Optimal Trajectories of a UAV Base Station Using Hamilton-Jacobi Equations

Marceau Coupechoux, Jérôme Darbon, Jean-Marc Kélif, and Marc Sigelle

Abstract—We consider the problem of optimizing the trajectory of an Unmanned Aerial Vehicle (UAV). Assuming a traffic intensity map of users to be served, the UAV must travel from a given initial location to a final position within a given duration and serves the traffic on its way. The problem consists in finding the optimal trajectory that minimizes a certain cost depending on the velocity and on the amount of served traffic. We formulate the problem using the framework of Lagrangian mechanics. We derive closed-form formulas for the optimal trajectory when the traffic intensity is quadratic (single-phase) using Hamilton-Jacobi equations. When the traffic intensity is bi-phase, i.e. made of two quadratics, we provide necessary conditions of optimality that allow us to propose a gradient-based algorithm and a new algorithm based on the linear control properties of the quadratic model. These two solutions are of very low complexity because they rely on fast convergence numerical schemes and closed form formulas. These two approaches return a trajectory satisfying the necessary conditions of optimality. At last, we propose a data processing procedure based on a modified K-means algorithm to derive a bi-phase model and an optimal trajectory simulation from real traffic data.

Index Terms—Cellular networks, UAV, Base station, trajectory optimization, optimal control.

1 Introduction

Unmanned Aerial Vehicles (UAV) are expected to play an increasing role in future wireless networks[1] [1], [2], [3], [4], [5]. UAVs may be seen as a new type of User Equipment for the cellular network, but they may also act as flying Base Stations (BS). A UAV BS can indeed complement the traditional cellular networks by providing enhanced coverage and capacity. Thanks to favorable channel conditions, it can provide a high data rate wireless service or serve as a relay to reach distant users outside the coverage of cellular networks. In disaster or emergency situations, a UAV provides a temporary infrastructure, which is fast, flexible and low cost, to retrieve and store vital information (survivors locations, emergency calls, disaster pictures, etc) and carry this information to a management center. It may also help rescue teams to communicate if the terrestrial infrastructure is damaged. In Internet of Things (IoT), Device-to-Device (D2D) or vehicular networks, UAVs can easily retrieve (or disseminate) information from (to) the devices. When there are temporary special events, like a football match or a live concert, terrestrial networks face a demand increase that may reduce the quality of service of the communications. Using on-demand UAVs is a flexible way of tackling this challenge.

- *M. Coupechoux is with LTCI, Telecom Paris, Institut Polytechnique de Paris, France. He has performed this work at LINCS laboratory. E-mail: marceau.coupechoux@telecom-paris.fr*
- *J. Darbon is with Brown University, US. He is supported by NSF-DMS-1820821. E-mail: jerome_darbon@brown.edu*
- *J.-M. Klif is with Orange Labs, Chtillon, France. E-mail: jeanmarc.kelif@orange.com.*
- *M. Sigelle is on leave from Telecom Paris, France. E-mail: marc.sigelle@gmail.com*

1. This work was partially supported by the European Unions Horizon 2020 Research and Innovation Program under the 5G!Drones project (Grant No. 857031).

In many of these applications, a path planning problem arises. On the one hand, a UAV shall serve as much traffic as possible (data traffic in a crowded area, number of IoT devices, number of emergency calls, etc.). On the other hand, its energy consumption, which depends on its velocity, has to be minimized. The problem we tackle in this paper is thus to find an optimal trajectory for a UAV BS that minimizes a certain cost depending on the velocity and on the amount of served traffic. We particularly focus on the scenario where the UAV is supposed to offload data traffic from the terrestrial cellular network in a region characterized by hot spots, i.e., locations of very high traffic demand. In the literature, several approaches to this problem rely on space or time discretization, on the formulation of mixed-integer non-convex problems and on complex approximation algorithms. On the contrary, we propose here a solution based on the Lagrangian mechanics framework and the use of Hamilton-Jacobi (HJ) equations. This approach leads to very low complexity algorithms. We show how from real measured data, we can apply our model and obtain optimal trajectories at a click speed.

1.1 Related Work

UAV trajectory optimization for networks has been tackled maybe for the first time in [6]. The model consists in a UAV flying over a sensor network from which it has to collect some data. The UAV can learn from previous experience, which is not assumed in our study.

The problem of optimally deploying UAV BSs to serve traffic demand has been addressed in the literature by considering static UAVs BSs or relays, see e.g. [7], [8]. The goal is to optimally position the UAV so as to maximize the data rate with ground stations or the number of served users. In these works, the notion of trajectory is either ignored or restricted to be circular or linear.

In robotics and autonomous systems, trajectory optimization is known as *path planning*. In this field, there are classical methods like Cell Decomposition, Potential Field Method, Probabilistic Road Map, or heuristic approaches, e.g. bio-inspired algorithms [9]. Authors of [10] have capitalized on this literature and proposed a path planning algorithm for drone BSs based on A* algorithm. The main goal of these papers is to reach a destination while avoiding obstacles. In [9], authors show that the main goal is decomposed into perception of the obstacles, localization of the robot, cognition and motion control. There is thus no notion of cost optimization along the trajectory. In [10], a UAV controlled by a cellular network selects a trajectory so as to avoid regions not covered by the terrestrial antennas. Again, there is no notion of cost optimization along the trajectory. On the contrary, in our work, we intend to minimize a certain cost function along the trajectory by controlling the velocity of the UAV.

This goal is studied in optimal control theory [11] and is applied for example in aircraft trajectory planning [12]. Most numerical methods in control theory can be classified in *direct* and *indirect* methods. In direct methods, the problem is transformed in a non linear programming problem using discretized time, locations and controls. Direct methods are heavily applied in a series of recent publications in the field of UAV-aided communications, see e.g. [13], [14], [15], [16], [17]. Formulated problems are usually non-convex. The standard approach is hence to rely on Successive Convex Approximation (SCA), which iteratively minimizes a sequence of approximate convex functions. SCA is known to converge to a Karush-Kuhn-Tucker solution under mild conditions [18] but the quality of the solution may heavily depend on the initial guess. Here, simple heuristics or solutions to the Travelling Salesman Problem (TSP) or the Pickup-and-Deliver Problem (PDP) can be used for finding an initial trajectory [19]. With direct methods, because of the discretization, the differential equations and the constraints of the systems are satisfied only at discrete points. This can lead to less accurate solutions than indirect methods and the quality of the solution depends on the quantization step [20]. Although every iteration of SCA has a polynomial time complexity, practical resolution time may dramatically increase with the quantization grid and the dimension of the problem.

On the other hand, indirect approaches relies on considering the Hamilton-Jacobi Partial Differential Equation associated to the optimal control problem (see e.g., [21], [22][chp. 10]). Several recent methods have been proposed to solve HJ Partial Differential Equations (PDE) in high dimensions. These include max-plus algebra methods [23], [24], dynamic programming and reinforcement learning [25], tensor decomposition techniques [26], sparse grids [27], model order reduction [28], polynomial approximation [29], optimization methods [30], [31], [32], [33] and neural networks [34], [35], [36], [37]. However, to our knowledge, these approaches above do not apply for the models considered in this paper.

In this paper, we consider certain indirect methods that provide analytical solutions for certain classes of optimal control problem as we have shown in a preliminary study [38]. Compared to this study, we have improved the gradient algorithm for bi-phase traffic, we have proposed a new low complexity algorithm for the bi-phase problem (the B-algorithm) and a data processing procedure to adapt real data to our quadratic model.

1.2 Contributions

Our contributions are the following:

- *Problem Formulation:* To the best of our knowledge, this is the first time, after our preliminary study [38], that the UAV BS trajectory problem is formulated using the formalism of Lagrangian mechanics and solved using Hamilton-Jacobi equations. This approach provides closed-form equations when the potential is quadratic and thus very low complexity solutions compared to existing solutions in the literature, see e.g. [13], [14], [15], [16].
- *Closed-form expression of the optimal trajectory with single phase traffic intensity:* When the traffic intensity map is made of a single hot spot or traffic hole, has a quadratic form (*single phase*), and is time-independent, closed form expressions for the optimal trajectory are derived. For a traffic hole, the trajectory is on an ellipse and corresponds to the case of an attractor in mechanics. It follows a hyperbola for a hot spot and corresponds to a repulsor in mechanics. Our contribution is here a reinterpretation of the data traffic as a positive potential field. Resulting trajectories have not been derived so far in the literature on Lagrangian mechanics [39].
- *Characterization of the optimal solution in bi-phase traffic intensity:* When the traffic map has two hot spots or traffic holes (*bi-phase*) whose regions are separated by interfaces and is time-independent, we derive necessary conditions to be fulfilled by the position and the instant at which the optimal trajectory crosses an interface (see Theorem 2.
- *A gradient algorithm for bi-phase traffic:* An in-depth analysis of convexity vs. non-convexity issues allows us to derive a gradient algorithm to solve the bi-phase problem (Algorithm 1). This algorithm finds a stationary point for the cost function. This algorithm has a complexity $O(1)$ at every iteration, whereas iterations of the sequential convex optimization technique have polynomial time complexity.
- *A new algorithm for the bi-phase optimization problem:* A new algorithm, called the B-algorithm (Algorithm 2), is proposed based on the linear control properties of the quadratic model. This algorithm relies on a bisection scheme the complexity of which is proportional to the logarithm of the desired precision and closed form formulas.
- *A data processing procedure:* We propose a method to preprocess real measured traffic data in order to derive a bi-phase quadratic model. This procedure is based on smoothing steps followed by a modified K-means algorithm adapted to our quadratic model (Algorithm 3). The optimal trajectory is computed in a region where real traffic data is available [40].

The paper is structured as follows. Section 2 gives the system model, its interpretation in terms of Lagrange mechanics, the problem formulation and preliminary results.

Section 3 characterizes the optimal trajectories for single- and bi-phase cases. Section 4 presents algorithms, Section 5 our data processing procedure and numerical experiments. Section 6 concludes the paper.

Notations: The usual Euclidean scalar product between $x \in \mathbb{R}^n$ and $y \in \mathbb{R}$ is denoted by $x \cdot y$. The Euclidean norm $\|x\|$ in $\mathbb{R}^n$ of $x \in \mathbb{R}^n$ is defined by $\|x\| := \sqrt{x \cdot x}$.

The set of matrices with m rows, n columns and real entries is denoted by $\mathcal{M}_{m,n}(\mathbb{R})$. The transpose of the $A \in \mathcal{M}_{m,n}(\mathbb{R})$ is denoted by $A^\dagger \in \mathcal{M}_{n,m}(\mathbb{R})$. We classically identify $\mathcal{M}_{m,1}(\mathbb{R})$ and $\mathcal{M}_{1,n}(\mathbb{R})$ as column vectors of $\mathbb{R}^m$ and row vectors of $\mathbb{R}^n$, respectively.

Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ defined by $f(x,y)$ where $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and $y = (y_1, \ldots, y_m) \in \mathbb{R}^m$. Let $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. We denote by $\frac{\partial f}{\partial x_i}(a,b)$ the partial derivative of f with respect to the variable x_i at $(a,b) \in \mathbb{R}^n \times \mathbb{R}^m$. We also introduce the notations

$$\nabla_x f(a,b) = (\frac{\partial f}{\partial x_1}(a,b), \ldots, \frac{\partial f}{\partial x_n}(a,b)) \in \mathbb{R}^n$$

and

$$\nabla_y f(a,b) = (\frac{\partial f}{\partial y_1}(a,b), \ldots, \frac{\partial f}{\partial y_m}(a,b)) \in \mathbb{R}^m$$

We also consider the following notation for partial Hessian matrices

$$\mathcal{M}_{m+n,m+n}(\mathbb{R}) \ni \nabla^2 f(a,b) = \begin{pmatrix} \nabla^2_{x,x} f(a,b) & \nabla^2_{x,y} f(a,b) \\ \nabla^2_{y,x} f(a,b) & \nabla^2_{y,y} f(a,b) \end{pmatrix}$$

where

$$\nabla^2_{x,x} f(a,b) \in \mathcal{M}_{n,n}(\mathbb{R}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(a,b) & \ldots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(a,b) \\ \vdots & \ldots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1}(a,b) & \ldots & \frac{\partial^2 f}{\partial x_n^2}(a,b) \end{pmatrix}$$

and Id_n denotes the identity matrix of $\mathcal{M}_{n,n}(\mathbb{R})$.

