Coverage Optimization with a Dynamic Network of Drone Relays

Edgar Arribas, Vincenzo Mancuso, Vicent Cholvi

Abstract—The integration of aerial base stations carried by drones in cellular networks offers promising opportunities to enhance the connectivity enjoyed by ground users. In this paper, we propose an optimization framework for the 3–D placement and repositioning of a fleet of drones with a realistic inter-drone interference model and drone connectivity constraints. We show how to maximize network coverage by means of an *extremal-optimization* algorithm. The design of our algorithm is based on a mixed-integer non-convex program formulation for a coverage problem that is NP-Complete, as we prove in the paper. We not only optimize drone positions in a 3–D space in polynomial time, but also assign flight routes solving an assignment problem and using a strong geometrical tool, namely *Bézier curves*, which are extremely useful for non-uniform and realistic topologies. Specifically, we propose to fly drones following *Bézier curves* to seek the chance of approaching to clusters of ground users. This enhances coverage over time while users and drones move. We assess the performance of our proposal for synthetic scenarios as well as realistic maps extracted from the topology of a capital city. We demonstrate that our framework is near-optimal and using *Bézier curves* increases coverage up to 47% while drones move.

Index Terms—Aerial networks, Relay, UAV, Mobile networks, Optimization, *Bézier curves*.

1 Introduction

Cellular networks are experiencing deep changes due to the advent of 5G technologies [1]. The need of flexible and adaptive management solutions, to address a highly mutable density of users, has allowed novel communication paradigms to emerge, e.g., device-to-device (D2D) communications, smart relay and the use of reconfigurable backhaul links controlled by Software-Defined Networking (SDN) tools [2]. Since the interest for mobile and topology-adaptive relays is now reviving—due to the techniques that make it doable in operational networks rather than just in theoretic speculations—in this paper we focus on the analysis of a key use-case: a fleet of coordinated drones carrying *aerial* relays meant to optimize cellular coverage.

The availability of broadband backhaul links allows manned and unmanned vehicles (e.g., drones) to carry mobile relays. The use of mobile relays brings unique opportunities to deploy adaptive and flexible networks that provide connectivity where fixed infrastructures lack operational connectivity [3]. Basic advantages from relays, like the reuse of cellular bands, have been studied by several authors, e.g., by Guo and O'Farrel [4], whose results illustrate how wireless networks are mainly impaired by interference. Other studies confirm that interference is commonly the main limiting factor also for the performance of relays operated over D2D sidelinks [5]. Hence, relays need to use dedicated frequencies in practice, which has the further advantage of simplifying resource allocation schemes [6].

Thus, we study the case of drones as relay stations transmitting on orthogonal frequencies with respect to ground base stations. However, due to limited spectrum resources, drones *do* interfere with each other and interference cannot be neglected both in users access and backhaul links, as we study in our work. These aspects have not been studied so far because existing works on drone relays focused on the optimal placement of a single drone in isolation, mainly under static scenarios, and neglect connectivity issues in the backhaul links between drones and ground base stations.

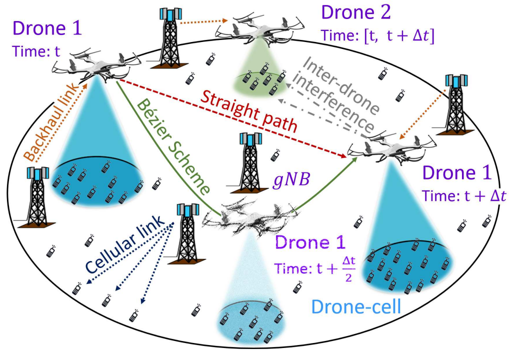


Figure 1. Reference scenario: multi-drone-aided network.

Although relay optimization can be used to support quality of service (QoS) in many different aspects, here we focus on providing support to user applications that require high probability to find the network available. This is important to guarantee essential communication means for voice and real time data communications. It is even more important for addressing the case of safety/emergency events and whenever notifying the status of users and having basic real-time communication tools is vital. These scenarios require to maximize the number of users that can be served, rather than maximizing on the allocated data rate or on the aggregate network rate. Therefore, we refer to QoS in terms of network availability with basic guaranteed rate.

Unlike other proposals, in this paper we focus on the optimization of 3–D hovering positions and flight routes for a fleet of drone relays aiding a ground cellular network,

- *E. Arribas is with IMDEA Networks Institute and Universidad Carlos III de Madrid (UC3M), Madrid, Spain. E-mail: edgar.arribas@imdea.org*
- *V. Mancuso is with IMDEA Networks Institute, Madrid, Spain. E-mail: vincenzo.mancuso@imdea.org*
- *V. Cholvi is with Universitat Jaume I (UJI) de Castelló, Spain. E-mail: vcholvi@uji.es*

as depicted in Figure 1. Drones are coordinated yet they mutually interfere. We optimize coverage based on the QoS offered by drones under realistic path-loss models for line-of-sight (LoS) and non-LoS (NLoS) communications and interference. Considering interference is key because it results in radically different coverage footprints and imposes strict constraints on the position of drones with respect to the position of ground base stations. We use *extremal-optimization* [7] and propose the On-demand Drone Coverage (OnDrone) algorithm, an *extremal-optimization* algorithm that computes near-optimally joint positions for drones, based on realistic assumptions on previous drone positions and interference, which is otherwise an intractable NP-Complete problem. We also propose for the first time the use of *Bézier curves* [8] for flight routes aiming to enhance communications over time. We assess the benefits of our optimization framework by (i) comparing OnDrone against the optimal solutions and state-of-the-art approaches for tractable cases; (ii) performing numerical simulations for larger networks with realistic topologies and environmental constraints; (iii) evaluating fleet repositioning using either *Bézier curves* or straight paths as drone routes. Our numerical results show that OnDrone is nearly optimal and outperforms state-of-the-art coverage solutions as proposed in [9] and [10]. Also, we show that the use of *Bézier curves* is key to boost coverage when studying drone repositioning in dense urban scenarios, and shows remarkable advantages over straight paths, as adopted in [11].

The main contributions of this paper are summarized:

- We propose a dynamic drone relay-aided network in which we maximize the coverage of ground users by means of aerial base stations with an interference-aware on-demand multi-drone coverage framework that accounts for both user access and backhaul links.
- We prove that the problem is NP-Complete.
- We propose OnDrone, a light-weight algorithm based on *extremal-optimization* that solves the problem *on-demand*.
- We propose the use of a strong geometrical tool to design the flight paths of drones: the *Bézier Scheme*.
- We assess our proposals in realistic scenarios and topologies in comparison with state-of-the art solutions and show the gain of our proposals.

The rest of the paper is structured as follows: Section 2 provides a discussion on related work regarding aerial networks and *extremal-optimization*. Section 3 presents the system model assumptions for the reference scenario and wireless channels, while Section 4 states and formulates the coverage problem, and shows that it is intractable. Section 5 details the overall optimization framework. Section 6 reports numerical results. Section 7 concludes the paper.

2 Related work

The usage of relays operating in the air space through mobile and non-terrestrial devices has been studied for several purposes, over different technologies. For instance, satellite networks [12] have already been deployed for several years. However, satellites aim to provide service to huge areas, typically at relatively low transmission rates. Moreover, satellites located hundreds of kilometers high are not able to adjust to ground users' topology, neither track the movement of small masses of users. In contrast, drone relay stations are able to dynamically change their position in real time at low altitudes, as the system evolves.

Google and Facebook carry the Loon [13] and Aquila [14] projects, respectively. They use aerial base stations mounted on high-altitude platforms—e.g., balloons—that fly several kilometers high. Balloons can drift slowly and provide basic network services to remote and rural areas. Instead, swarms of low-altitude drone relays can provide broadband connectivity and can reposition effectively on small time scales.

In [15], Zeng *et al.* study the use of aerial relays to relay traffic from two ground nodes whose links have been disrupted (due to big obstacles, environment or loss of infrastructures). Authors maximize throughput service and relay trajectories with an efficient algorithm that applies successive convex optimizations. In [16], Chen *et al.* extend this problem to the case where the relay network is offered by a swarm of multiple drones, and compare the effects of sending the traffic over one multi-hop link with using several two-hop links. In [17], Zhang *et al.* further focus on the multi-hop link of the aerial network to optimize the trajectories that maximize the end-to-end throughput and minimize transmit powers. These relay problems unveil the potentials of mounting relays on drones, and show clear use-cases for the applicability of such scenario. However, these problems are different to our approach, since they focus on the relay of traffic between two ground nodes. In contrast, we propose a drone relay-aided network that overlays the cellular network. Thus, we optimize the coverage service of users in a big region where a connected cellular infrastructure of ground base stations and mobile users exists, although it might become temporarily insufficient.

Moving drones to optimal locations to maximize various metrics, as coverage, is actively being investigated. Al-Hourani *et al.* [18] analytically model optimal altitude for one drone, to maximize coverage. Also, Hayajneh *et al.* [19] derive optimum drone altitude to minimize outages and bit-error rate. Mei *et al.* [20] propose a decentralized inter-cell interference coordination scheme to maximize the weighted sum-rate of one aerial station and all users, as well as the uplink cell association over multiple resources. Mozaffari *et al.* [21] found the optimal location for multiple *non-interfering* drones to minimize the total transmission power. The same authors also analyze the performance of a single-drone-aided cell in the presence of underlaid D2D users, using stochastic geometry [22]. Chen *et al.* [23] optimize the location of drone relays to provide aerial caching for mobile users that connect to drones by means of millimeter wave links. Andryeyev *et al.* [10] estimate drone positions to increase cellular capacity by means of a self-organization scheme based on *repulsion* from base stations and drones, and *attraction* by users. They use conventional ground path-loss models for wireless channels, which differ substantially from the actual—and more complex—air-to-ground signal propagation model we use here.

Optimal location of multiple interfering drones was studied by Mozaffari *et al.* [24] and Rohde *et al.* [25]. However, differently from us, the former characterizes the network in the presence of only two drones, while the last obviates the effects of LoS channel conditions. Although they shed light on the nature of the problem, using such strong assumptions is limiting for real drone deployments. Conversely, in a companion work [26], we propose an optimization framework for cellular networks aided by a fleet of interfering coordinated drones, although the target optimization and use cases differ from the approach used here in important aspects, and so do problem formulation and heuristics. In [26], the network is optimized in terms of resource allocation, so to maximize throughput *and* fairness in the presence of significant crowds. That work does not focus on path optimization and does not consider backhaul limitations. Instead, here we maximize the amount of users covered for use cases in which providing network availability is the key performance indicator. Differently from [26] here we do not need to optimize resource allocation in detail, and can simplify the drone relocation problem by imposing bounds on the guaranteed user data rate. Nonetheless, we show that the problem remains NP-Complete. Similarly, Kalantari *et al.* [27] also optimize user association and fair bandwidth allocation in one cell with multiple drones. They show that interference does not allow to run more than three drones (per cell), and propose to obviate the problem by using advanced interference cancellation. This approach could be combined with our schemes, although it would increase complexity significantly.

Fotouhi *et al.* [11] propose a distributed algorithm for autonomous control of drones, and analyze the benefits of repositioning for spectral efficiency using straight paths. However, the literature does not offer yet any clever scheme to design drone paths to assist communication networks. *Bézier curves* [8] have been used to plan drone routes in [28] for military purposes, to smooth drone routes that have to fly over several check-points. However, *Bézier curves* have not been proposed yet to improve communications performance, as we approach for the first time in this paper.

A complete network architecture that could support a coordinated fleet of drone relays is still under design. Petrolo *et al.* [29] have designed ASTRO, a software-defined network for tetherless coordination of autonomous drones. Sundaresan *et al.* [30] have designed SkyCore, a network module integrated into an end-to-end network architecture called SkyLite. That is a complete network architecture for autonomous drone relay coordination, which demonstrates that operating an aerial network of drone relays is feasible provided the correct optimizations, like coverage maximization (which they do not approach). The maximum coverage framework that we discuss in this manuscript would fit in and be an asset for the above mentioned architectures.

We show in Section 4 that finding exact optimal drone coverage locations is an NP-Complete problem, which is non-linear and non-convex, thus intractable with solvers that provide strictly optimal results. However, we adopt an *extremal-optimization scheme* [7] that let us find near-optimal solutions requiring a much lower number of iterations.

Roughly speaking, such schemes focus on, at each time step, picking the "least fit" element of a discrete set and change its value to the "best fit" in order to improve a given utility function. In our case, the utility function that we want to maximize is the coverage, and the elements that we use to achieve such maximization are the positions of the drones. Since the position of a given drone affects which users other drones will cover, and also influences how the QoS is impaired by interference, *extremal-optimization* represents a suitable choice for our optimum drone placement and repositioning framework. At this point, we remark that although *extremal-optimization* is a form of optimization originally introduced to be used in a static manner, in our work we use it to design a deployable algorithm for the location of drones that works in a dynamic manner.

3 System model

3.1 Reference scenario

We consider a ground surface $\mathcal{S}$ administrated by the ground network consisting of a set $\mathcal{G}$ of ground base stations, as shown in Figure 1.[1] We refer to ground base stations as *gNB*s, using the new 3GPP jargon for next generation base stations (BSs). In the region $\mathcal{S}$, the ground network provides service to a set $\mathcal{U}$ of user equipments (UEs), i.e., mobile users. In order to increase coverage, we consider that coverage assistance is provided by the presence of a finite set $\mathcal{D}$ consisting of D drone relay stations. Each drone is equipped with a mobile relay that gives access to UEs on an orthogonal downlink bandwidth with respect to the *gNB*s band. We refer to drones as aerial base stations (*ABS*s).

We assume that *gNB*s provide backhaul connectivity to *ABS*s over the reuse of the downlink spectrum used for *gNB*–UEs access links. Current *gNB*s provide cellular coverage through three sectors pointing mainly to the ground. We assume that in order to set backhaul *gNB*–*ABS* links, *gNB*s have an additional full dimensional antenna array that performs 3D–beamforming over clear LoS links[2], as suggested and studied in [31]. Therefore, access links *gNB*–UEs and backhaul links *gNB*–*ABS* do not interfere. Furthermore, *gNB*s equipped with this kind of antenna array are able to perform 3D–beamforming to several relays, and alternate transmissions over time slots on a millisecond scale. Hence, each *gNB* g can provide backhaul service to a limited number of *ABS*s, namely D_g.

The coverage of each *ABS* is an irregular ground area that depends on the drone height, cell environment and interference from other *ABS*s. The interference among *ABS*s directly affects the signal-to-interference-plus-noise ratio (SINR) that the ground users receive. The SINR depends on the air-to-ground path-loss model, which is based on the link LoS probability between drones and users. As described later, such path-loss model clearly differs from the conventional attenuation models used in ground cellular networks. Indeed, such a LoS-based path-loss and interference model for the communications channel provides a multi-*ABS* coverage framework for aerial networks, which is radically different from conventional frameworks for ground networks—as e.g., ground D2D networks [5]—and whose characteristics we study. In this framework, we

[1] New SDN tools designed to manage large networks are able to coordinate ground base stations to perform any network optimization [2].

[2] Based on the receiver location or instantaneous channel state information, 3D–beamforming allows to build directional beam-patterns that focus the transmission energy on the direction where the receiver is. This flexible technique helps to mitigate interference so to provide higher rates. 3D–beamforming is very useful for backhaul wireless links from one source to few relays, as in an aerial backhaul network.

consider that a *gNB* g and an *ABS* d can realistically serve a limited number of users, namely U_g and U_d, respectively. We further assume that channel bandwidth is equally split among the users that a BS serves, although more sophisticated schedulers could be easily adopted in the analysis.

With the above, we aim to find optimal locations for D drones, so to maximize the number of users covered by *gNB*s *and ABS*s with a guaranteed bandwidth. Besides, we identify two additional problems to support fleet repositioning: (i) deciding which drone flies to which position upon an optimization update and (ii) designing flight routes.

3.2 Air & Ground channels: path-loss and interference

We assume that the surface $\mathcal{S}$ is flat, so that the position of a UE $u \in \mathcal{U}$ is taken as an input and denoted by $\pi^u = (x^u, y^u, 0)$. The position of a *gNB* $g \in \mathcal{G}$ is denoted by $\Pi_g = (X_g, Y_g, h_g)$. The positions of all *ABS*s $d \in \mathcal{D}$ are the decision variables of the coverage problem, and denoted by $\Pi^d = (X^d, Y^d, h^d)$.

