and multiple drone scenarios. However, unlike our approach, it is not possible to change neither the truck's route nor the drone's trajectories for a customer in order to increase time savings.

Notice that, all these works do not consider the impact of winds on energy consumption since they are focused only on the time required to perform a delivery.

B. Drone Delivery Systems Influenced by the Wind

The impact of the wind on defining long-distance trajectories of aircraft has been considered in [9]. Instead, the impact of winds on small and battery powered drones' trajectory planning has not received much attention so far.

In [19], [20], [21], the wind is introduced in the model. Precisely, the drone trajectories are represented by paths in a weighted graph whose edge weights model the energy consumed by the drone, which depends on the carried payload

and the wind speed. In [20], [21] authors focus on finding which is the percentage of deliveries that can be accomplished with a given energy budget knowing the wind conditions on-the-fly. Authors of [22] extend the above solution to a multi-depot multi-drone delivery system. Whereas in [20], [21], the drone moves along predefined routes represented by the edges of a graph given in input to the problem, in this paper we must find the drone's trajectory, which is the output of the problem, by leveraging the impact of the wind.

A multi-drone delivery system dealing with wind, obstacle avoidance, and maximum energy budget, is also proposed in [23]. A subset of feasible deliveries is initially computed offline, and a depth-first search strategy is performed then to get the final mission plan. Their extended work [24] considers the same setting with many clusters, in which the problem is seen as a constraint satisfaction problem. In [25], a delivery scenario where the wind affects multiple drones is considered. Differently from our approach, each drone can perform multiple deliveries on the same mission. The focus is on optimizing the global mission plan. The wind influences the energy consumption, but the drone passively suffers the wind, i.e., it does not adapt the route to the wind, as we do.

Lastly, authors in [26] exploit the service paradigm to abstract drone's capabilities into a drone service. The main novelty is the creation of a test-bed that consists of an indoor replica of a city. A drone flies over it by using different polygonal trajectories and payloads. The wind conditions are simulated by a fan. The energy consumption is inferred from the collected data of 72 indoor flights. The authors focus on the study of how the shape (triangular, rectangular, hovering) of the drone's path acts on the energy consumption.

In conclusion, our work is different from the surveyed ones in many aspects. The most important difference is that in our study the battery powered drone actively reacts to the wind and searches for the minimum energy trajectory, i.e., the trajectory as closer to the tailwind as possible given the truck's movement. Computed the optimum trajectory, we compare how much energy is saved using the found trajectory with respect to the energy spent traversing the shortest trajectory, which is the best solution in absence of wind.

III. BACKGROUND

In the background of our solution there is the *wind triangle* concept. In air navigation, the wind triangle (see Figure 2) is a graphical representation of the relationship between the drone's motion and the wind. The *ground vector* represents the

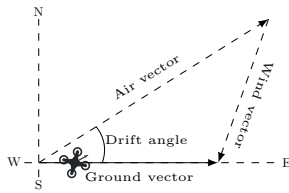


Fig. 2. The wind triangle.

motion of the drone over the ground, and it is the resultant of adding the *air vector*, i.e., the motion of the drone through the

air mass, and the *wind vector* [27]. Specifically, when there is a wind and the drone wants to follow a specific ground track, the air vector indicates the true heading and drone's speed. The difference in angle between heading and ground track is known as the *drift angle*. So, if the drone wants to follow the desired track, it must correct its heading of an angle depending on the wind direction. Each vector $\vec{z} = (z_d, z_s)$ in the wind triangle is characterized by a direction z_d and a speed z_s .

We first observe that in presence of wind, varying the ground direction also varies the air vector (see Figure 2). Since we aim to find the track that minimizes the energy spent by the drone (i.e., we try many possible ground directions) and since the energy spent to fly depends on the air vector (as we will see in Eq. (1)), our solution relies on the aforementioned wind triangle construction. The second observation is that by varying the ground track, the energy changes due to the variations in both the length of the trajectory to be traversed, and the air speed. In this work, we will show that sometimes, in presence of wind, it is convenient to leave the shorter drone's trajectory for a longer one that consumes less energy.

In the rest of this section, we explain how we derive the unitary energy for the ground trajectories.

A. The Relative Wind and Energy Model

The wind, characterized by speed and direction, is an important variable for evaluating the drone's energy consumption. Usually, the weather stations record the *meteorologic direction* of the wind ω_d^{me} , i.e., the direction from which the wind originates, assuming the North as the 0° direction. For instance, if the wind blows from the North to the South, a weather station records a direction of 0° , while if it blows from the East to the West, the weather station refers to 90° . From now on, to ease the notation, we indicate the angles without the degree symbol ($^\circ$). The meteorologic direction of the wind conventionally grows clockwise in a Cartesian coordinate system xOy whose x -axis is the North direction and the wind direction is given as the direction the wind is from. To conform the wind to the classical Cartesian coordinate system, we convert the meteorologic direction of the wind into the *mathematical direction* of the wind ω_d^{ma} . First, ω_d^{ma} defines the winds blowing to the origin O of the Cartesian coordinate system and not from it. The mathematical wind directions grow counterclockwise from the usual x -axis which is equal to the East direction. Hence, the conversion rule from the meteorological direction of the wind to the mathematical one is explained in [28], and it is as follows:

$$\omega_d^{\text{ma}} = \left| \underbrace{-\omega_d^{\text{me}} + 360}_{\text{clockwise}} + \underbrace{90}_{\text{phase}} - \underbrace{180}_O \right|_{360} = |-\omega_d^{\text{me}} + 270|_{360}.$$

Figure 3a shows the direction of the meteorological wind $\omega_d^{\text{me}} = 225$ that is directed towards O and the corresponding mathematical direction $\omega_d^{\text{ma}} = 45$ that has O as the source. In our work, from now on, we always refer to the mathematical direction of the wind.

Let us now define the *global wind* ω as the (mathematical) wind that blows in the delivery area. The global wind vector $\omega = (\omega_d, \omega_s)$ has *direction* ω_d and *speed* ω_s . The *relative wind*

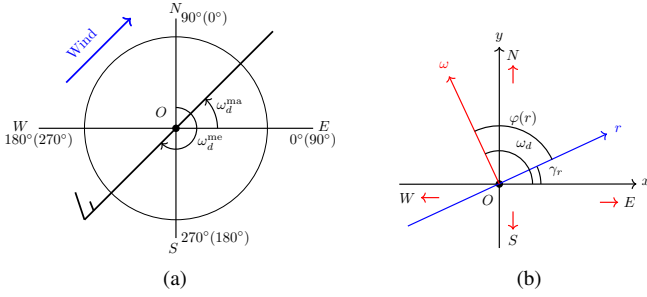


Fig. 3. Axes are labeled with the cardinal, mathematical, and meteorologic (in brackets) directions. Note the *phase* (equals to 90) which represents the offset North-East among the two ways for representing the wind (a); Relative wind $\varphi(\vec{r}) = 90$ on \vec{r} when $\gamma_r = 25$ and $\omega_d = 115$ (b).

on r $\varphi(\gamma_r)$ has direction $|\omega_d - \gamma_r|_{360}$ which is the difference mod 360 between the direction of the global wind and the direction of the drone's trajectory (see Figure 3b).

The energy for flying a unit of distance (e.g., 1m) on a line r mostly depends on the air speed of the drone, which in turn depends on the global wind $\omega = (\omega_d, \omega_s)$ and on the ground vector $\vec{r} = (\gamma_r, v_d)$, as seen in Figure 2. Precisely, the *air speed* a_s can be calculated as [10]:

$$a_s = \sqrt{s_N^2 + s_E^2}, \quad (1)$$

where $s_N = v_d - \omega_s \cos(\varphi(\gamma_r))$ and $s_E = -\omega_s \sin(\varphi(\gamma_r))$. Hence, when $\omega_s = 0$, it holds that $a_s = v_d$.

