IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 September 2023, accepted 15 October 2023, date of publication 18 October 2023, date of current version 23 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3325704

RESEARCH ARTICLE

Joint Spectrum and Computing Resource Allocation in Fog-Assisted Drone Communications for Ambiental Services

FARSHAD SHAMS, VINCENZO LOTTICI, AND FILIPPO GIANNETTI
Department of Information Engineering, University of Pisa, 56126 Pisa, Italy

Corresponding author: Farshad Shams (farshad.shams@ing.unipi.it)

This work was supported in part by the Project Smart Control of the Climate Resilience in European Coastal Cities (SCORE) funded by the European Commission's Horizon 2020 Research and Innovation Program under Grant 101003534, and in part by the Project Platform for Observations In Agricultural and eNvironmentAl fields (POIANA) funded by the University of Pisa under Grant DR n. 589.

ABSTRACT Fog computing allows for energy-efficient and low-latency offloading of computationally intensive tasks from wireless devices to nearby servers. The integration of this technology in drone communications enables the managing of challenging tasks such as the ones found in remote areas with complex civil protection environments, such as disaster areas and emergency zones. In this paper, we propose a joint resource allocation scheme that optimizes both radio and computational resources for fog-assisted drone communication networks. Each drone decides whether to execute its task locally on its edge node or offload it to a fog node deployed on the base station (BS). Our scalable solution effectively minimizes service latency and energy consumption jointly, while taking into account physical- and application-layer constraints. Specifically, we allocate the CPU frequency capacity of both the local edge node and the remote fog node, as well as link bandwidth. Wireless channels to access the BS are limited, so only the most beneficial drones offload their tasks, while others use their local edge nodes. We formulate the power dissipation of various electronic circuits in the network using practical models. To develop the bi-objective minimization for each drone, we apply the Tchebysheff theorem, which derives the Pareto boundary between the two objectives (service latency and energy consumption). The competition among drones is modeled using the non-cooperative game framework, and the existence and uniqueness of the Nash equilibrium (NE) are proven. NE is computed using an algorithm based on subgradient projection. Numerical results concerning both theoretical aspects and a practical case study are presented to corroborate the efficiency of the proposed solution.

INDEX TERMS Bi-objective optimization, drone communications, fog-assisted networks, non-cooperative game theory, pareto boundary, resource allocation, Tchebysheff method.

I. INTRODUCTION

The growth of the mobile applications industry has outpaced that of the electronic sector for wireless device CPUs and batteries. With the advent of new applications, people expect to run more computation-intensive tasks on their mobile devices. However, these tasks often consume vast amounts of energy, demand powerful computation capacity, and have strict delay constraints, while wireless devices are usually resource-constrained with limited computation capability and battery life, making such sophisticated applications impractical. To overcome these challenges, edge computing, and fog computing have been introduced. These technologies enable processing and storing data at the network's edge, bringing computing resources closer to the end-users, thereby significantly reducing latency for computation-intensive and time-sensitive applications. Edge computing involves computation at the edge of a wireless device network in computing

The associate editor coordinating the review of this manuscript and approving it for publication was Qiang Li.

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

servers named edge nodes. In contrast, fog computing is an extension of cloud computing and is a layer between the edge and the cloud. Fog nodes are more powerful than edge nodes. However, in traditional fog-assisted networks, computing servers are usually embedded in fixed base stations (BSs), making it challenging to meet the increasingly complex and dynamic computing demands.

Aerial computing has drawn extensive attention due to drones' mobility, flexibility, and maneuverability. Drones equipped with edge nodes can provide agile computing services to wireless devices. Additionally, drones can be quickly deployed to support wireless communications in remote areas with complex civil protection environments, such as disaster areas and emergency zones. The deployment of edge/fog nodes in drone communications has recently attracted wide attention from industry and academia.

In this work, we consider a single-cell fog-assisted network where one fog node is deployed on the BS to serve a number of drones, each of which is equipped with an edge node. Each drone decides whether to compute its task on its local edge node or offload it to the remote fog node. The fog node executes the offloaded tasks by sharing its computing resources with the drones. Our goal is to develop an efficient radio and computing resource allocation algorithm that jointly minimizes the energy consumption and service latency of all drones while considering their constraints and requirements. Radio resources refer to the bandwidth that is allocated to the drones that decide to offload their tasks, while computing resources include the frequency of the CPUs on the local edge nodes and the remote fog node.

A. MOTIVATION

Task offloading solutions in fog-assisted drone communications offer an opportunity to execute complex tasks on nearby fog nodes, thereby enhancing the computing capabilities of otherwise resource-constrained wireless devices. This opportunity, however, presents a critical challenge: the simultaneous minimization of service latency and energy consumption, two often conflicting objectives.

In fog-assisted drone communications, the balance between service latency and energy consumption is pivotal. Reducing latency can lead to increased energy consumption, while optimizing energy efficiency can slow down the service. The specific challenges in this context lie in the intricate interplay between physical and application layers, as well as the dynamic nature of drone tasks and movements. Achieving this balance is vital to ensure the efficient performance of latency-critical applications [1] and also to tackle one of the paramount challenges in wireless networks, that of reducing energy consumption while maintaining QoS, particularly service latency [2].

The need to optimize these conflicting goals gives rise to our approach: the derivation of a Pareto front between latency and energy. This Pareto front represents the efficient frontier of tradeoffs, providing a critical tool for network designers in the unique and complex domain of fog-assisted drone communications.

In the study presented in [3], the significance of task offloading in fog-assisted drone communication networks is emphasized, revealing potential benefits in terms of energy savings and decreased processing durations. However, given that their evaluations were performed without specific radio or computing resource allocation, their findings also point to an imperative need for a refined resource allocation strategy. This strategy should effectively address the intricate relationship between service latency and energy consumption, underscoring the importance of our proposed approach.

Given the complexities and limitations imposed by Shannon's law, and the drone communications potential for significant air-ground interference, the joint minimization of service latency and energy consumption becomes essential. Our work aims to address this challenge by proposing a resource allocation technique specifically designed for fog-assisted drone communications, with the goal of simultaneously minimizing these two crucial parameters.

In conclusion, efficient task offloading, mindful management of both radio and computing resources, and the intelligent minimization of both service latency and energy consumption are key motivators for our proposed approach. We believe that our work will not only enhance network performance but will also contribute significantly to the broader understanding and advancement of resource allocation techniques in the emerging field of fog-assisted drone communications.

B. RELATED WORKS

Fog-assisted networks are pivotal in resource allocation, aiming at different objectives:

Objective Function of Type (i): Minimizing overall energy consumption subject to a constraint on latency. Recent works investigate multi-cell networks [4], mobile terminals [5], [6], [7], stochastic network optimization [8], game theory [9], [10], and mixed-integer non-linear programming [11].

Objective Function of Type (ii): Minimizing service latency while considering energy constraints. Notable contributions address multi-tiered systems [12], [13], [14], multi-cell networks [15], and drone communications [16], [17]. Game theory [18], [19], and auction-based approaches [20] are used to model competition among devices.

Objective Function of Type (iii): Minimizing a weighted sum of energy and latency. This is tackled through machine learning [21], [22], [23], graph theory [24], genetic algorithms [25], blockchain [26], [27], and by decoupling offloading decisions [28], [29], [30].

For networks with multiple accessible fog nodes, various methods have been applied, such as game theory [9], genetic algorithms [25], Petri nets [31], and blockchain [26].

Game-theoretic approaches, including non-cooperative games [32], [33], [34], [35], and the Stackelberg game [36], are employed with techniques like the Bellman equations [33] and greedy algorithms [34].

In fog-assisted drone communications, recent research targets objective function types (*i*) [37], [38] and (*ii*) [16], [17]. However, in mentioned drone communications works, no radio and computing resource allocation is performed and the aim is choosing the best fog node to offload the task.

Overall, the review incorporates a diverse range of methodologies and objectives, synthesizing the recent literature in fog-assisted networks.

The review of the existing related works shows that the existing problem formulations suffer from the following major drawbacks:

1) Existing works did not consider a practical/industrial model for power consumption of electronic circuits of wireless devices. They usually set a fixed amount for it, while the power consumption of electronic circuits involved in communication, e.g., radio frequency (RF) and baseband (BB) electronic circuits, are variables and functions of the data rate. Based on the conducted study in [39], by considering practical models, the behavior of spectral and energy efficiencies changes significantly compared with considering non-practical models.
2) Existing works have not fully explored the potential of fog-assisted networks in achieving the joint minimization of both service latency and energy consumption objectives. The use of a weighted-sum objective function, which is commonly employed in existing works, does not result in the joint minimization of both objectives as they are functions of both radio and computing resource variables [40]. Furthermore, the choice of the weight value can greatly affect the results. Additionally, a weighted-sum function lacks a common measurement unit, making it an unsuitable network performance objective. In contrast, the Pareto boundary is a crucial tool for network designers in bi-objective networks, as it allows them to determine the minimum possible service latency while a fixed amount of energy is available, or vice versa [1]. However, the investigation of the Pareto boundary in existing works is inadequate, highlighting the need for a more comprehensive approach.
3) Another significant limitation of existing works is their failure to consider the capacity constraints of edge/fog nodes as wireless receivers, which can only serve a limited number of wireless devices at a time. This constraint arises due to the limited number of available wireless channels, as well as the physical limitations of the RF and BB electronic circuits.

C. MAIN CONTRIBUTIONS

This paper presents a novel resource allocation method that aims to overcome existing drawbacks in the literature on fog-assisted drone communications. The specific contributions of this work include:

1) A resource allocation method for the uplink direction of a single-cell fog-assisted wireless network is introduced. Drones either use an edge node to execute their tasks locally or can offload their computation to a more powerful fog node deployed on the BS. The proposed algorithm jointly minimizes service latency and energy consumption by considering the limited number of available wireless channels at the BS and efficiently assigning channels to drones.
2) Practical models are utilized to compute the energy consumption of both drones and the BS, encompassing all electronic circuits involved in communication and computing. Furthermore, the weighted Tchebysheff method [41] is applied to tackle the bi-objective minimization problem for each drone, deriving the Pareto boundary between service latency and energy consumption objectives. The Pareto boundary is of great importance to network designers and industry professionals as it provides a range of optimal solutions, allowing them to choose a solution that best fits their requirements.
3) The problem is formulated using a non-cooperative game to allocate radio and computing resources to competitive drones in an efficient manner. An algorithm based on subgradient projection is developed to compute the unique NE, taking into account the practical limitations of wireless channels and efficient utilization of available bandwidth and fog node CPU capacity.
4) Numerical results demonstrate that the proposed algorithm outperforms algorithms that do not optimize computing mode or network resource allocation, thereby showing the effectiveness of the proposed solution in minimizing service latency and energy consumption for each drone.

D. STRUCTURE OF THE PAPER

The rest of the paper is organized as follows. The system model and the problem statement are presented in Sect. II. In Sect. III, service latency and energy consumption objectives are formulated. To derive the latency-energy Pareto boundary for each drone, in Sect. IV, a tractable bi-objective optimization problem is formulated. The derived optimization problem is converted to an equivalent convex one in Sect. V. Then, in Sect. VI, the competition among drones is modeled using non-cooperative games, and an algorithm is developed to compute the unique NE point. Numerical results are provided in Sect. VII. Finally, concluding remarks are given in Sect. VIII.

II. NETWORK MODEL

In this section, we present the uplink direction of a single-cell wireless network focusing on the utilization of K rotatory wing drones as wireless transmitters, along with a fog node deployed on the Base Station (BS). The critical elements and their interactions of the network model are visually illustrated in Figure 1.

Drones and Tasks: We define each drone $k = 1, \dots, K$, with an edge node whose computational capacity is considerably less than the fog node on the BS. A computation task

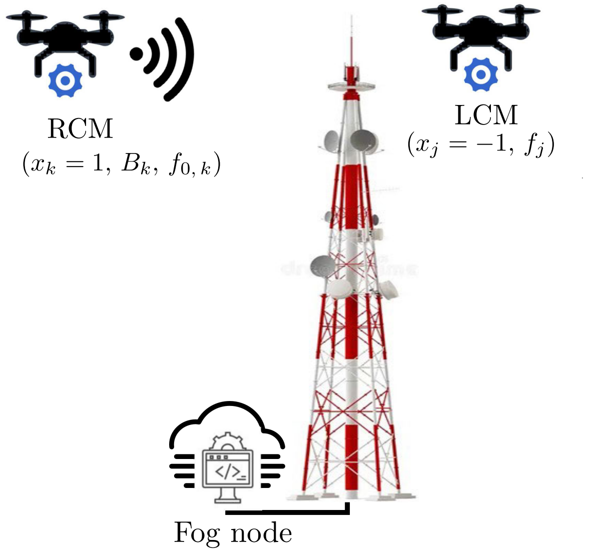


FIGURE 1. Network model.

for each drone k is expressed as a 2-tuple $(D_k, , C_k)$, with D_k indicating the task size in [bits] and C_k the required CPU cycles for one-bit data computation. These tasks are atomic and have strong dependence over different subtasks.

Computing Modes: The drones can operate in one of two computing modes: 1) *Local Computing Mode (LCM)*, utilizing local resources for task execution; and 2) *Remote Computing Mode (RCM)*, offloading tasks to a remote fog node for execution and result retrieval. The BS, limited to N simultaneous transmitting drones, employs virtual machine (VM) multiplexing and consolidation for parallel computation of offloaded tasks.

Resource Allocation: The network operator aims to efficiently allocate computing modes, bandwidth, and CPU resources among drones, minimizing joint service latency and energy consumption. This includes offload time, execution time, and the energy consumption of all involved circuits. We employ realistic practical models for energy consumption evaluation at both drones and the BS sides.

Communication and Interference: Drones communicate with the BS without interference as they are served in a non-overlapping frequency band. We introduce a distributed algorithm that adjusts the computing mode, bandwidth, and CPU frequencies for each drone, considering fixed and known drone transmit power levels during task offloading. The optimization of transmit power levels is neglected as the power required for the drone's flight operations is much greater than the power needed for data transmission, making the latter's consumption comparatively negligible [42], [43].