For all drone $d \in \mathcal{D}$, and for all user $u \in \mathcal{U}$, the horizontal distance between u and the ground projection of d is $r_{d,u} = \|(X^d, Y^d) - (x^u, y^u)\|$. The elevation of d is h^d. Due to the low altitude of drones—few hundreds of meters at most—the channel conditions of communications between a serving drone and an end-user are much affected by the LoS. Depending on whether the access link is free of obstacles (like buildings, traffic, etc.), the attenuation differs considerably [32]. Thus, the air-to-ground path-loss among *ABS*s and UEs depends on the probability of LoS, which is a complex function of the elevation angle between user u and drone d, according to the following expression:

$$P_{LoS}(d,u) = \frac{1}{1 + a \cdot e^{-b\left(\frac{180}{\pi}\arctan\left(\frac{h^d}{r_{d,u}}\right) - a\right)}}, \quad (1)$$

where a, b are parameters depending on the environment, i.e., number of buildings and large signal obstructions per unit area, building's height distribution, ratio of built-up area and clean surfaces, etc., as it has been derived in [18], based on the ITU recommendations [33]. In Eq. (1), the elevation angle (in radians) appears as $\theta_{d,u} = \arctan(h^d/r_{d,u})$. As $\theta_{d,u}$ approaches $\frac{\pi}{2}$, i.e., when the drone d hovers just above the user u, the probability of LoS reaches its maximum value. In Figure 2, we see that the positions of the drones directly affect blockage conditions of the *ABS*–UE acess links. Thus, the higher a drone hovers, the more likely is to have LoS. However, the strength of the signal gets also attenuated with the distance. For single-drone missions, there is an optimal altitude that maximizes coverage [18]. However, in a multi-drone scenario as the one we discuss in this paper, the effects on interference from other drones are a key additional issue to consider, one that makes the optimal drone hovering altitude depend on the positions and elevations of the rest of the drones. This also precludes the possibility to straightforwardly apply single-drone mission approaches to multi-drone scenarios, since the former are not designed to account for interference, as in [9].

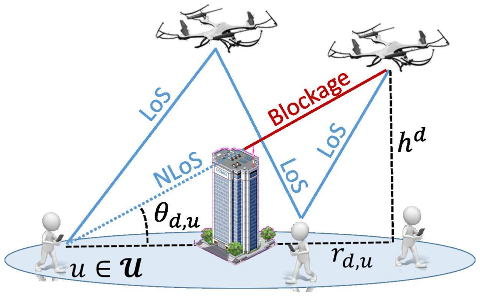


Figure 2. Reference illustration of LoS conditions.

3.2.1 Ground-to-ground channel

Ground cellular links, i.e., *gNB*–UE access links, operate over an OFMDA channel with access bandwidth $W_{\mathcal{A}}$. Hence, users scheduled by the same *gNB* do not suffer intra-cell interference. However, ground users may enjoy poor QoS due to the presence of inter-cell interference, from other close *gNB*s. The path-loss of these channels is based on large- and small-scale fading, as widely studied in literature [34] and shown in Table 1. We denote the SINR of ground access links (g,u) as $\Gamma^{\mathcal{A}}_{g,u}$ and impose that its minimum user access rate, i.e., $\frac{1}{U_g} W_{\mathcal{A}} \log_2(1+\Gamma^{\mathcal{A}}_{g,u})$, is above the guaranteed rate $R^{\mathcal{A}}_{\min}$.

3.2.2 Ground-to-air channel

In order to provide backhaul connectivity to *ABS*s, *gNB*s perform 3D–beamforming over clear LoS links. Hence, the attenuation that a signal from *gNB* g to *ABS* d suffers is:

$$\mathcal{L}_{\mathcal{B}}(g,d) = 10\eta_{\mathcal{B}} \log_{10}\left(\frac{4\pi f_{\mathcal{B}}}{c_l} \cdot \left\|\Pi_g - \Pi^d\right\|\right) + \mathcal{N}_{\sigma_{\mathcal{B}}}, \quad (2)$$

where $\eta_{\mathcal{B}} \approx 2$ is the path-loss exponent in free-space LoS links, $f_{\mathcal{B}}$ is the frequency of backhaul wireless links in Hz, c_l is the speed of light in m/s and $\mathcal{N}_{\sigma_{\mathcal{B}}}$ is a random gaussian variable with zero mean and $\sigma_{\mathcal{B}}$ standard deviation, modelling the effects of slow fading and shadowing.

3D–beamforming builds antenna patterns that radiate much of the energy over a main lobe with a half-power beam-width (HPBW) that may be wide, hence incurring high interference to other *ABS*s in LoS. Also, the formation of directional beam-patterns comes with the presence of side-lobes with non-negligible radiating power, that also incur (low) interference. Thus, depending on the radiating angle of other *gNB*s, a backhaul link may enjoy better or worse QoS, due to the presence of interference[3]. We denote the backhaul SINR of link (g,d) as $\Gamma^{\mathcal{B}}_{g,d}$, which is equal to:

$$\Gamma^{\mathcal{B}}_{g,d} = \frac{P^g_{Tx} \cdot G_g \cdot 10^{-\mathcal{L}_{\mathcal{B}}(g,d)/10}}{N^d + I^{\mathcal{B}}_{g,d}}, \quad (3)$$

where P^g_{Tx} is the transmission power of g, G_g is the antenna gain over the main lobe of the beam-pattern of g, N^d is the thermal noise, and most importantly, $I^{\mathcal{B}}_{g,d}$ is the actual interference that *ABS* d suffers from any other *gNB*, depending on the angle of their beam-patterns, i.e.:

$$I^{\mathcal{B}}_{g,d} = \sum_{g' \in \mathcal{G} \setminus \{g\}} P^{g'}_{Tx} \cdot G_{g'}(\phi_{g',d}) \cdot 10^{-\mathcal{L}_{\mathcal{B}}(g',d)/10}, \quad (4)$$

where $\phi_{g',d}$ is the angle between the direction of the main lobe of the antenna of g' and the position of *ABS* d. We impose that the minimum backhaul rate, $\frac{1}{D_g} W_{\mathcal{B}} \log_2(1+\Gamma^{\mathcal{B}}_{g,d})$, is above a rate $R^{\mathcal{B}}_{\min}$. $W_{\mathcal{B}}$ is the backhaul channel bandwidth.

[3] In general, since beamforming builds antenna patterns with directional main lobes, interference remains low for non-aligned *ABS*s

Table 1
Channel modelling

Channel	***Path-loss***	***Exponent***
Ground-to-Ground (*gNB*–UE)	$10\eta_G \log_{10}\left(\frac{4\pi f_G}{c_l}\right) +$ $10\eta_G \log_{10}\left(\lVert \Pi_g - \pi^u \rVert\right) + \mathcal{N}_{\sigma_G}$	$\eta_G > 2$
Ground-to-Air (*gNB*–*ABS*)	$10\eta_{\mathcal{B}} \log_{10}\left(\frac{4\pi f_{\mathcal{B}}}{c_l}\right) +$ $10\eta_{\mathcal{B}} \log_{10}\left(\lVert \Pi_g - \Pi^d \rVert\right) + \mathcal{N}_{\sigma_{\mathcal{B}}}$	$\eta_{\mathcal{B}} \approx 2$
Air-to-Ground (*ABS*–UE)	$10\eta_{\mathcal{A}} \log_{10}\left(\frac{4\pi f_{\mathcal{A}}}{c_l} \lVert \Pi^d - \pi^u \rVert\right) +$ $P_{LoS}(d,u) \cdot (\xi_{LoS} - \xi_{NLoS}) +$ $\xi_{NLoS} \;\; (+\mathcal{N}_{\sigma_{\mathcal{A}}} + \mathcal{R}_{\varsigma_{\mathcal{A}}})$	$\eta_{\mathcal{A}} = 2$

3.2.3 *Air-to-ground channel*

While ground cellular links suffer from conventional attenuation based on fast and slow fading, the path-loss of an *ABS*–UE link (d,u) differs notably and is affected by an *excess attenuation*, depending on the LoS likelihood presented in Eq. (1). The average attenuation is derived in [18] as:

$$\begin{aligned}\mathcal{L}_{\mathcal{A}}(d,u) \;\; &= \;\; 10\eta_{\mathcal{A}} \log_{10}\left(\frac{4\pi f_{\mathcal{A}}}{c_l} \cdot \sqrt{(h^d)^2 + r_{d,u}^2}\right) \\ &+ \; P_{LoS}(d,u) \cdot (\xi_{LoS} - \xi_{NLoS}) + \xi_{NLoS}, \qquad (5)\end{aligned}$$

where ξ_{LoS}, ξ_{NLoS} are the *excess path-loss* components in LoS and NLoS connections respectively, $\eta_{\mathcal{A}} = 2$ is the path-loss exponent and $f_{\mathcal{A}}$ is the carrier frequency in Hz. As surveyed in [35], we have based the air-to-ground path-loss on the average large-scale fading in order to perform the *ABS*–UE association in the optimization. Nevertheless, we further consider fast and slow fading, modelled as log-normal and Rayleigh distributions, respectively, in the implementation of the framework when we analyze the results. We denote the access link SINR as $\Gamma_{d,u}^{\mathcal{A}}$, which is equal to:

$$\Gamma_{d,u}^{\mathcal{A}} = \frac{P_{Tx}^{d} \cdot 10^{-\mathcal{L}_{\mathcal{A}}(d,u)/10}}{N_u + I_{d,u}^{\mathcal{A}}}, \qquad (6)$$

where P_{Tx}^{d} is the transmission power of the antenna integrated in the *ABS* d, N_u is the thermal noise, and most importantly, $I_{d,u}^{\mathcal{A}}$ is the actual interference that user u suffers from any other *ABS*s. Thus, the interference depends on the 3–D position of the rest of the *ABS*s, i.e.:

$$I_{d,u}^{\mathcal{A}} = \sum_{d' \in \mathcal{D} \setminus \{d\}} P_{Tx}^{d'} \cdot 10^{-\mathcal{L}_{\mathcal{A}}(d',u)/10}. \qquad (7)$$

We impose that the minimum rate, $\frac{1}{U_d} W_{\mathcal{A}} \log_2\left(1 + \Gamma_{d,u}^{\mathcal{A}}\right)$, is above $R_{\min}^{\mathcal{A}}$, where $W_{\mathcal{A}}$ is the access channel bandwidth.

In our system model, *ABS*s operate in orthogonal bandwidth with the cellular band. Thus, there is no interference between cellular users and drone-served users, which is commonly the main limiting factor in aided cellular networks, as e.g., *inband* D2D networks [5].

In Table 1 we gather the path-loss model used for each kind of channel, where f_G is the band used for *gNB*–UE access links and is equal to the *gNB*–*ABS* backhaul links, i.e., $f_G = f_{\mathcal{A}}$, and $\mathcal{R}_{\varsigma_{\mathcal{A}}}$ is a random Rayleigh variable with scale parameter $\varsigma_{\mathcal{A}}$. In Table 2 we summarize the interference suffered in each of the channels, as it has been described along this section. We have shadowed the table cells that imply presence of interference.

Table 2
Channel interference

	Ground-to- -Ground	**Ground- -to-Air**	**Air-to- -Ground**
Ground-to- -Ground	Inter-cell interference	Directional re-use	Orthogonal bands
Ground- -to-Air	Directional re-use	Low interference: 3D–beamforming	Orthogonal bands
Air-to- -Ground	Orthogonal bands	Orthogonal bands	Inter-drone interference

4 Multi-drone coverage framework

We aim to find optimal 3–D positions for a fleet of D drones in which the number of UEs under network coverage is maximum. The optimization is run at regular time intervals, considering every time the users as static, so that static drone positions solve the coverage problem. The coverage maximization provides a set of 3–D coordinates where drones have to fly during the time interval, provided a drone d has to fly towards a reachable destination, i.e., a point within the ball $\mathcal{S}_d$ of radius given by the drone speed times the duration of the optimization update interval. However, the output of the optimization does not necessarily coincide with the assignment of fleet destinations that also minimizes flight time. Hence, we will solve the assignment of fleet destinations as a secondary problem, given the optimal coordinates found by the primary problem.

Although it is possible to formulate the coverage maximization and the minimum flight time assignment in one optimization problem, we believe that it is more clear to decouple both problems and solve them separately, while having the same optimal solution. This is due to the fact that the set of optimal drone positions is not changed by solving the secondary problem and at least one feasible solution exists for the secondary problem, which is the output of the primary problem. Thus, we first present the optimal aerial coverage (Section 4.1) and then the assignment of fleet destinations (Section 4.2).

4.1 Optimal aerial coverage

Coverage Problem $\mathcal{C}$: *Given a fleet* $\mathcal{D}$ *of drone relays in a cellular network managed by a centralized orchestrator,* U *ground users, a height range* $[h_{\min}, h_{\max}]$ *for the ABSs, guaranteed coverage rates* $R_{\min}^{\mathcal{A}}$, $R_{\min}^{\mathcal{B}}$ *for access and backhaul channels respectively, and a maximum number of users* U_g *and* U_d *that gNBs and ABSs can cover, find the optimal 3–D positions* $\Pi^d = (X^d, Y^d, h^d)$ *of drones so to maximize the amount of users covered by ground- and drone-cells.*

Since the positions of drones, including their heights, affect the shape of the covered regions, we can mathematically formulate the Coverage Problem $\mathcal{C}$ to search for optimal values of the continuous decision variables $\Pi^d = (X^d, Y^d, h^d) \in \mathbb{R}^3$ that maximize the number of users under network coverage. Denoting by $C_{b,u}$ the binary variable that takes value 1 if BS $b \in \mathcal{G} \cup \mathcal{A}$ covers user u and 0 otherwise, and by $B_{g,d}$ the binary variable that takes value 1 if *gNB* g provides backhaul service to drone d and 0 otherwise, the formulation of the Coverage Problem $\mathcal{C}$ is:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2927335, IEEE Transactions on Mobile Computing

$$\begin{cases} \max\limits_{\{\Pi^d\}_{d\in\mathcal{D}}} \sum\limits_{u\in\mathcal{U}} \left(\sum\limits_{g\in\mathcal{G}} C_{g,u} + \sum\limits_{d\in\mathcal{D}} C_{d,u} \right); & \\ \text{subject to:} & \\ \textit{Access network constraints:} & \\ \frac{1}{U_b} W_{\mathcal{A}} \log_2\left(1+\Gamma^{\mathcal{A}}_{b,u}\right) \geq R^{\mathcal{A}}_{\min} \cdot C_{b,u}, & \forall b \in \mathcal{G}\cup\mathcal{D}, \forall u \in \mathcal{U}; \\ \sum\limits_{g\in\mathcal{G}} C_{g,u} + \sum\limits_{d\in\mathcal{D}} C_{d,u} \leq 1, & \forall u \in \mathcal{U}; \\ \sum\limits_{u\in\mathcal{U}} C_{g,u} \leq U_g; \quad \sum\limits_{u\in\mathcal{U}} C_{d,u} \leq U_d, & \forall g \in \mathcal{G}, \forall d \in \mathcal{D}; \\ \textit{Backhaul network constraints:} & \\ \frac{1}{D_g} W_{\mathcal{B}} \log_2\left(1+\Gamma^{\mathcal{B}}_{g,d}\right) \geq R^{\mathcal{B}}_{\min} \cdot B_{g,d}, & \forall g \in \mathcal{G}, \forall d \in \mathcal{D}; \\ \sum\limits_{g\in\mathcal{G}} B_{g,d} \leq 1, & \forall d \in \mathcal{D}; \\ \sum\limits_{d\in\mathcal{D}} B_{g,d} \leq D_g, & \forall g \in \mathcal{G}; \\ \textit{Access–backhaul constraints:} & \\ C_{d,u} \leq \sum\limits_{g\in\mathcal{G}} B_{g,d}, & \forall d \in \mathcal{D}, \forall u \in \mathcal{U}; \\ \sum\limits_{d\in\mathcal{D}} \sum\limits_{u\in\mathcal{U}} B_{g,d} \cdot C_{d,u} \leq U_g; & \forall g \in \mathcal{G}; \\ \textit{Drone air-space constraint:} & \\ \Pi^d \in \mathcal{S}_d, & \forall d \in \mathcal{D}. \end{cases} \tag{8}$$

Access network constraints: The first constraint guarantees that the access link rate between BS b and user u is above $R^{\mathcal{A}}_{\min}$ as soon as u accesses the network via b; the second constraint tells that a user cannot be covered by more than 1 BS; the third constraint accounts for the number of users that each BS (either *gNB* or *ABS*) can serve.