The energy also depends on other drone's features like the number of rotors, and the payload weight. There is no simple formula to calculate the unit energy as a function of the relative wind [10], so we introduce the compass rose to discretize the relative winds and to tabulate the energy for them. In this way, fixing the type of the drone (i.e., octocopter, quadcopter), the drone's ground speed v_d , the payload κ , and the speed of the global wind ω_s , we can pre-compute the energy to traverse 1m: one energy coefficient for each sector of the compass rose (e.g., see Table II in Section VI). For the sake of simplicity, when all the involved parameters are clear from the contexts, the *energy* $E(\vec{AB})$ will denote the energy spent by a drone to fly the ground segment \vec{AB} from A to B .

B. The Compass Rose

In general, weather stations store the different winds detected by applying a discretization. In other words, given a *compass rose*, the winds are grouped in a finite number of sectors. To define the sectors and the compass rose, we divide the turn angle at O of a conventional Cartesian coordinate system xOy into $4t$ sectors, each of width $\sigma = 180/2t$, where t is the *compass rose cardinality*. Therefore, σ defines the width of each sector. Conventionally, a *sector* S_i contains the winds whose relative wind direction verify:

$$\left(i \frac{180}{2t} ; (i+1) \frac{180}{2t} \right) \quad 0 \leq i \leq 4t - 1. \quad (2)$$

A *representative wind direction* ρ_i , with $0 \leq i \leq 4t - 1$, is associated to each sector S_i . The representative ρ_i is then used to compute the energy consumption of any relative wind direction that falls in S_i . The unit energy consumption for

any wind in S_i is denoted as μ_i . Basically, the compass rose is used to bound to $4t$ the number of relative winds and thus the possible energy levels, as we will see in the next section. Table II reports the values of the energy required by a octocopter with different payloads for the 12 winds of the compass rose used in our experiments when $\omega_s = 10\text{m/s}$ [10].

Having defined the energy consumption, we formally introduce the Minimum-energy Drone-trajectory Problem and explain how the drone can minimize the energy consumption.

IV. THE PROBLEM DEFINITION

In this work, we assume the *truck-drone tandem delivery system* which consists of a *truck* that carries drones in the delivery area. When the global wind is absent, i.e., $\omega_s = 0$, the speed of the relative wind is zero and the energy spent by the drone does not depend on the direction of the trajectory. Hence, in this case, the best solution for the drone is to select the shortest trajectory from the truck's route to the customer. In presence of wind, the less consuming trajectory is the one parallel to the wind (see Eq. (1)). However, there may be no trajectory parallel to the wind that starts from the truck's path and reaches the customer. In that case, the selection of the drone's trajectory that minimizes the energy is not trivial. In the rest of the section, we first describe the problem, then we introduce our solution when the truck's path is a single line.

A. The Minimum-energy Drone-trajectory Problem (MDP)

Let r be the line that represents the truck's path. So, the truck's path is represented by a vector \vec{r} whose direction is γ_r . Notice that, in this work we do not consider the truck's speed. Let us consider a delivery point P to be served by the drone. To serve P from the road r (see Figure 1), the truck selects the *take-off point* and *landing point* for the drone, called T and L , respectively. Thus, the drone detaches from the truck at T and flies on the line from T to P carrying the payload. After the delivery, the drone (without payload) moves back on the line from P to L , rendezvousing at the truck again. We select the trajectories making this assumption: the drone and the truck always move forward in the sense that the projection H of P onto the truck's path must follow T and precede L .

Our problem is to find the optimal composed trajectory $\vec{TP} \cup \vec{PL}$ such that the energy consumed by the drone to serve P is minimized under the assumption that $T \preceq H \preceq L$. We assume a constant wind during the whole delivery. We call this problem Minimum-energy Drone-trajectory Problem (briefly, MDP). In other words, we aim at minimizing the quantity $E(\vec{TP}) + E(\vec{PL})$. Changing T and L , the slope (i.e., direction) of the segments \vec{TP} and \vec{PL} changes, and so, by the wind triangle rule, the relative winds $\varphi(\vec{TP})$ and $\varphi(\vec{PL})$, and their relative energy, change.

In the next, we show how to solve MDP by selecting the optimum trajectory among a discrete set of candidate trajectories.

B. Towards the MDP Solution

Considering T and L fixed on the track's route r , we calculate the relative wind on the trajectories \vec{TP} and \vec{PL}

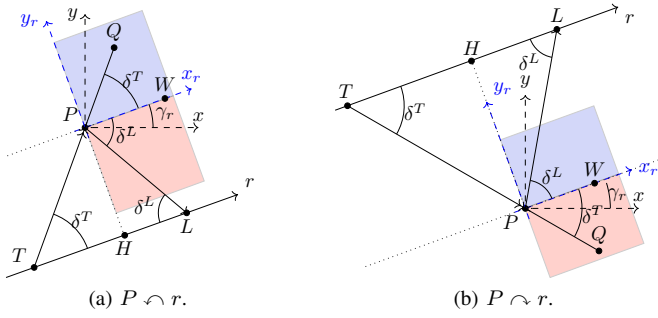


Fig. 4. The relative winds on \vec{TP} and \vec{PL} , where the quadrants Q_1 and Q_4 are highlighted in blue and red, respectively.

to find the energy consumption. First of all, we need to figure out whether P is to the right or left of the truck. This can be quickly done by evaluating the sign of the determinant of the associated matrix $M = [v(\vec{TH}), v(\vec{TP})]$, where $v(\vec{TP})$ and $v(\vec{TH})$ are the column vectors¹ associated to the drone's trajectories. Specifically, (i) P is on the left of truck (denoted as $P \cap r$) if $\det(A) > 0$; and (ii) P is on the right of truck (denoted as $P \cap r$) if $\det(A) < 0$.

In our discussion, we consider the Cartesian coordinate system xPy of the mathematical wind, with origin in P . Let H be the projection of P on r (see Figure 4). Let δ^T and δ^L be the angles formed, respectively, by the drone's trajectory \vec{TP} and \vec{PL} with the truck's route r . Namely, $\delta^T = \angle PTH$ and $\delta^L = \angle PLH$ can be easily computed as $\delta^T = \arctan(\overline{PH}/\overline{TH})$ and $\delta^L = \arctan(\overline{PH}/\overline{HL})$. Therefore, $\vec{TP} = \frac{\overline{PH}}{\sin(\delta^T)}$ and $\vec{PL} = \frac{\overline{PH}}{\sin(\delta^L)}$ (see Figure 4). So, it is evident that the length of the trajectories depends on the take-off or landing points and the slope of the truck's route r . Now, we can state the following.

Theorem 1. *The relative winds on \vec{TP} and \vec{PL} are:*

$$\begin{aligned} \varphi(\vec{TP}) &= \begin{cases} |\omega_d - (\gamma_r + \delta^T)|_{360} & \text{if } P \cap r \\ |\omega_d - (\gamma_r - \delta^T)|_{360} & \text{if } P \cap r \end{cases} \\ \varphi(\vec{PL}) &= \begin{cases} |\omega_d - (\gamma_r - \delta^L)|_{360} & \text{if } P \cap r \\ |\omega_d - (\gamma_r + \delta^L)|_{360} & \text{if } P \cap r \end{cases} \end{aligned} \quad (3)$$

Proof. We prove Eq. (3) when $P \cap r$ (see Figure 4a). Consider the rotated Cartesian system x_rPy_r with origin in P , x_r -axis parallel to r and oriented in the same direction as r . Recall that the y_r -axis forms a 90 counter-clockwise angle with x_r . Note that the angle between x and x_r has width γ_r . Observed that r and x_r are parallel, the segment \vec{TP} forms the same angle $\delta^T = \angle WPK$ on x_rPy_r , and the angle $(\gamma_r + \delta^T)$ on xPy . Hence, the direction of the relative wind on xPy is $\varphi(\vec{TP}) = \omega_d - (\gamma_r + \delta^T)$. Also the returning path \vec{PL} forms the angle $-\delta^L$ on x_rPy_r . Hence, the direction of the relative wind on xPy is $\varphi(\vec{PL}) = \omega_d - (\gamma_r - \delta^L)$.