Assumptions: Several assumptions guide our model. Small computation result sizes and high transmission power at the BS enable faster downloading. Continuous bandwidth and CPU frequency simplify our analysis, though discrete resources can also be considered. We assume that the drones are stationary during task offloading, a condition that aligns with specific real-world scenarios such as controlled environments, energy efficiency considerations, and maintaining data integrity. The stationary assumption facilitates the analysis by allowing for focused energy usage during computation, reducing the potential for interference, and offering compatibility with existing systems.

A. SERVICE LATENCY AND ENERGY CONSUMPTION OBJECTIVES

Let us denote by a binary variable x_k the computing mode of each drone k. For the sake of later convenience, we define $x_k \in \{-1, 1\}$ such that drone k is admitted for RCM if $x_k = 1$, and $x_k = -1$ otherwise. The limitation on the number of simultaneous transmitting drones imposes the constraint $\sum_{k=1}^{K}(1 + x_k)/2 \leq N$. For available CPU resources, let us denote by $\bar{f}_k$ and $\bar{f}_0$ in [CPU-cycle/s] the maximum computational capacity of each drone k and the fog node, respectively, where we have $\bar{f}_0$ is (much) larger than $\bar{f}_k$. We denote by $\bar{B}_0$ in [Hz] the whole amount of available bandwidth on the BS. A drone with LCM must only adjust its local CPU frequency cycle to f_k, which is upper-bounded by $\bar{f}_k(1 - x_k)/2$. That is, $0 \leq f_k \leq \bar{f}_k$ if $x_k = -1$ and $f_k = 0$ otherwise. The computational capacity of the fog node is allocated to the drones with RCM, $x_k = 1$. Those drones need to offload their tasks to the fog node on the BS at a fixed power level using its allocated bandwidth B_k. The constraints on the allocated bandwidth are represented by $\sum_{k=1}^{K} B_k \leq \bar{B}_0$ and $0 \leq B_k \leq \bar{B}_0(1 + x_k)/2$ for all k. That is, if $x_k = -1$, the second constraint forces the variable B_k to be equal to zero; otherwise, it is redundant with the first one. For a drone with $x_k = 1$, the fog node has to adjust the CPU frequency cycle of the assigned VM to $f_{0,k}$ subject to $\sum_{k=1}^{K} f_{0,k} \leq \bar{f}_0$ and $0 \leq f_{0,k} \leq \bar{f}_0(1 + x_k)/2$ for all k. These two constraints work as follows: if $x_k = -1$, from the second constraint $f_{0,k}$ is set to zero, otherwise, it is redundant with the first one. The critical variables are illustrated in Figure 1. For a drone k with LCM, the service latency includes only the time T_k^{loc} to execute the task on the local edge node. While, for a drone that is allowed to offload its task, the total service latency T_k^{rem} includes both: 1) the time $T_{k;\,\text{tx}}^{\text{rem}}$ it needs to offload its task to the BS and 2) the computing time $T_{k;\,\text{ex}}^{\text{rem}}$ the assigned VM takes to execute the task. Therefore, the total service latency of a drone k is computed by $T_k = T_k^{\text{rem}}(1 + x_k)/2 + T_k^{\text{loc}}(1 - x_k)/2$ in [s].

For a drone k with LCM, the energy consumption includes:

1) $E_{k;\,\text{ex}}^{\text{loc}}$: the energy consumed by the local edge node to execute the task of drone k,
2) $E_{k;\,\text{on}}^{\text{loc}}$: the power required to keep drone and local edge node functioning within the whole service latency T_k^{loc}. The drone includes both the power consumed by flight operation and control electronic circuits. The edge node includes the power consumed by different components, e.g., RAM and storage.

Thus, the total energy consumption for a drone with LCM is

$$E_k^{\text{loc}} = E_{k;\,\text{ex}}^{\text{loc}} + E_{k;\,\text{on}}^{\text{loc}} \qquad [\text{Joule}]. \tag{1}$$

For a drone k with RCM, the total energy the drone consumes to offload its task and the BS consumes to receive and execute the task include:

1) $E_{k;\,\text{tx}}^{\text{rem}}$: the total amount of power consumed over the duration of $T_{k;\,\text{tx}}^{\text{rem}}$ to offload the task. It includes both radiative transmit power level and the non-radiative power consumed by the baseband (BB) and radio frequency (RF) electronic circuits in the connected mode,
2) $E_{k;\,\text{rx}}^{\text{rem}}$: the power consumed by the electronic circuits of the BS for radio signal processing over $T_{k;\,\text{tx}}^{\text{loc}}$ to receive the task of drone k,
3) $E_{k;\,\text{ex}}^{\text{rem}}$: the energy consumed by the assigned VM on the fog node to execute the task of drone k,
4) $E_{k;\,\text{on}}^{\text{rem}}$: the power required to keep drones, the fog node, and the BS functioning within the whole service latency T_k^{rem}. For drones, it includes both the power consumed by flight operation and control electronic circuits. While on the edge node, fog node, and BS, it includes the power consumed by different electronic and computing components, e.g., RAM, storage, and cooling equipment.

Therefore, the total energy consumption for a drone with RCM reads as

$$E_k^{\text{rem}} = E_{k;\,\text{tx}}^{\text{rem}} + E_{k;\,\text{rx}}^{\text{rem}} + E_{k;\,\text{ex}}^{\text{rem}} + E_{k;\,\text{on}}^{\text{rem}} \qquad [\text{Joule}]. \tag{2}$$

We can formulate the total energy consumption of a drone k by $E_k = E_k^{\text{rem}}(1+x_k)/2 + E_k^{\text{loc}}(1-x_k)/2$ in [Joule]. As it can be seen, E_k represents the total amount of radiative and non-radiative power consumption at both drone and BS sides over the whole service latency.

B. LATENCY-ENERGY TRADEOFF

The circuit power for communication highly depends on the number of samples to be processed per second, i.e., the system sampling rate, which is proportional to the bandwidth. The amount of consumed power strongly depends on the signal processing load due to the coding, decoding, also backhauling of signals at the BS. This is also proportional to the data-rate and, in turn, to the bandwidth [39], [44], [45], [46], [47]. Instead, increasing bandwidth, based on Shannon's theorem, increases the data rate and, thus, decreases the time required to offload a task. In computing resources, on the one hand, increasing the CPU frequency of each VM reduces the execution time; on the other hand, it increases the energy consumption of the CPU. As it can be understood, for drones with RCM, the bandwidth, and the CPU frequency have contradictory effects on the service latency and energy consumption. That is, increasing these two parameters results in increasing the power consumption and simultaneously results in decreasing the service latency. The main trade-off is between service latency and energy consumption. They are interconnected through $f_{0,\,k}$ and B_k and are in essence conflicting. As in fog-assisted wireless networks, the key performance objectives are energy consumption and service latency; thus, we need to minimize both these performances jointly. In turn, we need to *jointly optimize* all $x_k, f_k, f_{0,\,k}$ and B_k for all drones while respecting the imposed constraints and limitations of all K drones, e.g., size of the tasks and wireless channel conditions, with the goal of joint minimization of service latency and energy consumption.

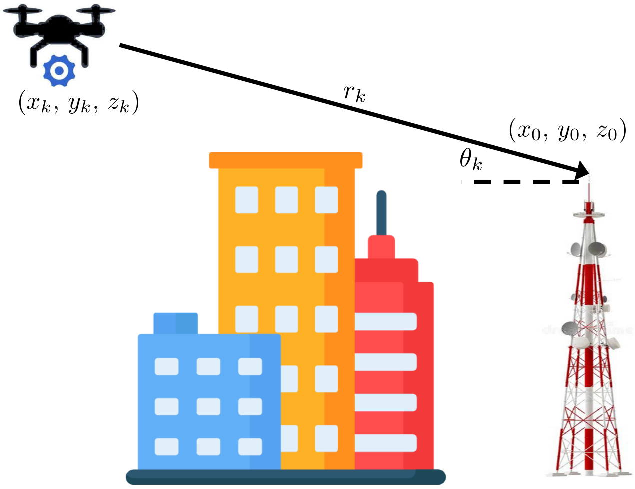


FIGURE 2. The parameters of channel model in drone communications.

C. WIRELESS COMMUNICATION MODEL

During offloading and executing a task, we assume all drones are stopped in the air. We denote the coordinates of BS antenna as $(x_0,\, y_0,\, z_0)$, while the coordinates of drone k is $(x_k,\, y_k,\, z_k)$. Moreover, we call r_k the Cartesian distance of drone k from the BS antenna. Inspired by [48] and [49], the propagation channel is modeled to account for the randomness associated with the line-of-sight (LoS) and non-line-of-sight (NLoS) communications links. To this end, a suitable model is given by [48] and [49]

$$\Pr_k(\text{LoS}) = \frac{1}{1 + \phi \exp\left(-\beta\left(\theta_k - \phi\right)\right)} \tag{3}$$

where ϕ and β are constant values that depend on the carrier frequency and type of environment (e.g., rural, urban, or dense urban), and θ_k [rad] is the elevation angle (see Figure 2) given by $\theta_k = \arcsin\left((z_k - z_0)/r_k\right)$ [rad]. Note that

$$R_k\left(B_k\right) = \sum_{m \in \{\text{LoS, NLoS}\}} \Pr_k\left(m\right) B_k \log_2\left(1 + \frac{p_k}{G_k(m)\, B_k N_0}\right) \quad [\text{bits/s}\,] \tag{5}$$

TABLE 1. Equations of the power consumptions P_k^{BB} and P_k^{RF} in [mW] [44].

Part	Variable	Model	Condition
P_k^{BB} in [mW]	$R_k(B_k)$ in [Mbit/s]	$34.5 + 0.87R_k(B_k)$	–
P_k^{RF} in [mW]	p_k in [dBm]	$23.6 + 0.78p_k$	$p_k \leq 0.2$ dBm
P_k^{RF} in [mW]	p_k in [dBm]	$45.4 + 17p_k$	0.2 dBm $\leq p_k \leq 11.4$ dBm
P_k^{RF} in [mW]	p_k in [dBm]	$1195 - 118p_k + 5.9p_k^2$	11.4 dBm $\leq p_k$

$\Pr_k$ (NLoS) $= 1 - \Pr_k$ (LoS). The path loss for LoS and NLoS links is mathematically modeled as [48] and [49]

$$\begin{cases} G_k\,(\mathrm{LoS}) = \eta_{\mathrm{LoS}} \left(\dfrac{4\pi f_c r_k}{c}\right)^{\alpha}, & \text{LoS link} \\ G_k\,(\mathrm{NLoS}) = \eta_{\mathrm{NLoS}} \left(\dfrac{4\pi f_c r_k}{c}\right)^{\alpha}, & \text{NLoS link} \end{cases} \tag{4}$$

where f_c is the carrier frequency, α is the path loss exponent, $\eta_{\mathrm{NLoS}} > \eta_{\mathrm{LoS}} > 1$ are the excessive path loss coefficients in the LoS and NLoS cases, and c is the speed of light.

For offloading a task, the transmit power level of each drone k is fixed at p_k. Since we assumed drones are stopped, we can also assume that G_k(LoS) and G_k(NLoS) are fixed and also we assume the duration of task offloading is large enough to set the expected value of the fast fading part of all wireless channels to one. In the above circumstances, the (ergodic) data rate of drone k for a specific bandwidth allocation profile $\mathbf{B} = \{B_k\}_{\forall k}$ is as (5), shown at the bottom of the previous page, with N_0 being the spectral density of the Gaussian noise in [W/Hz].

III. SERVICE LATENCY AND ENERGY CONSUMPTION FORMULATION

The goal is to assign the computing mode of the drones optimally and to allocate the available radio and computing resources among them to jointly minimize energy consumption and service latency of the drones. In the following, we compute different terms of the service latency and the energy consumption for each drone k.

A. SERVICE LATENCY

For each drone k admitted for RCM, the total service latency includes both the time necessary to offload and compute the task on the fog node. Otherwise, its service latency is only the time required to execute the task on its edge node.

1) COMPUTING T_k^{loc}

The duration time of local computing a task of D_k bits is computed by $C_k D_k / f_k$. To prevent division by zero, we add to the denominator a constant ε very close to zero, and rewrite the formula as

$$T_k^{\mathrm{loc}}\,(f_k) = \frac{C_k D_k}{f_k + \varepsilon} \qquad [\,\mathrm{s}]. \tag{6}$$

2) COMPUTING $T_{k;\,\mathrm{ex}}^{\mathrm{rem}}$

Similarly, the remote execution time $T_{k;\,\mathrm{ex}}^{\mathrm{rem}}$ on the fog node is computed by

$$T_{k;\,\mathrm{ex}}^{\mathrm{rem}}\,\left(f_{0,k}\right) = \frac{C_k D_k}{f_{0,k} + \varepsilon} \qquad [\,\mathrm{s}]. \tag{7}$$

3) COMPUTING $T_{k;\,\mathrm{tx}}^{\mathrm{rem}}$

For a drone k at data-rate $R_k(B_k)$ [bits/s], the time necessary to transfer D_k bits is computed by

$$T_{k;\,\mathrm{tx}}^{\mathrm{rem}}\,(B_k) = \frac{D_k}{R_k\,(B_k) + \varepsilon} \qquad [\,\mathrm{s}]. \tag{8}$$

4) COMPUTING T_k^{rem}

The total service for a drone k that is admitted to executing its task remotely is thus given by

$$T_k^{\mathrm{rem}}\,\left(B_k, f_{0,\,k}\right) = T_{k;\,\mathrm{ex}}^{\mathrm{rem}}\,\left(f_{0,\,k}\right) + T_{k;\,\mathrm{tx}}^{\mathrm{rem}}\,(B_k) \qquad [\,\mathrm{s}]. \tag{9}$$

Therefore, the total service latency of a drone k can be formulated by

$$T_k\,\left(x_k,\, B_k,\, f_k,\, f_{0,\,k}\right) = \begin{cases} T_k^{\mathrm{rem}}\,\left(B_k, f_{0,k}\right) & \text{if } x_k = 1; \\ T_k^{\mathrm{loc}}\,(f_k) & \text{otherwise}, \end{cases} \qquad [\,\mathrm{s}]. \tag{10}$$

B. ENERGY CONSUMPTION

Total energy consumption includes the energy consumption for offloading a task, the energy consumption by computation resources, and the energy consumption by flight operations and wireless network electronic circuits at both BS and drone sides. Here, first, we formulate different terms of energy consumption for RCM.