We shall see that the value function $S : \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R}$ will play a fundamental role in this paper. We use the notations (T_1, X_1, T_2, X_2) for S and therefore the partial derivatives of S at $(t_1, x_1, t_2, x_2) \in \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2$ are denoted as follows:

$$\frac{\partial S}{\partial T_1}(t_1, x_1, t_2, x_2)$$
$$\nabla_{X_1} S(t_1, x_1, t_2, x_2)$$
$$\frac{\partial S}{\partial T_2}(t_1, x_1, t_2, x_2)$$
$$\nabla_{X_2} S(t_1, x_1, t_2, x_2)$$

2 System Model and Lagrangian Mechanics Interpretation

2.1 System Model

We consider a network area characterized by a traffic density at position z and time t. We intend to control the trajectory and the velocity of a UAV base station, which is located in $z_0 \triangleq z(t_0)$ at t_0 and shall reach a destination $z_T \triangleq z(T)$ at T with the aim of minimizing a cost determined by the velocity and the traffic, defined hereafter by (1). The UAV is supposed to fly at a fixed altitude. We denote $a(t) = \frac{dz}{dt}(t)$ the velocity vector of the UAV BS along its trajectory. At (t,z), we assume that the UAV BS is able to cover an area, from which it can serve users (see Figure 1). The velocity of the UAV BS induces an energy cost. In this model, we control the velocity vector a of the UAV BS. The general form of the cost function is as follows

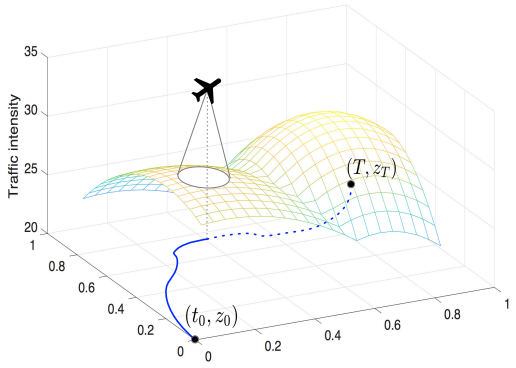


Fig. 1: A UAV Base Station travels from z_0 at t_0 to z_T at T and serves a user traffic characterized by its intensity.

$$\mathcal{L}(t,z,a) = \frac{K}{2}||a||^2 - u(t,z) \quad (1)$$

where the first term is a cost related to the velocity of the vehicle (K is a positive constant), and $\|\cdot\|$ denotes the usual Euclidean norm. The higher is the speed, the higher is the energy cost[2]. The second term is a *user traffic intensity*, i.e., the amount of traffic to be served by the UAV BS at (t,z). The traffic intensity is related to the amount of data users want to transmit. If all users have all the same requirement, traffic intensity is directly related to the number of users in an area. The goal of the UAV is to serve as much traffic as possible during the time duration T, while taking into account the energy cost. Note that a non-zero energy at null speed can be incorporated in the model by adding a constant. Without loss of generality, we assume that this constant is null.

Let $S(t_0, z_0, T, z_T)$ be the minimal total cost along any trajectory between z_0 at t_0 and z_T at T (also called *the action* in mechanics or *value function* in control theory). Let us define $\Omega(t_0, T)$ as the space of absolutely continuous functions from $[t_0; T]$ to $\mathbb{R}^2$. Our problem can now be formulated as follows

$$S(t_0, z_0, T, z_T) = \min_{a \in \Omega(t_0,T)} \int_{t_0}^{T} \mathcal{L}(s, z(s), a(s)) ds + J(z(T)) \quad (2)$$

where $\frac{dz}{dt}(t) = a(t)$, $z(t_0) = z_0$, and J is the terminal cost defined by $J(z) = 0$ if $z = z_T$ and $J(z) = +\infty$ otherwise. Note that knowing z_0 and the velocity $a(t)$ at any time instant t uniquely defines the trajectory. For simplicity reasons, we assume the existence and uniqueness of the optimal control $a^*(t)$ in (2) and denote the associated

2. The quadratic model represents the kinetic energy of a UAV of mass K. It is a good approximation of more realistic energy consumption models [14], [41], [42] provided that the UAV speed is not too low (typically not less than 10 m/s and 40 m/s for rotary-wing and fixed-wing UAVs, respectively). The parameter K can be chosen accordingly.

optimal trajectory $z^*(t)$. In a traffic map symmetric wrt. z_0 and z_T, the reader can convince himself that the uniqueness is not guaranteed.

2.2 Preliminary Results From Lagrangian Mechanics

We provide in this section important results from the Lagrangian mechanics for the convenience of the reader.

Definition 1 (Impulsion)**.** *The* impulsion *function is defined as* $p(t,z,a) := \nabla_a \mathcal{L}(t,z,a)$.

In the Newtonian classical framework that is used here (see (1)), the impulsion is the product of the particle mass by its velocity (hence the standard term "impulsion").

Definition 2. *The* Hamiltonian *function is defined as* $H(t,z,p) := \max_{a\in\mathbb{R}^2} p\cdot a - \mathcal{L}(t,z,a)$.

Lemma 1 (Euler-Lagrange Equations)**.** *Along the optimal trajectory* $z^*(t)$ *that starts from* z_0 *at* t_0 *and ends at* z_T *at* T*, we have*

$$\frac{d}{dt}\nabla_a \mathcal{L}(t,z^*(t),a^*(t)) = \nabla_z \mathcal{L}(t,z^*(t),a^*(t)) \tag{3}$$

or equivalently

$$\frac{dp}{dt}(t,z^*(t),a^*(t)) = \nabla_z \mathcal{L}(t,z^*(t),a^*(t)). \tag{4}$$

Proof. See Appendix A. □

The Euler-Lagrange equation is the first-order necessary condition for optimality and holds for every point on the optimal trajectory.

Definition 3. *A function* $f : \mathbb{R}^N \mapsto \mathbb{R}$ *is said to be* α*-homogeneous if* $f(\lambda\,\mathbf{x}) = |\lambda|^\alpha f(\mathbf{x})\ \forall \lambda \in \mathbb{R}$.

Lemma 2. *If the Lagrangian* $\mathcal{L}(t,z,a)$ *is time-independent and* α*-homogeneous in* z *and* a *for* $\alpha > 0$*, i.e.,* $\mathcal{L}(\lambda z,\lambda a) = |\lambda|^\alpha \mathcal{L}(z,a)$ *for all* $\lambda \in \mathbb{R}$*,* S *given by* (2) *reads*

$$S(t_0,z_0,T,z_T) = \frac{1}{\alpha}[z\cdot p]_{t_0}^T + J(z_T). \tag{5}$$

Proof. See Appendix B. □

Lemma 3 (Hamilton-Jacobi)**.** *Along the optimal trajectory, we have for* $t \in (t_0;T)$

$$\begin{aligned}\frac{\partial S}{\partial T_1}(t,z^*(t),T,z_T) &= H(t,z^*(t),-p^*(t)),\\ \frac{\partial S}{\partial T_2}(t_0,z_0,t,z^*(t)) &= -H(t,z^*(t),p^*(t)),\end{aligned} \tag{6}$$

where

$$p^*(t) = \nabla_a \mathcal{L}(t,z^*(t),a^*(t)) = \nabla_{X_1} S(t,z^*(t),T,z_T). \tag{7}$$

Proof. See Appendix C. □

From now, we assume that the Lagrangian is time-independent, i.e., $\mathcal{L}(t,z,a) = \mathcal{L}(z,a)$, and is an even function in a, i.e., $\mathcal{L}(z,-a) = \mathcal{L}(z,a)$. As a consequence H is time-independent and is an even function in p, i.e., we write $H(t,z,p) = H(z,p)$ and $H(z,-p) = H(z,p)$. This assumption is valid provided that the traffic intensity is approximately constant during the UAV flight duration.

3 Optimal Trajectory

In this section, we characterize the optimal trajectory when the traffic intensity is a quadratic form and also when it is made of two regions of quadratic form separated by an interface. We call these two cases *single-phase* and *bi-phase* intensities respectively. Both cases satisfy our assumptions on the Lagrangian with $\alpha = 2$. In practice, the single-phase case may correspond for example to a city quarter with a location of high traffic, e.g. a mall, a railway station, etc. The bi-phase case may correspond to two such locations of high traffic. More generally, the bi-phase is a more accurate model for approximating real measured data compared to the single-phase, as we will see in Section 5.

3.1 Single-Phase Optimal Trajectory

Assume that the traffic intensity is of the form $u(z) = \frac{1}{2}u_0||z||^2$. When $u_0 < 0$, the traffic intensity reaches a maximum at a geographical location that we designate as a *traffic hot spot* [43]. By analogy, when $u_0 > 0$, the traffic intensity reaches a minimum at a geographical location that we designate as a *traffic hole*. We disregard the case $u_0 = 0$ because it corresponds to a constant traffic intensity that is not of interest in this paper. Thus the cost function has the following form

$$\mathcal{L}(z,a) = \frac{1}{2}K||a||^2 - \frac{1}{2}u_0||z||^2. \tag{8}$$

Note that

$$p(z,a) = \nabla_a \mathcal{L}(z,a) = Ka. \tag{9}$$

3.1.1 Trajectory Equation

In this case, a closed form of the trajectory is obtained.

Theorem 1. *If* $u_0 < 0$*, the cost function is given by (10), and the optimal trajectory is*

$$z^*(t) = \frac{z_T \sinh(\omega(t-t_0)) + z_0 \sinh(\omega(T-t))}{\sinh(\omega(T-t_0))} \tag{11}$$

while the control is given by

$$a^*(t) = \omega\frac{z_T \cosh(\omega(t-t_0)) - z_0 \cosh(\omega(T-t))}{\sinh(\omega(T-t_0))} \tag{12}$$

where $\omega^2 = -\frac{u_0}{K}$.
If $u_0 > 0$*, the cost function is given by (13), the optimal trajectory is*

$$z^*(t) = \frac{z_T \sin(\omega(t-t_0)) + z_0 \sin(\omega(T-t))}{\sin(\omega(T-t_0))} \tag{14}$$

and the control is given by

$$a^*(t) = \omega\frac{z_T \cos(\omega(t-t_0)) - z_0 \cos(\omega(T-t))}{\sin(\omega(T-t_0))} \tag{15}$$

where $\omega^2 = \frac{u_0}{K}$.

Proof. See Appendix D. □

Corollary 1. *If the user traffic intensity is of the form* $u(t,z) = \frac{1}{2}u_0||z||^2 + u_0 z\cdot e + u_1$ *with* $u_1 \in \mathbb{R}$ *and* $e \in \mathbb{R}^2$*, then define* $\tilde{z} = z + e$*,* $\tilde{z}_0 = z_0 + e$*,* $\tilde{z}_T = z_T + e$ *and trajectories given in Theorem 1 are valid by replacing* z*,* z_0*,* z_T *by* $\tilde{z}$*,* $\tilde{z}_0$*,* $\tilde{z}_T$*, respectively. The cost function becomes:* $S(t_0,z_0,T,z_T) = \frac{1}{\alpha}[z\cdot p]_{t_0}^T + J(z_T) - u_1(T-t_0)$.

$$S(t_0, z_0, T, z_T) = \frac{K\omega}{2\sinh\omega(T-t_0)}\left((|z_0|^2+|z_T|^2)\cosh\omega(T-t_0) - 2z_0\cdot z_T\right) + J(z_T) \tag{10}$$

$$S(t_0, z_0, T, z_T) = \frac{K\omega}{2\sin\omega(T-t_0)}\left((|z_0|^2+|z_T|^2)\cos\omega(T-t_0) - 2z_0\cdot z_T\right) + J(z_T) \tag{13}$$

Corollary 2. *If the user traffic intensity is of the form $u(t,z) = \sum_i u_i||z-z_i||^2$ with $\sum_i u_i \neq 0$, then $u(t,z) = (\sum_i u_i)||z-z_b||^2 + \sum_i u_i||z_i-z_b||^2$ with $z_b = \frac{\sum_i u_i z_i}{\sum_i u_i}$. Define $\tilde{z} = z + z_b$, $\tilde{z}_0 = z_0 + z_b$, $\tilde{z}_T = z_T + z_b$, $\tilde{u}_0 = \sum_i u_i$ and trajectories given in Theorem 1 are valid by replacing z, z_0, z_T, u_0 by $\tilde{z}$, $\tilde{z}_0$, $\tilde{z}_T$, $\tilde{u}_0$ respectively.*

The system is thus equivalent to the one assumed in Theorem 1 by changing the origin of the locations to the barycentre z_b of the z_i.