The case where $P \cap r$ is depicted in Figure 4b. Eq. (3) can be proven similarly to the left case. \square

Hence, the relative wind depends on the wind direction, the direction of the truck's path, and take-off and landing points.

¹For instance, given $A = (1, 1)$ and $B = (3, 4)$, then $v(\vec{AB}) = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$.

We now analyze in which sector of the compass rose both $\varphi(\vec{TP})$ and $\varphi(\vec{PL})$ fall, when T and L move on r . Let us discuss the case $P \cap r$. When T and L move on r towards H (see Figure 4), it is easy to see that $\delta^T, \delta^L \in (0, 90]$. Precisely, $\delta^T = \delta^L = 90$ when $\overline{TH} = \overline{HL} = 0$, while δ^T and $\delta^L \rightarrow 0$ when both \overline{TH} and $\overline{HL} \rightarrow \infty$. Then, the take-off and landing drone's trajectories \vec{TP} and \vec{PL} scan, respectively, the first quadrant Q_1 (shown in blue) and the fourth quadrant Q_4 (shown in red) of the Cartesian coordinate system (see Figure 4) with origin in P and whose x -axis coincides with the relative wind $\varphi(\vec{r})$. Hence, observing that the relative wind on the truck's route is $\varphi(\vec{r}) = |\omega_d - \gamma_r|_{360}$, the take-off trajectories scan the compass rose starting from $\varphi(\vec{r})$ shown in blue in Figure 5, while the landing trajectories scan the compass rose starting from $\varphi(\vec{r})$ shown in red in Figure 5.

From that, Eq. (3) can be rewritten relatively to $\varphi(\vec{r})$:

$$\begin{aligned} \varphi(\vec{TP}) &= \begin{cases} |\varphi(\vec{r}) - \delta^T|_{360} & \text{if } P \cap r \quad Q_1 \\ |\varphi(\vec{r}) + \delta^T|_{360} & \text{if } P \cap r \quad Q_4 \end{cases} \\ \varphi(\vec{PL}) &= \begin{cases} |\varphi(\vec{r}) + \delta^L|_{360} & \text{if } P \cap r \quad Q_4 \\ |\varphi(\vec{r}) - \delta^L|_{360} & \text{if } P \cap r \quad Q_1 \end{cases} \end{aligned} \quad (4)$$

As explained in Section III-B, fixed T and P , the relative winds $\varphi(\vec{TP})$ and $\varphi(\vec{PL})$ experienced by the drone are given by wind representative ρ of the sector where they reside. The sectors visited when T and P move are listed starting from the sector S_τ where $\varphi(\vec{r})$ resides, i.e., S_τ , where $\tau = \lfloor |\varphi(\vec{r}) - 1|_{360}/\sigma \rfloor$. Note that τ is computed from $\varphi(\vec{r}) - 1$ instead of $\varphi(\vec{r})$ because sectors in Eq. (2) are defined *left-open* and *right-closed*. By the previous Eqs. (4) and (2), it holds:

Theorem 2. *Let $\tau = \lfloor |\varphi(\vec{r}) - 1|_{360}/\sigma \rfloor$. When $\varphi(\vec{TP})$ or $\varphi(\vec{PL})$ scan Q_1 , the drone spends the energy μ_i that depends on the winds S_i of the compass rose whose indices i are $\tau \leq i \leq \tau + t|_{4t}$. When $\varphi(\vec{TP})$ or $\varphi(\vec{PL})$ scan Q_4 , the drone spends the energy μ_i that depends on the winds S_i of the compass rose whose indices i are $|\tau - t|_{4t} \leq i \leq \tau$.*

Figure 5 shows two examples where $t = 3$ and so $\sigma = 30$ whose crossed quadrants are different in sectors and number. Specifically, when $\varphi(\vec{r}) = 0$, we cross the sectors S_0, S_1 , and S_2 in Q_1 , and S_{11}, S_{10} , and S_9 in Q_4 , examining so exactly $t = 3$ sectors per quadrant. Instead, with $\varphi(\vec{r}) = 50$ we scan one more sector than before, crossing S_1, S_2, S_3 , and S_4 in Q_1 , and S_1, S_0, S_{11} , and S_{10} in Q_4 . Table I reports the sectors and the angles $\delta \in (0, 90]$ associated to each sector of Q_1 and Q_4 when $\varphi(\vec{r}) = 0$ and $\varphi(\vec{r}) = 50$ (see Figure 5). Generalizing, when $|\varphi(\vec{r})|_\sigma = 0$ we cross exactly t sectors, instead when $|\varphi(\vec{r})|_\sigma \neq 0$ we scan $t + 1$ sectors. So, depending on which sector T and L fall in, the μ depends on the relative wind of the scanned sectors. Finally, note that changing r or the direction of the global wind has the same effect: $\varphi(\vec{r})$ changes and so the candidate sectors change.

The take-off/landing positions that delimit the i^{th} sector in Q_1 are those that form with γ_r the δ_i angle so defined:

$$\delta_i = \begin{cases} \begin{cases} -|\varphi(\vec{r})|_\sigma + (i + 1)\sigma & \text{if } i < t \\ 90 & \text{if } i = t \end{cases} & \text{if } |\varphi(\vec{r})|_\sigma \neq 0 \\ (i + 1)\sigma & \text{if } |\varphi(\vec{r})|_\sigma = 0 \end{cases} \quad (5)$$

A. The SINGLE Algorithm

The SINGLE algorithm optimally solves MDP when T and L are selected from a single road, requiring $\mathcal{O}(t)$ time and space, where t is the number of sectors for each quadrant. SINGLE has as input the wind vector (ω_d, ω_s) , the drone's speed v_d , the payload weight κ , the truck's road \vec{r} , the cardinality of sectors t in the compass rose, and the delivery point P . The pseudocode of SINGLE is sketched in Algorithm 1.

Algorithm 1: SINGLE $(P, \omega, t, \vec{r}, v_d, \kappa)$

```

1  $\mu_0^T, \dots, \mu_{4t-1}^T, \mu_0^L, \dots, \mu_{4t-1}^L \leftarrow \text{UNIT\_COSTS}(\omega)$ 
2  $\varphi(\vec{r}) \leftarrow |\omega_d - \gamma_r|_{360}$ 
3  $\tau \leftarrow \lfloor |\varphi(\vec{r}) - 1|_{360} / \sigma \rfloor$ 
4 if  $P \curvearrowright r$  then
5    $i^* \leftarrow \arg \min_{0 \leq i \leq t} \left\{ \frac{\mu_{|\tau+i|_{4t}}^T}{\sin(\delta_{|\tau+i|_{4t}}^T)} \right\}$ 
6    $j^* \leftarrow \arg \min_{0 \leq j \leq t} \left\{ \frac{\mu_{|\tau-j|_{4t}}^L}{\sin(\delta_{|\tau-j|_{4t}}^L)} \right\}$ 
7   let  $T$  such that  $\overline{TH} = \overline{HP} \cot(\delta_{|\tau+i^*|_{4t}}^T)$ 
8   let  $L$  such that  $\overline{HL} = \overline{HP} \cot(\delta_{|\tau-j^*|_{4t}}^L)$ 
9 if  $P \curvearrowleft r$  then
10   $i^* \leftarrow \arg \min_{0 \leq i \leq t} \left\{ \frac{\mu_{|\tau-i|_{4t}}^T}{\sin(\delta_{|\tau-i|_{4t}}^T)} \right\}$ 
11   $j^* \leftarrow \arg \min_{0 \leq j \leq t} \left\{ \frac{\mu_{|\tau+j|_{4t}}^L}{\sin(\delta_{|\tau+j|_{4t}}^L)} \right\}$ 
12  let  $T$  such that  $\overline{TH} = \overline{HP} \cot(\delta_{|\tau-i^*|_{4t}}^T)$ 
13  let  $L$  such that  $\overline{HL} = \overline{HP} \cot(\delta_{|\tau+j^*|_{4t}}^L)$ 
14 return  $T, L$ 
```