1) ENERGY CONSUMPTION OF RCM

a: COMPUTING $E_{k;\,\mathrm{tx}}^{\mathrm{rem}}$

The total radiative and non-radiative energy consumption for task offloading is evaluated based on the wireless terminal power consumption model proposed by [44] and [45]. In the adopted model, the energy consumption for a transmission includes transmitting power level and network electronic circuits. It is evaluated as

$$E_{k;\,\mathrm{tx}}^{\mathrm{rem}}\,(B_k) = \left(P_k^{\mathrm{c}} + P_k^{\mathrm{BB}} + P_k^{\mathrm{RF}} + p_k\right) T_{k;\,\mathrm{tx}}^{\mathrm{rem}}\,(B_k) \quad [\mathrm{Joule}] \tag{11}$$

where

TABLE 2. Hardware characterization constants.

Parameter	Symbol	Value
Radio frequency and digital signal processing	ω	10^{-14} Joule
Baseband processing	ν	10^{-14} Joule
Load-dependent power consumption	γ	10^{-15} Joule/bit
Fixed circuit power	δ	2.5 W

- p_k: quantifies the fixed radiative transmit power during the time required to offload the task.
- P_k^{c} is the power consumption of the active transmission chain in the connected mode (e.g. power amplifier). It is equal to 1.35 W, according to experimental assessments in [44].
- P_k^{BB} quantifies the power consumption of the BB components (e.g. encoder circuit). It depends on the uplink data rate $R_k(B_k)$, and thus B_k, following the corresponding equation in Table 1. Note that, in the corresponding equation in Table 1, the value of input parameter $R_k(B_k)$ is in [Mbits/s], and the result of P_k^{BB} is in [mW] that needs to be converted to [W] in (11).
- P_k^{RF} is the power consumption of radio RF components (e.g. modulation circuit). It depends on the transmission power p_k following the equations and conditions in Table 1. As in our system model, the value of p_k is fixed, and the value of P_k^{RF} is fixed.

b: COMPUTING $E_{k;\,\mathrm{ex}}^{\mathrm{rem}}$

Under the assumption of low CPU voltage, the energy consumption for each CPU cycle at frequency f is computed by $E^{cyc}(f) = \mu f^2$ [50], [51] in [Joule/CPU-cycle], where μ is a constant determined by the computation electronic circuits. Therefore, drone k's energy consumption to execute the task on its VM is computed by

$$E_{k;\,\mathrm{ex}}^{\mathrm{rem}}\left(f_{0,\,k}\right) = \mu f_{0,\,k}^2 C_k D_k \qquad \text{[Joule].} \tag{12}$$

c: COMPUTING $E_{k;\,\mathrm{rx}}^{\mathrm{rem}}$

To receive a task, the BS needs to consume power for the signal processing and decoding. This amount is proportional to the data-rate and the bandwidth. A realistic model is derived by [46] and [47] as follows

$$E_{k;\,\mathrm{rx}}^{\mathrm{rem}}(B_k) = (\delta + \omega B_k + \nu B_k + \gamma R_k\,(B_k))\, T_{k;\,\mathrm{tx}}^{\mathrm{rem}}(B_k) \qquad \text{[Joule]} \tag{13}$$

where $\omega, \nu, \gamma, \delta \geq 0$ are hardware characterization constants that are specified in Table 2.

d: COMPUTING $E_{k;\,\mathrm{on}}^{\mathrm{rem}}$

It includes the power required by flight operation and electronic circuits to keep the BS and the drone functioning. To quantify the power consumption incurred by flight operation, W_k, since it is assumed that drones are stopped during task offloading and execution, based on Newton's second law, only vertical force must be balanced with the drone's weight. That is, $W_k = m_k g$ with m_k being the drone k's mass in [Kg] and $g = 9.8$ m/s^2 being the gravity acceleration. The whole amount of non-radiative power consumption to keep different electronic circuits on edge nodes, the fog node and the BS functioning strictly depend on their architecture. For example, the power consumption of a RAM on an edge node strictly depends on its frequency and electronic structure. Work [52] provides a model for the power consumption of edge nodes of various architectures. Instead, the authors in [53] develop realistic power models for BSs with different architectures with a focus on component level, e.g., power amplifier and cooling equipment. Actually, the power consumption of some components, e.g., cooling equipment, strictly depends on the environmental condition, and its amount could even be zero. We denote by $P_{\mathrm{on}}^{\mathrm{rem}}$ the total amount of non-radiative power consumption on the drone, the fog node, and the BS. For $P_{\mathrm{on}}^{\mathrm{rem}}$ we will set a constant number based on the realistic models introduced in the existing literature, e.g., [52], [53], [54], and [55]. The amount of energy $E_{k;\,\mathrm{on}}^{\mathrm{rem}}$ is computed by the total power consumption W_k and $P_{\mathrm{on}}^{\mathrm{rem}}$ within the service latency of the RCM. Thus, we simply formulate $E_{k;\,\mathrm{on}}^{\mathrm{rem}}$ as

$$E_{k;\,\mathrm{on}}^{\mathrm{rem}}\left(B_k, f_{0,\,k}\right) = \left(W_k + P_{\mathrm{on}}^{\mathrm{rem}}\right) T_k^{\mathrm{rem}}\left(B_k, f_{0,\,k}\right) \qquad \text{[Joule].} \tag{14}$$

Thus, the total energy consumption of a drone for RCM is computed by

$$\begin{aligned} E_k^{\mathrm{rem}}\left(B_k, f_{0,\,k}\right) = E_{k;\,\mathrm{tx}}^{\mathrm{rem}}\,(B_k) + E_{k;\,\mathrm{ex}}^{\mathrm{rem}}\left(f_{0,\,k}\right) \\ + E_{k;\,\mathrm{rx}}^{\mathrm{rem}}(B_k) + E_{k;\,\mathrm{on}}^{\mathrm{rem}}\left(B_k, f_{0,\,k}\right) \qquad \text{[Joule].} \end{aligned} \tag{15}$$

2) ENERGY CONSUMPTION OF LCM

We formulate different terms of energy consumption for LCM as follows

a: COMPUTING $E_{k;\,\mathrm{ex}}^{\mathrm{loc}}$

Similar to $E_{k;\,\mathrm{ex}}^{\mathrm{rem}}$, the energy required to execute the task of a drone on the local edge node is computed by

$$E_{k;\,\mathrm{ex}}^{\mathrm{loc}}\,(f_k) = \mu f_k^2 C_k D_k \qquad \text{[Joule].} \tag{16}$$

b: COMPUTING $E_{k;\,\mathrm{on}}^{\mathrm{loc}}$

Similar to $E_{k;\,\mathrm{on}}^{\mathrm{rem}}$, the total amount of energy consumption incurred by flight operation and non-radiative power consumption of local electronic components on the drone is computed by

$$E_{k;\,\mathrm{on}}^{\mathrm{loc}}(f_k) = \left(W_k + P_{\mathrm{on}}^{\mathrm{loc}}\right) T_k^{\mathrm{loc}}(f_k) \qquad [\mathrm{Joule}], \qquad (17)$$

where we consider a fixed value for $P_{\mathrm{on}}^{\mathrm{loc}}$ in [W] based on the realistic power consumption models provided by [52] and [53]. Thus, the total energy consumption of a drone for LCM is computed by

$$E_k^{\mathrm{loc}}(f_k) = E_{k;\,\mathrm{ex}}^{\mathrm{loc}}(f_k) + E_{k;\,\mathrm{on}}^{\mathrm{loc}}(f_k) \qquad [\mathrm{Joule}]. \qquad (18)$$

Then, the total energy consumption is formulated by

$$E_k\left(x_k,\, B_k,\, f_k,\, f_{0,\,k}\right) = \begin{cases} E_k^{\mathrm{rem}}\left(B_k,\, f_{0,\,k}\right) & \text{if } x_k = 1; \\ E_k^{\mathrm{loc}}\left(f_k\right) & \text{otherwise,} \end{cases} \qquad [\mathrm{Joule}]. \qquad (19)$$

IV. JOINT LATENCY-ENERGY OPTIMIZATION PROBLEM

Given what we outlined about the network model, the service latency, and energy consumption objectives, we define our joint optimization problem here. The paper has to address this optimization problem for each drone k. The bi-objective optimization problem formulates the joint minimization of two objectives in (20), as shown at the bottom of the page, where $\mathbf{x} = \{x_k\}_{\forall k}$, $\mathbf{B} = \{B_k\}_{\forall k}$, and $\mathbf{f}_0 = \{f_{0,\,k}\}_{\forall k}$. As it can be seen, the drones are mutually coupled through variables $\mathbf{x}$, $\mathbf{f}_0$, and $\mathbf{B}$ in global constraints (20d) while all other constraints and the utility function are local for each drone k. We observe that the minimization problem (20) is not a standard optimization problem, so tackling it using conventional methods is impossible. In the following, we convert it to a standard problem to tackle it using conventional convexity programming methods.

A. REFORMULATION OF THE JOINT LATENCY-ENERGY OPTIMIZATION PROBLEM

As discussed above, the service latency and the energy consumption are in general contrasting objectives. Thus, simultaneously minimizing both objectives in (20) is impossible. Instead, one of the most widely-used methods is the weighted Tchebysheff method that minimizes the maximum between a weighted combination of the objectives [41]. To use the Tchebysheff method, first, we denote $\underline{T}_k$ and $\underline{E}_k$ as the minimum possible service latency and energy consumption, respectively, a drone k can achieve. The value of $\underline{T}_k$ is given by solving

$$\underline{T}_k = \min_{x_k \;\&\; B_k \le \overline{B}_0 \;\&\; f_k \le \overline{f}_k \& f_{0,\,k} \le \overline{f}_0} T_k\left(x_k,\, B_k,\, f_k,\, f_0\right) \qquad [\,\mathrm{s}]. \qquad (21)$$

By analyzing $T_k\left(x_k,\, B_k,\, f_k,\, f_0\right)$, it is easy to see that the service latency is a decreasing function of f_k, $f_{0,\,k}$, and $R_k(B_k)$. Moreover, inspecting the first derivative of the data-rate $R_k(B_k)$ with respect to B_k and exploiting that $y/(1+y) < \log(1+y)$ for any $y > 0$ shows that $R_k(B_k)$ is an increasing function of B_k. This yields, the minimum value $\underline{T}_k$ is achieved at the maximum values of f_k, $f_{0,\,k}$, and B_k. Therefore, we can write

$$\underline{T}_k = \begin{cases} \underline{T}_k^{\mathrm{rem}} \triangleq T_k^{\mathrm{rem}}\left(B_k = \overline{B}, f_{0,k} = \overline{f}_0\right) & \text{if } x_k = 1; \\ \underline{T}_k^{\mathrm{loc}} \triangleq T_k^{\mathrm{loc}}\left(f_k = \overline{f}_k\right) & \text{otherwise,} \end{cases} \qquad [\,\mathrm{s}]. \qquad (22)$$

Similarly, $\underline{E}_k$ is given by solving (23), as shown at the bottom of the page, where $\underline{E}_k^{\mathrm{loc}}$ is simply computed by setting its first-order derivative to zero. However, differently from $\underline{E}_k^{\mathrm{loc}}$, for $\underline{E}_k^{\mathrm{rem}}$ does not exist a closed form solution. We will discuss its convexity later and solve it using conventional optimization methods.

For the case of problem (20), the weighted Tchebysheff approach leads to considering the equivalent min-max problem in (24), as shown at the bottom of the next page, where $\eta \in (0,\, 1)$ can be interpreted as the "weight" that quantifies our desire to make each objective small or large, e.g., if we care much less about the service latency, we can take η small. It serves to find a balance between energy consumption and service latency in our model. From drone k's viewpoint, for any $\eta \in (0,\, 1)$, (24) has at least one solution that is Pareto optimal for (24) [41, Ch. 3], and

$$\min_{\mathbf{x},\, \mathbf{B},\, f_k,\, \mathbf{f}_0} \left[T_k\left(x_k,\, B_k,\, f_k,\, f_{0,\,k}\right),\quad E_k\left(x_k,\, B_k,\, f_k,\, f_{0,\,k}\right)\right] \qquad (20a)$$

$$\text{s.t.}\;\; x_k \in \{-1,\, 1\}; \qquad (20b)$$

$$0 \le f_k \le \overline{f}_k\left(1 - x_k\right)/2 \;\;\&\;\; 0 \le f_{0,\,k} \le \overline{f}_0\left(1 + x_k\right)/2 \;\;\&\;\; 0 \le B_k \le \overline{B}_0\left(1 + x_k\right)/2 \qquad (20c)$$

$$\sum_{k=1}^{K}\left(1 + x_k\right)/2 \le N \;\;\&\;\; \sum_{k=1}^{K} f_{0,\,k} \le \overline{f}_0 \;\;\&\;\; \sum_{k=1}^{K} B_k \le \overline{B}_0\,. \qquad (20d)$$

$$\underline{E}_k = \begin{cases} \underline{E}_k^{\mathrm{rem}} \triangleq \displaystyle\min_{B_k \le \overline{B}_0 \;\&\; f_{0,\,k} \le \overline{f}_0} E_k^{\mathrm{rem}}\left(B_k,\, f_{0,\,k}\right) & \text{if } x_k = 1; \\ \underline{E}_k^{\mathrm{loc}} \triangleq E_k^{\mathrm{loc}}\left(f_k = \min\left(\sqrt[3]{\left(W_k + P_{\mathrm{on}}^{\mathrm{loc}}\right)/2\lambda}\;,\; \overline{f}_k\right)\right) & \text{otherwise,} \end{cases} \qquad [\mathrm{Joule}]. \qquad (23)$$

solving (24) for all $\eta \in (0, 1)$ yields all the points on the Pareto boundary of (24) [41, Theorem 3.4.5]. On the other hand, the two extreme points $\eta = 0$ and $\eta = 1$ correspond to the single-objective minimization of the energy consumption and of the service latency, respectively. At the end, reformulate (24) in epigraph form, yields in the following equivalent problem

$$\min_{\mathbf{x}, \mathbf{B}, f_k, \mathbf{f}_0, y_k} \quad y_k \tag{25a}$$

$$\text{s.t.} \quad (20b) \;\&\; (20c) \;\&\; (20d); \tag{25b}$$

$$T_k\left(x_k, B_k, f_k, f_{0,k}\right) \leq \frac{y_k}{\eta} + \underline{T}_k; \tag{25c}$$

$$E_k\left(x_k, B_k, f_k, f_{0,k}\right) \leq \frac{y_k}{1-\eta} + \underline{E}_k; \tag{25d}$$

where y_k is an auxiliary slack variable. Problem (25) gives us the opportunity to tackle easier the problem at end.