Backhaul network constraints: The forth constraint guarantees that the backhaul link rate between *gNB* g and *ABS* d is above $R^{\mathcal{B}}_{\min}$ as soon as d connects to the network via g; the fifth constraint tells that each *ABS* cannot connect to more than 1 *gNB*; the sixth constraint limits the number of drones that each *gNB* g can serve to a maximum of D_g drones.

Access–backhaul constraints: The seventh constraint states that a user u can connect to a drone d only if d is under the coverage of some *gNB*. Hence, each drone that provides network access to at least one ground user is connected to the network via one backhaul link, so that, every ground user served by a drone is indeed attached to the cellular network. The eighth constraint states that the number of users covered by those drones attached to the same *gNB* g is limited by the maximum capacity of users U_g in g.

Drone air-space constraint: Finally, the air location of a drone d has to be within a 3–D region $\mathcal{S}_d \subseteq \mathcal{S} \times [h_{\min}, h_{\max}]$ which can be reached in the time interval used for optimization, depending on flight speed and current drone position.

Feasibility. The optimization problem in Eq. (8) is always feasible. For instance, consider a general instance of the problem. Take a random position for each *ABS* d inside their reachable regions $\mathcal{S}_d$. Now set all binary variables $\{C_{b,u}\}$, $\{B_{g,d}\}$ equal to 0. Since the SINR functions $\Gamma^{\mathcal{B}}_{g,d}$, $\Gamma^{\mathcal{A}}_{b,u}$ are always positive (see Eqs. (3), (6), respectively), the solution satisfies all the constraints. This solution provides a utility function of 0 users covered, but it is feasible.

Complexity. Problem $\mathcal{C}$ is NP-Complete because, as shown in Appendix A, the well-known NP-Complete *minimum geometric disk cover* (MGDC) problem [36] can be reduced, in polynomial time, to a particular case of Problem $\mathcal{C}$.

The first constraint on the user access rate, when $b \in \mathcal{D}$, is non-linear and very complex (also the forth constraint), since it depends on the air-to-ground path-loss shown in Eq. (5) for one link, but also for the interfering links from other drones. To make the constraint more visual and remark its non-linearity and complexity, we develop its expression for a drone d as follows:

$$\frac{\frac{K_1}{(h^d)^2+r^2_{d,u}} \cdot 10^{K_2 P_{LoS}(h^d, r_{d,u})}}{N_u + \sum\limits_{d' \in \mathcal{D}\setminus\{d\}} \frac{K_1}{(h^{d'})^2+r^2_{d',u}} \cdot 10^{K_2 P_{LoS}(h^{d'}, r_{d',u})}} \geq (2^{K_3}-1) \cdot C_{d,u}, \tag{9}$$

where the continuous variable $r_{d,u} = \|(x^u, y^u) - (X^d, Y^d)\|$ is the distance between user u and the ground projection of drone d, and $K_1 = P^d_{Tx} \cdot \left(\frac{c_l}{4\pi f_c}\right)^2 \cdot 10^{\frac{\xi_{NLoS}}{10}}$, $K_2 = \frac{\xi_{NLoS} - \xi_{LoS}}{10}$ and $K_3 = \frac{U_d \cdot R^{\mathcal{A}}_{\min}}{W_{\mathcal{A}}}$ are constant. In Eq. (9) we see that this constraint depends on the position decision variables (X^d, Y^d, h^d) not only as an attenuation from the distance, but they also affect the LoS probability, as shown in Eq. (1).

Unlike previous works like [3], [9], [37], in which the drone-service condition is based *only* on the attenuation or the Signal-to-Noise Ratio (SNR), in (8) we have formulated a novel 3–D drone placement optimization that accounts for the actual inter-drone interference suffered by ground users.

Eq (8) represents a Mixed-Integer Non-Convex Program (MINCP), which is not tractable with currently available optimizers dealing with problems that are, at least, convex. Since problem (8) presents a non-convex formulation mainly because of the attenuation depending on the LoS probability, we cannot apply any off-the-shelf optimizer to optimally solve this problem. In addition, the problem itself is NP-Complete, so we resort to a heuristic, as detailed in Section 5.

4.2 Assignment of fleet destinations

The second problem to solve when users move and drones have to be repositioned is an assignment problem. Since it does not matter which *ABS* goes to which destination (as long as such destination is reachable), we enforce each drone to fly towards the positions that minimize the aggregated flight-time by the fleet. Formally, we dispose of a fleet of D drones that must fly from source positions $\{\pi^d\}_{d\in\mathcal{D}}$ and reach target positions $\{\Pi^{d'}\}^D_{d'=1}$. Thus, we formulate the following *assignment problem*:

$$\begin{cases} \min\limits_{F_{d,d'}} U_{\text{fly}} = \sum\limits_{d\in\mathcal{D}} \sum\limits_{d'=1}^{D} \mathcal{T}\left(\pi^d, \Pi^{d'}\right) \cdot F_{d,d'}; & \\ \text{subject to:} & \\ \sum\limits_{d'=1}^{D} F_{d,d'} = 1, & \forall d \in \mathcal{D}; \\ \sum\limits_{d\in\mathcal{D}} F_{d,d'} = 1, & \forall 1 \leq d' \leq D; \\ F_{d,d'} \in \{0,1\}, & \begin{array}{l} \forall d \in \mathcal{D}; \\ \forall 1 \leq d' \leq D, \end{array} \end{cases} \tag{10}$$

where we have introduced the binary variable $F_{d,d'} \in \{0,1\}$ to denote whether drone d flies from π^d to $\Pi^{d'}$ or not. $\mathcal{T}\left(\pi^d, \Pi^{d'}\right)$ is the assignment weight, and depends on the time that drone d needs to fly from source position π^d to the destination $\Pi^{d'}$. The equality constraints ensure that each destination $\Pi^{d'}$ is reached by only one drone d, and that each drone d reaches only one destination $\Pi^{d'}$. The utility U_{fly} is used to minimize the flight time of the fleet of drones.

For simplicity, we assume that drones d fly at a constant speeds of v_d. Thus, the time needed to fly from π^d to $\Pi^{d'}$ is:

$$\mathcal{T}^*\left(\pi^d, \Pi^{d'}\right) = \left\| \pi^d - \Pi^{d'} \right\| / v_d. \tag{11}$$

However, each drone d can reach only those destinations with coordinates within their reachable region $\mathcal{S}_d$. Hence, we impose infinite required time for each drone d to reach all destinations $\Pi^{d'}$ that fall outside $\mathcal{S}_d$:

$$\mathcal{T}\left(\pi^d, \Pi^{d'}\right) = \begin{cases} \mathcal{T}^*\left(\pi^d, \Pi^{d'}\right), & \text{if } \Pi^{d'} \in \mathcal{S}_d; \\ +\infty, & \text{if } \Pi^{d'} \notin \mathcal{S}_d. \end{cases} \tag{12}$$

Hence, the optimization in Eq. (10) assigns to each drone d one destination $\Pi^{d'}$ such that d is able to reach $\Pi^{d'}$ and the aggregated flight time is minimized.

Feasibility. The optimization problem presented in Eq. (10) is always feasible and finite. For instance, assigning to each drone d the destination $\Pi^d = (X^d, Y^d, h^d)$ obtained as an output of the optimization problem of Eq. (8) provides a (finite) feasible solution.

Complexity. The optimization problem in Eq. (10) is a special case of mixed-integer linear program (MILP), one that can be solved efficiently in polynomial time through the *Hungarian method* [38], with complexity $\mathcal{O}(D^3)$. Therefore, this second problem related to dynamic networks is easy to address optimally in low-degree polynomial time.

5 Dynamic drone repositioning algorithms

So far we have discussed the optimization of the placement of a fleet of *ABS*s hovering over a ground cellular network (see Section 4.1), and the optimal flight assignment that minimizes the flight time (see Section 4.2), thus overlaying a legacy cellular network managed by an orchestrator. Nevertheless, since users move, the optimization is reconsidered periodically, with updated drone air-space constraints. Repositioning has to be run frequently, so that we need efficient heuristics. Next, we describe how to practically implement the optimization framework described so far.

5.1 `OnDrone`: an algorithm suit for on-demand drone coverage optimization

Coverage Problem $\mathcal{C}$ is NP-Complete, thus optima cannot be reliably solved on-demand for fast placement of drones, which is key for dynamic repositioning cases. Thus, even if the problem was optimally solvable, the need of having an efficient heuristic would remain. To this aim, we propose here an On-demand Drone Coverage (`OnDrone`) algorithm, based on an *extremal-optimization* algorithm (EOA) that runs in polynomial time. For benchmarking purposes, we further consider state-of-the-art proposals from [9] and [10].

5.1.1 `OnDrone` *for multi-drone coverage*

EOAs are evolutionary algorithms that restrict the search space and achieve near-optimal results in polynomial time [7]. EOAs are based on a fitness metric and, at each step, try to improve the configuration of the element of the system that yields the least contribution to the fitness metric. Therefore, EOA's principles perfectly match the nature of the coverage problem addressed. Specifically, the fitness function is the number of covered users and the least significant contribution comes from the drone that covers the least number of users. Such drone may be either far from users, where its transmissions are severely affected by the interference coming from the rest of *ABS*s, or in a position where the backhaul service is low. Thus, it is convenient to reposition such drone and increase the coverage.

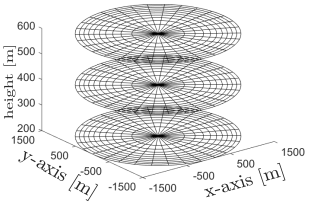


Figure 3. Cylindrical Lattice with $N_\rho = 10$, $M_\theta = 30$, $H = 3$.

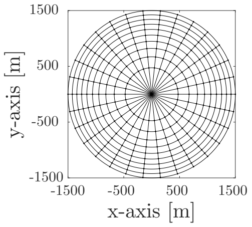


Figure 4. Circular base grid with $N_\rho = 10$, $M_\theta = 30$.

The search space for drone locations is restricted to a lattice, as shown in Figure 3. We derive a cylindrical lattice that contains the ground network surface $\mathcal{S}$ composed by the positions over which `OnDrone` moves the *ABS*s to get the best coverage utility. Those lattice points that fall outside the ground region $\mathcal{S}$ are discarded, so the design of the lattice applies to any shape of $\mathcal{S}$. We split the base into a grid of equal areas (see Figure 4), and the height into equidistant altitudes. This leads to a cylindrical lattice of equal volume subspaces. Specifically, we divide the square of the radius and the angle of the base grid in N_ρ and M_θ equal pieces. In polar coordinates, the resulting points and base grid are:

$$\begin{aligned} \rho_i &= \sqrt{i/N_\rho} \cdot R_C, \quad i = 1, ..., N_\rho; \\ \theta_j &= 2\pi \cdot (j-1)/M_\theta, \quad j = 1, ..., M_\theta; \\ \mathbb{G} &= \{(\rho_i, \theta_j) \in D(0, R_C) \mid 1 \le i \le N_\rho, 1 \le j \le M_\theta\}, \end{aligned}$$

where $D(0, R_C)$ is the closed disk in $\mathbb{R}^2$ centered at the origin and of radius R_C (where R_C is big enough to make $D(0, R_C)$ contain $\mathcal{S}$). In this way, the base area is divided into $N_\rho \times M_\theta$ regions with the same area $A = A_i = \frac{\pi}{M_\theta}(\rho_{i+1}^2 - \rho_i^2) = \frac{\pi R_C^2}{N_\rho M_\theta}$, for $i = 1, ..., N_\rho - 1$, which does not depend on i (see Figure 4). The height of the cylinder is divided into H equidistant segments in the interval between minimum and maximum drone hovering height, $h_{\min}$ and $h_{\max}$. In cylindrical coordinates, the resulting lattice is

$$\mathcal{L} = \{(\rho, \theta, h_k) \in \mathbb{G} \times \mathbb{R} \mid 1 \le k \le H\}. \tag{13}$$

The proposed EOA—`OnDrone`—begins with an initial feasible and suboptimal (random) implementation of the system. Then, it updates the positions of the *ABS*s providing worst individual contribution to the full performance, i.e., the drone covering less users. At each iteration, the "least fit" *ABS* is selected and *moved* to a reachable position where the system utility increases as much as possible, considering the coverage by ground- and drone-cells. Also, `OnDrone` provides a new position where the drone can be attached to a *gNB* that provides backhaul service with the guaranteed QoS. To decrease the probability of finding only local optima, we consider that if the worst performing drone cannot be moved to improve coverage, then we try to reposition the next worst-performing drone. `OnDrone` keeps moving the *ABS*s with lowest contribution until it does not find any better location for any *ABS*, or it reaches a maximum number of iterations $i_0 \in \mathbb{N}$. The optimality of this algorithm is studied in Section 6.

Complexity. Algorithm 1 reports the pseudocode of the proposed `OnDrone`, in order to target maximum ground-plus-air users coverage, thus approximating the optimal solution of Coverage Problem $\mathcal{C}$. The complexity of `OnDrone` can be evaluated as follows. At each iteration, one drone

Algorithm 1 OnDrone: On-demand Drone Coverage for 3-D Drone Placement

```
Require: Lattice $\mathcal{L}$, BSs $\mathcal{G} \cup \mathcal{D}$, users $\mathcal{U}$, Signal parameters.
1: Randomly place all $d$ at $\Pi^d \in \mathcal{L}$. Define $\Pi = \{\Pi^d\}_{d \in \mathcal{D}}$.
2: Compute number $U_d^{'}(\Pi)$ of UEs covered by all $g \in \mathcal{G}$ and $d \in \mathcal{D}$ $\big($If $\frac{1}{D_g} W_{\mathcal{B}} \log_2\big(1+\Gamma_{g,d}^{\mathcal{B}}\big) < R_{\min}^{\mathcal{B}}, \forall g \in \mathcal{G}$, then $U_d^{'}(\Pi) = 0\big)$.
3: Select $d_0 = \arg\min_{d \in \mathcal{D}}\{U_d^{'}(\Pi)\}$.
4: Take $\Pi^{d_0} = \arg\max_{\pi \in \mathcal{L}}\{\sharp$ of UEs covered if $d_0$ is at $\pi \mid \exists g \in \mathcal{G} : \frac{1}{D_g} W_{\mathcal{B}} \log_2\big(1+\Gamma_{g,d_0}^{\mathcal{B}}\big) \geq R_{\min}^{\mathcal{B}}$ in $\pi\}$.
5: If the same coverage remains when placing $d_0$ at $\Pi^{d_0}$, go back to step 3 ignoring unsuccessful $d_0$'s.
6: Place $d_0$ at $\Pi^{d_0}$ and set $\Pi \leftarrow \{\Pi^d\}_{d \in \mathcal{D}}$.
7: Go back to step 2 until:
     • Ground-plus-air coverage is no longer improved.
     • Maximum number of iterations $i_0$ is reached.
```

$d_0 \in \mathcal{D}$ is selected and repositioned. Such drone d_0 selects the 3–D position Π^{d_0} in the lattice $\mathcal{L}$ (see Eq. (13)) at which *gNB*s and *ABS*s cover more users as long as there exists $g \in \mathcal{G}$ providing the guaranteed backhaul QoS in that position. A user u is covered by an *ABS* d only if the user rate experienced is greater than minimal user access rate $R_{\min}^{\mathcal{A}}$ and d is covering at most U_d users, so that u enjoys the guaranteed QoS. Thus, the signal strength from d_0 and from the rest of the drones in $\mathcal{D} \backslash \{d_0\}$ must be checked. This means that the complexity of each iteration is $\mathcal{O}(|\mathcal{L}| \cdot D \cdot U)$, where $\mathcal{L}$ is the size of the lattice. Since i_0 is the maximum number of iterations needed for the algorithm to converge to a solution, the complexity of OnDrone is $\mathcal{O}(i_0 \cdot |\mathcal{L}| \cdot D \cdot U)$, where i_0 is constant and can be omitted.

We remark that, unlike the NP-Complete problem presented in the previous section, OnDrone requires few iterations. Indeed, we have performed all the experiments in this paper in few seconds in a personal computer, on average. As a matter of fact, OnDrone is intended to be used in an on-demand fashion, dynamically repositioning drones to adapt to user's moves over time. OnDrone is then practical and can be executed at the network orchestrator.