The SINGLE algorithm works as follows. Initially (Line 1), we perform a pre-processing phase to compute the unitary energy costs according to the payload weight. Precisely, with $0 \leq i \leq 4t - 1$, μ_i^T represents the energy spent for traversing a unitary distance when the global wind has speed ω_s , the drone moves in S_i with speed v_d and carries the payload κ , instead, μ_i^L is the same as μ_i^T but without payload. Then, we compute the relative wind $\varphi(\vec{r})$ on the road r according to the current global wind condition ω (Line 2), and also the wind sector with index τ (Line 3). At this stage, by exploiting Eq. (4) we identify the quadrants for T and L to scan in order to find the sub-routes with minimum energy. In other words, we find the best indices that return the optimal sectors for the take-off and landing sub-routes, i.e., i^* and j^* , respectively, starting from S_τ (Lines 5–6). Now, depending on the position of P with respect to the truck's road r , we can compute the optimal pair of points. If $P \curvearrowright r$ (Line 4), we calculate the best T searching in Q_1 and the best L in Q_4 , otherwise, if $P \curvearrowleft r$ (Line 9), we do the opposite reasoning. Eventually, we return the best pair T and L (Line 14).

B. The MULTI Algorithm

The MDP extension to multi-line assumes that the truck's path is a polygon. The sides of the polygon determine the set Π of different routes on which to run the SINGLE algorithm. In this case, we need to select not only the take-off and landing points, but also the side of the polygon where T and L reside.

Algorithm 2: MULTI $(P, \omega, t, \Pi, v_d, \kappa)$

```

1  $\mathcal{T} \leftarrow \emptyset, \mathcal{L} \leftarrow \emptyset$ 
2 foreach  $\pi \in \Pi$  do
3    $\{T, L\} \leftarrow \text{SINGLE}(P, \omega, t, \pi)$ 
4    $\mathcal{T} \leftarrow \mathcal{T} \cup T, \mathcal{L} \leftarrow \mathcal{L} \cup L$ 
5  $T \leftarrow -\infty, L \leftarrow +\infty$ 
6 for  $i \in 0, \dots, |\mathcal{T}|$  do
7   for  $k \in i, \dots, |\mathcal{L}|$  do
8     if  $E(\overline{\mathcal{T}[i]P}) + E(\overline{\mathcal{L}[k]P}) < E(\overline{TP}) + E(\overline{LP})$  then
9        $T \leftarrow \mathcal{T}[i], L \leftarrow \mathcal{L}[k]$ 
10 return  $T, L$ 
```

Moreover, T and L can be selected on two distinct sides. The pseudocode of MULTI is sketched in Algorithm 2.

Firstly, we build the two sets \mathcal{T} and \mathcal{L} which contain the set of feasible take-off and landing points computed for each route $\pi \in \Pi$ (Algorithm 2, Line 1–4). After this, we compute the optimal solution by searching among all the possible pairs (T, L) in \mathcal{T} and \mathcal{L} , paying attention that T precedes L (Line 6–8). The current best pair is then saved (Line 9) and eventually returned (Line 10). The algorithm requires $\mathcal{O}(|\Pi|^2 + |\Pi|t)$ time.

C. The FEASIBILITY Algorithm

The FEASIBILITY algorithm finds the two thresholds, i.e., P_{\max} and P_{\min} , used to establish the feasibility of a delivery P regardless of the wind conditions for a given drone's battery budget. The algorithm considers all the winds $\omega_d \in \Omega^2$. It takes $\mathcal{O}(|\Omega|t) = \mathcal{O}(t)$ time and $\mathcal{O}(1)$ space, where t is the number of sectors of the compass rose. The pseudocode of FEASIBILITY is sketched in Algorithm 3.

Algorithm 3: FEASIBILITY $(P, \omega, t, r, v_d, \kappa)$

```

1  $\mu_0^T, \dots, \mu_{4t-1}^T, \mu_0^L, \dots, \mu_{4t-1}^L \leftarrow \text{UNIT\_COSTS}(\omega)$ 
2  $\overline{HP}_{\max} \leftarrow 0, \overline{HP}_{\min} \leftarrow +\infty$ 
3 foreach  $\omega_d \in \Omega$  do
4    $\varphi(\vec{r}) \leftarrow |\omega_d - \gamma_r|_{360}, \tau \leftarrow \lfloor |\varphi(\vec{r}) - 1|_{360} / \sigma \rfloor$ 
5    $\Delta_T^* \leftarrow \min_{0 \leq j \leq t} \left\{ \frac{\mu_{|\tau-j|_{4t}}^T}{\sin(\delta_{|\tau-j|_{4t}}^T)} \right\}$ 
6    $\Delta_L^* \leftarrow \min_{0 \leq j \leq t} \left\{ \frac{\mu_{|\tau+j|_{4t}}^L}{\sin(\delta_{|\tau+j|_{4t}}^L)} \right\}$ 
7    $\overline{HP}_M(\omega_d) \leftarrow \frac{B}{\Delta_T^* + \Delta_L^*}$ 
8   if  $\overline{HP}_M(\omega_d) > \overline{HP}_{\max}$  then  $\overline{HP}_{\max} \leftarrow \overline{HP}_M(\omega_d)$ 
9   if  $\overline{HP}_M(\omega_d) < \overline{HP}_{\min}$  then  $\overline{HP}_{\min} \leftarrow \overline{HP}_M(\omega_d)$ 
10 return  $P_{\max}, P_{\min}$ 
```

Note that varying $\omega_d \in \Omega$ also varies the relative wind, but not the energy coefficients $\mu_0^T, \dots, \mu_{4t-1}^T, \mu_0^L, \dots, \mu_{4t-1}^L$. Only different sectors of the compass rose will be considered for different relative winds, as explained before. In our experiments, FEASIBILITY is then extended considering not only the truck's path \vec{r} but all the lines Π of the sides of the truck's polygon path. Note that changing r , although the energy coefficient of the compass rose are the same, the trajectory angles will change, as explained before.

²This is implemented by varying ω_d with regular steps between 0 to 360.

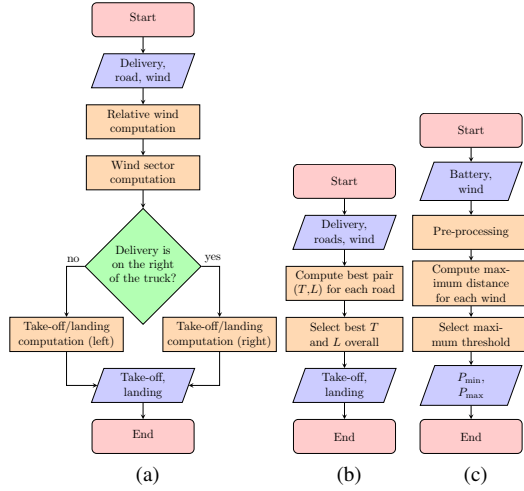


Fig. 6. Flowchart of SINGLE (a), MULTI (b), and FEASIBILITY (c).

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our algorithms when solving MDP. Specifically, we perform two kinds of evaluation: *numerical* in Section VI-B, and *simulation* in Section VI-C. In the previous, we simply run our proposed algorithms by varying all the aforementioned parameters, and by relying on our ad-hoc and coded environment. In the latter, instead, we rely on the open air simulator called BlueSky [29].