B. SELECTION OF TASK COMPUTING MODE

In the minimization problem (25), for a drone with LCM $x_k = -1$, the left hand side of constraints (25c) and (25d) are replaced by $T_k^{\text{loc}}(f_k)$ and $E_k^{\text{loc}}(f_k)$, respectively. Otherwise, for RCM, they become $T_k^{\text{rem}}\left(B_k, f_{0,k}\right)$ and $E_k^{\text{rem}}\left(B_k, f_{0,k}\right)$, respectively. We introduce y_k^{loc} and y_k^{rem} as the equivalents of y_k for $x_k = -1$ and $x_k = 1$, respectively. For $x_k = -1$, problem (25) becomes a simplified minimization problem as follows

$$\min_{f_k, y_k^{\text{loc}}} \quad y_k^{\text{loc}} \tag{26a}$$

$$\text{s.t.} \quad 0 \leq f_k \leq \overline{f}_k; \tag{26b}$$

$$T_k^{\text{loc}}(f_k) \leq \frac{y_k^{\text{loc}}}{\eta} + \underline{T}_k^{\text{loc}}; \tag{26c}$$

$$E_k^{\text{loc}}(f_k) \leq \frac{y_k^{\text{loc}}}{1-\eta} + \underline{E}_k^{\text{loc}}; \tag{26d}$$

where $\underline{T}_k^{\text{loc}}$ and $\underline{E}_k^{\text{loc}}$ come from (22) and (23), respectively. The constraints and the objective function in (26) depend only on local variables of drone k, so each drone can solve problem (26) independently to find its optimal values $(f_k^*, y_k^{\text{loc},*})$ without interacting with the other drones. For $x_k = -1$, the value of $T_k^{\text{loc}}(f_k^*)$ and $E_k^{\text{loc}}(f_k^*)$ are computed and used to find the Pareto boundary between service latency and energy consumption at a given η.

From the network's viewpoint, it is obvious that the preferable mode is the LCM. A drone will opt for the RCM only if it can achieve better performance by considering the weight η. We can define a threshold to select the best computing mode for a task as follows[1]

$$\nu_k^{\text{loc},*} = \max\left[\eta T_k^{\text{loc}}(f_k^*),\; (1-\eta)\, E_k^{\text{loc}}(f_k^*)\right]. \tag{27}$$

and similarly, we define the following *variable* for RCM:

$$\nu_k^{\text{rem}} = \max\left[\eta T_k^{\text{rem}}\left(B_k, f_{0,k}\right),\; (1-\eta)\, E_k^{\text{rem}}\left(B_k, f_{0,k}\right)\right] \tag{28}$$

As a decision strategy, a drone k will select the RCM only if ν_k^{rem} achieves a value less than the fixed threshold $\nu_k^{\text{loc},*}$. This translates to checking whether the resource allocation optimization problem can guarantee the following constraint

$$\Delta\nu_k \triangleq \nu_k^{\text{rem}} - \alpha\nu_k^{\text{loc},*} \leq 0 \tag{29}$$

where the positive constant α is a network-centric parameter. Setting a positive value close to zero for α forces the drones to choose the LCM. Conversely, assigning a large value to α will result in the drones selecting RCM. Assigning RCM to all drones is a realistic approach when executing tasks requires data integration across multiple activities or interaction with a centralized database. This approach is applicable in various fields, such as national security, health monitoring environments, and disaster management [56]. To fully exploit the network resources and achieve optimal performance, the parameter α should be set to a value around one. By imposing constraint (29) and tuning α, we can assign available wireless channels and fog nodes to the most beneficial drones for task offloading.

Therefore, we introduce the minimization problem shown in (30), as shown at the bottom of the next page, to address this issue. In order to ensure that constraint (30d) can be satisfied, we need to add a new constraint (30d) to the joint minimization problem in equation (25). Additionally, we must add constraints to (30e) for the variable ν_k^{rem}. This converts the optimization problem in (25) into the minimization problem presented in (30). Here, ε is a small positive constant close to zero that ensures $-1 \leq \varepsilon\Delta\nu_k \leq 1$. If $\Delta\nu_k \leq 0$, constraint (30d) is converted to $0 \leq (1 - x_k)/2$. Maximizing x_k in the objective function results in x_k equaling one. Otherwise, if $\Delta\nu_k > 0$, constraint (30d) is converted to $0 < (1 - x_k)/2$, and therefore x_k will be set to -1. Thus, x_k will only equal one if condition (29) is met.

C. ACCESS STRATEGY TO THE BS

In the case of $K > N$, it is possible for constraint (29), which governs the task computing mode, $\Delta\nu_k \leq 0$, to be

[1] Note that, we cannot use $y_k^{\text{loc},*}$ instead of $\nu_k^{\text{loc},*}$ since, for example, the relation $T_k^{\text{loc}}(f_k^*) - \underline{T}_k^{\text{loc}} \leq T_j^{\text{loc}}(f_j^*) - \underline{T}_j^{\text{loc}}$ does not necessarily result in $T_k^{\text{loc}}(f_k^*) \leq T_j^{\text{loc}}(f_j^*)$.

$$\min_{\mathbf{x}, \mathbf{B}, f_k, \mathbf{f}_0} \quad \max\left[\eta\left(T_k\left(x_k, B_k, f_k, f_{0,k}\right) - \underline{T}_k\right),\quad (1-\eta)\left(E_k\left(x_k, B_k, f_k, f_{0,k}\right) - \underline{E}_k\right)\right] \tag{24a}$$

$$\text{s.t.} \quad (20b) \;\&\; (20c) \;\&\; (20d), \tag{24b}$$

satisfied by more than N drones. This scenario implies that there are not enough wireless channels to offload all tasks concurrently. The optimal solution is to allocate N wireless channels to N drones that can benefit the most and achieve the minimum ν_k^{rem}. In this regard, for drones with $\Delta\nu_k \leq 0$, the values of ν_k^{rem} are sorted in ascending order, and only the first N drones are assigned to RCM, while the rest are assigned to LCM. To enable individual computing mode selection for each drone, while considering the values of ν_j^{rem} of the other drones with $j \neq k$, we devise the following strategy.

For a drone k where the constraint $\Delta\nu_k \leq 0$ is feasible, it checks whether it is in the first N most beneficial drones for task offloading. This is determined by comparing the number of opponents $j \neq k$ with $\Delta\nu_j \leq 0$ and $\nu_j^{\text{rem}} \leq \nu_k^{\text{rem}}$ to the number of available wireless channels N. If the number of such opponents is greater than or equal to N, it means that the drone k is not beneficial for the task offloading and sets $x_k = -1$. To solve this access strategy problem for each drone k, we introduce minimization problem (31), as shown at the bottom of the page, where the drone optimizes x_k individually, taking into consideration the values of $\{\nu_j^{\text{rem}}\}_{\forall j}$. To improve readability, we introduce the optimization problem in (31) as a separate problem. However, it will actually be integrated into the main problem defined in (30) without any conflicts between the constraints and objectives. When $K > N$, the minimization problem for each drone k can be defined as shown in (31). Here, $\mathbf{n}_k = n_{kj}$ for all $j \neq k$ are binary slack variables, $\boldsymbol{\nu} = \{\nu_k^{\text{rem}}\}_{\forall k}$, and ε is again a small positive constant close to zero. For a drone $j \neq k$, if both conditions $\Delta\nu_j \leq 0$ and $\nu_j^{\text{rem}} \leq \nu_k^{\text{rem}}$ are met, then constraints (31c) and (31d) will be converted to $0 \leq (1-n_{kj})/2$. Maximizing n_{kj} results in it being set to 1. On the other hand, if either of the conditions does not hold, then the constraint will be converted to $0 < (1-n_{kj})/2$, and n_{kj} will be set to -1. In this way, $\sum_{j\neq k}(1+n_{kj})/2$ gives the number of drones $j \neq k$ with $\Delta\nu_j \leq 0$ that will be more beneficial than drone k for task offloading. Drone k will be assigned for task offloading only if both conditions $\sum_{j\neq k}(1+n_{kj})/2 \leq N-1$ and $\Delta\nu_k \leq 0$ are satisfied. This can also be verified in constraint (31e). It is worth noting that constraint (30d) can be substituted with (31e). Lastly, it is important to note that the drones are mutually coupled through variables $\boldsymbol{\nu}$ in the global constraint defined in (31c). Summarizing the problem at the end, the minimization problem of each drone is defined as

If $K \leq N$: Problem (30)	If $K > N$: Problem [(30), (31)]

This approach provides a fair and efficient way to allocate tasks among multiple drones with limited wireless resources. By allowing each drone to individually select its computing mode based on the availability of wireless channels and the remaining computation workload of other drones, the

$$\min_{\mathbf{B},\,\mathbf{x},\,\mathbf{f}_0,\,\nu_k^{\text{rem}},\,y_k^{\text{rem}}} \quad y_k^{\text{rem}} + \nu_k^{\text{rem}} - x_k \tag{30a}$$

$$\text{s.t.} \quad \text{Global constraints in (20d)}; \tag{30b}$$

$$x_k \in \{-1,\ 1\}; \tag{30c}$$

$$\max\left(\varepsilon\Delta\nu_k,\ 0\right) \leq \left(1-x_k\right)/2; \tag{30d}$$

$$T_k^{\text{rem}}\left(B_k,\ f_{0,k}\right) \leq \frac{\nu_k^{\text{rem}}}{\eta} \quad \& \quad E_k^{\text{rem}}\left(B_k,\ f_{0,k}\right) \leq \frac{\nu_k^{\text{rem}}}{1-\eta}; \tag{30e}$$

$$0 \leq f_{0,k} \leq \overline{f}_0\left(1+x_k\right)/2 \quad \& \quad 0 \leq B_k \leq \overline{B}_0\left(1+x_k\right)/2; \tag{30f}$$

$$T_k^{\text{rem}}\left(B_k,\ f_{0,k}\right) \leq \frac{y_k^{\text{rem}}}{\eta} + \underline{T}_k^{\text{rem}}; \tag{30g}$$

$$E_k^{\text{rem}}\left(B_k,\ f_{0,k}\right) \leq \frac{y_k^{\text{rem}}}{1-\eta} + \underline{E}_k^{\text{rem}}; \tag{30h}$$

$$\min_{x_k,\,\boldsymbol{\nu},\,\mathbf{n}_k} \quad -\sum_{j\neq k} n_{kj} \tag{31a}$$

$$\text{s.t.} \quad n_{kj} \in \{-1,\ 1\}; \qquad \forall j \neq k \tag{31b}$$

$$\varepsilon\Delta\nu_j \leq \left(1-n_{kj}\right)/2; \qquad \forall j \neq k \tag{31c}$$

$$\varepsilon\left(\nu_j^{\text{rem}} - \nu_k^{\text{rem}}\right) \leq \left(1-n_{kj}\right)/2; \qquad \forall j \neq k \tag{31d}$$

$$\max\left(\varepsilon\left(\sum_{j\neq k}\left(1+n_{kj}\right)/2 - N + 1\right),\ \varepsilon\Delta\nu_k,\ 0\right) \leq \left(1-x_k\right)/2 \tag{31e}$$

approach maximizes the total offloading benefit while avoiding congestion and ensuring fairness. The proposed method can be used in various applications, such as surveillance, search and rescue, and environmental monitoring, where multiple drones need to coordinate their tasks to achieve a common goal.

V. CONVEXIFICATION OF THE MINIMIZATION PROBLEM

The goal of our resource allocation problem is to share available bandwidth and the CPU capacity of the fog node among drones. As discussed above, the impacts of the variables B_k and $f_{0,\,k}$ on different terms of service latency and energy consumption are contrasting and so, we need to jointly minimize both objectives to achieve the best tradeoff between them. To this end, in Sect. VI, we will model the competition among drones to achieve common resources using game theory. Before, as it is needed by any optimization algorithm, we check the convexity of the minimization problem of each drone k. The convexity of the terms $T_k^{\text{loc}}(f_k)$, $E_k^{\text{loc}}(f_k)$, $E_{k;\,\text{ex}}^{\text{rem}}\left(f_{0,\,k}\right)$, and $E_{k;\,\text{on}}^{\text{rem}}\left(B_k, f_{0,\,k}\right)$ are simple to verify by inspecting the second order derivative.