From (10), we can easily show that the cost function S is a decreasing function of u_0.

3.1.2 Traffic Hot Spot, Traffic Hole

We assume that there is a hot spot or a traffic hole located in z_h and that the traffic intensity is of the form $u(t,z) = \frac{1}{2}u_0||z-z_h||^2 + u_1 = \frac{1}{2}u_0||z||^2 - u_0 z\cdot z_h + \frac{1}{2}u_0||z_h||^2 + u_1$. We can apply Corollary 1 with $e = -z_h$.

Figure 2a shows optimal trajectories when z_h is a hot spot, i.e., for $u_0 < 0$, and different values of K. The starting point is z_0 and the destination is z_T. Dots are shown at regular time intervals: close-by dots indicate a low speed, while distant dots are due to a higher speed. When K increases, the velocity cost increases and the trajectories tend to the straight line between z_0 and z_T, which minimizes the speed. As the energy cost is high, the UAV almost ignores the traffic intensity to minimize the total cost. When K is small, the UAV can go fast to z_h, reduces its speed in the vicinity of the hot spot to serve more traffic and then goes fast to the destination (in order to decrease the cost function (1) by increasing its traffic contribution). In several scenarios, the UAV shall reach a target and hover above it for some time: this is captured in our model by giving the UAV a sufficient time budget T. We then obtain a trajectory similar to the one of Figure 2a when K is small. If the UAV is constrained to come back to its initial position, setting $z_0 = z_T$ makes the UAV going towards z_h and coming back using a straight line.

Figures 2b and c show optimal trajectories when z_h is a traffic hole, i.e., for $u_0 > 0$. In Figure 2b, T is smaller than the period of the ellipse, i.e., $\frac{2\pi}{\omega} > T$. When K decreases, the UAV can spend more time in the areas of higher traffic intensity. In Figure 2c, T is larger than the period. In this case, the trajectory follows one period of the ellipse whose equation is given by (14) plus a part of the same ellipse from z_0 to z_T.

3.2 Bi-Phase Trajectory Characterization

We now consider a traffic intensity (or potential) consisting in two quadratic functions separated by an interface $\mathcal{I}$ of equal potentials delimiting two regions 1 and 2. The interface is defined by an equation $f(z) = C$, where C is a constant and f is a differentiable function. We assume that the optimal trajectory crosses the interface only once, at position ξ and time τ[3].

The impulsion p^* is defined everywhere on the optimal trajectory between (t_0, z_0) and (T, z_T). The following notations will be used in the sequel: $p^- = p^*(\tau^-) = \lim_{s\to\tau s<\tau} p^*(s)$, $p^+ = p^*(\tau^+) = \lim_{s\to\tau s>\tau} p^*(s)$, $H^- = H_1(\xi, p^*(\tau^-))$, $H^+ = H_2(\xi, p^*(\tau^+))$.

Theorem 2. *The position and time (ξ, τ) at interface crossing are such that*

$$p^- - p^+ - \mu\,\nabla_z f(\xi) = 0 \tag{16}$$
$$H^+ - H^- = 0 \tag{17}$$
$$f(\xi) = C \tag{18}$$

for some Lagrange multiplier $\mu \in \mathbb{R}$

Proof. See Appendix E. □

Equation (17) expresses the fact the energy is conserved when crossing the interface. One can show that actually the energy is conserved along the whole trajectory. Equation (16) is related to the conservation of the tangential component of the impulsion at the interface. Equation (18) is the interface equation at ξ. One can show that under the assumption of equal potential on the interface, the kinetic energy, the impulsion, and the velocity vector are conserved across the interface.

3.3 Bi-phase optimal trajectory uniqueness and value function convexity issues

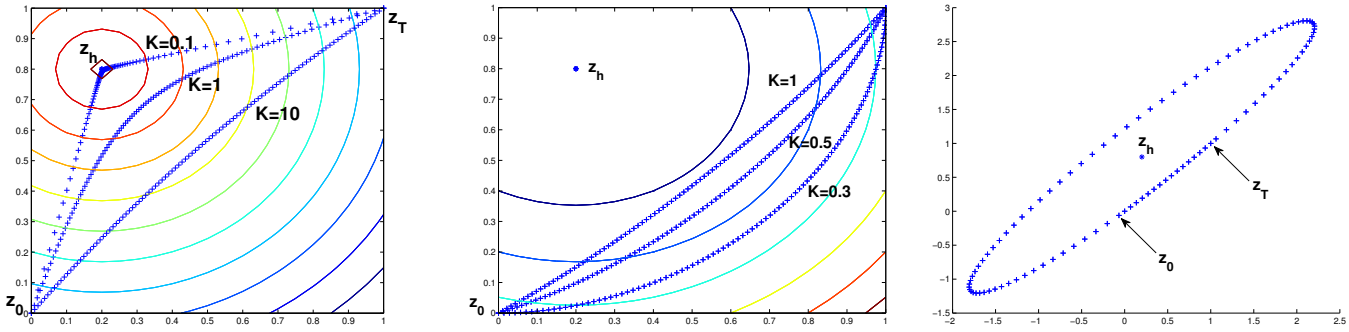
Looking for uniqueness/non-uniqueness of optimal bi-phase trajectories leads naturally to study the convexity of the bi-phase total cost. As stated in Appendix E this cost is additive and satisfies the dynamic programming principle which reads

$$\bar{S}(\Theta = \tau,\, \Xi = \xi \mid t_0, z_0, T, z_T) = S_1(t_0, z_0, \tau, \xi) + S_2(\tau, \xi, T, z_T) \tag{19}$$

where S_1 and S_2 are themselves minimal.

Hamilton-Jacobi equations applied to each cost component allows to study their first- an second- order differential properties such as convexity. For instance in the chosen quadratic model, each single-phase (minimal) total cost $S_1(t_0, z_0, \tau, \xi)$, $S_2(\tau, \xi, T, z_T)$ given by Corollary 1 and (10) is convex (since it is a positive quadratic form with respect to

3. The possibility of crossing several times the interface or having more than two phases leads to combinatorial complications that are left for further work. This assumption prevents the UAV to come back to its initial position.



(a) Traffic hot spot. (b) T smaller than the ellipse period. (c) T larger than the ellipse period.

Fig. 2: (a) Traffic hot spot ($u_0 < 0$) and (b)(c) Traffic hole ($u_0 > 0$). Circles are iso-traffic levels.

the spatial coordinates) with respect to the spatial position ξ, but not necessarily so with respect to the interface crossing time τ. We consider in what follows a general form of the single-phase value function between (t_1, x_1) and (t_2, x_2) that is denoted by $S(t_1, x_1, t_2, x_2)$. We also note by ω the pulsation, $\phi = \omega\,(t_2 - t_1)$ is the *temporal* phase and $p_1,\ p_2$ are the the initial and final impulsions as derived from formula (12).

Theorem 3. *i) The Hessian of the single-phase cost wrt. joint variables $\psi_i = (T_i, X_i)_{i=1,2}$ is*

$$\mathcal{H}(\psi_i) = \nabla^2_{T_i,X_i}\, S(t_1, x_1, t_2, x_2) = \left(\begin{array}{c|c} \alpha & {\Pi_i}^\dagger \\ \hline \Pi_i & K g\, Id_2 \end{array}\right) \quad (20)$$

with

$$\left\{\begin{array}{l} \alpha = \dfrac{\partial^2 S}{\partial {T_i}^2} = \dfrac{\omega\, p_1 \cdot p_2}{K\ \sinh\phi} \\ g = \omega\ \coth\phi\ > 0 \\ \Pi_i = -\dfrac{\omega\, p_{3-i}}{\sinh\phi} \in \mathbb{R}^2 \end{array}\right.$$

where p_i is the impulsion at time T_i, i.e., at extremity X_i ($i = 1, 2$ and $j = 3 - i$).
ii) At most one eigenvalue of $\mathcal{H}(\psi_i)$ can be negative and $\alpha < 0$ is a sufficient condition for this to hold, namely implying local non-convexity of the (single-phase) value function.
iii) The two-phase Hessian has a similar structure as (20) *and writes wrt. variable $\Psi = (\Theta, \Xi)$*

$$\bar{\mathcal{H}}(\Theta = \tau, \Xi = \xi) = \mathcal{H}_1(\chi_2) + \mathcal{H}_2(\chi_1) = \left(\begin{array}{c|c} \alpha & \Pi^\dagger \\ \hline \Pi & K h\, Id_2 \end{array}\right) \quad (21)$$

with

$$\left\{\begin{array}{l} \alpha = \dfrac{\partial^2 S_1}{\partial {T_2}^2}(t_0, z_0, \tau, \xi) + \dfrac{\partial^2 S_2}{\partial {T_1}^2}(\tau, \xi, T, z_T) \\ h = \omega_1\ \coth\phi_1 + \omega_2\ \coth\phi_2 \\ \Pi = -\left[\dfrac{\omega_1\, p(t_0)}{\sinh\phi_1} + \dfrac{\omega_2\, p(T)}{\sinh\phi_2}\right] \end{array}\right.$$

where $\phi_1 = \omega_1\ (\tau - t_0$, $\phi_2 = \omega_2\ (T - \tau)$. It may thus be non-convex as well.

Proof. See Appendix F subsections F.1 and F.2. □

In the single-phase case, the sufficient condition *ii)* i.e., $\alpha < 0$, supports first a *physical* interpretation (see Fig. 3b) and also a *geometrical* one (see next Theorem 4 and Fig. 3c).

Theorem 4. *Let $u = \frac{x_1+x_2}{2}$ and $v = \frac{x_2-x_1}{2}$ ($\|u\|$ is the distance from the hotspot to the center of x_1 and x_2, and $\|v\|$ is the half-length of $[x_1 x_2]$.) The next condition is sufficient for* ii) *to hold:*

$$\frac{\|\ v\ \|}{\|\ u\ \|} < \frac{\cosh\phi - 1}{1 + \cosh\phi} = \tanh^2\frac{\phi}{2} \Rightarrow \alpha = \frac{\partial^2 S}{\partial {t_i}^2} < 0.$$

Proof. We combine the formulas (40) and (41) in (42) to obtain

$$\begin{array}{rcl} \alpha & = & \dfrac{w}{K}\ \dfrac{p_1 \cdot p_2}{\sinh\phi} \\ & = & \dfrac{K\,\omega^3}{\sinh^3\phi}[(\|\ x_1\ \|^2 + \|\ x_2\ \|^2)\ \cosh\phi \\ & & -x_1\ .\ x_2\ (1 + \cosh^2\phi)] \end{array} \quad (22)$$

which can be rewritten as

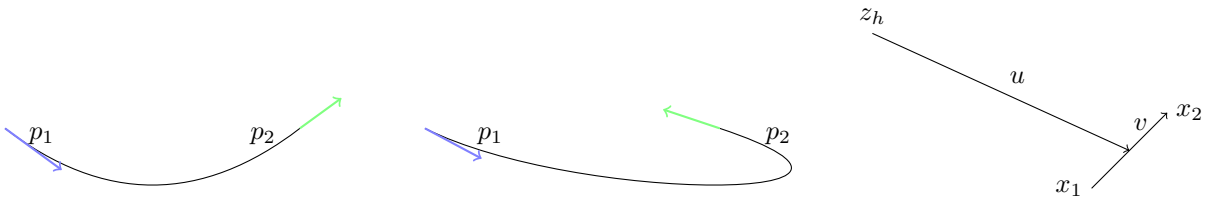
$$\begin{array}{rcl} \alpha & = & \dfrac{K\,\omega^3}{\sinh^3\phi}[-\ \|\ u\ \|^2\ \ (1 - \cosh\phi)^2 + \\ & & \|\ v\ \|^2\ \ (1 + \cosh\phi)^2)]. \end{array}$$

□

This non-convexity condition for the single-phase value function interprets as: long phase ($\tanh\frac{\phi}{2} > \frac{1}{2}$) and long distances between the hotspot z_h to both initial and final positions x_1, x_2, relatively to their mutual distance ($\frac{\|u\|}{\|v\|} \ggg 1$). First simulations show that when this case happens for both phases, the total cost is indeed non-convex and several local optimal solutions may exist.