A. Settings and Parameters

We implemented our algorithms in Python language version 3.7, and run all the instances on an Intel i7-10genK computer with 16GB of RAM. We consider a delivery area that consists of a circle of radius 5km whose origin coincides with the Cartesian coordinate system. When considering the *single line* (SL) scenario, we simply generate a line inside the area. In this case, the truck travels on a straight line (e.g., a highway), and deliveries take place in the area in front of such line. On the other hand, when we consider the *multi-line* (ML) scenario, we randomly generate a convex polygon with 8 sides that represent the closed path (route) of the truck to/from the depot. In other words, there are 8 contiguous roads that the truck traverses in the delivery area. By setting the starting point of the truck's route as one of the vertices of the polygon, the route follows the sequence of vertices traveled in a clockwise direction. In this case, the truck circumnavigates the area where the deliveries occur. Also, $n = 20$ deliveries P_1, \dots, P_{20} are uniformly generated inside the delivery area.

We also consider two types of winds in the evaluation: *exhaustive winds* (EW) and *real winds* (RW). In EW, we vary ω_d with regular steps and consequently the relative wind $\varphi(\vec{r})$ on r varies. In RW, the wind varies according to the collected data at different hours of the day in two locations in Corsica (France) during winter days. In general, such a region is characterized by low-intensity inland wind and high-intensity sea wind. When considering the EW scenario, we vary the wind speed $\omega_s = \{10, 20\}$ m/s and the wind direction which varies according to a step of 15° . Moreover, we set the wind cardinality to $t = 3$ affecting so the wind sectors and classes.

Concerning the parameters of the drone, we set the speed of the drone $v_d = \{10, 20, 30\}$ m/s, the weight of the payload $\kappa = \{2, 6\}$ kg, and the battery budget $B = \{2.5, 5, 10\}$ MJ when dealing with the feasibility.

Also, as a reference, we compare our solution (minimum energy trajectory) with respect to the shortest trajectory, on both the single line (SINGLE-ST) and multi-line (MULTI-ST) scenarios, where ST denotes “shortest trajectory” (i.e., the one with $\delta = 90$) that is optimal when the wind is absent. So, our numerical evaluations compare the energy required in presence of wind with the energy required in absence of wind.

B. Numerical Evaluation

In this section, we perform a numerical evaluation concerning the proposed algorithms. We will consider the compass rose formed by 12 sectors and hence $\sigma = 30$ (case $t = 3$). In Table II, we report the different values of μ_i that have been used in the experiments. We tabulated these values since there is no simple formula to calculate them [10].

TABLE II
OCTOCOPTER ENERGY CONSUMPTION μ_i FIXING $v_d = 10$ m/s AND $\omega_s = 10$ m/s, AND WITH DIFFERENT PAYLOADS κ .

S_i	payload κ			S_i	payload κ		
	0	2	6		0	2	6
0	0.123	0.151	0.212	6	0.567	0.602	0.677
1	0.148	0.177	0.239	7	0.532	0.567	0.640
2	0.221	0.251	0.316	8	0.442	0.474	0.545
3	0.446	0.478	0.549	9	0.218	0.247	0.313
4	0.534	0.569	0.642	10	0.147	0.175	0.238
5	0.567	0.602	0.677	11	0.123	0.151	0.212

1) *Single Line Exhaustive Winds Scenario*: In Figure 7, we focus on the SL-EW scenario. In the x -axis we report the scanned wind directions ω_d , while in the y -axis we report the ratio among the energy required for performing a delivery P in absence of wind (by invoking SINGLE-ST), and the optimal energy required in presence of wind (by invoking SINGLE).

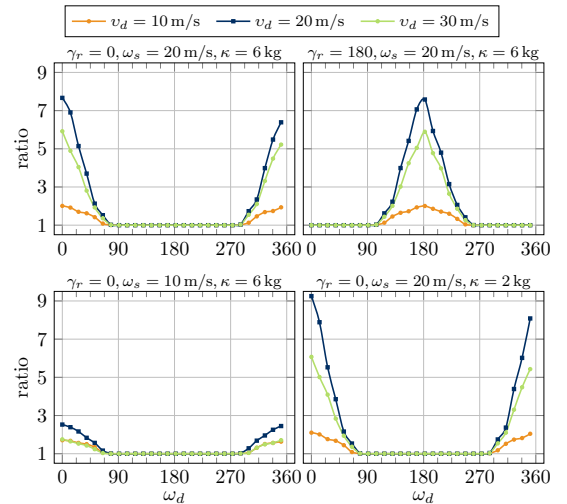


Fig. 7. SL-EW scenario: SINGLE vs SINGLE-ST.

In the first row of Figure 7 we initially see that by fixing a configuration of the speed and payload of the drone, and

varying the direction of the truck's route γ_r , the results are the same. In fact, they have just shifted by 180° . So, changing the direction of the road, the compass rose just rotates. Namely, for instance, the values on the y -axis when $\omega_d = 45$ and $\gamma_r = 0$ are exactly the same when $\omega_d = 45 + 180$ and $\gamma_r = 180$. Also the lengths of the minimum trajectories are the same because the δ angles coincide being $|\varphi(0)|_{30} = |\varphi(180)|_{30} = 0$. Focusing on the first plot with $\gamma_r = 0$, we can see that when the wind $\omega_d \in [90, 270]$, the ratio is 1, which means that the two solutions work the same. In other words, the best solution provided by SINGLE forces the drone to fly perpendicularly like SINGLE-ST. Eventually, if there is a trajectory that starts from r and reaches P which is "more" tailwind (i.e., parallel or almost parallel to the global wind), SINGLE finds much better solutions with respect to the shortest perpendicular ones. In particular, it is interesting to see how much energy can be saved when v_d varies in these circumstances.

In Figure 7, where $\omega_s = 20\text{m/s}$, when $v_d = 20\text{m/s}$ the minimum trajectory consumes $1/8$ of the energy in absence of wind, when $v_d = 30\text{m/s}$, $1/6$, and when $v_d = 10\text{m/s}$, $1/2$. This behavior is expected and comes from the adopted drone's energy reported for completeness in Section A: the energy does not change linearly with v_d as it seems at first glance in Eq. (9). Indeed, the energy gain also depends on the air-speed a_s and as such on the relative difference between v_d and ω_s as reported in Eq. (1). The gain is emphasized when such difference is positive, and it is reduced when such difference is negative. When we decrease the wind speed ω_s from 20m/s to 10m/s (i.e., plots of the first column), the energy saved is less because the head/tail component decreases. Finally, for the payload weight (i.e., comparing $\kappa = 6\text{kg}$ and $\kappa = 2\text{kg}$), the energy saved is more when the payload is lighter.

2) *Number of Sectors Analysis:* In Figure 8, we analyze how the number of sectors can impact on the performance.

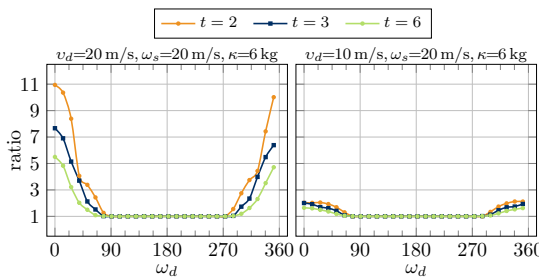


Fig. 8. SL-EW scenario: comparison between different number of sectors.

Both the x -axis and y -axis have the same meaning as those in Figure 7. We varied the value of t . With $t = 2$, we have 4 different classes and therefore sectors with a width of 45° ; with $t = 3$, we have 6 different classes and sectors with a width of 30° ; and finally, with $t = 6$, we have 12 different classes and sectors with a width of 15° .