A. CONVEXIFICATION OF BINARY VARIABLES $\mathbf{x_k}$ AND $\mathbf{n_k}$

We know that a binary variable is inherently non-convex. In order to release this issue on $\Lambda_k = \{x_k, \mathbf{n}_k\}$, we use the method presented by [57]. We denote a slack vector variable $\mathbf{v} = \{v_k, \mathbf{v}_k\} \in \mathbb{R}^K$ with $\mathbf{v}_k \triangleq \{v_{kj}\}_{j \neq k}$ where v_k corresponds to x_k and v_{kj} to n_{kj}. Following the method, the following constraints need to be added

$$\|\mathbf{v}\|_2^2 \leq K \quad \& \quad -\mathbf{1} \leq \Lambda_k \leq \mathbf{1} \quad \& \quad v_k x_k + \mathbf{v}_k^T \mathbf{n}_k = K. \tag{32}$$

The minimization problem can be solved iteratively over $\mathbf{x}$ and $\mathbf{v}$ alternatively. The convergence of the algorithm is guaranteed at the point $\mathbf{v} = \Lambda_k$. At each iteration q, first, given $\mathbf{v}^q$, we solve the minimization problem as a function of Λ_k to obtain Λ_k^{q+1}. Then, the following optimal solution for $\mathbf{v}$ is defined [57]

$$\mathbf{v}^{q+1} = \begin{cases} \sqrt{N}\Lambda_k^{q+1}/\|\Lambda_k^{q+1}\|_2, & \Lambda_k^{q+1} \neq 0; \\ \text{any } \Lambda_k \text{ with } \|\Lambda_k\|_2^2 \leq N, & \text{otherwise.} \end{cases} \tag{33}$$

B. CONVEXITY CHECK OF $T_k^{\text{rem}}(B_k, f_{0,\,k})$

The term $T_{k;\,\text{ex}}^{\text{rem}}\left(f_{0,k}\right)$ is a convex function of $f_{0,\,k}$. By neglecting inessential constant terms with respect to B_k, the term $T_{k;\,\text{tx}}^{\text{rem}}(B_k)$, from convexity viewpoint, has the form $1/R_k(B_k)$. By exploiting the second-order derivative, we find that $R_k(B_k)$ is a concave function in B_k. As $R_k(B_k)$ is a strictly positive function, its reciprocal is a convex one [40]. Therefore, $T_k^{\text{rem}}\left(B_k, f_{0,\,k}\right)$ is jointly convex with respect to both variables $f_{0,\,k}$ and B_k.

C. CONVEXITY CHECK OF $T_{k;\,\text{tx}}^{\text{rem}}(B_k)$

By neglecting inessential constant terms with respect to B_k, the term $T_{k;\,\text{tx}}^{\text{rem}}(B_k)$ has the form $1/R_k(B_k)$ that is a convex function.

D. CONVEXITY CHECK OF $E_{k;\,\text{rx}}^{\text{rem}}(B_k)$

In $E_{k;\,\text{rx}}^{\text{rem}}(B_k)$ (13) we set $R_k(B_k)/(R_k(B_k) + \varepsilon) = 1$. By neglecting inessential constant terms with respect to B_k, the energy consumption $E_{k;\,\text{rx}}^{\text{rem}}(B_k)$ has the form $B_k/R_k(B_k)$. By inspecting the second order derivative and exploiting that $1/(1+x) < \log(1+1/x) < 1/x$ for $x > 0$, it can be seen that $B_k/R_k(B_k)$ is a concave function. To tackle the non-convexity of $E_{k;\,\text{rx}}^{\text{rem}}(B_k)$, inspired by [58] and [59], we substitute it by an equivalent convex function as

$$E_{k;\,\text{rx}}^{\text{rem}}(B_k;\, c) = (\omega + \nu)\left(c - \frac{B_k}{R_k\left(B_k\right)}\right) + \gamma D_k \tag{34}$$

where $c > 0$ is given and at each iteration t it is being updated iteratively by

$$c^{t+1} = \frac{2B_k^t}{R_k\left(B_k^t\right)}. \tag{35}$$

Therefore, in the RCM, the total energy consumption of each drone is converted to the following convex function

$$\begin{aligned} E_k^{\text{rem}}\left(B_k, f_{0,\,k};\, c\right) = E_{k;\,\text{tx}}^{\text{rem}}\left(B_k\right) + E_{k;\,\text{ex}}^{\text{rem}}\left(f_{0,\,k}\right) \\ + E_{k;\,\text{rx}}^{\text{rem}}(B_k;\, c) + E_{k;\,\text{on}}^{\text{rem}}\left(B_k, f_{0,\,k}\right). \end{aligned} \tag{36}$$

Now, the formula $\underline{E}_k$ (23) is rewritten as the following convex problem

$$\underline{E}_k = \begin{cases} \min\limits_{B_k \leq \overline{B}_0 \;\&\; f_{0,\,k} \leq \overline{f}_0} E_k^{\text{rem}}\left(B_k, f_{0,\,k};\, c\right) & \text{if } x_k = 1; \\ \min\limits_{f_k \leq \overline{f}_k} E_k^{\text{loc}}\left(f_k\right) & \text{otherwise,} \end{cases} \quad [\text{Joule}]. \tag{37}$$

Then, constraint (30h) is converted to

$$E_k^{\text{rem}}\left(B_k, f_{0,\,k};\, c\right) \leq \frac{y_k^{\text{rem}}}{1-\eta} + \underline{E}_k. \tag{38}$$

For convexification, while most existing papers in the literature use the SCA method, there is always a gap between the original non-convex function and the converted convex one. In general, the gap is unknown and it could be large since the SCA method replaces a non-convex function with its first-order Taylor series. It is important to emphasize that, here, $E_k^{\text{rem}}\left(B_k, f_{0,\,k};\, c\right)$ is an equivalent convex function of $E_k^{\text{rem}}\left(B_k, f_{0,\,k}\right)$ and there is no any gap between them. Therefore, our convexification process does not result in optimality degradation.

E. CONVEX OPTIMIZATION PROBLEM OF EACH DRONE

Here, we define the feasible set of each drone. We integrate optimization problems (30) and (31) and define the convex feasible domain of each drone. Since min function is concave and max function is convex [40], we can see the convexity of the constraints in (30) and (31). We denote by $\mathcal{F}$ and $\mathcal{L}_k$ the global and local constraints related to drone k, respectively, which are defined by

$$\mathcal{F} \triangleq \{(20\text{d}) \;\&\; (31\text{c}) \;\&\; (31\text{d})\} \tag{39}$$

and

$$\mathcal{L}_k \triangleq \{(30d)\ \&\ (30e)\ \&\ (30f)\ \&\ (30g)\ \&\ (31e)\ \&\ \&\ (32)\ \&\ (38)\}. \tag{40}$$

Finally, the minimization problem of each drone k is rewritten to the following convex one

$$\min_{\mathbf{B},\,\mathbf{x},\,\mathbf{f}_0,\,\mathbf{v},\,\mathbf{n}_k,\,\boldsymbol{\nu}} \quad y_k^{\text{rem}} + \nu_k^{\text{rem}} - x_k - \sum_{j\neq k} n_{kj} \tag{41a}$$

$$\text{s.t.}\quad \mathcal{F}\ \&\ \mathcal{L}_k. \tag{41b}$$

Unlike sub-problem (26), in convex sub-problem (41), a drone k needs to exchange the information about the variables $(B_k, x_k, f_{0,k}, \nu_k^{\text{rem}})$ to handle the global constraints in $\mathcal{F}$. In the next section, we will model the interaction among the drones in (25) using non-cooperative game theory and introduce an *iterative distributed algorithm* to compute the NE. For later convenience, we denote $w_k \triangleq (B_k, x_k, f_{0,k}, \nu_k^{\text{rem}})$ as the variables of drone k which are being exchanged with other drones and by $\mathbf{w} \triangleq (\mathbf{B}, \mathbf{x}, \mathbf{f}_0, \boldsymbol{\nu})$ as the global variables through which the drones are mutually coupled. We also denote by $\psi_k \triangleq (\mathbf{w}, \mathbf{n}_k, \mathbf{v})$ as the state of each drone k in sub-problem (41). We further denote the feasible set of ψ_k by

$$\mathcal{S}_k \triangleq \left\{\psi_k \in \mathbb{R}_+^{6K-1} : \text{All the constraints in } \{\mathcal{F}\ \&\ \mathcal{L}_k\}\right\}. \tag{42}$$

The convex optimization problem (feasible set) of each drone k is thus rewritten as

$$\min_{\psi_k}\ u_k(\psi_k) \tag{43a}$$

$$\text{s.t.}\ \psi_k \in \mathcal{S}_k \tag{43b}$$

where $u_k(\psi_k)$ denotes the objective function of drone k in (41). As optimization problem (43) is convex, based on the weighted Tchebysheff theorem, for each $\eta \in (0, 1)$ it has a unique solution that is (strong) Pareto-optimal [41, Ch. 3]. Therefore, by solving optimization problem (43) for all values of η in the range of $(0, 1)$, we will obtain the Pareto boundary between the service latency and energy consumption for a drone.

The present study investigates the problem of joint radio and computing resource allocation in fog-assisted drone communications, with a focus on modeling the competition among drones to access the available resources. To this end, we adopt a non-cooperative game-theoretic framework in our analysis. The key difference between a cooperative and a non-cooperative approach lies in the extent to which the players can be induced to cooperate. In a non-cooperative game, cooperation is typically achieved as a result of self-optimization, as unilateral deviations are not beneficial for the players. Conversely, in a cooperative game, the players are willing to cooperate and seek to reach a mutually beneficial agreement. However, we note that algorithms based on cooperative games suffer from significant computational complexity and scalability issues, especially when the number of variables and constraints is large [60], [61]. Given that our minimization problem (43) involves a large number of variables and constraints, we need to opt for a non-cooperative game-theoretic framework, which is better suited to handle such scenarios. Specifically, we choose a non-cooperative game-theoretic model as it allows for efficient and scalable solutions to the resource allocation problem, even in scenarios with a large number of variables.

VI. NON-COOPERATIVE GAME FORMULATION

For a given η, a decentralized resource allocation algorithm looks for the solution of the following K problems that are mutually coupled through the variables $\mathbf{w}$ in global constraints $\mathcal{F}$:

$$\arg\min_{\psi_k \in \mathcal{S}_k(\mathbf{w}_{\setminus k})}\ u_k(\psi_k) \tag{44}$$

where $\mathcal{S}_k\left(\mathbf{w}_{\setminus k}\right)$ is the feasible set $\mathcal{S}_k$ (42) given $\mathbf{w}_{\setminus k} \triangleq \{w_j\}_{\forall j\neq k}$. That is the values of w_js of all the competitors of drone k. Note that in (44) we write $u_k(\psi_k)$ instead of, as usual in the literature, $u_k\left(\psi_k;\ \mathbf{w}_{\setminus k}\right)$ since $\mathbf{w}_{\setminus k}$ does not impact the objective function but the constraints in $\mathcal{S}_k$. We begin by modeling the interactions between K drones as a non-cooperative game wherein each drone k wishes to minimize its utility function $u_k(\psi_k)$ while satisfying the associated constraints.

A. GAME FORMULATION

We denote our non-cooperative game by

$$\mathcal{G} \triangleq \left\{\{k\}_{k=1}^{K},\ \left\{\mathcal{S}_k\left(\mathbf{w}_{\setminus k}\right)\right\}_{\forall k},\ \{u_k(\psi_k)\}_{\forall k}\right\} \tag{45}$$

The K coupled problems in (44) define the best-response dynamics (BRD) of the game, while the solution of the k-th problem in (44) is the k-th drone's best-response to the other drones. We recall that, the best response $\mathsf{br}_k(\mathbf{w}_{\setminus k})$ of each drone k to $\mathbf{w}_{\setminus k}$ is defined as

$$\mathsf{br}_k\left(\mathbf{w}_{\setminus k}\right) \triangleq \arg\min_{\psi_k \in \mathcal{S}_k(\mathbf{w}_{\setminus k})}\ u_k(\psi_k). \tag{46}$$

In general, a non-cooperative game may have zero, one, or more (pure) equilibria. It is not guaranteed that the BRD converges to an equilibrium even if one or more equilibria exist. Therefore, it is essential to establish the existence and uniqueness of an equilibrium and whether the BRD converges to an equilibrium. However, unlike conventional non-cooperative games, our scenario is more challenging because the drones' strategy sets are mutually coupled through the global constraints in $\mathcal{F}$. To address this challenge, we use the category of *generalized non-cooperative games* [62], [63], [64]. To proceed further, we recall that $\boldsymbol{\psi}^\star = \{\psi_k^\star\}_{\forall k}$ is a pure-strategy generalized NE (GNE) vector of the game $\mathcal{G}$, where each element $\psi_k^\star$ is the BRD $\mathsf{br}_k(\mathbf{w}_{\setminus k}^\star)$ to the coupled strategies $\mathbf{w}_{\setminus k}^\star$ chosen by the competitors. Any fixed point of the BRD is a GNE of the game.

B. ANALYSIS OF THE EQUILIBRIA

More restrictive conditions (than classical non-cooperative games) have to be fulfilled for a unique GNE to exist and for the best-response to converge. All this is addressed in the following.

Lemma 1 (Existence and Uniqueness of GNE): The game $\mathcal{G}$ admits a unique nonempty set of GNE points.

Proof: Observe that the existence of a GNE is guaranteed under the following first three assumptions and the uniqueness under the fourth condition [63]:

1) *Existence:* The drones' feasible action set $\mathcal{S}_k\left(\mathbf{w}_{\setminus k}\right)$ are nonempty, closed, convex, and contained in some compact set $\mathcal{C}_k$ for all feasible values for the vector $\mathbf{w}_{\setminus k}$.
2) *Existence:* The sets $\mathcal{S}_k\left(\mathbf{w}_{\setminus k}\right)$ vary continuously with $\mathbf{w}_{\setminus k}$ (in the sense that the graph of the set-valued correspondence $\mathbf{w}_{\setminus k} \rightarrow \mathcal{S}_k\left(\mathbf{w}_{\setminus k}\right)$ is closed).
3) *Existence:* Each drone's utility function is continuously differentiable in ψ_k given $\mathbf{w}_{\setminus k}$.
4) *Uniqueness:* The feasible set of each drone, minimization problem (43), is convex on ψ_k given $\mathbf{w}_{\setminus k}$.