5.1.2 *Seq: Sequential Multi-Placement*

In addition to OnDrone, we have also developed a simple heuristic that finds feasible solutions to the Coverage Problem $\mathcal{C}$ in polynomial time. We base this algorithm on the Efficient 3-D Placement—hence the name of the algorithm—scheme derived in [9], and thus we adapt it to Sequential Multi-Placement (Seq) in order to support aerial networks with more than one *ABS*. In [9], the authors model the presence of one drone providing coverage in one single cell, i.e., they maximize coverage for single drone missions. They model a circular drone footprint, which allows to formulate a convex optimization problem due to the presence of *only* one drone. We adapt the proposal in [9] to place *ABS*s one by one, according to realistic interference metrics *not originally considered* in that work. Seq will be used as a benchmark for OnDrone.

We build the Seq heuristic by induction as follows: Since the framework proposed in [9] only considers drone-coverage, we first compute the amount of users covered by ground-cells. Second, we select one *ABS* and maximize coverage as described in [9] (i.e., using Efficient 3-D Placement) for the remaining non-covered users, namely $\mathcal{U}^{'}$. Here, the placement space for drones is restricted to those positions where at least one *gNB* provides backhaul connectivity with the guaranteed QoS. We denote as $\mathcal{U}_1$ to the set of users that are covered by the first selected *ABS*. For this first *ABS*, there are no interference issues.

Let $i > 1$ and assume that we have located $i-1$ *ABS*s and that we want to locate the i-th *ABS*. Assume that $\mathcal{U}_{i_k}$ are the sets of UEs that each previous i_k-th *ABS* covers at the moment of its placement. Then, Seq finds a 3–D position at which the i-th *ABS* covers more users from the set $\mathcal{U}^{'} \backslash \bigcup_{1 \leq i_k < i} \mathcal{U}_{i_k}$ and at least one *gNB* provides the requested backhaul QoS. Thus, the i-th *ABS* aims to cover the maximum number of users that are not covered yet.

Seq ends when the D-th *ABS* is located. After this, the algorithm computes the actual number of users served according to interference (in both the backhaul and access network). Hence, Seq has the same objective as OnDrone: covering the maximum number of users according to QoS guarantees. We report the pseudocode in Algorithm 2.

At each iteration $i > 1$, the i-th *ABS* sequentially selects the best position for it based on Efficient 3-D Placement from [9] maximizing its own coverage, no matter how its position affects the coverage of the remaining *ABS*s or the already set backhaul links. Since placing the new i-th *ABS* adds interference to the system, the previous $i-1$ drone-cells shrink, and cover less UEs than the ones originally intended by the Seq choice.

Complexity. The complexity of Seq is evaluated as follows. Seq makes D steps, one per each drone. At each step d, Seq defines the final position of drone $d \in \mathcal{D}$. In [9], the authors do not propose any algorithm for placing the drone, but they solve a convex mixed-integer non-linear program with a convex optimizer. Such optimizer does not run an algorithm with polynomial-time complexity. Instead, it performs a combination of interior-point methods with a branch&bound search [39]. Hence, we opt for approximating their problem through a search on the lattice $\mathcal{L}$. As in OnDrone, checking whether a user is covered requires to check the signal strength of the serving drone along with the interfering drones, i.e., D signal strengths. The complexity of this process is $\mathcal{O}(|\mathcal{L}| \cdot D \cdot U)$. After the last

Algorithm 2 Seq: Sequential Multi-Placement

```
Require: BSs $\mathcal{G} \cup \mathcal{D}$, users $\mathcal{U}$, and Signal parameters.
Compute the number of users covered by ground-cells.
Find $\Pi^{d_1} \in \mathbb{R}^3$ where $d_1$ covers more users from $\mathcal{U}^{'}$ and $\exists g \in \mathcal{G} \mid \frac{1}{D_g} W_{\mathcal{B}} \log_2\big(1+\Gamma_{g,d_1}^{\mathcal{B}}\big) \geq R_{\min}^{\mathcal{B}}$.
Define the set of users covered by $d_1$: $\mathcal{U}_1 \subseteq \mathcal{U}^{'}$.
for $2 \leq i \leq D$
    Find $\Pi^{d_i} \in \mathbb{R}^3$ where $d_i$ covers more UEs in $\mathcal{U}^{'} \backslash \bigcup_{1 \leq i_k < i} \mathcal{U}_{i_k}$ such that $\exists g \in \mathcal{G} \mid \frac{1}{D_g} W_{\mathcal{B}} \log_2\big(1+\Gamma_{g,d_i}^{\mathcal{B}}\big) \geq R_{\min}^{\mathcal{B}}$ in $\Pi^{d_i}$.
    Define the set of UEs served by $d_i$: $\mathcal{U}_i \subseteq \mathcal{U}^{'} \setminus \bigcup_{1 \leq i_k < i} \mathcal{U}_{i_k}$.
end for
Derive the actual covered UEs according to interference.
```

iteration, the algorithm checks which users are actually covered, since those users that at some iteration i were covered, may no longer enjoy the guaranteed QoS because of the repositioning of other drones in successive iterations. This check has a complexity of $\mathcal{O}(D^2 \cdot U)$. Thus, the complexity of the `Sequential Multi-Placement` algorithm is $\mathcal{O}(|\mathcal{L}| \cdot D \cdot U + D^2 \cdot U)$, that is similar to the complexity of `OnDrone` because $|\mathcal{L}| \geq D$, since drones cannot be co-located and so the number of possible distinct drone positions cannot be smaller than the number of drones (indeed, in a well designed system, $|\mathcal{L}| \gg D$).

5.1.3 `RA`: a `Repulsion-Attraction` scheme

Here we briefly describe the `Repulsion-Attraction` (`RA`) scheme derived in [10]. `RA` is a multi-drone placement scheme in which several self-organized *ABS*s dynamically change their position to track clusters of users. The approach is based on the assumption that *ABS*s will be attracted by the presence of users in the ground, and will be repulsed by *gNB*s and other *ABS*s in order to avoid interference.

Complexity. `RA` consists into maximizing a non-integer metric without any constraints. Hence, one can apply standard methods like line-search or trust-regions methods for unconstrained optimization. Such methods have low complexity and their performance depends on the target tolerance on the error. Moreover they converge really quickly [39], with a linear dependance on the number of iterations $i_{\mathtt{RA}}$, i.e., the complexity is $\mathcal{O}(i_{\mathtt{RA}})$.

5.2 *Bézier* flight routes

The last problem to solve consists in designing drone's trajectories. The output of `OnDrone` (Section 5.1), and the *Hungarian method* (Section 4.2), provide the source and destination for each drone carrying an *ABS*. Therefore, we now design a route optimization scheme.

While drones fly, both the backhaul and users association change. Initially, each *ABS* attaches to a *gNB* with the required QoS and starts the flight. Once the backhaul QoS level is low due to long distance or interference impairment, the *ABS* sets a backhaul link with a new *gNB* with the guaranteed QoS. Similarly, while a drone flies, UEs attach to that *ABS* in case that the minimum access data link rate (i.e., QoS) is guaranteed. Upon arrival to the destination, the association is already optimal in terms of coverage.

On the one hand, drones have high aerial mobility and fly over a ground cellular network in a 3–D space, without many restrictions of walls, streets or vehicles. On the other hand, drones hovering over regions with good QoS from some *gNB*s or under-populated regions with a big surface may lead to under-utilized *ABS*s and low coverage, depending on the topology of users. Furthermore, if a drone is not fast enough, it might occur that when the drone arrives at the destination the users topology has changed too much so the destination is no longer optimal, and the network needs to be re-optimized. To avoid such undesired effects, and knowing that drones may have to be redirected while flying towards a destination, we propose drone paths following *Bézier curves* [8], instead of commonly assumed straight lines, as adopted in [11]. Indeed, using *Bézier curves* allows to deflect drone trajectories towards areas with higher user density, so to enhance drone coverage and enable unique coverage opportunities *while* drones seek their optimal position. Since we leverage on the notion of *Bézier curve*, Appendix B provides some background on the subject.

Algorithm 3 *Bézier Scheme*

```
Require: $\mathcal{P}_0 = \{\pi^d, \Pi^d\}$, $\omega$, $B$, $\tau = (1+\alpha) \cdot \lambda\left(\beta^{\mathcal{P}_0}(t)\right)$.
$\mathcal{U}^{\mathcal{P}_0} = \mathcal{U} \cap \mathcal{S}^{\omega}\left(\beta^{\mathcal{P}_0}(t)\right)$.
$k = 0$.
while $|\mathcal{P}_k| < B$ & $\lambda\left(\beta^{\mathcal{P}_k}(t)\right) < \tau$, do:
    for $u \in \mathcal{U}^{\mathcal{P}_k}$, $g_u^{\mathcal{P}_k} = |\{u' \in \mathcal{U}^{\mathcal{P}_k} \mid \|u' - u\| \leq \omega/2\}| / \left(\pi\omega^2/4\right)$.
    $u_{k+1} = \arg\max_{u \in \mathcal{U}^{\mathcal{P}_k}} \left\{ g_u^{\mathcal{P}_k} \mid \lambda\left(\beta^{\mathcal{P}_k \cup \{u\}}(t)\right) \leq \tau \right\}$.
    $\mathcal{P}_{k+1} = \mathcal{P}_k \cup \{u_{k+1}\}$,  $\mathcal{U}^{\mathcal{P}_{k+1}} = \mathcal{U}^{\mathcal{P}_k} \cap \mathcal{S}^{\omega}\left(\beta^{\mathcal{P}_{k+1}}(t)\right)$.
    $k = k + 1$.
end while
$\mathcal{P} = \mathcal{P}_{k-1}$.
```

In our proposal, we define the set of anchor points for our *Bézier*-based flight path and use the standard *de Casteljau* algorithm [8] to derive the *Bézier curve* corresponding to the selected anchor points (see Appendix B for details).

We obtain the set of anchor points inductively, as detailed next. Let π^d and Π^d be source and destination of a drone $d \in \mathcal{D}$, let $\omega > 0$ be the width for the two-sided offset region of the curves and let $B > 1$ be the maximum number of anchor points for the *Bézier curve*. B is determined as the density of users covered per drone-cell, since in case a drone cannot cover more than B users, it does not make sense that such drone wishes to deflect its path attracted by more than B users. We take $\omega = 2R_d$, where R_d is the maximum range at which drone d can provide coverage in its optimal position. We define as the initial set of anchor points $\mathcal{P}_0$ both nodes: $\mathcal{P}_0 = \{\pi^d, \Pi^d\}$. Thus, we define the *Bézier curve* $\beta^{\mathcal{P}_0}(t)$ for $\mathcal{P}_0$, which is the segment joining π^d and Π^d, computed with the *de Casteljau* algorithm [8]. Now, we iteratively modify the current *Bézier curve* until we derive the final *Bézier* path. Given a curve $\beta(t)$, we denote its length as $\lambda(\beta(t))$, and take $\lambda\left(\beta^{\mathcal{P}_0}(t)\right)$ as a reference length for the final *Bézier curve*. Indeed, we build a *Bézier Scheme* such that the obtained curve is not longer than $\tau = (1+\alpha) \cdot \lambda\left(\beta^{\mathcal{P}_0}(t)\right)$, for a given $\alpha > 0$. α is determined as the fraction of the time interval in which a drone would have already arrived to its destination in case of following a straight path. Hence, we make sure that a drone reaches its destination but has the flexibility to deflect its path to improve coverage. Let $\mathcal{S}^{\omega}(\beta(t))$ be the two-sided offset region of width ω that results from stroking a curve $\beta(t)$, and let $\mathcal{U}^{\mathcal{P}_0} = \mathcal{S}^{\omega}\left(\beta^{\mathcal{P}_0}(t)\right) \cap \mathcal{U}$ be the set of users that are inside the offset region of $\beta^{\mathcal{P}_0}(t)$. We denote by $g_u^{\mathcal{P}_0}$ the gravity of a user $u \in \mathcal{U}^{\mathcal{P}_0}$ and define it as the density of users in a disk centered in u with radius $\omega/2$:

$$g_u^{\mathcal{P}_0} = \left|\{u' \in \mathcal{U}^{\mathcal{P}_0} \mid \|u - u'\| \leq \omega/2\}\right| \Big/ \left(\pi\omega^2/4\right). \tag{14}$$

The first user u_1 selected as anchor point is the one with highest gravity in $\mathcal{U}^{\mathcal{P}_0}$ such that the resulting *Bézier curve* is not longer than $\tau = (1+\alpha) \cdot \lambda\left(\beta^{\mathcal{P}_0}(t)\right)$, i.e.:

$$u_1 = \arg\max_{u \in \mathcal{U}^{\mathcal{P}_0}} \left\{ g_u^{\mathcal{P}_0} \mid \lambda\left(\beta^{\mathcal{P}_0 \cup \{u\}}(t)\right) \leq \tau \right\}. \tag{15}$$

Now, $\mathcal{P}_1 = \mathcal{P}_0 \cup \{u_1\}$ defines a new *Bézier curve*, $\beta^{\mathcal{P}_1}(t)$. The sorting of positions in the sets $\mathcal{P}_k$ is important, since

Table 3
Summary of Algorithms' Complexity

Algorithm	*Complexity*
Optimal Aerial Coverage (Section 4.1)	NP-Complete
Assignment of Fleet Destinations (Section 4.2)	$\mathcal{O}(D^3)$
OnDrone (Section 5.1.1)	$\mathcal{O}(\lvert\mathcal{L}\rvert \cdot D \cdot U)$
Seq (Section 5.1.2)	$\mathcal{O}(\lvert\mathcal{L}\rvert \cdot D \cdot U + D^2 \cdot U)$
RA (Section 5.1.3)	$\mathcal{O}(i_{\mathtt{RA}})$
Bézier Scheme (Section 5.2)	$\mathcal{O}(B^3)$
Overall Optimization Framework with OnDrone (Section 5.3)	$\boldsymbol{\mathcal{O}(\lvert\mathcal{L}\rvert \cdot D \cdot U) + \mathcal{O}(D^3)}$

each order defines a different curve. Thus, we sort the points in increasing order according to the distance to the source, π^d. Finally, $\mathcal{U}^{\mathcal{P}_1} = \mathcal{U}^{\mathcal{P}_0} \cap \mathcal{S}^{\omega}(\beta^{\mathcal{P}_1}(t))$ is the updated set of users for next iteration. Then, we keep selecting anchor points for the *Bézier curve* while they exist, until the length of the *Bézier curve* exceeds τ or the maximum number B of anchor points is reached. At each iteration k, we build a new curve from $\mathcal{P}_{k-1}$, $\mathcal{U}^{\mathcal{P}_{k-1}}$ and $\beta^{\mathcal{P}_{k-1}}(t)$.

Complexity. We show the derived *Bézier Scheme* in Algorithm 3. Its complexity can be evaluated as follows. There is an iteration of the *de Casteljau* algorithm to derive each *Bézier curve* $\beta^{\mathcal{P}_k}(t)$. The complexity of the *de Casteljau* algorithm is quadratic with the number of anchor points of the *Bézier curve* that it builds [8]. The *Bézier Scheme* runs at most B iterations, and at each iteration k it uses $k+2 = \lvert\mathcal{P}_k\rvert \leq B$ anchor points. Thus, the complexity of the *Bézier Scheme* is $\mathcal{O}(B^3)$, where B is the maximum number of anchor points allowed.

5.3 Overall complexity

In Figure 5 we show the flow chart of the proposed optimization framework for drone-aided dynamic cellular networks. As we have fully detailed in the paper, first we solve, through OnDrone, the framework for maximizing network coverage in polynomial time. OnDrone outputs the optimal 3–D positions at which drones must locate, given their previous positions. However, drones do not move instantaneously. The *Hungarian method* solves then the assignment problem that minimizes the distance flown in polynomial time, and outputs the source and destination of each drone concretely. Finally, our *Bézier Scheme* designs the flight routes of drones in order to deflect their paths towards clusters of users and increase the coverage efficiency over the time interval. This is another polynomial time algorithm. Therefore, the overall optimization framework runs in polynomial time. The overall complexity of our framework for fast repositioning of drone-aided cells is the result of summing the complexities of the schemes that compose such framework, which is shown in Figure 5. When the 3–D placement optimization is performed by OnDrone, the complexity is $\mathcal{O}(\lvert\mathcal{L}\rvert \cdot D \cdot U) + \mathcal{O}(D^3) + \mathcal{O}(B^3) = \mathcal{O}(\lvert\mathcal{L}\rvert \cdot D \cdot U) + \mathcal{O}(B^3)$, since $\lvert\mathcal{L}\rvert \gg U \gg D$.