4 Algorithms

Previous propositions allow us to propose several numerical algorithms seeking optimal trajectories. In this section, we present two algorithms aiming this goal: a gradient descent algorithm and a bisection search method based on the linear control property (12).



(a) $\alpha = \dfrac{\partial^2 S}{\partial {t_i}^2} > 0$ (b) $\alpha = \dfrac{\partial^2 S}{\partial {t_i}^2} < 0$ (c) $u = \dfrac{x_1 + x_2}{2}$, $v = \dfrac{x_2 - x_1}{2}$

Fig. 3: Non-convexity vs. convexity of the single-phase cost at $(t_i, x_i), i = 1, 2$: (a) possibly convex. (b) the initial and final impulsions (p_1, p_2) form an *obtuse* angle $\Rightarrow$ the total cost is *non-convex*. (c) same property if long time and long distance of both x_1, x_2 to the hotspot z_h.

4.1 A Gradient Descent Algorithm GRAD-ALGO

In this section, we propose GRAD-ALGO (Algorithm 1) which is an alternated optimization-based algorithm relying on the procedures described in this section. The algorithm proceeds by iterations using two nested loops (the repeat-until-loop in steps 9-16 and the for-loop in steps 11-14). The inner loop assumes a fixed ξ and optimizes τ using a Newton descent (step 12). The descent is based on the calculation of the Hessian and the Hamiltonians (step 13 and procedure in steps 4-7). Moreover, a certain point B is computed (step 13 and procedure in steps 4-7). The outer loop assumes a fixed τ and optimizes ξ by projecting the point B on the interface (step 10). It stops when certain stopping criteria are met (step 16). We now detail the two optimization procedures and the stopping criteria.

4.1.1 Procedure for seeking an optimal ξ given a fixed τ

From Hamilton-Jacobi equations (Lemma 3 and Appendix C) the gradient of the total cost function with respect to ξ is $p^*(\tau^-) - p^*(\tau+)$. Equation (9) says that in the Newtonian framework the impulsion is proportional to the control variable a. Equation (12) says that in the quadratic model the velocity vector is, at any time a linear combination of *centered* initial and final positions. Then the searched gradient appears to be an *affine* function of ξ which reads

$$\begin{aligned} p^*(\tau^-) - p^*(\tau+) &= \nabla_{X_2} S_1(t_0, z_0, \tau, \xi) + \nabla_{X_1} S_2(\tau, \xi, T, z_T) \\ &= Kh\,(\xi - B) \end{aligned} \quad (23)$$

Scalar h and point B (where the spatial gradient cancels at fixed τ i.e., $p^*(\tau^-) = p^*(\tau+)$) verify

$$h = \omega_1 \coth(\omega_1(\tau - t_0)) + \omega_2 \coth(\omega_2(T - \tau)) \quad (24)$$

$$\begin{aligned} B = \frac{1}{h}\big[&\omega_1 z_{h1} \coth(\omega_1(\tau - t_0)) + \omega_2 z_{h2} \coth(\omega_2(T - \tau)) \\ &+ \frac{\omega_1(z_0 - z_{h1})}{\sinh(\omega_1(\tau - t_0))} + \frac{\omega_2(z_T - z_{h2})}{\sinh(\omega_2(T - \tau))}\big] \end{aligned} \quad (25)$$

The equation involving the Lagrange multiplier (16) now reads

$$Kh\,(\xi - B) - \mu\,\nabla_\xi f(\xi) = 0 \quad (26)$$

and shows that the optimal location ξ^* is the *orthogonal projection* of B on the interface $\mathcal{I}$ (step 10 of Algorithm 1).

4.1.2 Procedure for seeking an optimal τ given a fixed ξ

We use the result of Theorem 2. As also shown by Hamilton-Jacobi equations (Lemma 3 and Appendix C), the gradient of S with respect to τ is given by $H_2(\xi, p^*(\tau^+)) - H_1(\xi, p^*(\tau^-))$. The Hamiltonians are easily computed in each phase by applying the classical Newton formula at given location (z, t) and read $H(p, z, t) = \dfrac{\| p \|^2}{2\,K} + V(z)$ (procedure in steps 4-7 of Algorithm 1). We then update τ by using a simple gradient descent scheme (see step 12 of Algorithm 1).

4.1.3 Stopping criteria

The search procedure is performed until the impulsion and the Hamiltonian have converged with a given accuracy (step 16 of Algorithm 1). For this purpose, we consider the function g:
$(\mathbb{R}^n \setminus \{0\})^2 \to \mathbb{R}$ defined as $g(x, y) = \dfrac{\| x - y \|}{\inf(\| x \|, \| y \|)}$ which measures a relative "discrepancy" between x and y. Now let us give ϵ_p and ϵ_H and define the stop criteria as follows

$$\texttt{STOP} = g(p^+, p^-) \,/\, \epsilon_p < 1 \;\;\&\;\; g(H^+, H^-) \,/\, \epsilon_H < 1. \quad (27)$$

Complexity: Algorithm 1: if $\alpha^* = \dfrac{\partial^2 \bar{S}}{\partial \Theta^2}(\tau^*, \xi*) \neq 0$ then the convergence of Newton descent on τ is quadratic (see [44] for instance). If $\alpha^* = 0$ say, $H_2(\xi, p^*(\tau^+)) - H_1(\xi, p^*(\tau^-)) \sim (\tau - \tau^*)^\nu$ with $\nu \geq 2$, the convergence is linear (see [44] for instance). We take $M_\tau = 10$ (in accordance with $\epsilon_p = \epsilon_H = 2.10^{-4}$) i.e., $\nu \approx 2$. The global complexity is $O(N_{step}\, M_\tau)$ and usually $N_{step} = 2$.

4.2 The B-curve algorithm B-ALGO

The B-curve algorithm aims to overcome the non-convexity issues developed previously. It proceeds as follows. Consider the B point defined in formula (25). We notice that for each $\tau \in [t_0, T]$ this point is defined univocally knowing all parameters (t_0, T, z_0, z_T, z_h) and all (quadratic) traffic profiles, so that we can see B in (25) as a function of τ, denoted $B(\tau)$. It can be shown that this function $\tau \mapsto B(\tau)$ is continuous in the interval $[t_0, T]$ and that $\lim_{\tau \to t_0^+} B(\tau) = z_0$, $\lim_{\tau \to T^-} B(\tau) = z_T$.

Assume now that the two optimal sub-trajectories are such that crossing time $\tau \in [t_0, T]$ and interface position ξ verify $\xi = B(\tau)$. Then by formula (23) the spatial gradient of S at point ξ is $p^*(\tau^-) - p^*(\tau+) = 0$. This implies that the *kinetic* components of both Hamiltonians are equal at the interface.

Algorithm 1 GRAD-ALGO An uncoupled projected gradient descent algorithm

1: **Input:** precisions ϵ_p, ϵ_H M_τ = number of gradient descent iterations on τ
2: **Init:** starting position $\xi \leftarrow \xi_0$ e.g., = $\frac{z_0+z}{2}$, $\tau \leftarrow \tau_0$ e.g., = $\frac{t_0+T}{2}$, $N_{step} = 0$
3: **Output:** (τ, ξ)
4: **procedure** TWO_PHASE(τ, ξ)
5: **computes:** the current bi-phase trajectory given by $(t_0, z_0) \rightarrow (\tau, \xi) \rightarrow (T, z_T)$
6: **return** $(B, H^+, H^-, \bar{\mathcal{H}})$ = (the B-point (25), the two phase Hamiltonians and the total Hessian at current location (τ, ξ) (19, 21)
7: **end procedure**
8: $(B, H^+, H^-, \bar{\mathcal{H}}) =$ TWO_PHASE(τ, ξ)
9: **repeat**
10: $\xi \leftarrow \text{Proj}_{\mathcal{I}}(B)$ (26)
11: **for** $m = 1, \ldots, M_\tau$ **do**
12: $\tau \leftarrow \tau - \dfrac{(H^+ - H^-)}{\alpha}$ with $\alpha = \bar{\mathcal{H}}_{11} = \dfrac{\partial^2 \bar{S}}{\partial \Theta^2}(\tau, \xi)$ (Newton descent wrt. τ)
13: $(B, H^+, H^-, \bar{\mathcal{H}}) =$ TWO_PHASE(τ, ξ)
14: **end for**
15: $N_{step} = N_{step} + 1$
16: **until** STOP (27)

Now since ξ belongs to the interface, both phase *potentials* (traffics) are equal by definition, so that the *total* Hamiltonian is also conserved at the interface: This is the optimality condition required with respect to time τ.

To summarize, in these conditions, both the Hamiltonian and the total impulsion are conserved at interface $\mathcal{I}$, i.e., local optimality conditions hold for the total value function. It is also worth noticing that the related Lagrange multiplier appearing in (26) now simply vanishes: $\mu = 0$. The proposed B-algorithm consists then in seeking the intersection of the B-curve $\mathcal{B} = \{B(\tau), \tau \in [t_0, T]\}$ with the interface $\mathcal{I}$ (Algorithm 2). For this, we first select precision ϵ_B, then proceed by bisection and check the stopping criterion

$$\text{STOP\_B} = \frac{|t_2 - t_1|}{|b - a|} \frac{1}{\epsilon_B} < 1. \quad (28)$$

To go in more details, the algorithm starts by initializing two time instants t_1 and t_2 such that $B(t_1)$ and $B(t_2)$ are on either side of the interface (step 3). Then it proceeds by iterations to perform the bisection procedure (steps 8-15). In step 9, we set the new time instant τ to the middle of the interval $[t_1; t_2]$ in order to perform one iteration of the binary search. The corresponding location ξ on the B-curve is also found. If ξ is on the same side of the interface as $B(t_2)$ (step 10), then t_2 is moved to τ (step 11), otherwise t_1 is moved to τ (step 12). A loop invariant is that $B(t_1)$ and $B(t_2)$ are always on either side of the interface. Because of the binary search, the length of the interval $[t_1; t_2]$ is divided by 2 at every iteration. The loop is stopped when the length of the interval is sufficiently small (step 15).

Complexity: Algorithm 2: if $\epsilon_B = \dfrac{1}{2^m}$ then the bisection algorithm converges in m iterations. Its complexity is thus $O(\log \dfrac{1}{\epsilon_B})$. We chose $\epsilon_B = 2.10^{-4} \approx \dfrac{1}{2^{12}}$ and 12 iterations are indeed sufficient to provide a trajectory with optimality conditions holding at this (relative) precision.

Algorithm 2 the B-curve bisection algorithm

1: **Input:** initial/final time search interval $[a, b] \in [t_0, T]$, precision ϵ_B fixed by the user
2: **Output:** (ξ, τ)
3: **Init:** $t_1 \leftarrow a$, $t_2 \leftarrow b$, the algorithm only starts if IN_ZONE2($B(a)$) $\neq$ IN_ZONE2($B(b)$)
4: **procedure** IN_ZONE2(z)
5: $u_i(z)$ = (time-stationary) traffic $u(z)$ from hotspot z_{hi} at point z $(i = 1, 2)$
6: **return** $(u_2(z) > u_1(z))$
7: **end procedure**
8: **repeat**
9: $x_1 \leftarrow B(t_1)$, $x_2 \leftarrow B(t_2)$, $\tau \leftarrow \dfrac{t_1 + t_2}{2}$, $\xi \leftarrow B(\tau)$
10: **if** (IN_ZONE2(ξ) == IN_ZONE2(x_2)) **then**
11: $t_2 \leftarrow \tau$
12: **else** $t_1 \leftarrow \tau$
13: **end if**
14: **compute stop criterion** STOP_B (28)
15: **until** STOP_B
16: $\xi \leftarrow B(\tau)$

Algorithm 3 Data preprocessing

1: First smoothing: data aggregation
2: Second smoothing: LOWESS
3: Normalization
4: K-means with quadratic models

5 NUMERICAL EXPERIMENTS

5.1 From Measurements to Quadratic Profile

In this section, we explain how from measured or estimated traffic load, we can derive a quadratic model that will allow us to apply our framework.