The discretized angle (in more classes) makes the energy consumption calculation more accurate, and therefore the used representative angle is closer to the actual experienced angle by the drone (i.e., when $t = 6$). However, when there are more sectors to deal with, the running time for computing the

correct angles in presence of wind increases accordingly. On the other hand, when the number of classes is small, the energy consumption calculation is less accurate. So, for this reason, we decided to use a good trade-off between performance and complexity, relying on the average case with $t = 3$.

3) *Single Line Real Winds Scenario:* In Figure 9, we focus on the SL-RW scenario by using real data of winds collected in a few locations in Corsica, France. In the x -axis we report the time (hours) of a winter day, while in the y -axis we report the same ratio as already explained in Figure 7.

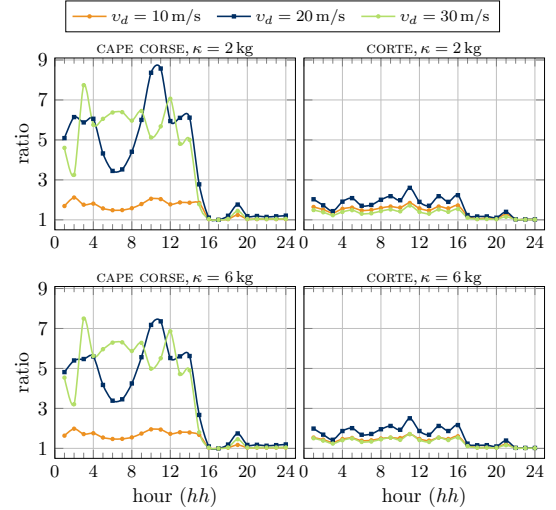


Fig. 9. SL-RW scenario: SINGLE vs SINGLE-ST.

We considered two cities that are differently affected by the wind, i.e., Cape Corse and Corte, during a whole day, from hour 1 to hour 24, during the winter season. Cape Corse is close to the sea where the wind strongly blows, while Corte is an inland city, and therefore the wind is much more moderate. In the first row of Figure 9 we report the results when the payload is 2kg , while in the second one when the payload is 6kg . The most interesting thing to observe is the impact of the wind in the first two-thirds of the day in Cape Corse. So, due to the strong wind in Cape Corse, the drone took advantage of the wind in its favor thus saving a lot of energy.

A counter-intuitive but correct behavior can be observed in Cape Corse when the drone's speed is the highest, i.e., $v_d = 30\text{m/s}$. In the previous Figure 7 we had seen that when the drone flies very fast, the energy saved is less. However, in Cape Corse the recorded wind from 4AM to 8AM was severely strong (even more than $30\text{m/s} \approx 110\text{km/h}$) and the drone, although flying at $v_d = 30\text{m/s}$, was able to save more energy than when it flies at the "most suitable" speed $v_d = 20\text{m/s}$. So, the energy saving depends on the relationship of the wind and drone's speed, and not only on drone's speed. Namely, the tailwind component is a function of ω_s in Eq. (1).

In Corte, however, since the wind was almost absent, and therefore the unitary energy consumption was almost the same for all the wind sectors, SINGLE prefers shorter routes, if not the shortest, to save energy. Comparing the results when the payload changes, also in this case the energy saved is more when the payload to carry is less.

4) *Multi-line Scenario*: In Figure 10, we focus on the ML case on both EW (first row) and RW (second row) scenarios. On each plot, in the x -axis we report the i^{th} randomly generated delivery point P_i . The plots to the left report the energy, while the ones to the right report the length of the routes. The used parameters are $\omega_s = 20\text{m/s}$, $\kappa = 6\text{kg}$.

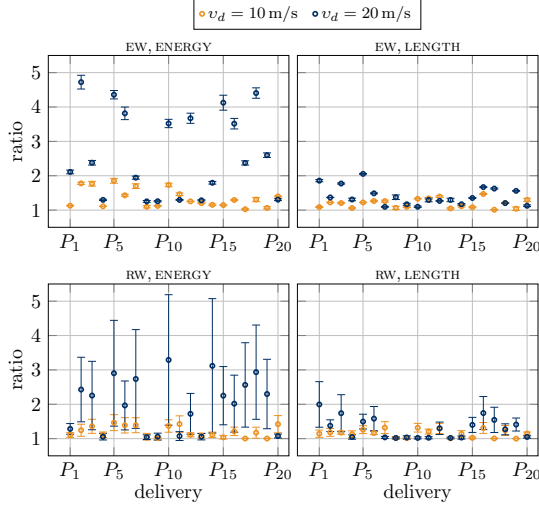


Fig. 10. SL-EW/RW scenarios: MULTI vs MULTI-ST.

In Figure 10, when considering the ML-EW scenario (first row) with all the winds ω_d , in the y -axis we report the *average* ratio (along with the 95% confidence interval) of the performance between MULTI-ST and MULTI in terms of energy (left plot) and length of paths (right plot). We can observe that on average, the energy saving of the 20 deliveries, when the drone's speed $v_d = 10\text{m/s}$, is pretty stable. The distribution of the deliveries shows that the ratio is between 1 and 2, which means that, considering all the winds, the shortest trajectory is not that bad. However, when the drone's speed increases to $v_d = 20\text{m/s}$, the same distribution of points is more variable. On average, the ratio is around 3, but many values are close to 5. In these cases, the wind helps in saving energy. About the length of the paths, the ratio is between the length of the paths of MULTI and the length of the paths of MULTI-ST. It is interesting to see that when $v_d = 20\text{m/s}$, for the randomized delivery P_5 the energy saved was really large, and the length of the path is more than twice the shortest one. On average, for this speed, the paths are around 50% longer than the shortest ones. In Figure 10, when considering the RW scenario (second row), in the y -axis we report the same ratios just explained for the first row but considering all the real winds evaluated from 10 consecutive days in Corsica. The saving is high when $v_d = 20\text{m/s}$. This trend also impacts the length of the paths (second row, right plot).

5) *Feasibility Analysis*: In Figures 11–12, we focus on the feasibility analysis in the SL/ML-EW scenario.

In Figure 11, we report the in the x -axis the relative wind of the truck $\varphi(r)$, while in the y -axis the value P_{\max} which depends on the relative wind. Here, we consider $\omega_s = 10\text{m/s}$ and $\kappa = 6\text{kg}$, while we vary the size of the battery 5MJ to the left, and 10MJ to the right. Not surprisingly, the values

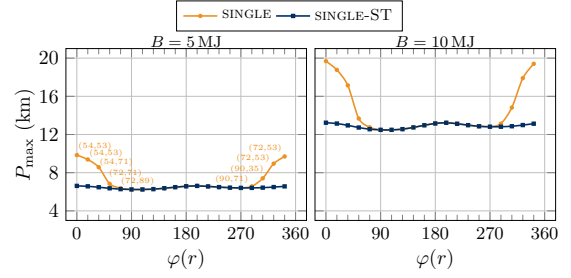


Fig. 11. SL-EW scenario: furthest deliveries that can be served in presence of wind (SINGLE) and in absence of wind (SINGLE-ST) given $\varphi(r)$.

of P_{\max} with twice the battery are doubled because having more energy available means flying more distance. About the case with 5MJ of energy, the farthest distance that the drone can reach is around 10km whose pair of take-off and landing angles is (54, 53) when $\varphi(r) = 0$ (in orange). On the same relative wind, the drone could have gone at a maximum of 7km far away from the truck's road exploiting the shortest trajectory (in blue). We did not report the pairs of angles to the plot on the right since they match the ones on the left.