In Table 3 we summarize the complexities of the detailed algorithms, including the *overall optimization framework*.

5.4 Orchestration of the optimization framework

The possibility to integrate the above described framework into a communication system raises several alternatives regarding where the different algorithms can be executed. First, there is the issue of deciding where the drones must be located at each time instant. This task (i.e., to execute OnDrone for drone placement) should be assigned to a device on which all drones have direct communication and therefore can easily know what must be their destination. Therefore, the *gNB*s seem to be an appropriate place for the orchestration of drones' positions. In alternative, and considering current trends in 5G networks architecture design, the orchestrator can be a software slice in the MEC, which is the edge-cloud computing platform of future 5G cellular networks and which resides just next to base stations [40].

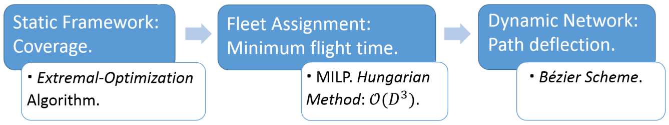

Static Framework: Coverage.
• Extremal-Optimization Algorithm.
Fleet Assignment: Minimum flight time.
• MILP. Hungarian Method: $\mathcal{O}(D^3)$.
Dynamic Network: Path deflection.
• Bézier Scheme.

Figure 5. Flow chart of the drone-aided dynamic network.

The second issue consists in designing drone's trajectories, provided it must change its current location. However, the *Bézier curves* used for the flight routes have already designed in a discretized manner with the *de Casteljau* algorithm, using short straight segments to build a *Bézier curve*. At this point, we note that currently, most drones are already capable of autonomously travel to concrete positions following straight lines (see for instance [41]). So, drones can follow such short segments (using their originally integrated traveling mechanisms) without significantly deviating from the flight route provided by the *Bézier scheme*. In that case, those responsible for such tasks (i.e., obtain the discretized *Bézier curves* and follow the corresponding straight segments) are the drones themselves.

6 Experimental Results

Here we numerically assess the performance of our multi-drone optimization framework. We assess the coverage offered by the network, with the assistance of a fleet of drones, in a circular surface. We compare optimal placement results yielded by OnDrone (presented in Section 5.1.1) with the ones obtained with Seq (based on [9] and described in Section 5.1.2) and with the RA scheme (from [10], and described in Section 5.1.3). We also compare the results with the optimal achieved by means of Monte Carlo simulations, since computing exact optima is not doable in networks with as few as a fistful of UEs, *gNB*s, and drones. Besides, we compare our scheme to a modified OnDrone that neglects interference (referred as "iNeg" in the figures). Hence, we assess the importance of introducing interference in the analysis. We also compare coverage results while drones reposition following the *Bézier Scheme* (presented in Section 5.2) with a simpler *Straight Scheme*, as adopted in [11], which consists in moving drones over straight paths towards a certain destination identified with the *Hungarian method* (see Section 4.2). Since RA has been designed in [10] to dynamically track UEs, we compare also the *Bézier* and *Straight Schemes* when the path planning is based on RA.

We mainly study the placement and repositioning of drones in synthetic and realistic scenarios, and the effects of interference and LoS on user coverage over time, under four classes of environmental scenarios: *suburban*, *urban*,

Table 4
Environmental parameters for the computation of LoS probability

Environment	***suburban***	***urban***	***dense***	***high-rise***
ξ_{LoS} [dB]	0.1	1	1.6	2.3
ξ_{NLoS} [dB]	21	20	23	34
a	4.88	9.61	12.08	27.23
b	0.43	0.16	0.11	0.08

dense and *high-rise*. These four environments correspond to different densities of elements (e.g., buildings) that affect the LoS probability. Moreover we study three distinct cases of deployment scenarios, in which the location of users follows different distributions:

- *PPP*: We place UEs in a circular surface, according to a Poisson point process.
- *Cheese*: We define a surface that includes certain regions which are not accessible to UEs, and locate UEs uniformly random in the rest of the network. Then, we have a surface with empty areas (like in a Swiss cheese). This users' distribution is typical of public gardens or areas with restricted zones.
- *Capital*: We also run our numerical evaluation over a simplified map of the center of a dense capital city, Madrid, considering main zones of people affluence and no users in indoor installations.

In the *PPP* and *Cheese* scenarios, we locate *gNB*s according to the same distribution. Heterogeneous synthetic distributions would be of interest for traffic demand-based optimizations [42], which is out of the scope of this paper, and we therefore do not consider them. However, in the *Capital* scenario, we consider the actual locations of those *gNB*s that belong to the main network operator in the city. In all experiments, the number of iterations required by `OnDrone` was very small (see Appendix C.1).

Table 4 gathers the parameters for the air-to-ground path-loss model and the probability of LoS described in Eqs. (1) and (5), depending on the density of the environment that we consider in the numerical simulations. These parameters are obtained based on the number of buildings and large signal obstructions per unit area, building's height distribution, ratio of built-up area and clean surfaces, etc., as it has been derived in [18], based on the ITU recommendations [33]. Such parameters allow to differentiate the main four environmental conditions.

Table 5 gathers the parameters that, unless otherwise specified, we have used for the network model, regardless of the simulation environment. We take a circular surface of $R_C = 1.5$ km of radius where there are 10 *gNB*s, and a height range between $h_{\min} = 60$ m and $h_{\max} = 600$ m for drones. This is a generally doable height range,[4] since lower values would be too close to ground (and, e.g., vehicles or even people) and higher elevation would be affected by high-speed winds which are unsafe for an aerial network of simple drones. However, in our numerical evaluation the actual maximum drone altitude is rather determined by the environment density. For instance, in a *high-rise* environment, although high altitudes increase the probability of LoS, the attenuation is much stronger, so that drones need to fly closer to the ground. In contrast, the *suburban* or *urban* environments do not suffer strong attenuation, so that drones can fly higher. However, a high altitude turns into links with higher LoS probability, thus yielding more interference for far ground users.

The power transmission from *ABS*s is 10 dBm, as adopted in [43], [44], [45], which is notably lower than the usual 44 dBm used for *gNB*s in the ground network (as we adopt). This is because *ABS*s have much higher LoS probability than ground base stations and do not use omnidirectional antennas, and hence require much less power. Using higher *ABS* transmission power, as 25 or 44 dBm, may provide better coverage due to better signal strength, although also provides less resilience to interference impairment, as we discuss in our results. Moreover, *ABS*s are carried by flying drones, which spend most of their energy into hovering, flying towards desired positions at a given speed, and carrying the weight of the communication equipment. This poses serious constraints on transmission power, as evaluated in [43]. Hence, we have chosen to use a 10 dBm of power transmission for analyzing our framework and algorithms. The guaranteed user data rate is 0.72 Mbps, which guarantees video streaming with 240p resolution [46], and allows 360p and 480p resolution in many cases over the MPEG-4 standard. It also allows adequate videoconferencing quality using video compression [47]. Such guaranteed rate, with customary 20 MHz bands and assuming that no more than 100 users can attach to a BS, corresponds to guaranteeing that the SINR is higher than $\gamma_{\mathcal{A}} = 10.9$ dB, according to the Shannon capacity formula (see Appendix C.2 for a discussion on the minimum data rate experienced under coverage). Besides, such SINR value allows for a 16QAM modulation and a coding rate of 1/2 in LTE communications, as derived in [48], although we also study the impact of using other guaranteed rates. The carrier frequency used by *gNB*s (either for backhaul channels between *gNB*s and *ABS*s or access channels between *gNB*s and UEs) and for the access channel between *ABS*s and users are 1815.1 MHz and 2630 MHz, respectively. These channel parameters are as in the LTE/LTE-A network provided by Movistar in Madrid. As it has been discussed in [31], the antenna patterns of *gNB*s for backhaul connectivity are directional with a HPBW of 65 degrees, according to the 3GPP technical specifications [49]. For dynamic experiments, we slot time into intervals of length $T = 60$ s. This means that every minute, we re-optimize the network by means of `OnDrone` (or with any of the benchmark schemes, as `Seq` or `RA` schemes) and immediately re-direct drone flights for dynamic repositioning using the solution of our assignment problem and a flight route computed with either the *Bézier* or the basic *Straight Scheme* from [11]. Users move according to the random way-point model (RWP)[5] (see [50], [51]) with an average speed of 2 m/s. We update user's positions every second, while drones fly at a constant speed of 15 m/s over a continuous path. Moreover, such flight speed allows for low drone energy consumption, according to the energy consumption model for aerial aircrafts derived in [43].

[4] Such under-kilometer altitudes comply with current possibilities of commercial drones. For instance, DJI Phantom 4 has an elevation range of few thousands of meters, according to its commercial specifications.

[5] The RWP model is one of the most studied and used mobility models to assess mobile networks. It is simple and has wide availability.

Table 5
System and simulation parameters

Parameter	***Value***
Circular surface radius R_C	1500 m
Height range, $[h_{min}, h_{max}]$	$[60, 600]$ m
gNB, *ABS* Tx power, P^d_{Tx}, P^g_{Tx}	10 dBm, 44 dBm
Users guaranteed QoS	0.72 Mbps
*gNB*s Carrier frequency, $f_{\mathcal{B}}$	1815.1 MHz
*ABS*s Carrier frequency, $f_{\mathcal{A}}$	2.63 GHz
gNB, *ABS* Bandwidths	20 MHz
Thermal Noise Power	-174 dBm/Hz
HPBW of *gNB*s	65 degrees
Time interval length T	60 secs
UEs, Drones speed	2 m/s, 15 m/s
Lattice dimensions N_ρ, M_θ, H	20, 40, 40
Monte Carlo runs per instance	10^7
Instances of simulations	1000

We use MATLAB R2018a to simulate channel conditions and mobility of drones and UEs. For small networks, we have performed Monte Carlo simulations to search for the optimal drone placement. For each network instance, we have run 10^7 random positions for the fleet and taken the settings for the best coverage. Such simulations are very time-consuming even for small networks, yet we have observed that such number of Monte Carlo runs per instance provides a coverage output that is hard to improve, since the output placement provided remains invariable for a large number of additional runs, as we have preliminary tested. We have simulated every scenario 1000 times in order to derive the statistics shown in the figures.

As mentioned earlier, the high complexity of the exact solution of the Coverage Problem $\mathcal{C}$, allows us to find optima only for small instances of the problem, with a reduced number of *gNB*s, *ABS*s and UEs. Instead, as further discussed in Appendix C.1, OnDrone only requires a few iterations to converge. Thus, for realistically larger deployments, we only show the results obtained with OnDrone, and compare the results to what achieved by Seq based on [9], the RA scheme [10], and a modified OnDrone scheme that neglects inter-*ABS* interference ("iNeg" in the figures). We have evaluated the coverage performance with denser lattices in OnDrone and observed that the results cannot be significantly improved (since they are already close to optimal, as shown in the comparison with the optimal placements), while imposing more computational complexity. We also evaluate the impact of *Bézier* routes vs. straight flying routes [11]. The flight assignment problem is solved optimally and efficiently by the *Hungarian method* [38], so we do not comment on its performance. At the end of the section we also provide a summary of the lessons learnt from our performance evaluation study.

6.1 Coverage optimization

Here we numerically evaluate the coverage performance at a precise time instant. We pictorially show the drone footprints and indicate the altitude, computed with the different schemes that target coverage maximization, namely with the optimum (i.e., with its approximation obtained by means of Monte Carlo simulations), with OnDrone, and with Seq. With OnDrone, we clearly see in Figure 6 that *ABS*s are positioned very close to the optimal positions, while in Figure 7 we see that the Seq scheme locates drones in much distinct positions (as well as the RA scheme, not shown here to keep the figure clear). Indeed, Seq makes a greedy decision at each step for a given drone. Hence, Seq finds a good position for such drone knowing the interference incurred by the previously located drones. However, the additional interference from drones located afterwards is ignored, which results on shrunk coverage areas as the final performance. Also, the altitudes provided by Monte Carlo simulations and OnDrone are only slightly different. In general, *ABS*s avoid locations already covered by *gNB*s.

Another important aspect to pay attention to is the shape of the drone footprints. Unlike in currently used models [3], [21], [22], [27], the area served by a drone is not circular, due to inter-drone interference, which is not considered in the mentioned works. In Figure 8, we show the average number of covered users in a network with 100 users distributed in the ground surface according to a *PPP*, and different fleet sizes. Solid bars represent the total amount of covered users, while stripped bars represent users covered by *ABS*s. The figure shows that OnDrone approximates well the optimal solution (within a mere 1% from reaching the optimal) both to total coverage and *ABS* coverage, while neglecting interference leads to very inaccurate results, getting worse as the fleet size increases (and hence backhaul and inter-drone interference increases). In the figure, we also see that Seq only covers around 80% of the optimal coverage, depending on the fleet size. RA is not even able to outperform the coverage results from Seq, and its performance soon decays due to interference issues in the presence of as few as 5 *ABS*s. This shows that although it is practical, the intuition behind RA is not accurate enough for optimizing coverage.

In Figure 9 we further compare the coverage achieved when considering the four reference environments of Table 4, for the same *PPP* case discussed above. The LoS likelihood between *ABS*s and UEs decreases in denser environments. Thus, drone-cells shrink, and *ABS*s cover less users. Indeed, the figure shows a factor ∼6 between the *ABS* coverage achievable in *suburban* and *high-rise* environments, and also that the ground network handles higher total coverage percentages as the environment grows denser, at least for small fleet sizes. Here, we again see the high accuracy of OnDrone in comparison to the optimum searched by means of Monte Carlo simulations, in both total and *ABS* coverage. As before, Seq and RA provide coverage noticeably lower. We also see that neglecting interference is very counter-productive for *ABS* coverage in low dense environments, since there is higher LoS probability and links are attenuated easily from interfering signals. The opposite behavior is observed with RA, which provides reasonable results for *suburban* scenarios (better than Seq) but then its performance decays with denser environments (and RA becomes the worst scheme). Results derived in the *Cheese* and *Capital* scenarios are qualitatively similar to those discussed above for the *PPP* scenario. Thus, we omit the results here.

To give a performance sample of OnDrone for larger fleets, Figures 10 and 11 show the placement of 8 and 6 *ABS*s in a *dense Cheese* and *Capital* scenario, respectively. Here, drones tend to follow UEs distribution, avoiding empty areas and regions covered by *gNB*s. Indeed, OnDrone avoids also overlapping drone-cells, thus incurring low inter-*ABS* interference and being able to cover more users.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2927335, IEEE Transactions on Mobile Computing

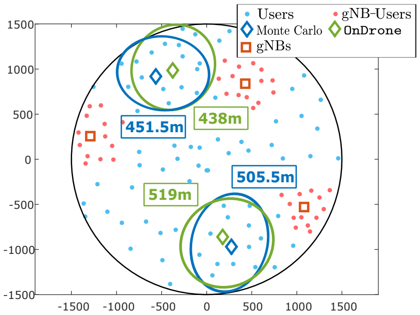


Figure 6. Drone 3–D placement. $D = 2$, $U = 100$. Scenario: *urban, PPP*.

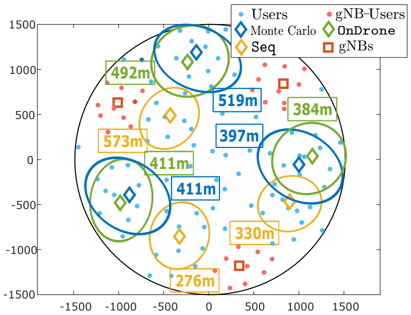


Figure 7. Drone 3–D placement. $D = 3$, $U = 100$. Scenario: *dense, PPP*.

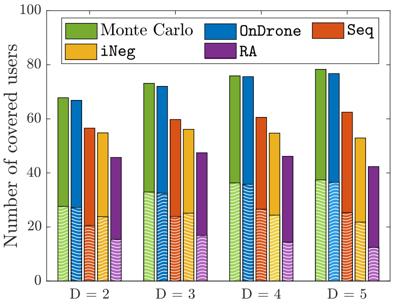


Figure 8. Comparison of algorithms on total coverage (solid bars) and *ABS* coverage (stripped bars). U=100. Scenario: *dense, PPP*.