To illustrate the procedure, we extract data from the open data set presented in [40]. Traffic data (in number of bytes) has been collected from an operational cellular network in a medium-size city in China and is recorded for every base station and every hour. For our experiment, we extract a rectangle region $[X_{\min}, X_{\max}] \times [Y_{\min}, Y_{\max}]$, where $X_{\min} = 111$, $X_{\max} = 111.12$, $Y_{\min} = 13.12$, $Y_{\max} = 13.22$ are the minimum and maximum longitude and latitude respectively (real figures have been anonymized). This corresponds approximately to a rectangle of 11 km$\times$13 km with 400 base stations having an average cell range of 337 m. The traffic of the 22th August 2012 between 5 and 6pm is illustrated in Figure 4a. We assume that this traffic is representative of the traffic intensity when the drone is launched.

In order to fit this raw data to our model, we follow the pre-processing steps shown in Algorithm 3. The first smoothing consists in aggregating the traffic data on a grid of 50 steps in both longitude and latitude directions.

The resulting elementary regions should correspond approximately to the drone coverage. The result is shown in Figure 4b. The data exhibits a very high variability with very high peaks around few locations.

The second smoothing is a Locally Weighted Scatterplot Smoothing (LOWESS) [45]. We use here the Matlab function `fit` with the option "Lowess". The choice of the smoothing parameter α, i.e., the proportion of data points used for every local regression, has a decisive impact on the result. Increasing α has the effect of averaging out the different peaks. In our specific scenario, $\alpha = 0.25$ yields Figure 5a with two local maxima. With $\alpha = 0.5$, we obtain a single maximum. In step 3 of the pre-processing, the traffic is normalized between 0 and 1 with no influence on the optimal trajectory.

The final pre-processing step is an adaptation of the classical K-means algorithm (see Algorithm 4) to fit to quadratic models. Inputs are the data points obtained after the normalization, K_c, the number of clusters (or hot spots), K_n the number of nearest neighbors and M the number of iterations. Every cluster is associated to a quadratic function (in our case, we have $K_c = 2$). Every data point j is associated to a cluster and has a related label L_j in $\{1, ..., K\}$. For every data point, a list of nearest neighbors is built (step 4). An arbitrary initial labelization is chosen (step 6). The algorithm then proceeds by iterations (steps 7-16). At every iteration, if a point j has some neighbor with a different label (step 9), a best new label is found for j (step 10-13) in terms of quadratic error e_k. The error e_k measures the difference between the data points and the K_c-quadratic model, which fits a quadratic function to every cluster assuming that j has label k (steps 18-31). In Algorithm 4, $\texttt{KNN}(K_n, X, Y)$ is a procedure that finds the K_n nearest neighbors of (X, Y) with respect to the Euclidian distance. We use the Matlab implementation `knnsearch`. $\texttt{NLLS}(\mathcal{L})$ is a non-linear least square method that fits data points in $\mathcal{L}$ to a quadratic function of the form $\frac{1}{2}u_0|z - z_h|^2 + u_1$. The Matlab implementation is based on a trust-region approach of the Levenberg-Marquardt Algorithm.

Complexity: Algorithm 4: Searching the K_n-nearest neighbors of a data point using k-d trees takes $O(K_n \log J)$ in average and $O(K_n J)$ in the worst case. Steps 3-5 has thus a complexity of $O(K_n J \log J)$ in average and $O(K_n J^2)$ in the worst case. Initial labelization is a simple linear partitioning of the 2D space in K_c zones and is thus performed in $O(J)$. In the main loop, there are at most $O(MJK_c)$ calls to the function `FIT`. The Levenberg-Marquardt Algorithm requires $O(\epsilon^{-2})$ iterations to reach an ϵ-approximation of a stationary point of the objective function [44], [46]. The overall complexity of Algorithm 4 is thus $O(K_n J^2 + MJK\epsilon^{-2})$.

The K-means partition obtained after $M = 12$ iterations is shown in Figure 5b. The final fits are shown in Figures 7a and 7b for $(K_c, \alpha) = (2, 0.25)$ and $(K_c, \alpha) = (1, 0.5)$, respectively. Figure 6 shows the quadratic error as a function of the number of iterations of the K-means algorithm for $K_c = 1$ and $K_c = 2$. The error is constant for $K_c = 1$ as there is only one iteration, which performs the non-linear least square fitting for the single cluster. For $K_c = 2$, we distinguish two cases: $K_n = 5$ and $K_n = \infty$. In the former case, only the 5 nearest neighbors of a data point are inspected to decide if a relabelization should be performed.

Algorithm 4 K-means with quadratic models

1: **Input:** K_c (number of clusters), J (number of measurement points) (X_j, Y_j, Z_j), $j = 1, \ldots, J$ (coordinates and estimated traffic load Z_j in (X_j, Y_j)), K_n (number of nearest neighbors), M (number of iterations)
2: **Output:** z_{hk}, u_{0k}, u_{1k} $k = 1, \ldots, K_c$ (hot spot characteristics), e (quadratic error)
3: **for** $j = 1, \ldots, J$ **do**
4: $\quad \mathcal{K}_j \leftarrow \texttt{KNN}(K_n, X_j, Y_j)$
5: **end for**
6: $L \leftarrow$ Initial labelization
7: **for** $m = 1, \ldots, M$ **do**
8: $\quad$ **for** $j = 1, \ldots, J$ **do**
9: $\quad\quad$ **if** $\exists j' \in \mathcal{K}_j$ s.t. $L_{j'} \neq L_j$ **then**
10: $\quad\quad\quad$ **for** $k = 1...K_c$ **do**
11: $\quad\quad\quad\quad e_k, z_{hl}, u_{0l}, u_{1l}, \quad l = 1, \ldots, K_c \leftarrow \texttt{FIT}((X, Y, Z), j, L, k, K_c)$
12: $\quad\quad\quad$ **end for**
13: $\quad\quad\quad L_j \leftarrow \arg\min_k e_k$
14: $\quad\quad$ **end if**
15: $\quad$ **end for**
16: **end for**
17: **return** z_{hk}, u_{0k}, u_{1k} $k = 1, \ldots, K_c$, global quadratic error
18: **procedure** $\texttt{FIT}((X, Y, Z), j, L, k, K_c)$
19: $\quad L_j \leftarrow k$
20: $\quad$ **for** $l = 1,...,K_c$ **do**
21: $\quad\quad \mathcal{L}_l \leftarrow \{(X_i, Y_i, Z_i) | L_i = l\}$
22: $\quad\quad z_{hl}, u_{0l}, u_{1l} \leftarrow \texttt{NLLS}(\mathcal{L}_l)$
23: $\quad$ **end for**
24: $\quad e_k \leftarrow 0$
25: $\quad$ **for** $j = 1,...,J$ **do**
26: $\quad\quad \tilde{Z}_j \leftarrow \max_{l=1,\ldots,K_c} \frac{1}{2}u_{0l}|z - z_{hl}|^2 + u_{1l}$
27: $\quad\quad e_k \leftarrow e_k + |\tilde{Z}_j - Z_j|^2$
28: $\quad$ **end for**
29: $\quad e_k \leftarrow e_k / J$
30: $\quad$ **return** $e_k, z_{hl}, u_{0l}, u_{1l}, l = 1, \ldots, K_c$
31: **end procedure**

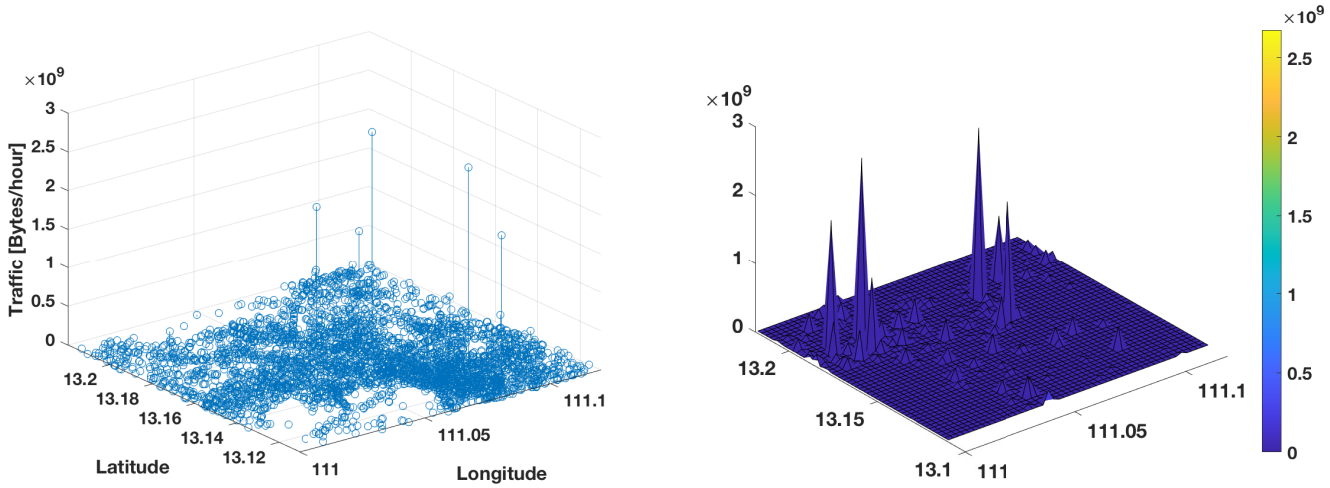
In the later case, relabelization is systematically considered. Increasing K_n increases the complexity of every iteration but provides a faster convergence. Choosing a single-phase model provides a simpler model though less accurate.

In our model, traffic intensity is supposed to be known a priori. If traffic intensity parameters, u_0, u_1, z_h for every phase are known to belong to some uncertainty intervals, a robust optimization can be performed by sampling the cartesian product of uncertainty sets and computing the maximum cost using our low complexity algorithms.

5.2 Trajectory optimization results

5.2.1 Estimation procedure of parameters K and T

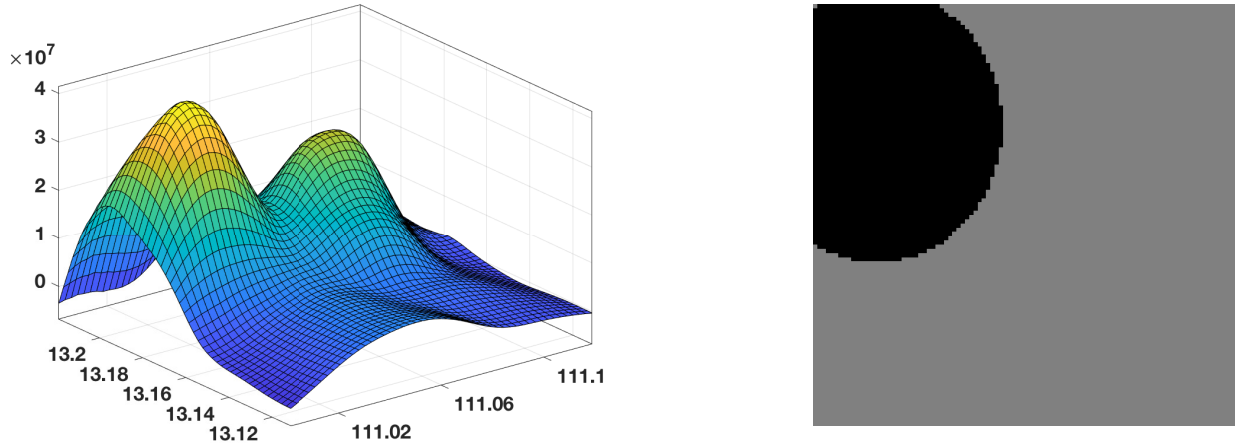
As in most similar algorithms, a 'prior' estimation of parameters is necessary since the optimal results strongly depend on them. Traffic parameters have been estimated just above, so the required parameters are first the mass K representing trade-off between the kinetic term and the traffic term in Lagrangian (1) and then the total duration time T which has also a significant influence on the



(a) Raw traffic data on the 22th Aug. 2012 6pm [40].

(b) First smoothing: data aggregation at drone coverage level.

Fig. 4: Data preprocessing: raw data and first smoothing.



(a) Second smoothing: LOWESS with $\alpha = 0.25$.