In our setting, as discussed above, the constraints in $\mathcal{S}_k$ are convex. It is also simple to discuss that the set $\mathcal{S}_k\left(\mathbf{w}_{\setminus k}\right)$ is nonempty, closed, convex, and bounded for each $\mathbf{w}_{\setminus k}$. Moreover, each of them varies continuously in their variables. The non-convex constraints are convexified in Sect. V. Furthermore, as the utility functions of the drones are linear functions and independent of each other, they are continuously differentiable and convex functions. ■

In the following, we introduce an iterative algorithm to compute the unique GNE and prove its convergence. The following parts are derived from the results in [65] and [66].

C. COMPUTE THE UNIQUE GNE

Here, we represent the best-response algorithm to converge the unique GNE of our game $\mathcal{G}$. For convenience, first, we represent global convex constraints $\mathcal{F}$ in the standard form of convex constraints and we denote them by the vector $\mathbf{F}(\mathbf{w}) \leq \mathbf{0}$. Let us also denote by $\mathbf{F}(w_k;\ \mathbf{w}_{\setminus k})$ as a function of w_k given the values of $\mathbf{w}_{\setminus k}$. We introduce the following minimization problem for each drone k:

$$\min_{\psi_k}\ u_k\left(\psi_k\right) + \boldsymbol{\lambda}^T \mathbf{F}(w_k;\ \mathbf{w}_{\setminus k}) \tag{47a}$$

$$\text{s.t.}\ \ \psi_k \in \mathcal{S}_k\left(\mathbf{w}_{\setminus k}\right) \tag{47b}$$

From the duality theory [40] point of view, the vector $\boldsymbol{\lambda}$ is the Lagrangian multipliers associated with the global constraints $\mathbf{F}(w_k;\ \mathbf{w}_{\setminus k})$. By doing this, we have incorporated the global constraints as a part of the utility function. We will show that the drones, by distributively optimizing (47), obtain the GNE point. Equivalently, the BRD for each drone k is represented by

$$\mathbf{br}^t\left(\mathbf{w}_{\setminus k}\right) \triangleq \underset{\psi_k \in \mathcal{S}_k\left(\mathbf{w}_{\setminus k}\right)}{\arg\min}\ u_k\left(\psi_k\right) + \boldsymbol{\lambda}^T \mathbf{F}\left(w_k;\ \mathbf{w}^{t-1}_{\setminus k}\right) \tag{48}$$

Algorithm 1 Subgradient Projection Algorithm

```
solve convex problem (26) to obtain $\left(f_k^*,\ y_k^{\text{loc},*}\right)$ for
  all $k = 1, \ldots, K$;
initialize $t = 0$; step size sequence $\{\gamma^t\}$; $\boldsymbol{\lambda}^t \geq \mathbf{0}$; a
  feasible $\mathbf{x}^t$, $\mathbf{B}^t$, and $\mathbf{f}_0^t$;
repeat
    compute $\mathbf{br}_k^{t+1}\left(\mathbf{w}^t_{\setminus k}\right) =$
      $\underset{\psi_k \in \mathcal{S}_k\left(\mathbf{w}^t_{\setminus k}\right)}{\arg\min}\ u_k\left(\psi_k\right) + \boldsymbol{\lambda}^{t,T}\mathbf{F}\left(w_k;\ \mathbf{w}^t_{\setminus k}\right)\ \forall k$;
    update $\boldsymbol{\lambda}^{t+1} = \left[\boldsymbol{\lambda}^t + \gamma^t \mathbf{F}\left(\mathbf{w}^{t+1}\right)\right]^+$;
    set $t = t + 1$; //Iteration number;
until (convergence of $\boldsymbol{\lambda}$);
set $f_k^* = f_k^*\left(1 - x_k^t\right)/2$;  $f_{0,k}^* = f_{0,k}^t$;  $B_k^* = B_k^t$;  $\forall k$;
return $(f_k^*,\ f_{0,k}^*,\ B_k^*)$  $\forall k$; //GNE obtained at
  $\boldsymbol{\lambda}^t = \boldsymbol{\lambda}^\star$;
```

Algorithm 2 Solving $\mathbf{br}_k^{t+1}\left(\mathbf{w}^t_{\setminus k}\right)$ in Algorithm (1)

```
initialize $\mathbf{v}$ and $c$ by feasible values;
repeat
    repeat
        solve $\mathbf{br}_k^{t+1}\left(\mathbf{w}^t_{\setminus k}\right)$ while fixing $\mathbf{v}$, $\boldsymbol{\nu}$, $\mathbf{B}$ and $\mathbf{f}_0$;
        update $\mathbf{v}$ as in (33);
    until ($\Lambda_k = \mathbf{v}$);
    solve $\mathbf{br}_k^{t+1}\left(\mathbf{w}^t_{\setminus k}\right)$ while fixing $\mathbf{v}$ and $\Lambda_k$;
    update $c$ as in (35);
until (convergence of $\Lambda_k$ and $c$);
```

We denote by $\boldsymbol{\lambda}^\star$ as the optimal value of $\boldsymbol{\lambda}$. Of course, the value of $\boldsymbol{\lambda}^\star$ in (47) is unknown a priori, but it can be found by, for example, subgradient method [67], that is an iterative algorithm. Specifically, we can design a double-loop algorithm: *i)* in the inner loop, given the Lagrangian multiplier $\boldsymbol{\lambda} = \boldsymbol{\lambda}^t$ and the global variables of the competitors $\mathbf{w}_{\setminus k} = \mathbf{w}^t_{\setminus k}$, one distributively computes the unique optimal solution of (47) for each drone k and obtain $(w_k^{t+1},\ \mathbf{n}_k^{t+1},\ \mathbf{v}^{t+1})$; *ii)* in the outer loop, the Lagrangian multiplier $\boldsymbol{\lambda}^t$ is updated according to a subgradient-based projection method. Optimization problem (47) in the inner loop can be solved using standard convex optimization techniques, e.g. log-barrier method [40], or using existing optimization tool such as CVX [68] that is a Matlab-base convex optimization framework. We summarize this double-loop algorithm in Algorithm 1. The parameter γ^t is the tth step size. There are some well-known step size rules to guarantee the convergence to the value of $\boldsymbol{\lambda}^\star$. For example, one can use [67]

$$\gamma^t \to 0; \qquad \sum_{t=1}^{\infty} \gamma^t = \infty; \qquad \sum_{t=1}^{\infty} \left(\gamma^t\right)^2 < \infty. \tag{49}$$

One typical example is $\gamma^t = a/(b+t)$, where $a,\ b > 0$.

The flowchart in Figure 3 provides a visual representation of the integrated structure of Algorithms 1 and 2, illustrating the iterative process and convergence criteria involved in the problem transformation. According to our convexification methods in Sect. V, in the inner-loop optimization (Line 4) of Algorithm 1, there are two loops: one loop on the iteration for the convexification of the binary variables $\Lambda = \{x_k,\ \mathbf{n}_k\}$, and the second loop on the iteration for the parameter c in (35). The procedure is summarized in Algorithm 2. The best-response of $\mathsf{br}_k^{t+1}\left(\mathbf{w}_{\backslash k}^t\right)$ is computed using a standard convex optimization solution. In the inner-loop optimization (Line 4) of Algorithm 1, the only information required by the t-th individual problem is the aggregate trading vector of all the competitors, $\mathbf{w}_{\backslash k}^t$. In the outer loop of the algorithm, to update the Lagrangian vector $\boldsymbol{\lambda}^t$ (Line 5), the computing server needs to collect the optimized value of $\mathbf{w}^t$. For the convergence proof, we refer the reader to [65] and [66]. The proof is based on the features of potential games.

VII. NUMERICAL RESULTS

In this section, we present the simulation results to evaluate the performance of the proposed algorithm.

A. NETWORK SETUP

Consider the system model described in Section II, with system parameters as: $N_0 = -174$ dBm/Hz, $p_k = 10$ W, $\overline{f}_k = 1.2$ GHz, $\overline{f}_0 = 12$ GHz, the path loss exponent $\alpha = 2$, $\eta_{\text{NLoS}} = 23$ dB, $\eta_{\text{LoS}} = 3$ dB, $\psi = 11.95$, $\beta = 0.14$, and $f_c = 2.4$ GHz. The mass of a drone is set by $m_k \sim \texttt{Unif}[3,\ 4.5]$ Kg. We set the size of each task by $D_k \sim \texttt{Unif}[4,\ 10]$ MBytes and the complexity of each task by $C_k \sim \texttt{Unif}[100,\ 400]$ CPU-cycle/bit. Based on assessments in [52] and [55], practical amounts to $P_{\text{on}}^{\text{loc}} \sim \texttt{Unif}[2,\ 3.5]$ W and $P_{\text{on}}^{\text{rem}} \sim \texttt{Unif}[3,\ 5]$ W are set. By reading the specifications of various CPUs [69], [70], two types of CPUs, denoted by {CPU-A, CPU-B}, are chosen with power consumption coefficients $\mu = \{10^{-25},\ 10^{-22}\}$, respectively.

B. THEORETICAL PERFORMANCE

Figure 4 demonstrates the Pareto boundary between energy consumption and service latency for a network with $K = 4$ drones randomly located in a 70 m radius cell, considering different configurations with settings $\overline{B}_0 = \{1,\ 5\}$ MHz and two different CPUs. Each figure reports four curves. The blue solid and dashed lines depict the results of the proposed Algorithm 1 with variations $N = 4$ and $N = 2$, respectively, and by setting $\alpha = 1$ in computing strategy condition (29). The green dotted curve depicts the result of the proposed algorithm when $N = 4$ drones are forced to select RCM. To set $x_k = 1$ for all k, it is enough to set a

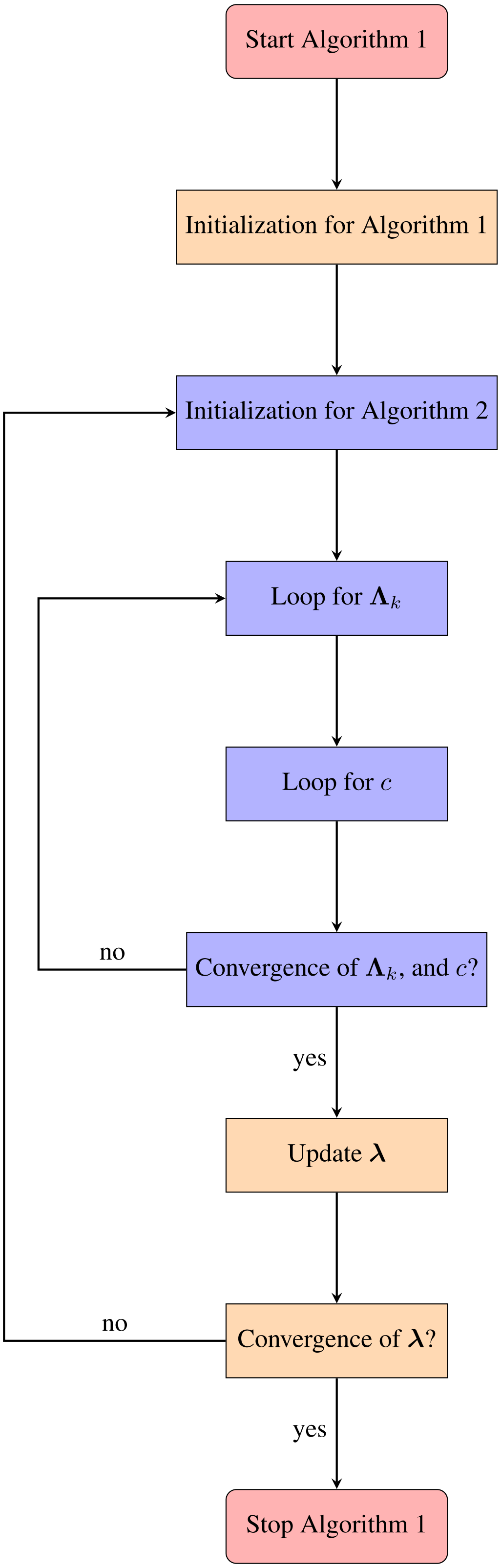
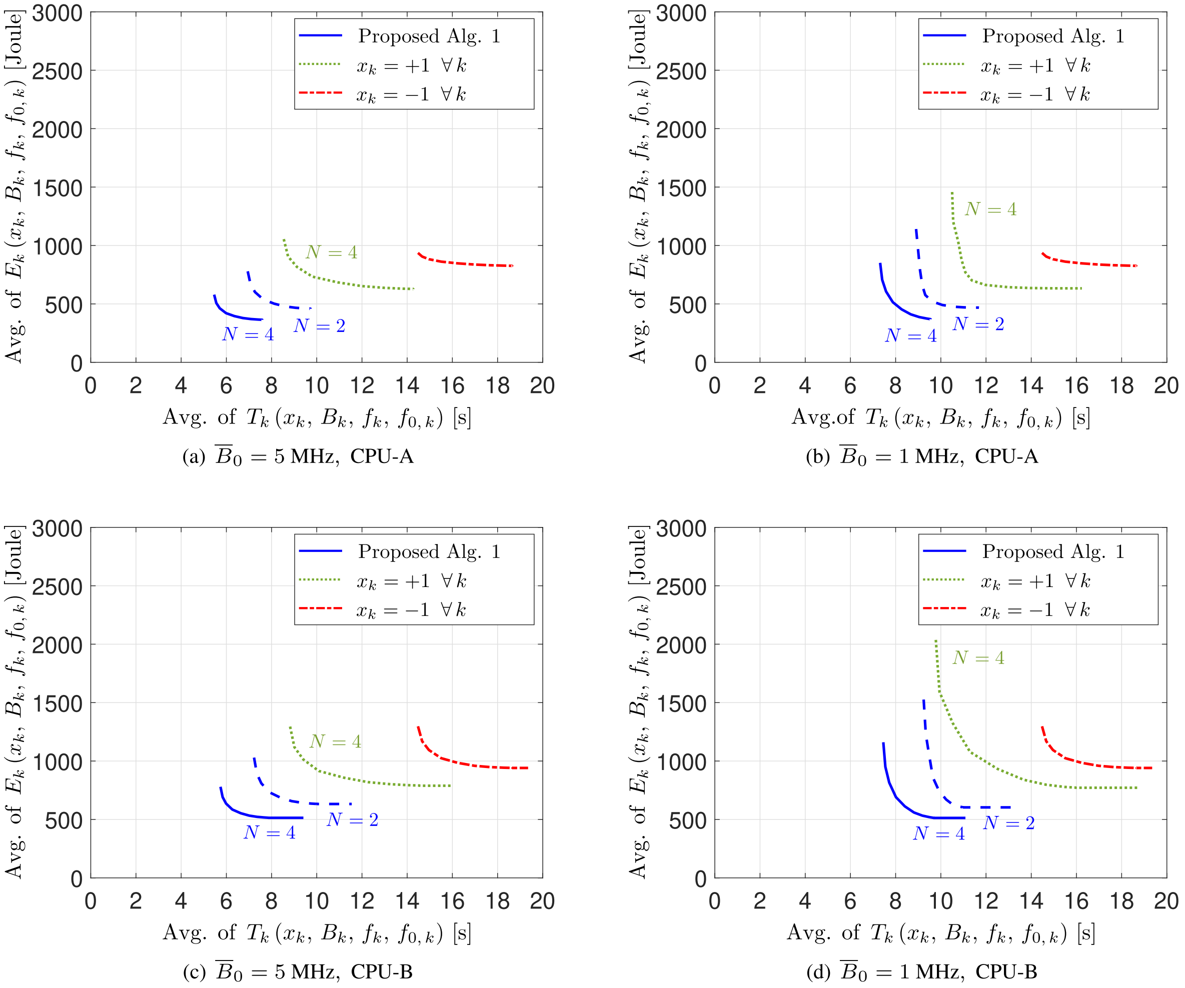