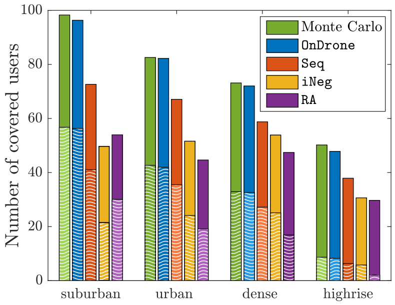


Figure 9. Impact of environment on total coverage (solid bars) and *ABS* coverage (stripped bars) coverage. $D = 3$, $U = 100$. Scenario: *PPP*.

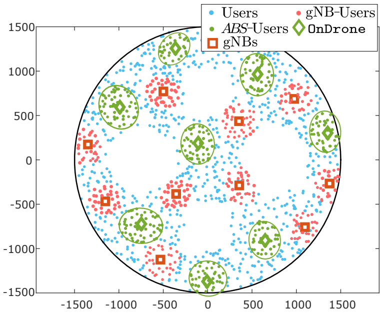


Figure 10. Drone 3–D placement. $D = 8$, $U = 1000$. Scenario: *dense, Cheese*.

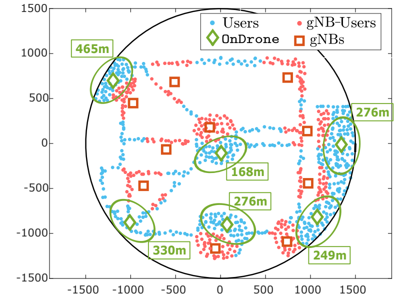
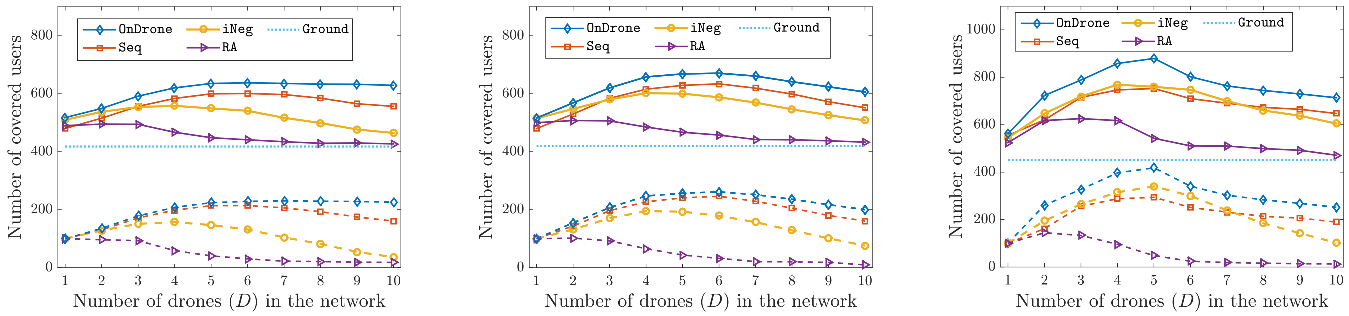


Figure 11. Drone 3–D placement. $D = 6$, $U = 1000$. Scenario: *dense, Capital*.

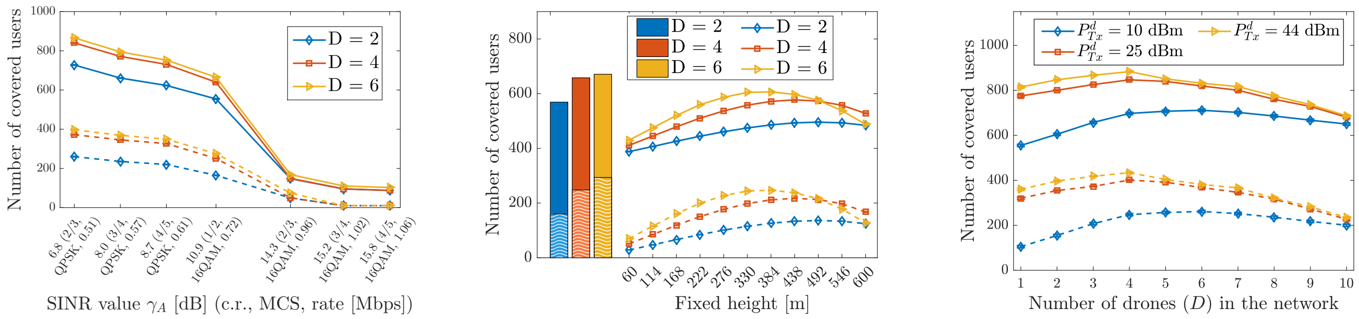


(a) Scenario: *dense, PPP*.

(b) Scenario: *dense, Cheese*.

(c) Scenario: *dense, Capital*.

Figure 12. Total coverage (solid lines) and *ABS* coverage (dashed lines) for $U = 1000$ UEs.



(a) Impact of SINR value γ_A in total coverage (solid lines) and *ABS* coverage (dashed lines). The x axis indicates SINR values in dB, and then, in between parentheses, the corresponding coding rate (c.r.), MCS, and guaranteed user data rate (in Mbps) with a 20 MHz channel and with at most 100 users per BS.

(b) Impact of fixed height h^d in total coverage (solid bars and solid lines) and *ABS* coverage (stripped bars and dashed lines). The bars show coverage with adaptive height.

(c) Impact of drone transmission power P^d_{Tx} in total coverage (solid lines) and *ABS* coverage (dashed lines) for $D = 2, ..., 10$.

Figure 13. Study of tunable network parameters: Guaranteed bandwidth, fixed drone height h^d, and drone transmission power P^d_{Tx}. $U = 1000$ UEs. Scenario: *dense, Cheese*.

In Figure 12 we further study the impact of the fleet size on coverage, with $U = 1000$ users, for each scenario. Here, solid lines represent total coverage and dashed lines report the portion of users that would be served by *ABS*s. We also show coverage performance in absence of drones, labelled as Ground in the figures. First, for the *PPP* scenario in Figure 12(a), adding more drones increases total and *ABS* coverage because there are more *ABS*s that can cover larger areas with limited interference. Each analyzed scheme behaves significantly different. When the fleet size becomes larger than $D = 7$, the coverage remains stable or even slightly decays for OnDrone. Here, we notice that the larger the fleet is, the more interference issues appear in both, the backhaul and the access network. OnDrone is able to maintain a stable coverage by reallocating positions to drones to have good backhaul connectivity while providing stable and wide coverage to users, thanks to the design based on *extremal-optimization*. However, Seq does not have a design that allows a reconfiguration of the aerial network. Hence, it suffers more from interference in the access and backhaul sides. The figure also shows that neglecting interference leads to very poor coverage, as well as with the RA scheme. In both cases, adding drones becomes soon counter-productive. While we have adapted in Seq the scheme proposed in [9] to account for interference, the RA scheme proposed in [10] does not target any specific interference metric in the computation of its *repulsion* component. This explains why RA behaves poorly with as few as 4 or more *ABS*s in a *dense* scenario. Instead, as we have checked numerically, when the interference is neglected—or approximated by a constant value—the coverage apparently never stops increasing with the fleet size. In fact, without interference, having more drones implies covering more non-interfering drone-cells. However, in reality, it happens that *there is an exact number of drones that maximizes coverage, depending on the environment*.

In Figures 12(b) and 12(c) we show the same type of graph for the *Cheese* and *Capital* deployment, respectively. The *Cheese* case confirms that OnDrone is a good option for irregular deployments, in which the area to cover in the ground surface is smaller than the complete circular surface, so drone positions are more packed. Here, we observe more clearly the effect that increasing the fleet size is not beneficial for coverage purposes, since the surface $\mathcal{S}$ is irregular, so that users are more packed and hence it is harder for *ABS*s to fully avoid interference. Indeed, Seq is not able to avoid interference as good as OnDrone, and suffers larger performance drops with more than 5 drones, as well as the scheme neglecting interference. RA appears to be able to manage interference issues, although with no good coverage results overall. In the *Capital* case, no matter the adopted scheme, once the optimum number of *ABS*s is reached, it is hard to add more drones without incurring interference, although in this scenario Ondrone substantially overcomes the rest of schemes. This is due to the fact that users are concentrated in relatively small areas and nearby drones can interfere large masses of users. In any case, OnDrone largely outperforms any other benchmarking schemes also in realistic networks as the one extracted from a dense capital city (Madrid).

So far we analyzed our framework in comparison with significant state-of-the-art proposals, and shown that our approach provides significant gain. We now show, in Figures 13(a)–13(c), users' coverage obtained with OnDrone when tuning a few key parameters. Specifically, in a *dense Cheese* deployment, we consider that case in which the guaranteed user data rate varies based on the SINR value $\gamma_{\mathcal{A}}$, the drone height becomes fixed, and the *ABS* power transmission increases, respectively. We show in any case both total and *ABS* coverage. In Figure 13(a), we consider fleet sizes of $D = 2, 4, 6$ drones with different user data rate guarantees. Such SINR values correspond to the MCS (Modulation and Coding Scheme) values marked in the figure, according to [48]. Here, we observe that $\gamma_{\mathcal{A}} = 10.9$ dB is a good election: on the one side, lower $\gamma_{\mathcal{A}}$ values provide higher total and *ABS* coverage since the QoS requirement is less strict, but the MCS is only QPSK, which renders a considerably lower final user throughput; on the other side, the highest QoS requirements lead to much better MCS and coding rate ("c.r." in the figure), but here the QoS requirement gets too strict so that coverage performance falls notably. In Figure 13(b), we fix the drone height to altitudes between 60 and 600 meters, for fleet sizes of $D = 2, 4, 6$ drones. Moreover, in the left side of the figure we show a histogram with the total and *ABS* coverage results when the height is left as an adaptive choice of OnDrone, as intended by the framework proposed in this paper. The results show that, depending on the fleet size, there is an optimal fixed height for the fleet where signal strength is good and the interference impairment remains stable, hence providing the best coverage. However, the difference with respect to the case in which the height is adaptive is around 20% for *ABS* coverage, which supports the idea that flexible non-uniform altitudes are convenient for drone-aided networks. In Figure 13(c), we analyze the impact of transmission power. We have selected three typical values for P^d_{Tx}: 10 dBm [43], [44], 25 dBm [16], [52], and 44 dBm [25] (besides, 44 dBm is the power transmission used for *gNB*s in cellular networks). Here, we clearly see that with 25 and 44 dBm the coverage is significantly increased due to better signal strength from the serving drone. However, fleet size increases, the framework is not able to keep the coverage stable and the interference quickly impairs coverage performance. Conversely, a 10 dBm power transmission allows the framework to combat the interference from multiple *ABS*s and keep the coverage stable. This shows that lower power transmissions make the framework more resilient. Moreover, since aerial networks are very energy-limited [43], using high power transmission needs to guarantee that the performance is more efficient. Note also that the difference in Watts from 10 to 25 dBm is more than 96%, while the corresponding coverage improvement is only 30% in the best case.

6.2 Continuous repositioning

Having assessed the basic properties and performance figures of OnDrone for drone placement in Section 6.1, we now consider flight routes. Specifically, we study the performance of OnDrone and RA (specifically proposed for dynamic cases) with repositioning routes computed with either the *Bézier Scheme* or with the *Straight Scheme* every $T = 60$ s, in two practical and realistic scenarios: *Cheese* and *Capital*. Irregular topologies like *Cheese* and *Capital* are more interesting to study with respect to regular *PPP* cases because they clearly provide visual fact of the importance of our *Bézier Scheme* or alike schemes using deflected routes when several regions have really low densities of users.

In order to assess dynamic topologies, we consider a random initial position of drones in the network. In the successive time intervals, the source position of each drone is the last location it was occupying in the previous interval.

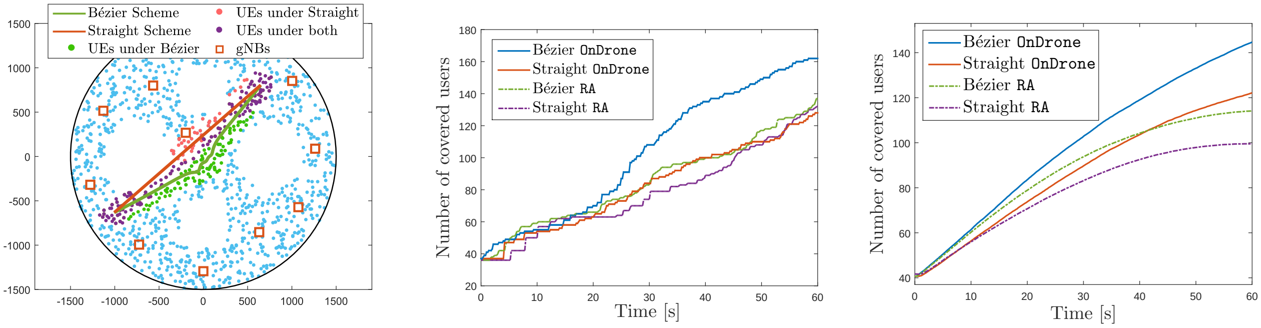
In order to compute the *Bézier curves* used as flight routes, we keep adding anchor points and run iterations of the *de Casteljau* algorithm in the *Bézier Scheme* until the segments of the piece-wise curve approximating the *Bézier curve* are all shorter than 3 m, which corresponds to flight segments of 0.2 s. Hence we can consider the network as practically static in each of such segments, and solve the corresponding coverage problem.

In Figures 14 and 15 we analyze the performance of continuous *ABS* repositioning with the *Bézier Scheme*, in comparison to the *Straight Scheme* in the *dense Cheese* and *high-rise Capital* scenarios, respectively. In both cases, we show an illustration of drone trajectories (subfigure (a)) with the corresponding *ABS* coverage over time (subfigure (b)) and the average *ABS* coverage for the scenario obtained with longer and repeated simulation runs (subfigure (c)).

In Figure 14(a) we show an instance of a network surface where four empty regions have no user (or few users) demanding for coverage, and 1 drone provides coverage assistance. While a drone following straight lines can be barely used during the full time interval since (i) it traverses part of the empty regions and (ii) it does not avoid ground cells covered by fixed *gNB*s, the *Bézier Scheme* deflects drone routes towards regions with denser population, avoiding empty regions and *gNB*-served users. The resulting gain of a simple example is quantified in Figure 14(b). The figure depicts the count of *ABS*-covered users over time for 60 s (the count is updated at each segment of the piece-wise curve approximating the *Bézier* path, every 0.2 s). In contrast with the *Straight Scheme*, in this example the *Bézier Scheme* provides a coverage gain from the beginning of the interval, and shows a gain of 26% after about 45 seconds. The gain remains high until the drone ends its path. The figure also shows that `OnDrone` largely outperforms `RA`. Indeed, *Bézier* and *Straight Schemes* used with `RA` most of the time perform worse than `OnDrone` with the *Bézier Scheme*. These coverage performance gaps vary depending on the instance. Hence, to fairly compare each scheme, in Figure 14(c) we report the *average* cumulated *ABS* coverage with each scheme, calculated over the entire numerical simulation of the scenario and over the tested initial positions. On average, by using *Bézier* flight paths in combination with `OnDrone`, at the end of an interval T, *ABS* coverage increases by a remarkable 18% compared to *Straight Scheme*. Using *Bézier* paths is beneficial also when using the `RA` placement algorithm. However, since the coverage guarantees of `RA` are much weaker than with `OnDrone`, the achievable coverage is lower.