(b) K-means partition.

Fig. 5: Data preprocessing: second smoothing and partitions in 2 phases.

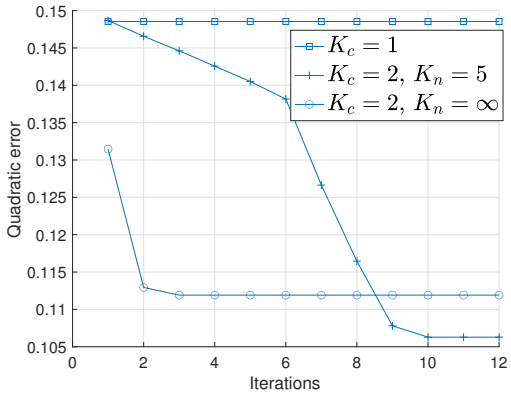
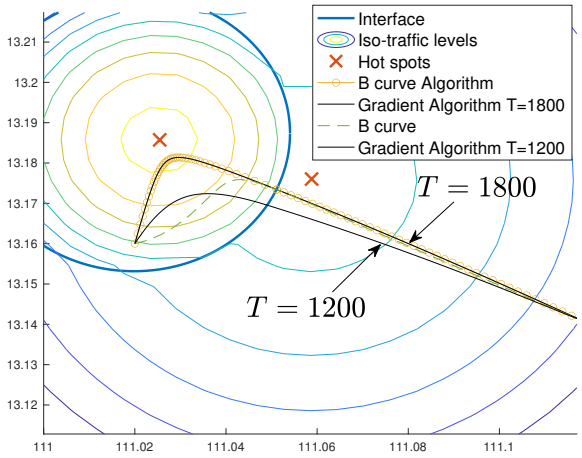
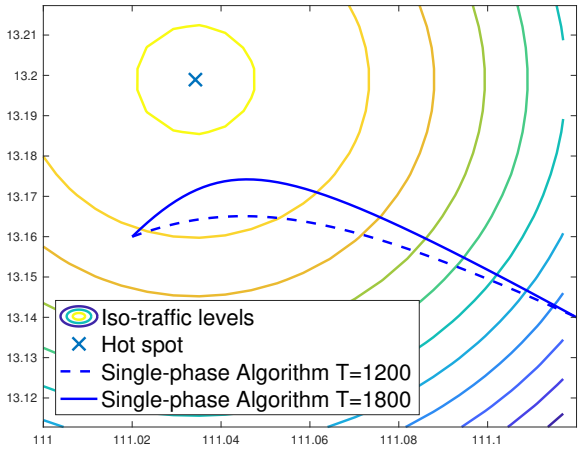


Fig. 6: Quadratic error between normalized smoothed data and quadratic models.

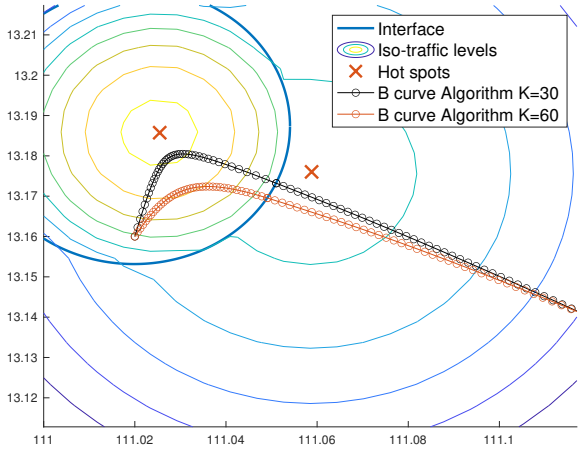


(a) $K = 60$, $T = 1200$ and 1800, 2 phases.

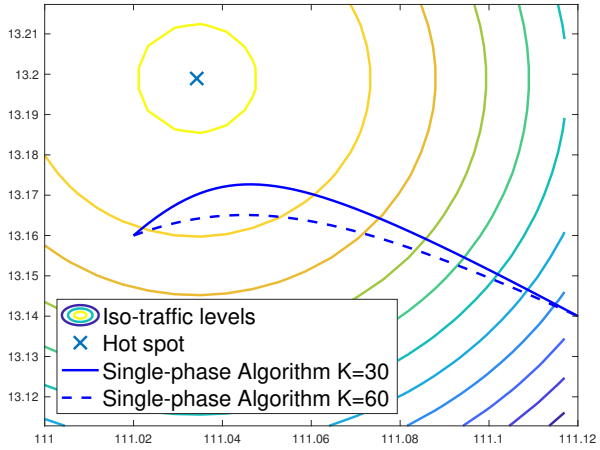


(b) $K = 60$, $T = 1200$ s and 1800 s, 1 phase.

Fig. 7: Influence of T on the optimal trajectory.



(a) $K = 30$ and 60, $T = 1200$, 2 phases.



(b) $K = 30$ and 60, $T = 1200$ s, 1 phase.

Fig. 8: Influence of K on the optimal trajectory.

optimal trajectory (by convention $t_0 = 0$). This procedure is developed in Appendix G and yields in our case: $T \approx 850$, $K = 50 \times r^2$ with $r = \frac{40\times10^6}{360\times100}$ and we select: $T = 1200, 1800s$ and $K = 30, 60 \times r^2$.

5.2.2 Results

Optimal trajectories are shown in Figs. 7 and 8 and related numerical results in Table 1. Length of various trajectories and related average velocities are given in Table 2. The following main comments can be made on these results:

- First the obtained trajectories for GRAD-ALGO and B-ALGO satisfy the optimal conditions, as "relative" discrepancies between impulsions and between Hamiltonians at the interface are indeed below selected precision of 2.10^{-4}, yielding a "convenient" full trajectory.
- Then GRAD-ALGO is stable due to our choice of initial and terminal drone positions (not too close to the interface and not too far from their related hotspots) and of time intervals (not too large *temporal* phases). Thus, we are far from the non-convexity conditions expressed in Theorems 3 and 4. The total hessian is checked to be positive at each iteration. Hence, a large (spatio-temporal) convergence basin resulting into exact convergence of the GRAD-ALGO (as well as the B-ALGO) towards the *unique* optimal solution.
- As already noted for the single-phase case (paragraph 3.1.2), decreasing mass K enables the drone to collect more traffic. In fact, during the allowed time interval, the drone will get closer to the hotspot with *maximal* traffic (zone 1) in order to decrease the total cost. Increasing time interval ($T \nearrow$) will produce the same tendancy, allowing the drone to spend more time near

ξ	$\mathbf{S}$	H^-	ϕ^-	ϕ^+	τ
a) $K = 60 \times r^2$, $T = 1200$ $\rightarrow S_{1ph} = -221.4103$					
B-ALGO $(Niter = 12)$					
$(111.048, 13.170)$	-168.5713	0.8131	1.9803	0.9236	538.69
GRAD-ALGO $(Niter = 5)$					
$(111.048, 13.170)$	-168.5713	0.8131	1.9802	0.9237	538.68
b) $K = 60 \times r^2$, $T = 1800$ $\rightarrow S_{1ph} = -637.9077$					
B-ALGO $(Niter = 12)$					
$(111.0511, 13.1734)$	-641.6388	0.8332	4.1764	0.9273	1136.09
GRAD-ALGO $(Niter = 2)$					
$(111.0511, 13.1734)$	-641.6388	0.8332	4.1763	0.9273	1136.08
c) $K = 30 \times r^2$, $T = 1200$ $\rightarrow S_{1ph} = -399.1474$					
B-ALGO $(Niter = 12)$					
$(111.051, 13.173)$	-393.1217	0.8313	3.7936	0.9289	729.71
GRAD-ALGO $(Niter = 2)$					
$(11.051, 13.173)$	-393.1217	0.8313	3.7933	0.9290	729.65

TABLE 1: Table of results with B-ALGO and GRAD-ALGO for $K = 30, 60 \times r^2$, $T = 1200, 1800s$ and $r = \frac{40\times10^6}{360\times100} \approx 1111.1111$ (corresponding to scaled spatial unity of 100m). Also, S_{1ph} denotes the total single-phase cost (i.e., assuming only one, effective hotspot).

case	bi-phase		single-phase	
	L (km)	$\bar{V}$(m/s)	L (km)	$\bar{V}$(m/s)
a)	12.41	10.34	11.71	9.76
b)	13.80	7.66	12.53	6.96
c)	13.64	11.37	12.37	10.30

TABLE 2: Table of trajectory lengths and related average speeds.

the hotspot with higher traffic z_{h1}, see Figs. 7a and 8a.

- Since the drone has enough time to pick up traffic located close to hotspots, this is reflected in an average drone velocity about half its nominal value (see Table 2).
- Although sometimes a bit low in this particular scenario, observed average speeds (see Table 2) are compatible with the quadratic approximation of the energy consumption for rotary-wing UAVs (typically more than 10 m/s [14], [41]). The quadratic model is less accurate when the UAV comes close to the hotspot (at low speed), while it is a good approximation when it leaves the hotspot to reach the destination (at higher speed). Improving the model to take into account the portion of the trajectory with lower speed is left for future work.

On a Mac Book Pro 2.7 GHz Dual-Core Intel Core i5 using Scilab 6.02 (and Java 8), the CPU time with GRAD-ALGO is 0.24 s (5 iterations) and with B-ALGO is 0.023 s (12 iterations)[4].

4. A direct comparison with SCA-based algorithms proposed in the literature [13], [14], [15], [16], [17] would require a significant modification of these algorithms, which is beyond the scope of this paper. These studies indeed consider UAVs serving a finite set of ground stations, while our model assumes a continuous traffic map.

6 CONCLUSION

In this paper, we propose a Lagrangian approach to solve the UAV base station optimal trajectory problem. When the traffic intensity exhibits a single phase, closed-form expressions for the trajectory and speed are derived from Hamilton-Jacobi equations. When the traffic intensity exhibits multiple phases, we characterize the crossing time and location at the interface. We propose two low-complexity algorithms for the bi-phase time-stationary traffic case that provide optimal crossing time and location on the interface and fulfill the necessary conditions of optimality. At last, we present a data processing procedure based on a modified K-means algorithm that derives a single-phase or bi-phase quadratic model from real traffic data. Further extensions of this work are envisioned to generalize the approach to three or more hot-spots and to consider multi-drone coordinated trajectories.

REFERENCES

[1] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on uav communications for 5g and beyond," *Proceedings of the IEEE*, vol. 107, no. 12, pp. 2327–2375, 2019.

[2] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[3] A. Sharma, P. Vanjani, N. Paliwal, C. M. Basnayaka, D. N. K. Jayakody, H. C. Wang, and P. Muthuchidambaranathan, "Communication and networking technologies for UAVs: A survey," *Journal of Network and Computer Applications*, vol. 168, p. 102739, oct 2020.

[4] D. S. Lakew, A. Masood, and S. Cho, "3d uav placement and trajectory optimization in uav assisted wireless networks," in *2020 International Conference on Information Networking (ICOIN)*, 2020, pp. 80–82.

[5] E. H. P. 5G!Drones, "Use case specifications and requirements," Tech. Rep. D1.1, Dec. 2019.

[6] B. Pearre and T. X. Brown, "Model-free trajectory optimization for wireless data ferries among multiple sources," in *IEEE Globecom Workshops*, Dec 2010, pp. 1793–1798.

[7] V. Sharma, M. Bennis, and R. Kumar, "Uav-assisted heterogeneous networks for capacity enhancement," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1207–1210, June 2016.

[8] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy tradeoff in ground-to-uav communication via trajectory design," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6721–6726, 2018.

[9] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016.

[10] T.-Y. Chi, Y. Ming, S.-Y. Kuo, C.-C. Liao *et al.*, "Civil uav path planning algorithm for considering connection with cellular data network," in *IEEE Intl. Conf. on Computer and Information Technology (CIT)*, June 2012, pp. 327–331.

[11] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2011.

[12] D. Delahaye, S. Puechmorel, P. Tsiotras, and E. Féron, "Mathematical models for aircraft trajectory design: A survey," in *Air Traffic Management and Systems*. Springer, 2014, pp. 205–247.

[13] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for uav-enabled mobile relaying systems," *IEEE Transactions on Communications*, vol. 64, no. 12, pp. 4983–4996, Dec 2016.