Start Algorithm 1
Initialization for Algorithm 1
Initialization for Algorithm 2
Loop for $\boldsymbol{\Lambda}_k$
Loop for c
no
Convergence of $\boldsymbol{\Lambda}_k$, and c?
yes
Update $\boldsymbol{\lambda}$
no
Convergence of $\boldsymbol{\lambda}$?
yes
Stop Algorithm 1

FIGURE 3. Flowchart representing the integrated structure of Algorithms 1 and 2.



(a) $\overline{B}_0 = 5$ MHz, CPU-A

(b) $\overline{B}_0 = 1$ MHz, CPU-A

(c) $\overline{B}_0 = 5$ MHz, CPU-B

(d) $\overline{B}_0 = 1$ MHz, CPU-B

FIGURE 4. Pareto boundary between Energy consumption [Joule] and service latency [s] with $K = 4$.

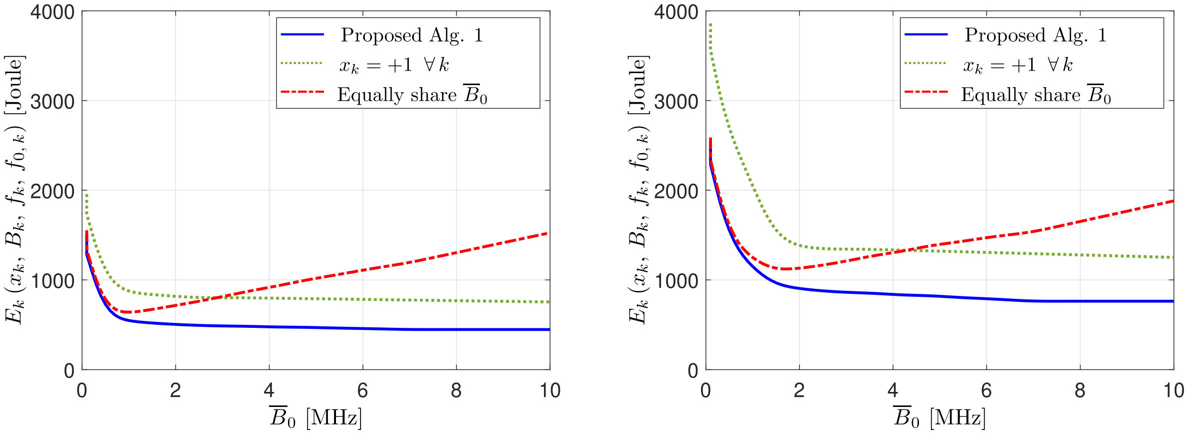
large positive number for α in strategy condition (29) and executing Algorithm 1 to optimize the variables $(B_k, f_{0,k})$. The red dash-dotted curve depicts the case when all drones are forced to LCM. In the case, $x_k = -1$ for all k, each drone has to solve the minimization problem (26) individually to optimize f_k. It is obvious that changing $\overline{B}_0$ does not impact red curves; those are impacted only by changing the type of CPU (changing power consumption coefficient μ).

The proposed algorithm outperforms both forced RCM and LCM, especially with an increased number of wireless channels, giving more flexibility to satisfy computing strategy condition (29). In cases where all drones choose RCM (green dotted curve), energy consumption is higher, and offloading time is greater than the proposed solution. Conversely, forced LCM (red dash-dotted curve), as the task is executed by a relatively low-capacity CPU, increases task execution time. Interestingly, when η is close to one, increasing the $\overline{B}_0$ from 1 to 5 MHz leads to a decrease in energy consumption, whereas it does not impact the range of service latency.

Figures (5) and (6) depict energy consumption and service latency against available bandwidth $\overline{B}_0$ in a network with CPU-B and with $\eta = 0.01$ and $\eta = 0.99$, in the Tchebysheff problem formulation (24). The results affirm that the proposed game theory-based solution consistently achieves lower energy consumption and service latency than non-optimized networks. Small $\overline{B}_0$ results in higher values, while in optimized scenarios, drones do not necessarily utilize the entire bandwidth, leaving some unused.

Comparing optimized bandwidth assignment (blue solid curve) with non-optimized scenarios, it can be seen that increased $\overline{B}_0$ reduces energy consumption to 0.6 of the green dotted curve (forced RCM) and to 0.3 of the red dash-dotted curve (equal bandwidth division among drones) at $\overline{B}_0 = 10$ MHz.

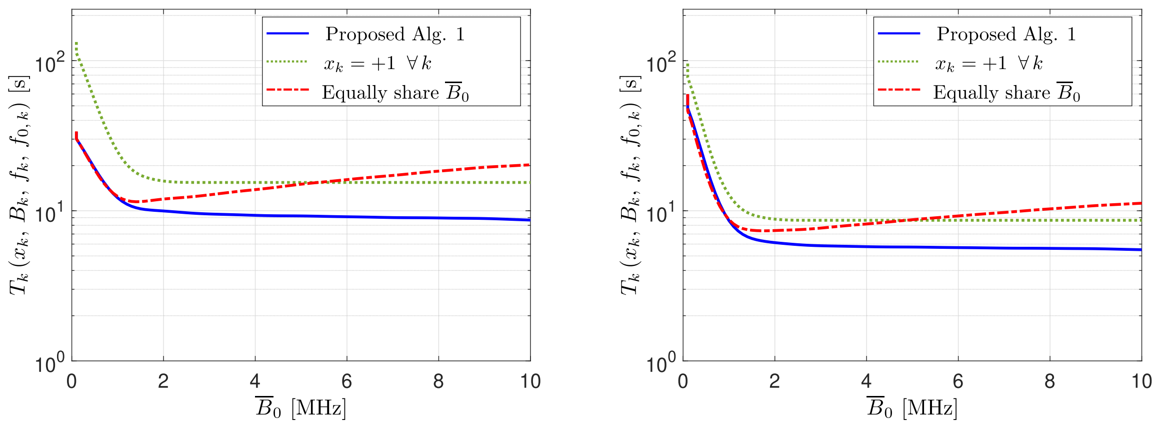
From Figure 6, increasing available bandwidth significantly improves service latency in optimized scenarios. The achieved latency by the blue solid curves reaches 0.5 of the green dotted curves and 0.4 of the red dash-dotted curves.



(a) $\eta = 0.01$

(b) $\eta = 0.99$

FIGURE 5. Energy as a function of bandwidth with $K = N = 4$ and CPU-B.



(a) $\eta = 0.01$

(b) $\eta = 0.99$

FIGURE 6. Service latency as a function of bandwidth with $K = N = 4$ and CPU-B.

These comprehensive results underline the efficiency and robustness of the proposed solution, demonstrating substantial improvements in energy consumption and service latency over conventional and non-optimized computing modes, even with variations in bandwidth and computing strategy.

C. COMPARATIVE DISTINCTION

While our proposed algorithm focuses on the unique objective of jointly minimizing service latency and energy consumption, it may appear natural to seek a comparison with existing works. However, the specific formulation of our problem, combined with the exacting convexification techniques we employed, sets our work apart in a foundational manner. It is imperative to note that due to the lack of a gap in our convexification method, outperforming the result we achieved is inherently infeasible. This ensures we attain the absolute global optimum without compromising efficiency, a criterion not directly comparable with many prevalent methods. Nevertheless, we have illustrated the advantages of our approach by contrasting it with networks where the computing modes of drones are not optimized. While this comparison might seem intuitive, it highlights the significant gains realized through our methodology. We believe that these distinctions, complemented by the pioneering techniques introduced, underscore the novelty and substantial contributions of our research.

D. CASE STUDY

As a case study, we consider a network with $K = N = 4$, $\overline{B}_0 = 5$ MHz, and CPU type of CPU-B. We consider two distances from the drones to the BS, short and long distances, and two types of computational tasks, light, and heavy tasks. As it is described in Table 3, drones 1 and 2 are close to the BS, while the other two drones are at a long distance. Drones 1 and 3 have light tasks to execute, while the other two drones have heavy ones. Figure 7 depicts the service latency as a function of the transmit power level p_k. As it can be seen, at a low transmit power level, the service latency does not change. This means all drones select the LCM. This happens

TABLE 3. Case study with four drones.

Drone	#1	#2	#3	#4
Task (D_k, C_k)	Light	Heavy	Light	Heavy
Distance r_k	Short	Short	Long	Long

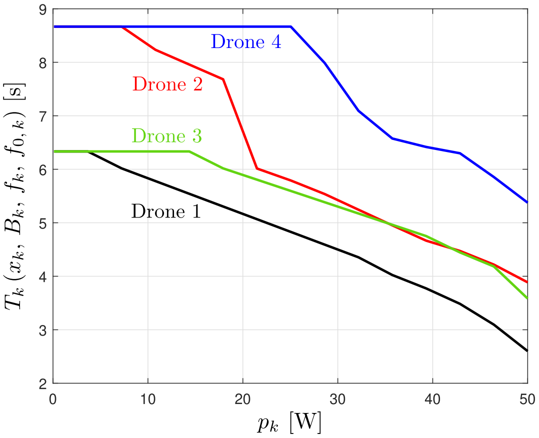


FIGURE 7. Service latency as a function of the transmit power with $K = N = 4$ and CPU-B.

since, at a low transmit power level the achieved data rate is low and, in turn, offloading the task requires a relatively long time, so selecting the RCM is not beneficial. As the transmit power level increases, the time required to offload the task decreases, so selecting the RCM to exploit the powerful CPU on the fog node could be beneficial. It is obvious that this happens at a lower transmit power level for drones at a short distance to the BS and the drones with light tasks. For drone 4, at a long distance with a heavy task, the RCM is preferred only at a very high transmit power level.

VIII. CONCLUSION

In this paper, we introduced a resource allocation method for fog-assisted drone communication networks, targeting the joint minimization of energy consumption and service latency while factoring in both physical and application layer constraints. Our approach enables drones to choose between local computing mode (LCM) and remote computing mode (RCM) as per their requirements. The resource allocation algorithm designed assigns bandwidth for drones in RCM and modulates the CPU frequency of edge and fog nodes to balance latency and energy usage. We incorporated realistic models for drone electronic circuit power consumption. Utilizing the Tchebysheff theorem, we established a bi-objective minimization to pinpoint the latency-energy Pareto boundary and ensured convexity for each drone's minimization problem. We modeled drone interactions through a non-cooperative game and introduced an algorithm, based on subgradient projection, to determine the unique Nash equilibrium. Simulations show our algorithm outperforms networks with non-optimized drone computing modes or evenly distributed bandwidth.

REFERENCES

[1] Z. Han and K. J. R. Liu, *Resource Allocation for Wireless Networks: Basics, Techniques, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[2] E. Hossain, V. K. Bhargava, and G. P. Fettweis, *Green Radio Communication Networks* (Wiley Series in Telecommunications). Cambridge, U.K.: Cambridge Univ. Press, 2012.

[3] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 128–134, Feb. 2017.

[4] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[5] V. Marbukh, "Towards fog network utility maximization (FoNUM) for managing fog computing resources," in *Proc. IEEE Int. Conf. Fog Comput. (ICFC)*, Jun. 2019, pp. 195–200.

[6] S.-S. Lee and S. Lee, "Poster abstract: Deep reinforcement learning-based resource allocation in vehicular fog computing," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Paris, France, Apr. 2019, pp. 1–6.

[7] X. Chen, S. Leng, K. Zhang, and K. Xiong, "A machine-learning based time constrained resource allocation scheme for vehicular fog computing," *China Commun.*, vol. 16, no. 11, pp. 29–41, Nov. 2019.

[8] X. Gao, X. Huang, S. Bian, Z. Shao, and Y. Yang, "PORA: Predictive offloading and resource allocation in dynamic fog computing systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 72–87, Jan. 2020.