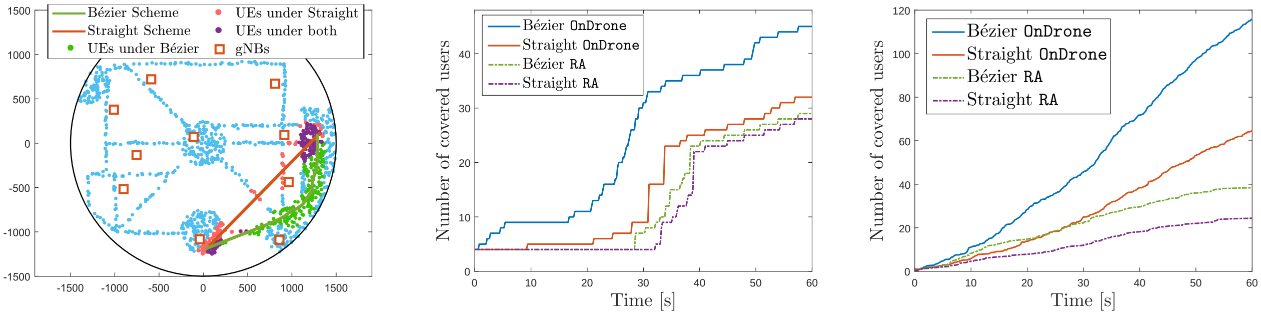
In Figure 15(a) we show a snapshot of drone trajectories optimized with the *Bézier Scheme* over *high-rise* environmental conditions in the *Capital* scenario with 1000 UEs located uniformly at random over the city, plus five masses of 200 UEs. Again, to make the presentation visual and simpler, we use a case with a single drone, although the results for more drones are similar. In the figure, we also report the corresponding flight path computed with the *Straight Scheme*. We observe that the *ABS* flies from the bottom towards the right side of the city, where there are more users. Moreover, the *Bézier* path avoids two *gNB*s in order not to overlap *ABS* coverage with *gNB* coverage. In addition, although not shown to keep the figure clear, the *Bézier* path meets two masses of 200 users on its way to the final destination, where another mass of 200 users is targeted also by the *Straight Scheme*. However, the path followed under the *Straight Scheme* does not avoid coverage overlapping with *gNB*s and includes regions with very low density of users, hence missing key coverage opportunities. Figure 15(b) quantifies the gain due to deflected drone paths for an example over these conditions. The figure shows that the *Bézier Scheme* considerably increases the cumulated number of users covered since the beginning of the interval, reaching a coverage efficiency increase of 33% with respect to the *Straight Scheme*, and with minimal route adjustments. As done in the *dense Cheese* scenario, to fairly compare the performance obtained with either *Bézier* or straight paths, in Figure 15(c) we report the *average* gain in terms of the count of users covered since the beginning of a time interval T. We see that, on average, by using *Bézier* flight paths in combination with the `OnDrone` placement algorithm, at the end of an interval T the *ABS* coverage is increased by a remarkable 47%. Using *Bézier* paths is beneficial also when using the `RA` placement algorithm. However, since the coverage guarantees of `RA` are much weaker than with `OnDrone`, the achievable coverage is much lower.

To show the repositioning operation of multiple drones with `OnDrone` and the *Bézier Scheme*, in Figure 16(a) we show the trajectories of 4 drones in the *high-rise Capital* scenario, during an interval of duration $10T$. Here, we capture the main behaviours of our repositioning schemes. The figure reflects backhaul *gNB* associations by showing multi-color paths, each color corresponding to a different *gNB* indicated on the map with a marker of the same color. Drones at the bottom and left side start at a suboptimal position. On their path, they deflect over more populated zones and follow streets that connect the origin and final destinations. Once they get to optimal positions, they remain hovering the zone, adjusting according to changes in the presence of users. The drone starting at the top behaves similarly, although it does not benefit from following any street from source to destination because (i) this street is partially covered by the yellow *gNB* and (ii) user density is low. The remaining drone has the longest path and traverses several regions. First, it flies upwards to comply with the optimal position of the first stages, but soon deflects its path towards the center of the city where a new optimal position is identified, and remains hovering this central region over the rest of the time. We also see that when this drone is very close to the red *gNB*, it does not switch its backhaul association because the yellow *gNB* provides enough connectivity while the drone coming from the top has no alternative but to attach to the red *gNB*. Pictorially showing correlation between drone trajectories and user movements is not simple with the scenarios presented throughout this paper. However, for a simple case with a dozen of users



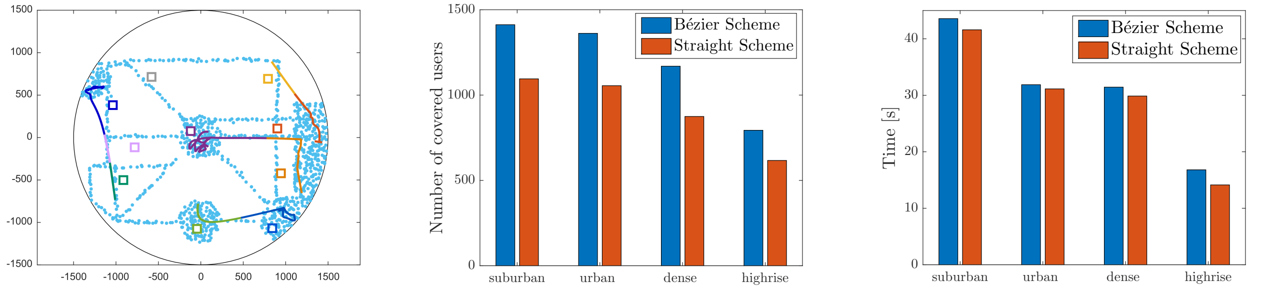
(a) Example of drone trajectories. (b) Covered UEs over time in one instance. (c) Average covered UEs over time.

Figure 14. Continuous repositioning. $D = 1$, $U = 1000$. Scenario: *dense, Cheese*.



(a) Example of drone trajectories. (b) Covered UEs over time in one instance. (c) Average covered UEs over time.

Figure 15. Continuous repositioning. $D = 1$, $U = 2000$. Scenario: *high-rise, Capital*.



(a) *Bézier* trajectories. Each solid line represents one drone. Trajectory segments are colored with the color of the backhaul associated *gNB*. Scenario: *high-rise*. (b) Average number of covered users per minute. *Bézier Scheme* vs. *Straight Scheme*. (c) Average time spent per user per minute in coverage. *Bézier Scheme* vs. *Straight Scheme*.

Figure 16. Continuous repositioning during 10 minutes, $D = 4$, $U = 2000$. Scenario: *Capital*.

and one *ABS*, we refer the reader to the illustrative example described in Appendix C.3.

Finally, in Figures 16(b) and 16(c) we analyze, for the same experiment with four repositioning drones, the *ABS* coverage count (i.e., the average number of users covered by drones in a one-minute time window) and the average time during which an *ABS*-covered user receives service within the same one-minute time window. We use the *Capital* scenario in all possible environmental density cases, with 2000 users in total. Figure 16(b) shows that the *Bézier Scheme* provides a coverage gain around 25% with respect to using straight paths. More interestingly, the denser environments present higher coverage gains. While the *urban* environment provides a gain of 25%, the *dense* environment increases such gain up to a remarkable 33%. Therefore, deflected paths are more important in denser scenarios, like in historical districts of old cities and in modern downtowns.

The number of seconds spent under *ABS* coverage, reported in Figure 16(c), tells the quality of coverage opportunities offered by drones *on the move*. The figure shows little coverage time differences between the *Bézier* and *Straight Schemes*, with some improvements observed with the former. Therefore, we can conclude that the increased coverage count offered by using deflected flight paths is not obtained at the expenses of the time spent by users under coverage. In general, we remark that although the *Bézier Scheme* needs more time to reach the optimal placement identified with OnDrone, it outperforms significantly approaches based on straight flights for *ABS*s.

6.3 Discussion and lesson learnt

Our analysis has shown that optimizing the position of a fleet of drones in a coordinated manner is unfeasible with standard solvers. However, using *extremal-optimization* techniques, we have seen that it is possible to achieve nearly-optimal results in polynomial time. Thus, it is doable to run drone repositioning on a minute time scale. Indeed, our results show that with realistic topologies, drones are able to follow mobile masses of people over time, or either react quickly to changes, thanks to OnDrone.

The performance of `OnDrone` that we have proposed and validated depends on the thickness of the lattice used to reduce the space of options used to seek (near-)optimal drone positions. Denser lattices would result in improved accuracy, although at the expense of computational cost. With as much as 10 drones and a few thousands of users, it is possible to achieve accurate near-optimal results on a subsecond time scale using commodity hardware, as we have done for our numerical evaluation. For larger fleets, more powerful hardware is required, which is not a big problem for base stations and definitely not a problem for cellular back-offices, e.g., in the MEC of future 5G networks [53].

Our numerical results have quantitatively shown the importance of integrating a fleet of drones in a cellular network. The presence of inter-drone interference is key in the optimization of drone positions, so that it cannot be neglected, contrary to what so far used in the literature. The presence of interference makes the optimal number of drones finite, with no advantages coming from deploying unnecessarily dense fleets.

We have also seen that repositioning is a key component of the overall drone-aided network framework. Repositioning requires solving not only for 3–D drone positions, but also finding a flight assignment and deflect flight routes. We have used the *Hungarian method* to solve the assignment problem optimally and *Bézier curves* to obtain very efficient and dynamic trajectories that offer coverage opportunities to many users without reducing their time under drone coverage. Such dynamic behavior is key for network surfaces in which some regions cannot host users that can benefit of the presence of ABSs, e.g., in forests or in indoor installations in residential and commercial areas, and also to avoid flying over those ground regions already served by gNBs.

Indeed, our dynamic *Bézier Scheme* presents remarkable results, and we have observed that it enhances a lot coverage experienced in realistic topologies, especially in densely populated cities with *dense* or *high-rise* profiles. Where main avenues and landmarks attract users, our *Bézier Scheme* allows the drones to easily follow masses of users on selected paths and areas.

In general, we have shown that it is feasible to have an autonomous dynamic aerial network that reorganizes itself to optimize network coverage.

7 Conclusions

In this paper we have proposed a novel optimization framework for drone-aided cellular networks in terms of user coverage under guaranteed signal quality. The framework is novel because it addresses multiple coordinated drones as well as legacy gNBs and analytically accounts for complex interference expressions caused by drone transmissions under non-negligible and variable LoS probability. Specifically, we have studied the integration of a finite fleet of drones acting as *aerial base stations* connected to and aiding a ground network of gNBs, from/to which they are able to relay traffic. Since cellular users can move, we have shown that implementing a solution for our framework requires solving three important subproblems: finding optimal positions of drones at a given time instant, map drones onto coverage points, and plan deflected flight routes, to optimize network coverage also while moving the drones.

The coverage problem is a non-linear, non-convex and mixed-integer NP-Complete problem, so it is not possible to handle it with conventional off-the-shelf optimizers. The main issue in our novel formulation lies in the intertwined nature of interference, drone positioning, and LoS probability. Hence, we have proposed `OnDrone`, an *extremal-optimization* algorithm that performs near-optimally in low-order polynomial time for various user topologies and under different density of LoS obstructions, from *suburban* to *high-rise* scenarios. `OnDrone` outperforms state of the art mechanisms because they do not address the root causes of interference. Interestingly, we have found out that unnecessarily large drone fleets would only have negative impacts on coverage, due to interference. Besides, we have unveiled that optimally mapping drones onto coverage targets is doable in negligible time and, most importantly, we have introduced and evaluated a novel and dynamic scheme to compute intelligent drone trajectories upon repositioning. With our *Bézier Scheme*, it is possible to deflect flight routes of drones so to dramatically improve coverage performance (up to 47% in our experiments) in a variety of scenarios, including in real, dense and high-populated cities.

Acknowledgments

The work of E. Arribas is partially supported by the FPU15/02051 grant from the Spanish Ministry of Education, Culture and Sports (MECD). The work of V. Mancuso is supported by the Ramón y Cajal grant (ref: RYC-2014-16285) from the Spanish Ministry of Economy and Competitiveness and by the DiscoEdge grant (TIN2017-88749-R) from the Spanish Ministry of Science, Innovation and Universities.

References

[1] W. Mohr, “5G empowering vertical industries,” Tech. Rep., 2016.

[2] M. Maternia *et al.*, “5G PPP use cases and performance evaluation models,” 5G-PPP, Tech. Rep., Apr. 2016, White Paper.

[3] I. Bor-Yaliniz and H. Yanikomeroglu, “The new frontier in RAN heterogeneity: Multi-tier drone-cells,” *IEEE Communications Magazine*, vol. 54, no. 11, 2016.

[4] W. Guo and T. O’Farrell, “Relay deployment in cellular networks: Planning and optimization,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 8, 2013.

[5] E. Arribas and V. Mancuso, “Multi-path D2D leads to satisfaction,” in *IEEE WoWMoM*, 2017, pp. 1–7.

[6] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Wireless communication using unmanned aerial vehicles (UAVs): Optimal transport theory for hover time optimization,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8052–8066, 2017.

[7] S. Boettcher and A. Percus, “Optimization with extremal dynamics,” *Phys. Rev. Lett.*, vol. 86, pp. 5211–5214, 2001.

[8] P. Casteljau, “Mathématiques et CAO. Vol. 2: formes à pôles,” *Hermes*, 1985.

[9] R. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, “Efficient 3-D placement of an aerial base station in next generation cellular networks,” in *ICC*. IEEE, 2016.

[10] O. Andryeyev and A. Mitschele-Thiel, “Increasing the cellular network capacity using self-organized aerial base stations,” in *ACM DroNet 2017*, pp. 37–42.

[11] A. Fotouhi, M. Ding, and M. Hassan, “Dynamic base station repositioning to improve spectral efficiency of drone small cells,” in *WoWMoM, 2017*. IEEE, pp. 1–9.

[12] R. Ferrús, H. Koumaras, O. Sallent, G. Agapiou, T. Rasheed, M.-A. Kourtis, C. Boustie, P. Gélard, and T. Ahmed, “SDN/NFV-enabled satellite communications networks: Opportunities, scenarios and challenges,” *Physical Communication*, vol. 18, pp. 95–112, 2016.

[13] S. Katikala, “Google$^{\text{TM}}$Project Loon,” *InSight: Rivier Academic Journal*, vol. 10, no. 2, pp. 1–6, 2014.

[14] T. Simonite, “Meet Facebooks stratospheric Internet drone,” *MIT Technology Review. Retrieved June*, vol. 8, p. 2016, 2015.

[15] 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, 2016.

[16] Y. Chen, N. Zhao, Z. Ding, and M.-S. Alouini, “Multiple UAVs as Relays: Multi-Hop Single Link Versus Multiple Dual-Hop Links,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 6348–6359, 2018.

[17] G. Zhang, H. Yan, Y. Zeng, M. Cui, and Y. Liu, “Trajectory Optimization and Power Allocation for Multi-Hop UAV Relaying Communications,” *IEEE Access*, vol. 6, pp. 48 566–48 576, 2018.

[18] A. Al-Hourani, S. Kandeepan, and S. Lardner, “Optimal LAP altitude for maximum coverage,” *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[19] A. Hayajneh, S. Zaidi, D. McLernon, and M. Ghogho, “Optimal dimensioning and performance analysis of drone-based wireless communications,” in *GC Wkshps*. IEEE, 2016.

[20] W. Mei, Q. Wu, and R. Zhang, “Cellular-connected UAV: Uplink association, power control and interference coordination,” *arXiv preprint arXiv:1807.08218*, 2018.

[21] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Optimal transport theory for power-efficient deployment of unmanned aerial vehicles,” in *IEEE ICC, 2016*.

[22] ——, “Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, 2016.

[23] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, “Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046–1061, 2017.

[24] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Drone small cells in the clouds: Design, deployment and performance analysis,” in *IEEE GLOBECOM*, 2015.

[25] S. Rohde and C. Wietfeld, “Interference Aware Positioning of Aerial Relays for Cell Overload and Outage Compensation,” in *IEEE VTC, 2012*.

[26] E. Arribas, V. Mancuso, and V. Cholvi, “Fair Cellular Throughput Optimization with the Aid of Coordinated Drones,” in *IEEE INFOCOM WKSHPS – MiSARN*, pp. 1–6,.

[27] E. Kalantari, I. Bor-Yaliniz, A. Yongacoglu, and H. Yanikomeroglu, “User association and bandwidth allocation for terrestrial and aerial base stations with backhaul considerations,” in *IEEE PIMRC,2017*, pp. 1–6.

[28] O. Sahingoz, “Generation of bezier curve-based flyable trajectories for multi-UAV systems with parallel genetic algorithm,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 499-511, 2014.

[29] R. Petrolo, Y. Lin, and E. Knightly, “ASTRO: Autonomous, sensing, and tetherless networked drones,” in *ACM DroNet, 2018*, pp. 1–6.

[30] M. Moradi, K. Sundaresan, A. Chai, S. Rangarajan, and M. Mao, “SkyCore: Moving Core to the Edge for Untethered and Reliable UAV-based LTE networks,” in *ACM MobiCom, 2018*.

[31] Y. Zeng, J. Lyu, and R. Zhang, “Cellular-Connected UAV: Potentials, Challenges and Promising Technologies,” *IEEE Wireless Communications*, vol. 26, no. 1, pp. 120–127, 2019.