[14] Y. Zeng and R. Zhang, "Energy-efficient uav communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, June 2017.

[15] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-uav enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[16] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing uav," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[17] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "Uav trajectory planning for data collection from time-constrained iot devices," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 34–46, 2020.

[18] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operations research*, vol. 26, no. 4, pp. 681–683, 1978.

[19] J. Zhang, Y. Zeng, and R. Zhang, "Uav-enabled radio access network: Multi-mode communication and trajectory design," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5269–5284, 2018.

[20] O. von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, no. 1, pp. 357–373, Dec 1992.

[21] G. Barles, A. Briani, and E. Trélat, "Value function for regional control problems via dynamic programming and pontryagin maximum principle," *Mathematical Control and Related Fields*, vol. 8, no. 3/4, p. 509, 2018.

[22] L. C. Evans, *Partial differential equations*, 2nd ed., ser. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2010, vol. 19.

[23] M. Akian, S. Gaubert, and A. Lakhoua, "The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis," *SIAM Journal on Control and Optimization*, vol. 47, no. 2, pp. 817–848, 2008.

[24] W. McEneaney, *Max-plus methods for nonlinear control and estimation*. Springer Science & Business Media, 2006.

[25] A. Alla, M. Falcone, and L. Saluzzi, "An efficient DP algorithm on a tree-structure for finite horizon optimal control problems," *SIAM Journal on Scientific Computing*, vol. 41, no. 4, pp. A2384–A2406, 2019.

[26] S. Dolgov, D. Kalise, and K. Kunisch, "A tensor decomposition approach for high-dimensional Hamilton-Jacobi-Bellman equations," *arXiv preprint arXiv:1908.01533*, 2019.

[27] W. Kang and L. C. Wilcox, "Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations," *Computational Optimization and Applications*, vol. 68, no. 2, pp. 289–315, 2017.

[28] A. Alla, M. Falcone, and S. Volkwein, "Error analysis for POD approximations of infinite horizon problems via the dynamic programming approach," *SIAM Journal on Control and Optimization*, vol. 55, no. 5, pp. 3091–3115, 2017.

[29] D. Kalise, S. Kundu, and K. Kunisch, "Robust feedback control of nonlinear PDEs by numerical approximation of high-dimensional Hamilton-Jacobi-Isaacs equations," *arXiv preprint arXiv:1905.06276*, 2019.

[30] Y. T. Chow, J. Darbon, S. Osher, and W. Yin, "Algorithm for overcoming the curse of dimensionality for state-dependent Hamilton-Jacobi equations," *Journal of Computational Physics*, vol. 387, pp. 376 – 409, 2019.

[31] J. Darbon and S. Osher, "Algorithms for overcoming the curse of dimensionality for certain Hamilton–Jacobi equations arising in control theory and elsewhere," *Research in the Mathematical Sciences*, vol. 3, no. 1, p. 19, Sep 2016.

[32] I. Yegorov and P. M. Dower, "Perspectives on characteristics based curse-of-dimensionality-free numerical approaches for solving Hamilton–Jacobi equations," *Applied Mathematics & Optimization*, pp. 1–49, 2017.

[33] Y. T. Chow, J. Darbon, S. Osher, and W. Yin, "Algorithm for overcoming the curse of dimensionality for certain non-convex HamiltonJacobi equations, projections and differential games," *Annals of Mathematical Sciences and Applications*, vol. 3, no. 2, pp. 369–403, 2018.

[34] T. Nakamura-Zimmerer, Q. Gong, and W. Kang, "Qrnet: Optimal regulator design with lqr-augmented neural networks," *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1303–1308, 2021.

[35] J. Han, A. Jentzen, and W. E, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, 2018.

[36] J. Darbon, G. P. Langlois, and T. Meng, "Overcoming the curse of dimensionality for some Hamilton–Jacobi partial differential equations via neural network architectures," *Research in the Mathematical Sciences*, vol. 7, no. 3, Jul. 2020.

[37] J. Darbon and T. Meng, "On some neural network architectures that can represent viscosity solutions of certain high dimensional HamiltonJacobi partial differential equations," *Journal of Computational Physics*, vol. 425, p. 109907, 2021.

[38] M. Coupechoux, J. Darbon, J.-M. Kélif, and M. Sigelle, "Optimal trajectories of a uav base station using lagrangian mechanics," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 626–631.

[39] J.-L. Basdevant, *Variational principles in physics*. Springer Science & Business Media, 2006.

[40] X. Chen, Y. Jin, S. Qiang, W. Hu, and K. Jiang, "Analyzing and modeling spatio-temporal dependence of cellular traffic at city scale," in *Communications (ICC), 2015 IEEE International Conference on*, 2015.

[41] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, apr 2019.

[42] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3417–3442, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8675384/

[43] A. Nika, A. Ismail, B. Y. Zhao, S. Gaito, G. P. Rossi, and H. Zheng, "Understanding and Predicting Data Hotspots in Cellular Networks," *Mobile Networks and Applications 2015 21:3*, vol. 21, no. 3, pp. 402–413, oct 2015. [Online]. Available: https://link.springer.com/article/10.1007/s11036-015-0648-6

[44] M. Bierlaire, *Optimization: Principles and Algorithm*. EPFL Press, 2015.

[45] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American statistical association*, vol. 74, no. 368, pp. 829–836, 1979.

[46] K. Ueda and N. Yamashita, "On a global complexity bound of the

levenberg-marquardt method," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 443–453, 2010.
[47] A. Fotouhi, M. Ding, and M. Hassan, "Dronecells: Improving 5g spectral efficiency using drone-mounted flying base stations," *arXiv preprint arXiv:1707.02041*, 2017.

Marceau Coupechoux received the Engineer degree from Telecom Paris in 1999 and University of Stuttgart in 2000, the Ph.D. degree from Institut Eurecom in 2004, the Habilitation degree from University Pierre et Marie Curie in 2015. He is a Professor at Telecom Paris and a Professeur Charg de Cours at Ecole Polytechnique. From 2000 to 2005, he was with Alcatel-Lucent. In 2011-2012, he was a Visiting Scientist with the Indian Institute of Science, Bengaluru, India. He has been a General Co-Chair of WiOpt 2017 and Gamenets 2019. In the Computer and Network Science Department of Telecom Paris, he is working on wireless and cellular networks, focusing mainly on performance evaluation, optimization and resource allocation.

Jrme Darbon received his PhD from Telecom Paris in 2005. He is currently an associate professor at the Division of Applied Mathematics of Brown University and on leave from CNRS. His current research interests include imaging sciences, optimization, optimal control and solving Hamilton-Jacobi PDEs in high dimensions. He was one of the main co-organizer of a semester long program at IPAM on "High Dimensional Hamilton-Jacobi PDEs" in Spring 2020.

Jean-Marc Kelif obtained the Engineer degree in Materials and Solid State Physics Sciences from the University of Paris XIII (Villetaneuse, France) in 1984, and his Ph.D. from Ecole Nationale Suprieure des Tlcommunications in Paris (ENST, Telecom Paris), France. Since 1993, he has been with Orange Labs (Research Centre of Orange). His current research interests include the performance evaluation, modeling, dimensioning and optimization of wireless telecommunication networks.

Marc Sigelle graduated as an Engineer from Ecole Polytechnique in 1975 and from Telecom Paris in 1977. He obtained a PhD from Telecom Paris in 1993. He worked first at Centre National d'Etudes des Télécommunications in physics and computer algorithms. Since 1989 he has been working at Telecom Paris in image processing, his main topics of interests being the restoration and segmentation of images using stochastic/statistical models and methods (Markov Random Fields, Bayesian networks, Graph Cuts in relationships with Statistical Physics). In the past few years he has been involved in the transfer of this knowledge to the modeling and optimization (namely, stochastic) of wireless networks.

APPENDIX A
PROOF OF LEMMA 1

In a neighborhood of the optimal trajectory, the first order variation of S is null

$$\begin{aligned}\delta S &= \int_{t_0}^{T} \delta\mathcal{L}(t,z,a)\,dt \\ &= \int_{t_0}^{T} [\nabla_z\mathcal{L}(t,z,a)\cdot\delta z(t) + \nabla_a\mathcal{L}(t,z,a)\cdot\delta a(t)]\,dt\end{aligned} \tag{29}$$

Noting that $\delta a = \delta\frac{dz}{dt} = \frac{d(\delta z)}{dt}$ and integrating by part the second term in the integral of δS yields

$$\begin{aligned}\int_{t_0}^{T} \nabla_a\mathcal{L}(t,z,a)\cdot\frac{d(\delta z)}{dt}\,dt &= [\delta z(t)\cdot\nabla_a\mathcal{L}(t,z,a)]_{t_0}^{T} \\ &\quad - \int_{t_0}^{T}\delta z(t)\cdot\frac{d}{dt}\nabla_a\mathcal{L}(t,z,a)\,dt\end{aligned} \tag{30}$$

Note that $[\delta z\frac{\partial\mathcal{L}}{\partial a}]_{t_0}^{T} = 0$ because z_0 and z_T are fixed. Equating δS to zero gives

$$0 = \int_{t_0}^{T}\left[\nabla_z\mathcal{L}(t,z,a) - \frac{d}{dt}\nabla_a\mathcal{L}(t,z,a)\right]\cdot\delta z(t)\,dt \tag{31}$$

As this should be true for every δz, $\mathcal{L}$, z_0 and z_T, we obtain the first result. Now assume that we have the optimal $a(t)$, the condition for variable $z(T)$ to be optimal is

$$\begin{aligned}\delta S &= [\delta z(t)\cdot\nabla_a\mathcal{L}(t,z,a)]_{t_0}^{T} + \nabla J(z(T))\cdot\delta z(T) \\ &= \nabla_a\mathcal{L}(z(T),T,a(T))\cdot\delta z(T) + \nabla J(z(T))\cdot\delta z(T) \\ &= 0\end{aligned} \tag{32}$$

Note that z_0 is fixed and so δz in z_0 is null. We thus obtain the second result of the lemma.

APPENDIX B
PROOF OF LEMMA 2

As $\mathcal{L}(z,a)$ is an homogeneous function of z and a, we have: $\mathcal{L}(\lambda z,\lambda a) = |\lambda|^{\alpha}\mathcal{L}(z,a)$ for all λ (in our case with $\alpha = 2$). Deriving this expression with respect to λ, setting $\lambda = 1$, and noting that $a = \dot{z}$ we obtain

$$z\cdot\frac{\partial\mathcal{L}(z,\dot{z})}{\partial z} + \dot{z}\cdot\frac{\partial\mathcal{L}(z,\dot{z})}{\partial z} = \alpha\mathcal{L}(z,\dot{z}). \tag{33}$$

Using (4) and (33), we have: $z\cdot\frac{dp}{dt} + \dot{z}\cdot p = \alpha\mathcal{L}$ or equivalently $\frac{d(p\cdot z)}{dt} = \alpha\mathcal{L}$. We can now integrate the cost function (2) along the optimal trajectory as follows

$$\begin{aligned}S(t_0,z_0,T,z_T) &= \frac{1}{\alpha}\int_{t_0}^{T}\frac{d(p\cdot z)}{dt}(t)\,dt + J(z_T) \\ &= \frac{1}{\alpha}\left(p(T)\cdot z_T - p(t_0)\cdot z_0\right) + J(z_T)\end{aligned}$$

APPENDIX C
PROOF OF LEMMA 3

We assume that an optimal trajectory exists and we apply the principle of optimality on the optimal trajectory between $(t,z^*(t))$ and $(t+h,z^*(t)+ah)$, where $h>0$, to obtain

$$\begin{aligned}S(t,z^*(t),T,z_T) &= \min_a[h\mathcal{L}(t,z^*(t),a) \\ &\quad + S(t+h,z^*(t)+ah,T,z_T)] \\ &= \min_a[h\mathcal{L}(t,z^*(t),a) + S(t,z^*(t),T,z_T) \\ &\quad + ha\cdot\nabla_{X_1}S(t,z^*(t),T,z_T) \\ &\quad + h\frac{\partial S}{\partial T_1}(t,z^*(t),T,z_T)].\end{aligned} \tag{34}$$

$$\begin{aligned}\frac{\partial S}{\partial T_1}(t,z^*(t),T,z_T) &= -\min_a[a\cdot\nabla_{X_1}S(t,z^*(t),T,z_T) \\ &\quad + \mathcal{L}(t,z^*(t),a)] \\ &= \max_a[-a\cdot\nabla_{X_1}S(t,z^*(t),T,z_T) \\ &\quad - \mathcal{L}(t,z^*(t),a)] \\ &= H(t,z^*(t),-\nabla_{X_1}S(t,z^*(t),T,z_T)).\end{aligned}$$

By using the same approach between $t-h$ and t, we deduce in the same way equation (6) when the final time T is varying.