[9] Y. Wang, J. Yang, X. Guo, and Z. Qu, "A game-theoretic approach to computation offloading in satellite edge computing," *IEEE Access*, vol. 8, pp. 12510–12520, 2020.

[10] T. Fang, D. Wu, J. Chen, and D. Liu, "Cooperative task offloading and content delivery for heterogeneous demands: A matching game-theoretic approach," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 2, pp. 1092–1103, Jun. 2022.

[11] S. Tong, Y. Liu, M. Cheriet, M. Kadoch, and B. Shen, "UCAA: User-centric user association and resource allocation in fog computing networks," *IEEE Access*, vol. 8, pp. 10671–10685, 2020.

[12] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.

[13] M. Mukherjee, S. Kumar, Q. Zhang, R. Matam, C. X. Mavromoustakis, Y. Lv, and G. Mastorakis, "Task data offloading and resource allocation in fog computing with multi-task delay guarantee," *IEEE Access*, vol. 7, pp. 152911–152918, 2019.

[14] Y.-W. Hung, Y.-C. Chen, C. Lo, A. G. So, and S.-C. Chang, "Dynamic workload allocation for edge computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 3, pp. 519–529, Mar. 2021.

[15] J. Plachy, Z. Becvar, E. C. Strinati, and N. D. Pietro, "Dynamic allocation of computing and communication resources in multi-access edge computing for mobile users," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 2089–2106, Jun. 2021.

[16] J. Yao and N. Ansari, "Online task allocation and flying control in fog-aided Internet of Drones," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5562–5569, May 2020.

[17] H. A. Alharbi, B. A. Yosuf, M. Aldossary, J. Almutairi, and J. M. H. Elmirghani, "Energy efficient UAV-based service offloading over cloud-fog architectures," *IEEE Access*, vol. 10, pp. 89598–89613, 2022.

[18] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 771–786, Apr. 2019.

[19] S. Jošilo and G. Dán, "A game theoretic analysis of selfish mobile computation offloading," in *Proc. IEEE Conf. Comput. Commun.*, Atlanta, GA, USA, May 2017, pp. 1–9.

[20] T. Bahreini, H. Badri, and D. Grosu, "Mechanisms for resource allocation and pricing in mobile edge computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 667–682, Mar. 2022.

[21] K. Wang, X. Wang, X. Liu, and A. Jolfaei, "Task offloading strategy based on reinforcement learning computing in edge computing architecture of Internet of Vehicles," *IEEE Access*, vol. 8, pp. 173779–173789, 2020.

[22] C. Zhang, H. Zhao, and S. Deng, "A density-based offloading strategy for IoT devices in edge computing systems," *IEEE Access*, vol. 6, pp. 73520–73530, 2018.

[23] X. Huang, S. Leng, S. Maharjan, and Y. Zhang, "Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9282–9293, Sep. 2021.

[24] J. Chen, Y. Yang, C. Wang, H. Zhang, C. Qiu, and X. Wang, "Multitask offloading strategy optimization based on directed acyclic graphs for edge computing," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9367–9378, Jun. 2022.

[25] X. Xu, Y. Xue, X. Li, L. Qi, and S. Wan, "A computation offloading method for edge computing with vehicle-to-everything," *IEEE Access*, vol. 7, pp. 131068–131077, 2019.

[26] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: Blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4187–4195, Jun. 2020.

[27] X. Huang, X. Liu, Q. Chen, and J. Zhang, "Resource allocation and task offloading in blockchain-enabled fog computing networks," in *Proc. IEEE 94th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2021, pp. 01–05.

[28] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.

[29] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[30] X. Lyu, H. Tian, W. Ni, Y. Zhang, P. Zhang, and R. P. Liu, "Energy-efficient admission of delay-sensitive tasks for mobile edge computing," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2603–2616, Jun. 2018.

[31] Q. Zhang, F. Machida, and E. Andrade, "Performance bottleneck analysis of drone computation offloading to a shared fog node," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Charlotte, NC, USA, Oct. 2022, pp. 216–221.

[32] K. Li, "A game theoretic approach to computation offloading strategy optimization for non-cooperative users in mobile edge computing," *IEEE Trans. Sustain. Comput.*, early access, Sep. 5, 2018, doi: 10.1109/TSUSC.2018.2868655.

[33] B. Liu, H. Xu, X. Zhou, and Z. Han, "Stackelberg differential game based resource allocation in wireless networks with fog computing," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Changchun, China, Aug. 2019, pp. 406–410.

[34] Y. Jie, M. Li, C. Guo, and L. Chen, "Game-theoretic online resource allocation scheme on fog computing for mobile multimedia users," *China Commun.*, vol. 16, no. 3, pp. 22–31, Mar. 2019.

[35] H. M. Birhanie, S.-M. Senouc, M. A. Messous, A. Arfaoui, and A. Kies, "A stochastic theoretical game approach for resource allocation in vehicular fog computing," in *Proc. IEEE 17th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2020, pp. 1–2.

[36] S. Jošilo and G. Dán, "Wireless and computing resource allocation for selfish computation offloading in edge computing," in *Proc. IEEE Conf. Comput. Commun.*, Paris, France, Apr. 2019, pp. 2467–2475.

[37] Y. Luo, Q. Hu, Y. Wang, J. Wang, O. Alfarraj, and A. Tolba, "Revenue optimization of a UAV-fog collaborative framework for remote data collection services," *IEEE Access*, vol. 8, pp. 150599–150610, 2020.

[38] A. Gupta and S. K. Gupta, "Intelligent collaboration of multi-agent flying UAV-fog networking for better QoS," in *Proc. Int. Conf. Electr., Comput., Commun. Mechatronics Eng. (ICECCME)*, Nov. 2022, pp. 1–6.

[39] Y. Chen, S. Zhang, S. Xu, and G. Y. Li, "Fundamental trade-offs on green wireless networks," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 30–37, Jun. 2011.

[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[41] M. Kaisa, *Nonlinear Multiobjective Optimization* (International Series in Operations Research & Management Science), vol. 12. Boston, MA, USA: Kluwer Academic Publishers, 1999.

[42] A. Filippone, *Flight Performance of Fixed and Rotary Wing Aircraft*, 1st ed. Oxford, MI, USA: Amer. Inst. Aeronaut. Astronaut., 2006.

[43] T. Dietrich, S. Krug, and A. Zimmermann, "An empirical study on generic multicopter energy consumption profiles," in *Proc. Annu. IEEE Int. Syst. Conf. (SysCon)*, Montreal, QC, Canada, Apr. 2017, pp. 1–6.

[44] A. R. Jensen, M. Lauridsen, P. Mogensen, T. B. Sørensen, and P. Jensen, "LTE UE power consumption model: For system level energy and performance optimization," in *Proc. IEEE Veh. Technol. Conf.*, Yokohama, Japan, Sep. 2012, pp. 1–5.

[45] M. Lauridsen, L. Noël, T. B. Sørensen, and P. Mogensen, "An empirical LTE smartphone power model with a view to energy efficiency evolution," *Intel Technol. J.*, vol. 18, no. 1, pp. 172–193, Mar. 2014.

[46] C. Schlegel and C. B. Winstead, "Energy limits of message-passing error control decoders," in *Proc. Int. Zurich Seminar Commun. (IZS)*, Zurich, Switzerland, Feb. 2014.

[47] A. Mammela and A. Anttonen, "Why will computing power need particular attention in future wireless devices?" *IEEE Circuits Syst. Mag.*, vol. 17, no. 1, pp. 12–26, 1st Quart., 2017.

[48] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[49] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–5.

[50] *Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor*, Intel$^{®}$ Core$^{\text{TM}}$, Santa Clara, CA, USA, Mar. 2004.

[51] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI signal Process. Syst. Signal, Image Video Technol.*, vol. 13, nos. 2–3, pp. 203–221, Aug. 1996.

[52] E. Ahvar, A.-C. Orgerie, and A. Lebre, "Estimating energy consumption of cloud, fog, and edge computing infrastructures," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 2, pp. 277–288, Apr. 2022.

[53] O. Arnold, F. Richter, G. Fettweis, and O. Blume, "Power consumption modeling of different base station types in heterogeneous cellular networks," in *Proc. Future Netw. Mobile Summit*, Florence, Italy, Jun. 2010, pp. 1–8.

[54] B. Kopras, F. Idzikowski, and P. Kryszkiewicz, "Power consumption and delay in wired parts of fog computing networks," in *Proc. IEEE Sustainability ICT Summit (StICT)*, Montreal, QC, Canada, Jun. 2019, pp. 1–8.

[55] A. Mourato, D. Duarte, I. Pinto, and P. Vieira, "A novel and realistic power consumption model for multi-technology radio networks," *URSI Radio Sci. Bull.*, vol. 2018, no. 364, pp. 20–29, Mar. 2018.

[56] F. Shams, A. Cerone, and R. D. Nicola, "On integrating social and sensor networks for emergency management," in *Proc. Softw. Eng. Formal Methods (SEFM) Workshops*, York, U.K., Sep. 2015.

[57] G. Yuan and B. Ghanem, "Binary optimization via mathematical programming with equilibrium constraints," 2016, *arXiv:1608.04425*.

[58] K. Shen and W. Yu, "Fractional programming for communication systems—Part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, May 2018.

[59] Z. Wang, L. Vandendorpe, M. Ashraf, Y. Mou, and N. Janatian, "Minimization of sum inverse energy efficiency for multiple base station systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–7.

[60] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, MA, USA: MIT Press, 1994.

[61] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory Wireless Commun. Networks: Theory, Models, and Applications*, 1st ed. New York, NY, USA: Cambridge Univ. Press, 2012.

[62] G. Debreu, "A social equilibrium existence theorem," *Proc. Nat. Acad. Sci. USA*, vol. 38, no. 10, pp. 886–893, Oct. 1952.

[63] T. Ichiishi, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*, 1st ed. New York, NY, USA: Academic Press, 1983.

[64] A. Mas-Colell, M. Whinston, and J. Green, *Microeconomic Theory*. Oxford U.K.: Oxford Univ. Press, 1995.

[65] Y. Yang, F. Rubio, G. Scutari, and D. P. Palomar, "Multi-portfolio optimization: A potential game approach," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5590–5602, Nov. 2013.

[66] D. Paccagnan, B. Gentile, F. Parise, M. Kamgarpour, and J. Lygeros, "Distributed computation of generalized Nash equilibria in quadratic aggregative games with affine coupling constraints," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 6123–6128.

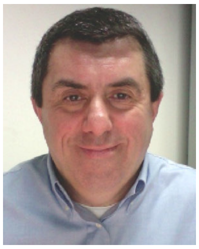
[67] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and Optimization*. New York, NY, USA: Athena Sci., 2003.

[68] M. Grant and S. Boyd. (Aug. 2012). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.0*. [Online]. Available: http://cvxr.com/cvx

[69] Intel$^{®}$ Core$^{\text{TM}}$. (Aug. 2022). *X-Series Processor Family*. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/processors/core/x.html

[70] AMD. (Aug. 2022). *Ryzen^{TM} Embedded Family*. [Online]. Available: https://www.amd.com/en/products/embedded-ryzen-series

FARSHAD SHAMS received the Ph.D. degree from the IMT School for Advanced Studies, Lucca, Italy, in 2012. In 2018, he received the H2020 Marie Skłodowska-Curie Actions (MSCA) Individual Fellowships (IF) REDESIGN, which he carried out with CentraleSupélec, University Paris-Saclay, Gif-sur-Yvette, France. He is currently a Researcher and a Developer with the Faculty of Information Engineering, University of Pisa, Pisa, Italy. His research interests include multi-objective resource allocation techniques for wireless communications and tensor-based signal processing for 6G RAN.



VINCENZO LOTTICI received the M.S. degree (cum laude) in electronic engineering from the University of Pisa, in 1985. He spent a research period with the international center, from 1987 to 1993, worked on signal processing algorithms for passive sonar systems. Since 1993, he has been with the Department of Information Engineering, University of Pisa, where he is currently an Associate Professor of communication systems. He has authored more than 150 journal and conference papers and participated in several international and national research projects, funded by EU, ESA, and MIUR. He has been involved in the ETSI standardization activity of the mobile radio system TETRA release2. His research interests include the broad area of signal processing for terrestrial and satellite communications, with an emphasis to synchronization techniques, dynamic resource allocation and waveform adaptation, cooperative and cognitive wireless networks, ultra wide-band communications, and compressive sensing techniques. He received the Best Thesis Award from the University of Pisa in 1986. From 2013 to 2019, he joined the editorial board of *EURASIP Journal on Advances Signal Processing*, and *EURASIP Journal on Wireless Communications and Networking*, since 2019.

FILIPPO GIANNETTI received the Laurea degree from the University of Pisa, Pisa, Italy, in 1989, and the Ph.D. degree in electronic engineering from the University of Padua, Padua, Italy, in 1993. He spent a research period with TELETTRA (currently ALCATEL), Vimercate, Milan, Italy, from 1988 to 1989, working on SONET/SDH radio modems. In 1992, he was with the European Space Agency Research and Technology Centre (ESA/ESTEC), Noordwijk, The Netherlands, where he was engaged in several activities in the field of digital satellite communications. Since 1993, he has been with the Department of Information Engineering, University of Pisa, where he is an Associate Professor of telecommunications. He has involved in the NEFOCAST Project. He has authored more than 170 journal and conference papers and is a co-inventor of several patents, jointly developed with ESA. His main research interests are in the broad area of wireless terrestrial and satellite communications, with special attention to digital modulations, error-correcting codes, multiple-access techniques, digital modem architecture design, radio resource allocation, signal processing for communications, radio propagation, and radiolocation techniques. He is an Editorial Board Member of *EURASIP Journal on Wireless Communications and Networking*.

• • •

Open Access funding provided by ‘Università di Pisa’ within the CRUI CARE Agreement