[32] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, “Modeling air-to-ground path loss for low altitude platforms in urban environments,” in *IEEE GLOBECOM, 2014*, pp. 2898–2904.

[33] ITU-R, “Propagation data and prediction methods required for the design of terrestrial line-of-sight systems,” 2015.

[34] J. Seybold, *Introduction to RF propagation*. John Wiley & Sons, 2005.

[35] A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, “A survey of channel modeling for UAV communications,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2804-2821, 2018.

[36] G. Das, R. Fraser, A. Lóopez-Ortiz, and B. Nickerson, “On the discrete unit disk cover problem,” *International Journal of Computational Geometry & Applications*, vol. 22, no. 05, pp. 407–419, 2012.

[37] J. Chen and D. Gesbert, “Optimal positioning of flying relays for wireless networks: A los map approach,” in *IEEE ICC, 2017*.

[38] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[39] W. Sun and Y.-X. Yuan, *Optimization theory and methods: nonlinear programming*. Springer Science & Business Media, 2006, vol. 1.

[40] L. Zanzi, F. Giust, and V. Sciancalepore, “M^{2}EC: A multi-tenant resource orchestration in multi-access edge computing systems,” in *IEEE WCNC, 2018*, April 2018, pp. 1–6.

[41] S. Hayat, E. Yanmaz, and R. Muzaffar, “Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2624–2661, Fourthquarter 2016.

[42] M. Mirahsan, R. Schoenen, and H. Yanikomeroglu, “Hethetnets: Heterogeneous traffic distribution in heterogeneous wireless cellular networks,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2252–2265, 2015.

[43] Y. Zeng and R. Zhang, “Energy-efficient UAV communication with trajectory optimization,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[44] B. V d Bergh, A. Chiumento, and S. Pollin, “LTE in the sky: trading off propagation benefits with interference costs for aerial nodes,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 44–50, 2016.

[45] J. Lyu, Y. Zeng, and R. Zhang, “Cyclical multiple access in UAV-aided communications: A throughput-delay tradeoff,” *IEEE Wireless Communications Letters*, vol. 5, no. 6, pp. 600–603, 2016.

[46] W. Pan and G. Cheng, “QoE assessment of encrypted YouTube adaptive streaming for energy saving in Smart Cities,” *IEEE Access*, vol. 6, pp. 25 142–25 156, 2018.

[47] W. Li, “Overview of fine granularity scalability in MPEG-4 video standard,” *IEEE Transactions on circuits and systems for video technology*, vol. 11, no. 3, pp. 301–317, 2001.

[48] A. Asadi and V. Mancuso, “On the compound impact of opportunistic scheduling and D2D communications in cellular networks,” in *ACM MSWiM, 2013*, pp. 279–288.

[49] 3GPP TR38.901, “Study on channel model for frequencies from 0.5 to 100 GHz, V14.0.0.”

[50] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile computing*. Springer, 1996, pp. 153-181.

[51] S. Mao, “Fundamentals of communication networks,” in *Cognitive Radio Communications and Networks*. Elsevier, 2010, pp. 201–234.

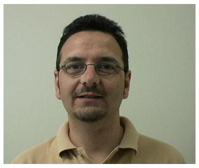
[52] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017.

[53] F. Giust, V. Sciancalepore, D. Sabella, M. C. Filippou, S. Mangiante, W. Featherstone, and D. Munaretto, “Multi-access edge computing: The driver behind the wheel of 5G-connected cars,” *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 66–73, 2018.

[54] T. F. Hain, A. L. Ahmad, S. V. R. Racherla, and D. D. Langan, “Fast, precise flattening of cubic Bezier path and offset curves,” *Computers & Graphics*, vol. 29, no. 5, pp. 656–666, 2005.

Edgar Arribas obtained his BSc in Mathematics from Universitat de València (UV) in 2015. After completing an MSc in Telematic Engineering at Universidad Carlos III de Madrid (UC3M) in 2016, he has become a PhD student at IMDEA Networks Institute and UC3M, funded by the MECD FPU15/02051 grant. He works on optimization of dynamic relay in wireless networks.

Vincenzo Mancuso is Research Associate Professor at IMDEA Networks, Madrid, Spain, and recipient of a Ramon y Cajal research grant of the Spanish Ministry of Science and Innovation. Previously, he was with INRIA (France), Rice University (USA) and University of Palermo (Italy), from where he obtained his Ph.D. in 2005. His research focus is on analysis, design, and experimental evaluation of opportunistic wireless architectures and mobile broadband services.



Vicent Cholvi graduated in Physics from the University of Valencia, Spain and received his doctorate in Computer Science in 1994 from the Polytechnic University of Valencia. In 1995, he joined the Jaume I University in Castellón, Spain where he is currently a Professor. His interests are in distributed and communication systems.

Appendix A
Proof of NP-Completeness

Theorem 1. *The Coverage Problem $\mathcal{C}$ is NP-Complete.*

Proof. We claim that the coverage problem $\mathcal{C}$ is NP-Complete, since the *minimum geometric disk cover* (MGDC) problem can be reduced, in polynomial time, to a particular case of the Coverage Problem $\mathcal{C}$. Since the *MGDC* problem is a very well known NP-Complete problem [36], then $\mathcal{C}$ is also NP-Complete. We prove such claim in what follows.

Let $\mathcal{M}$ be the *MGDC* problem. From the statement of problem $\mathcal{M}$, we have a set of points $\mathcal{U}$ (users in our case) and a fixed radius $R > 0$. Thus we want to minimize the number of disks with radius R that cover all the points of $\mathcal{U}$. We consider a specific set of instances of the Coverage Problem $\mathcal{C}$ in which the interference is negligible, all drones hover at the same elevation and are fast enough to reach any position in the surface in the time interval (i.e., $\mathcal{S}_d = \mathcal{S}$). In this way, the ground coverage area of each *ABS* is a circle with a radius that depends on the elevation. Let's use elevation h_R at which the ground coverage area has the radius R of the *MGDC*. Thus, we show that this is a particular set of instances of the coverage problem that cannot be deterministically solved in polynomial time.

Assume, *reductio ad absurdum*, that we can solve problem $\mathcal{C}$ in polynomial time for all its instances. Then, we present in Algorithm 4 an algorithm that solves any instance of the *MGDC* problem in polynomial time. Specifically, the *MGDC* problem that finds the minimal number of disks of radius R that cover all points in a set of 2–D coordinates can be solved by a linear search of the minimum number of disks. The search proceeds by adding a disk per iteration. It verifies that a given number of disks covers all points by solving the Coverage Problem $\mathcal{C}$ for the special instances described above. If the solution of the Coverage Problem $\mathcal{C}$ covers all points, the algorithm stops since we have found the minimum number of disks that cover all points. Note that in Algorithm 4, the iteration counter d (which is also the number of disks to use in one iteration) is bounded by the number of points $|\mathcal{U}|$. Thus, Algorithm 4 has a polynomial complexity, since we are assuming that problem $\mathcal{C}$ is solvable in polynomial time, which is a contradiction with the fact that the *MGDC* problem is NP-Complete.

Since *MGDC* is NP-Complete [36], the claim follows. □

Appendix B
Bézier curves for drone flight paths

Bézier curves are an outstanding geometrical tool that deflects the trajectory of an *ABS* close to clusters of UEs while targeting the optimum positioning. *Bézier curves* are smooth, endpoint interpolators—i.e., the curve begins and ends in a provided source and destination—and are contained in the convex hull of a set of *anchor points*. The curve is *attracted* by *anchor points*, which in our case are specific users. Below, we mathematically define a *Bézier curve*.

Definition 1. *Given a set of points $\mathcal{P} = \{P_k\}_{k=0}^{n} \subset \mathbb{R}^m$, the resulting Bézier curve is:*

$$\beta^{\mathcal{P}}(t) = \sum_{k=0}^{n} \binom{n}{k} \cdot P_k \cdot t^k \cdot (1-t)^{n-k}, \quad t \in [0,1]. \tag{16}$$

Algorithm 4 Solution for the *Minimum geometric disk cover* problem based on the solution of the Coverage Problem $\mathcal{C}$.

```
Require: Set of points $\mathcal{U}$, radius $R>0$.
$d \leftarrow 1$.
found ← False.
while not found
    Solve problem $\mathcal{C}$ for the set $\mathcal{U}$ and fixed elevation $h_R$.
    if problem $\mathcal{C}$ covers all points $\mathcal{U}$, then found ← True.
    $d \leftarrow d+1$.
end of while
```

From the definition, we see that the *Bézier curve* $\beta^{\mathcal{P}}(t)$ begins at the source $\beta^{\mathcal{P}}(t=0) = P_0$, and ends at the destination $\beta^{\mathcal{P}}(t=1) = P_n$. The variable range for variable t can be adapted with a linear transformation to modify the speed of the curve. Since we are interested only in the flight path, i.e., the trajectory of the curve, we consider $t \in [0,1]$.

We need to obtain the offset region of the *Bézier curve* (a.k.a. stroking the curve). Since *Bézier* offset curves are not analytically obtainable [54], we use the *de Casteljau* algorithm [8]. This is the main recursive and numerically stable method to draw *Bézier curves* from a set of anchor points $\mathcal{P}$, by approximating the curve with small segments. We use *Bézier curves* for the ground projection of the trajectories, while the elevation of drones varies linearly from source to destination heights, indicated as h_{src} and h_{dest}. The actual 3–D trajectory of a drone with anchor points $\mathcal{P}$ is:

$$\vartheta^{\mathcal{P}}(t) = \Big(\beta^{\mathcal{P}}(t), \;\; h_{src} + t \cdot (h_{dest} - h_{src})\Big), \quad t \in [0,1]. \tag{17}$$

Finally, we stroke $\beta^{\mathcal{P}}(t)$ by stroking the segments obtained by the *de Casteljau* algorithm.

Appendix C
Notes on OnDrone operation

C.1 Convergence of the `OnDrone` algorithm

Notwithstanding the complexity of the problem formulation expressed in Eq. (8), our near-optimal heuristic, `OnDrone`, requires few iterations. We report in Figure 17 a typical example of `OnDrone` iterations for each of the three deployment types studied in the manuscript, with 1000 users and 6 drones. The figure plots coverage vs. number of iterations. Coverage increases fast in the initial iterations and then flattens until the `OnDrone` stopping condition is reached. In each of the deployment scenarios of this case, we do not need more than 14 iterations to *converge*. With up to 10 drones and 2000 users, we have observed that convergence never required more than a few tens of iterations.

C.2 Guaranteed user data rate

The coverage optimization problem addressed in this paper aims to maximize the amount of ground users that are covered, either by *gNB*s or by *ABS*s, with a guaranteed service availability that allows for a minimum user data rate. Hence, as soon as we determine parameters as the maximum number of users U_g and U_d allowed to attach to a *gNB* g or *ABS* d, respectively, and the data rates $R^{\mathcal{A}}_{\min}$ and $R^{\mathcal{B}}_{\min}$, there is a user data rate guaranteed for each covered user. In this paper, we mainly use an SINR value

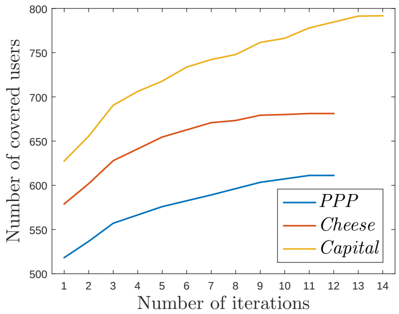


Figure 17. Convergence of `OnDrone` with $D = 6$ *ABS*s and $U = 1000$ UEs. Scenario: *dense*.

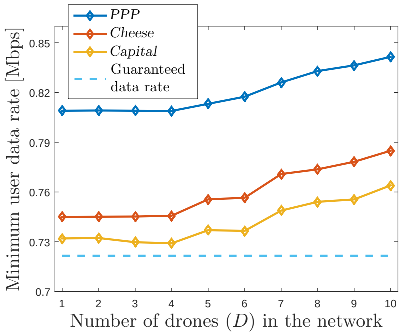


Figure 18. Minimum user data rate achieved in comparison with the user guaranteed rate. $U = 1000$. Scenario: *dense*.

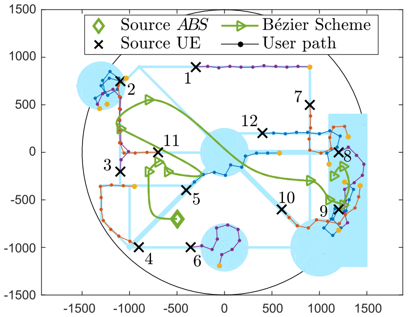


Figure 19. Network dynamics in a small scenario during 10 minutes. $D = 1$, $U = 12$. Scenario: *high-rise, Capital*.

of 10.9 dB because it allows to use a 16QAM MCS and a reasonable coding rate of $1/2$, so that covered users can enjoy a decent network access service. Indeed, we have evaluated in Figure 13(a) the impact of varying the SINR values studied in [48], and observed that higher values let coverage decay considerably. In contrast, lower values do not present much relevant impact on coverage, albeit they guarantee worse data rates. Moreover, the maximum allowed number of users served by a *gNB* or an *ABS* is set here to 100 users, which reflects the capacity scale of current 3GPP-compliant cells in terms of active users per BS sector.

In Figure 18, we show the minimum data rate experienced by covered users. The rate is computed with the Shannon formula, using simulated SINR values for a network of 1000 UEs with up to 10 *ABS*s. With the selected SINR thresholds (10.9 dB) and maximum number of users per base station (100), and with a bandwidth of 20 MHz, the guaranteed data rate is 0.72 Mbps, which is plotted as an horizontal dotted line in the figure.

On the one side, since in the *PPP* scenario the distribution of users is uniform, the footprint of *gNB*s and *ABS*s is not sufficient to cover the maximum allowed number of users. Hence, resources are split among less users, so that they experience rates well above the guaranteed one. On the other side, scenarios like *Cheese* and, more clearly, *Capital* present a non-uniform distribution of users, with groups that form spontaneously. Hence, as previously seen in Figure 12, coverage is higher than for the *PPP* scenario. The result is that both *gNB*s and *ABS*s tend to cover almost as many users as they are allowed to cover, and minimum rates approach the guaranteed one. In all scenarios, increasing the number of drones is beneficial because the resulting number of users per base station diminishes, on average.

The analysis of other scenarios, thresholds, limits in the number of users per base stations and in the backhaul, and environmental conditions lead to very similar qualitative conclusions, so we do not include those results in the figure.

Overall, here we have shown that the good performance discussed for our coverage optimization schemes, are not obtained in change of poor or, what would be worst, uncontrollably low data rates. In contrast, the guaranteed rate computed based on a fistful of optimization constraints in our problem formulation, results in a realistically close approximation for the lower bound of users performance.

C.3 User mobility and drone trajectories

In order to evaluate the relation between user mobility and drone trajectories, in Figure 19 we picture a small scenario with 12 UEs and 1 *ABS*. We depict both the *ABS* movement *and* the users movement, during 10 minutes. Here, to further simply presentation, we have took out *gNB*s.

In the figure, trajectories are marked with positions sampled once per minute, although the actual movement followed by users is more erratic, according to the random way-point model simulated with a one-second time resolution. An X marks the starting point for each user, while a green diamond indicates the initial position of the drone.

If we analyze the trajectories minute by minute, we can observe the evolution of optimum coverage over time. For instance, at the beginning, the drone only covers users 4, 5 and 6, which is optimum at that stage (other users are far and there are no larger groups to cover). However, after one minute, these three users walk away from each other making not possible to cover them altogether, so that the *ABS* immediately reacts and flies where it can cover other three users. Note that, on its first-minute trajectory, the drone deflects its path towards user 4 to cover it momentarily. The effects caused by using the *Bézier Scheme* are evident throughout the entire trajectory of the drone. During the following minutes, the drone hovers on the left side of the map, until the users begin to gather in the area on the right part of the map. Hence, the drone quickly reacts to this change and gets repositioned on this region, where it stays for the last few minutes, eventually covering four users.

From what presented in this appendix, the proposed `OnDrone` heuristic, with the help of *Bézier* trajectories, is suitable for fast reconfiguration and therefore allows to dynamically adjust network coverage at the same time-scale at which the topology of users evolves. The presence of users plays the role of "attractor" for both optimizing the position of drones and for shaping the trajectories followed while repositioning.