APPENDIX D
PROOF OF THEOREM 1

From (3) and (8), we obtain the following ordinary differential equation of second degree: $\ddot{z} = -\frac{u_0}{K}z$. If $\frac{u_0}{K} > 0$, we define $\omega^2 = \frac{u_0}{K}$ and we look for an optimal trajectory of the form: $z(t) = A\cos(\omega t) + B\sin(\omega t)$. If $\frac{u_0}{K} < 0$, we look for an optimal trajectory of the form: $z(t) = A\cosh(\omega t) + B\sinh(\omega t)$ with $\omega^2 = -\frac{u_0}{K}$. Let us denote $z_0 = z(t_0)$ and $a_0 = a(t_0)$ the initial conditions for z and $\dot{z}$. Take the case $\frac{u_0}{K} < 0$. Using the derivative of $z(t)$ and identifying terms, we obtain: $z(t) = z_0\cosh\omega(t-t_0) + \frac{a_0}{\omega}\sinh\omega(t-t_0)$. At $t = T$, we have also: $z_T = z_0\cosh\omega(T-t_0) + \frac{a_0}{\omega}\sinh\omega(T-t_0)$, from which we deduce

$$a(t_0) = \frac{\omega(z_T - z_0\cosh\omega(T-t_0))}{\sinh\omega(T-t_0))}, \tag{35}$$

$$a(T) = \frac{\omega(-z_0 + z_T\cosh\omega(T-t_0))}{\sinh\omega(T-t_0))}. \tag{36}$$

when $u_0 < 0$. In a similar way, we have

$$a(t_0) = \frac{\omega(z_T - z_0\cos\omega(T-t_0))}{\sin\omega(T-t_0)}, \tag{37}$$

$$a(T) = \frac{\omega(-z_0 + z_T\cos\omega(T-t_0))}{\sin\omega(T-t_0)}, \tag{38}$$

when $u_0 > 0$. Injecting $a(t_0) = a_0$ in the equation of the trajectory provides the result. For the computation of S, we now use the result of Lemma 2 as our cost function is 2-homogeneous. From equation (5), we see that only initial and final conditions are required to compute the cost function. Recall now that $p = Ka$. Injecting the equations of $a(t_0)$ and $a(T)$ in (5), we obtain the result for the cost function.

APPENDIX E
PROOF OF THEOREM 2

The optimal trajectory between (z_0,t_0) and (z_T,T) can be decomposed in two sub-trajectories that are themselves

optimal between (z_0, t_0) and (ξ, τ) on the one hand and between (ξ, τ) and (z_T, T) on the other hand, by the principle of optimality.

In region 1, the optimal cost up to τ is $S_1(t_0, z_0, \tau, \xi) = \int_{t_0}^{\tau} \mathcal{L}(z^*(s), a^*(s))ds$. Using Hamilton-Jacobi, we obtain $\frac{\partial S_1}{\partial T_2}(t_0, z_0, \tau, \xi) = -H_1(\xi, p^*(\tau^-))$.

Similarly, the optimal cost in region 2 is $S_2(\tau, \xi, T, z_T) = \int_{\tau}^{T} \mathcal{L}(z^*(s), a^*(s))ds$ and $\frac{\partial S_2}{\partial T_1}(\tau, \xi, T, z_T) = H_2(\xi, p^*(\tau^+))$.

A necessary condition for the optimality of τ is thus $\frac{\partial S_1}{\partial T_2}(t_0, z_0, \tau, \xi) + \frac{\partial S_2}{\partial T_1}(\tau, \xi, T, z_T) = 0$, that is $H_1(\xi, p^*(\tau^-)) = H_2(\xi, p^*(\tau^+))$.

A necessary condition for the optimality of ξ in the total cost under the constraint $f(\xi) = C$ is also $\mu \nabla_z f(\xi) = \nabla_{X_2} S_1(t_0, z_0, \tau, \xi) + \nabla_{X_1} S_2(\tau, \xi, T, z_T) = p^*(\tau^-) - p^*(\tau^+)$ where μ is a Lagrange multiplier associated to the constraint and where the second equality comes from equation (7) of Hamilton-Jacobi. Thus we obtain precisely equation (16).

APPENDIX F
STRUCTURE OF THE HESSIAN

F.1 Proof of Theorem 3

Recall that we study the second-order differentiability properties of single-phase value function $S(t_1, x_1, t_2, x_2)$. From Eq. (1) we study Hessians of the form $\mathcal{M}_{1+2,1+2}(\mathbb{R}) \ni \nabla^2\sigma = \begin{pmatrix} \frac{\partial^2}{\partial {T_i}^2}\,\sigma & \frac{\partial}{\partial T_i}\nabla_{Xi}\,\sigma \\ \nabla_{Xi}\frac{\partial}{\partial T_i}\,\sigma & \nabla^2_{X_i,X_i}\,\sigma \end{pmatrix}$, where σ stands for $S(t_1, x_1, t_2, x_2)$, and $i = 1, 2$. It is clear by inspecting the symmetries of (10) with respect to the spatial coordinates that $\nabla^2_{X_1,X_1} S(t_1, x_1, t_2, x_2) = \nabla^2_{X_2,X_2} S(t_1, x_1, t_2, x_2) = Kg\ \mathrm{Id}_2$, where $g = \omega\ \coth\phi$ and $\phi = \omega\ (t_2 - t_1)$ is the *temporal* phase. Concerning the symmetry with respect to the time variables, one also finds easily that

$$\frac{\partial^2 S}{\partial {T_1}^2}(t_1, x_1, t_2, x_2) = \frac{\partial^2 S}{\partial {T_2}^2}(t_1, x_1, t_2, x_2).$$

The Hessian structure with respect to the variable $\chi_i = (T_i, X_i)_{i=1,2} \in \mathbb{R} \times \mathbb{R}^2$ is thus

$$\mathcal{H}(\chi_i) = \left(\begin{array}{c|c} \alpha & {\Pi_i}^\dagger \\ \hline \Pi_i & Kg\ \mathrm{Id}_2 \end{array}\right) \text{ with } \left\{ \begin{array}{l} \alpha = \frac{\partial^2 S}{\partial {T_1}^2} = \frac{\partial^2 S}{\partial {T_2}^2} \\ g = \omega\ \coth\phi \\ \Pi_i = \frac{\partial}{\partial T_i}\nabla_{X_i} S \end{array}\right. \tag{39}$$

where the vector Π_i and real scalar α remain to be determined. Now recall that initial and final impulsions p_1 and p_2 as given by (35) are

$$p_1 = K\ \omega\ \frac{-x_1 \cosh\phi + x_2}{\sinh\phi}, \tag{40}$$

$$p_2 = K\ \omega\ \frac{-x_1 + x_2 \cosh\phi}{\sinh\phi}. \tag{41}$$

In fact the expressions of Π_i and α appearing in Hessian (39) result both from the computation of $\frac{\partial p_i}{\partial T_i}$ for $i = 1,\ 2$.

Using the expression of the temporal phase, the differentiation of (40) and (41) with respect to T_1 and T_2 leads to $\Pi_1 = \frac{\partial}{\partial T_1}\nabla_{X_1} S = \frac{\partial p_1}{\partial T_1} = -\ \omega\ \frac{p_2}{\sinh\phi}$ and $\Pi_2 = \frac{\partial}{\partial T_2}\nabla_{X_2} S = \frac{\partial p_2}{\partial T_2} = -\ \omega\ \frac{p_1}{\sinh\phi}$.

Now, the second partial derivative of value function S with respect to time α reads $\alpha = \frac{\partial^2 S}{\partial {T_1}^2} = -\frac{\partial H}{\partial T_1}$, where the last equality follows for the Hamilton-Jacobi PDE.

Recall that the Hamiltonian has the form $H(z, p, t) = \frac{\parallel p \parallel^2}{2\ K} + V(z)$.

Then differentiating it with respect to the *temporal phase* ϕ for *fixed* extremities x_1, x_2 leads to $\delta H = \frac{p_1 \cdot \delta p_1}{K} = \frac{p_2 \cdot \delta p_2}{K} = -\frac{p_1 \cdot p_2}{K\ \sinh\phi}\ \delta\phi$ that is

$$\alpha = \frac{\partial^2 S}{\partial {T_1}^2} = \frac{\partial^2 S}{\partial {T_2}^2} = -\ \omega\ \frac{\partial H}{\partial \phi} = \frac{\omega}{K}\ \frac{p_1 \cdot p_2}{\sinh\phi}. \tag{42}$$

Now from (19) the two-phase Hessian is simply the sum of the two single-phase Hessians.

F.2 Diagonalizing the Hessian

Since the Hessians of single- and two-phase total cost have the same structure, we consider a general Hessian of the form (39). Its characteristic polynom is easily developed as $\det(\mathcal{H} - \nu\ \mathrm{Id}_3) = (Kg - \nu)^2\ (\alpha - \nu) - (Kg - \nu) \parallel \Pi \parallel^2$.

Since from (39) one has $\forall(\theta,\, x) \in \mathbb{R} \times \mathbb{R}^2$, $\mathcal{H}(\theta,\, x) = (\alpha\ \theta + \Pi \cdot x,\, \Pi\ \theta + Kg\ x)$, the three real eigenvalues $(\nu_i)_{i=0:2}$ and related eigenvectors $Y_i = (\theta_i, x_i)$ satisfy:

i) For $\nu_0 = Kg > 0$, then $Y_0 = (\theta = 0,\, x_0)$ with $x_0 \perp \Pi$, i.e., we have a "space" eigenvector $Y_0 \in \{0\} \times \mathbb{R}^2$.

ii) For $(\nu_i)_{i=1,2}$ such that $(Kg - \nu_i)\ (\alpha - \nu_i)\ - ||\Pi||^2 = 0$, then $Y_i = (\theta_i,\, x_i = \Pi)_{i=1,2}$. We have two (mutually orthogonal) eigenvectors orthogonal to Y_0 (since $\Pi \perp x_0$) with

$$\nu_1\ \nu_2 = \alpha\ Kg\ - ||\Pi||^2$$

and

$$\nu_1 + \nu_2 = \alpha + Kg$$

Thus ν_1 and ν_2 cannot be simultaneously negative since this would imply both that $\alpha < -Kg < 0 \Rightarrow \det(\mathcal{H}) < 0$ and $\det(\mathcal{H}) = \nu_0\ \nu_1\ \nu_2\ > 0$! However, the condition $\alpha\ Kg\ < \parallel \Pi \parallel^2$ (e.g. induced by the *sufficient* condition $\alpha < 0$) implies that *one* of the eigenvalues ν_1 or ν_2 is strictly negative, thus there is a *local non-convexity* of the Hessian matrix $\mathcal{H}$. This property holds for a single-phase total cost as well as for the two-phase case.

APPENDIX G
ESTIMATION OF PARAMETERS

The estimation of the simulation parameters is based on a typical drone cell speed $\bar{V} = 20$ m/s and autonomy of about 28 min [47]. We test our procedure for spatially-scaled data, where the convenient spatial unity is 100 m so that the scaling ratio is $r = \frac{40.10^6}{360} \times 100 \approx 1111.1111$. For estimating T in seconds, we approximate the trajectory length by $L \approx 1.5\times \parallel z_0 - z_T \parallel$, which provides $T = \frac{\bar{V}}{L}$.

For estimating K, we impose that the phase in each sub-trajectory should satisfy $\phi_i = \omega_i T_i < 10$, $i = 1, 2$. Since in a spatial scaling by factor r: $u_0^i \leftarrow \frac{u_0^i}{r^2}$, $K \leftarrow \frac{K}{r^2}$ ($K\omega_i^2 = u_0^i$ at any scale), going back into the original frame therefore implies that $K \leftarrow Kr^2$.