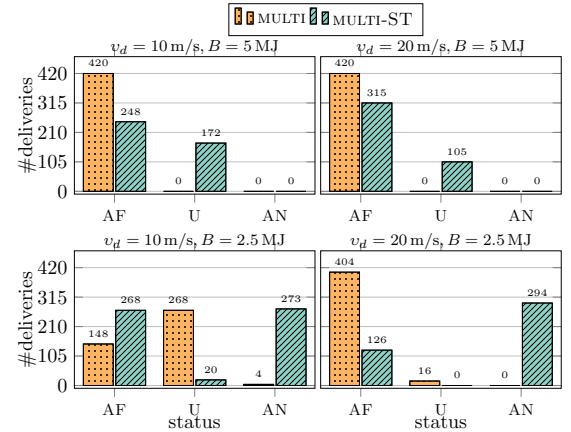


Fig. 12. ML-EW scenario: status of deliveries.

In Figure 12, we report the in the x -axis the *status* of the deliveries taking into account P_{\max} and P_{\min} , while in the y -axis the number of deliveries for that particular status, in the ML-EW scenario. Here, we consider the size of the battery $B = 2.5\text{MJ}$ to the left, and 5MJ to the right. A delivery can have one of the following status: *always feasible* (AF), if it is closer than P_{\min} ; *always non-feasible* (AN), if it is farther than P_{\max} ; *unknown* (U), otherwise. It is really interesting to observe the number of U and AF when comparing MULTI and MULTI-ST. In fact, when relying on MULTI it is possible to perform at least 35% of deliveries more than those done with MULTI-ST. Moreover, the number of non-feasible deliveries dramatically drops down when performing MULTI.

C. Simulation Evaluation

For evaluating our algorithms on a simulated environment, we rely on BlueSky. BlueSky is an open Air Traffic Simulator (ATS), and is meant as a tool to perform research on Air

Traffic Management and Air Traffic Flows [29]. At the best of our knowledge, BlueSky is the only simulator in which it is possible to simulate the flight of different aircraft by injecting the presence of the wind. Although BlueSky can model aircraft under the effect of the wind, there is not any available option that computes the actual energy consumption of them. So, we created an ad-hoc plug-in for BlueSky (still at a very preliminary stage though) that can estimate the energy consumed when a drone flies. When a flight simulation is done, BlueSky creates a log file (in CSV format) in which each line comprises different fields, like GPS coordinates, drone's speed, global wind, and so on. Basically, we parse the log file and for each two consecutive lines, we extract the two GPS coordinates, so that we can compute the actual flown distance and the drone's heading. Accordingly, knowing the wind experienced by the drone, we can then *precisely* estimate the energy consumption of it by using Eq. (10) in [10] (for completeness, reported in Eq. (10) in Section A). Note that, the equation in [10] considers the current wind (i.e., $t = 90$) and so the energy consumption calculation is much accurate (and slower to compute) than our proposed model.

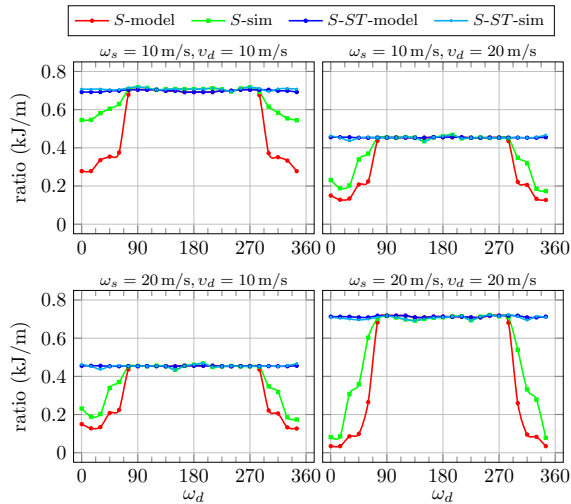


Fig. 13. The simulated algorithms under BlueSky: *S* stands for SINGLE.

In Figure 13, we report the in the x -axis the wind direction, while in the y -axis the energy consumption per unit of distance (i.e., kJ/m). The energy per unit of distance is calculated through the ratio *total flown distance* to *total energy consumed*. Here, we have to specify two different cases: the results shown under the suffix “model” are determined exactly as done before in the numerical evaluation, while the ones under the suffix “sim” are determined inside the BlueSky simulator. The drone's altitude is 40m above the ground. We set the other parameters with respect to the settings in Section VI-A.

The simulated results confirm the numerical ones. The difference among the simulated and ideal perpendicular paths (SINGLE-ST model and sim) is almost zero, while slightly different is the situation with the non-perpendicular ones. We also observe that the largest energy consumption per meter occurs when the wind and the drone's speed match (i.e., $\omega_s = v_d$). In general, one can observe that the saved energy is

less in the simulated environment than in the ideal numerical environment. Except the case with $\omega_s = v_d = 10\text{m/s}$, the gap in terms of required energy per unit of distance is $\leq 0.1\text{kJ/m}$ in favor of the ideal numerical model. This is because the simulator uses more complicated trajectories for the aircraft while flying between two points. We have indeed observed that the drone does turning maneuvers when changing its direction.

VII. CONCLUSIONS

For the first time, to the best of our knowledge, we adapt the trajectory of a drone to the wind. We consider a truck-drone tandem delivery system. The drone actively reacts to the wind by adopting the “most tailwind” available trajectory between the truck's path and the delivery. Two solutions are proposed for solving MDP, i.e., when the truck moves on a line (e.g., a highway) in front of the deliveries, and when the truck moves on a polygon that delimits a convex area where the deliveries take place. Currently, our proposed model presents some limitations. For instance, if the wind dynamically changes during a delivery, the drone does not update its path in order to save energy. Moreover, the discretization in multiple wind classes determines approximated angles if we worked in the continuous space, and sometimes the energy estimations differ from the actual ones. Nevertheless, our proposed model has the undoubted value of showing the positive role that the wind can play in the drone flight.

In the future, we would like to confirm our findings by using other simulators, and eventually extending our investigation to a test-bed made by off-the-shelf drones. We will also include discussions on what can go wrong, e.g., time of day/night when winds can be unstable and how to include flight time scheduling to manage different winds at different hours.

APPENDIX DRONE-WIND ENERGY MODEL

The motion of drones is regulated by physical properties [10]. The total required thrust is $\mathbb{T} = Wg + F_D$, where W is the total weight of the drone, g is the gravitational constant, and $F_D = \frac{1}{2}\rho a_s^2 C_D A$ is the total drag force, where ρ is the air density, a_s is the drone's air speed (see Eq. (1)), $A = \pi R^2$ is the cross sectional area (R is the rotor radius), and C_D is the drag coefficient. Having computed \mathbb{T} , we estimate the required power \mathbb{P} for a steady flight, which is $\mathbb{P} = \mathbb{T}(v_d \sin(\alpha) + s_i)$, where $\alpha = \arctan(F_D/Wg)$ is the pitch angle, and s_i , which is the induced velocity required for a given thrust \mathbb{T} , can be obtained by solving the implicit equation [30] $s_i = s_h^2 / \sqrt{(v_d \cos(\alpha))^2 + (v_d \sin(\alpha) + s_i)^2}$, where $s_h = \sqrt{\mathbb{T}/2\rho A}$ is the induced velocity at hover [30]. Note that, F_D and so α and \mathbb{P} depend on the drone's direction γ_r . So, fixed $\omega = (\omega_d, \omega_s)$, the unitary energy $\mu(\gamma_r)$ of travel along a segment r with direction γ_r is calculated as follows:

$$\mu_\omega(\gamma_r) = \mathbb{P}/v_d. \quad (9)$$

Therefore, the energy consumed for traversing one edge γ_r of length $\lambda(\gamma_r)$ can be expressed as:

$$d_\omega(\gamma_r) = \mu_\omega(\gamma_r)\lambda(\gamma_r). \quad (10)$$

REFERENCES

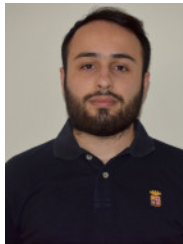
- [1] S. Ahirwar, R. Swarnkar, S. Bhukya, and G. Namwade, "Application of drone in agriculture," *Intl. Journal of Current Microbiology and Applied Sciences*, vol. 8, no. 01, pp. 2500–2505, 2019.
- [2] T. S. de Smet, A. Nikulin *et al.*, "Successful application of drone-based aeromagnetic surveys to locate legacy oil and gas wells in cattaraugus county, new york," *Journal of Applied Geophysics*, vol. 186, no. C, 2021.
- [3] A. M. Seid *et al.*, "Multi-agent drl for task offloading and resource allocation in multi-uav enabled iot edge network," *Trans. on Network and Service Management*, vol. 18, no. 4, pp. 4531–4547, 2021.
- [4] F. B. Sorbelli, S. K. Das, C. M. Pinotti, and G. Rigoni, "A comprehensive investigation on range-free localization algorithms with mobile anchors at different altitudes," *Elsevier PMC*, vol. 73, p. 101383, 2021.
- [5] A. Trotta, F. D. Andreagiovanni, M. Di Felice *et al.*, "When uavs ride a bus: Towards energy-efficient city-scale video surveillance," in *INFOCOM 2018*. IEEE, 2018, pp. 1043–1051.
- [6] F. B. Sorbelli, F. Corò, S. Das, L. Palazzetti, and C. Pinotti, "On the scheduling of conflictual deliveries in a last-mile delivery scenario with truck-carried drones," *Pervasive and Mobile Comp.*, p. 101700, 2022.
- [7] F. B. Sorbelli, C. M. Pinotti, and G. Rigoni, "On the evaluation of a drone-based delivery system on a mixed euclidean-manchattan grid," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [8] D. Schneider, "The delivery drones are coming," *IEEE Spectrum*, vol. 57, no. 1, pp. 28–29, 2020.
- [9] B. Girardet, L. Lapasset, D. Delahaye, and C. Rabut, "Wind-optimal path planning: Application to aircraft trajectories," in *13th Intl. Conf. on Control Automation Robotics & Vision*. IEEE, 2014, pp. 1403–1408.
- [10] J. K. Stolaroff, C. Samaras, E. R. O'Neill *et al.*, "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery," *Nature communications*, vol. 9, no. 1, pp. 1–13, 2018.
- [11] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation: Emerging Technologies*, vol. 54, pp. 86–109, 2015.
- [12] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458, 2018.
- [13] N. Boysen, S. Fedtke, and S. Schwerdfeger, "Last-mile delivery concepts: a survey from an operational research perspective," *OR Spectrum*, pp. 1–58, 2020.
- [14] L. Palazzetti, C. M. Pinotti, and G. Rigoni, "A run in the wind: favorable winds make the difference in drone delivery," in *17th Intl. Conf. on Distr. Comp. in Sensor Systems (DCOSS)*. IEEE, 2021, pp. 109–116.
- [15] N. A. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *ERIM Report Series Reference No. ERS-2015-011-LIS*, 2018.
- [16] J. C. de Freitas and P. H. V. Penna, "A variable neighborhood search for flying sidekick traveling salesman problem," *Intl. Trans. in Operational Research*, vol. 27, no. 1, pp. 267–290, 2020.
- [17] G. C. Crişan and E. Nechita, "On a cooperative truck-and-drone delivery system," *Procedia Computer Science*, vol. 159, pp. 38–47, 2019.
- [18] N. Boysen, D. Briskorn, S. Fedtke, and S. Schwerdfeger, "Drone delivery from trucks: Drone scheduling for given truck routes," *Networks*, vol. 72, no. 4, pp. 506–527, 2018.
- [19] T. Nguyen and T.-C. Au, "Extending the range of delivery drones by exploratory learning of energy models," in *Autonomous Agents and MultiAgent Systems*. IFAAMAS, 2017, pp. 1658–1660.
- [20] F. B. Sorbelli, F. Corò, S. K. Das, and C. M. Pinotti, "Energy-constrained delivery of goods with drones under varying wind conditions," *IEEE Trans. on Intelligent Transportation Systems*, 2020.
- [21] L. Palazzetti, "Routing drones being aware of wind conditions: a case study," in *17th Intl. Conf. on Distr. Comp. in Sensor Systems (DCOSS)*. IEEE, 2021, pp. 343–350.
- [22] A. Khanda, F. Corò, F. B. Sorbelli, C. M. Pinotti, and S. K. Das, "Efficient route selection for drone-based delivery under time-varying dynamics," in *18th MASS*, 2021.
- [23] A. Thibbotuwawa, G. Bocewicz *et al.*, "Planning deliveries with uav routing under weather forecast and energy consumption constraints," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 820–825, 2019.
- [24] —, "Uav mission planning resistant to weather uncertainty," *Sensors*, vol. 20, no. 2, p. 515, 2020.
- [25] G. Radzki *et al.*, "A proactive approach to resistant uav mission planning," in *Conf. on Automation*. Springer, 2020, pp. 112–124.
- [26] J. Janszen, B. Shahzaad *et al.*, "Constraint-aware trajectory for drone delivery services," in *ICSOC*. Springer, 2021.
- [27] T. Kirschstein, "Energy demand of parcel delivery services with a mixed fleet of electric vehicles," *Cleaner Engineering and Technology*, vol. 5, p. 100322, 2021.
- [28] J. M. Wallace and P. V. Hobbs, *Atmospheric science: An introductory survey*. Amsterdam: Elsevier Academic Press., 2006.
- [29] J. M. Hoekstra and J. Ellerbroek, "Bluesky atc simulator project: an open data and open source approach," in *7th Intl. Conf. on Research in Air Transportation*, vol. 131. FAA/USA/Europe, 2016, p. 132.
- [30] W. Johnson, *Helicopter theory*. Courier Corporation, 2012.



Francesco Betti Sorbelli is currently an assistant professor at the University of Perugia, Italy. He received the Bachelor and Master degrees *cum laude* in Computer Science from the University of Perugia in 2007 and 2010, respectively, and his Ph.D. in Computer Science from the University of Florence, Italy, in 2018. He was a postdoctoral researcher at the Missouri University of S&T, USA, in 2020, and at the University of Perugia in 2018, 2021–2022. His research interests include algorithms design, combinatorial optimization, unmanned vehicles.



Federico Corò is currently a postdoctoral researcher in the Computer Science Department of the University of Perugia, Italy. He earned a PhD in Computer Science at the Gran Sasso Science Institute in L'Aquila, Italy, in 2019. Previously, he was a postdoctoral researcher at La Sapienza University of Rome, Italy (2019–2020), and at the Missouri University Science and Technology, USA (2021–2022). His research interests include unmanned vehicles, combinatorial optimization, and the design and efficient implementation of algorithms.



Lorenzo Palazzetti received the Bachelor and Master degrees in Computer Science from the University of Perugia, Italy, in 2018 and 2020, respectively. He is currently a third year PhD student under the supervision of Prof. Cristina M. Pinotti at the University of Florence, Italy. His research interests include design and analysis of algorithms, combinatorial optimization, unmanned vehicles.



Cristina M. Pinotti received the Master degree *cum laude* in Computer Science from the University of Pisa, Italy, in 1986. In 1987–1999, she was Researcher with the National Council of Research in Pisa. In 2000–2003, she was Associate Professor at the University of Trento. Since 2004, she is a Full Professor at the University of Perugia. Her current research interests include the design and analysis of algorithms, combinatorial optimization, emerging technologies, wireless sensor networks, and communication networks.



Giulio Rigoni received the Bachelor and Master degrees in Computer Science from the University of Padua, Italy, in 2012 and 2017, respectively, and his Ph.D. in Computer Science from the University of Florence, Italy, in 2022. Currently, he is a postdoc researcher at the University of Padua, Italy. He is a member of the SPRITZ Security & Privacy Research Group since 2018. His research interests include unmanned vehicles, security and machine learning.