# Joint Allocation of Computing and Wireless Resources to Autonomous Devices in Mobile Edge Computing

Slađana Jošilo and György Dán
ACCESS Linnaeus Center, School of Electrical Engineering and Computer Science
KTH, Royal Institute of Technology, Stockholm, Sweden, E-mail: {josilo, gyuri@kth.se}

## ABSTRACT

We consider the interaction between mobile edge computing (MEC) resource management and wireless devices that offload computationally intensive tasks through shared wireless links to edge cloud servers, so as to minimize their completion times. We model the interaction between the devices and the operator that optimizes the allocation of the wireless and computing resources as a Stackelberg game. We show that a pure strategy Stackelberg equilibrium exists, and we provide an efficient algorithm for computing equilibrium allocations. Our simulation results show that joint optimization of the wireless and computing resources can provide a significant reduction of completion times at little increase in computational complexity compared to a system where resource allocation is not optimized.

## 1 INTRODUCTION

Mobile edge computing (MEC) brings computing resources close to the network edge, and is expected to meet the low response time requirements of emerging computationally intensive applications [1, 2]. It is thus considered to be a key enabler for emerging IoT applications, including automated surveillance, augmented reality, autonomous vehicles, and industrial monitoring [3, 4]. Despite the proximity of the computing resources, when many devices use the limited wireless and computing resources simultaneously, the application response times could deteriorate [5, 6]. Thus, in order to ensure low response times, the wireless and the computing resources will have to be managed jointly.

The joint management of wireless and computing resources is inherently challenging for various reasons. First, the computational tasks generated by different devices might differ significantly in terms of the amount of the input data needed to be transmitted, and in terms of the computational complexity. Hence, resource management has to accommodate heterogeneous wireless and computing requirements. Second, the devices competing for the resources are often autonomous entities, and thus the resource allocation should be compatible with the devices' individual interests. Third, in case of a dense deployment of wireless access points and multiple edge clouds, the offloading choices of individual devices have to be coordinated so as to achieve good overall performance, while respecting the interests of the devices and of one or more cloud providers.

Most previous works followed a centralized approach for managing the wireless and computing resources, so as to maximize the performance of the devices in terms of response time and energy consumption, without considering the devices' interests [7–9]. Recognizing the importance of the potential autonomy of the devices [10, 11], a few recent works proposed to coordinate the offloading decisions of the devices in a decentralized manner based on game theoretical models [12–15], but without considering the objectives of the cloud providers. There is thus a lack of a framework for modeling the joint resource allocation problem of cloud providers and autonomous devices, and we lack efficient algorithms for managing the wireless and computing resources in a way that is compatible with the interests of operators and devices alike.

In this work we propose to bridge this gap. We consider the interaction between an operator that jointly optimizes the allocation of wireless and cloud computing resources and autonomous devices that aim at minimizing the response times of their own applications. We formulate the problem as a multiple leader common follower Stackelberg game, played by the devices as leaders and the operator as follower. We provide a closed form solution for the operator's best response, and we show that a pure Stackelberg equilibrium exists. Our constructive proof is based on transforming a player specific congestion game into a weighted congestion game, and serves as an efficient algorithm for coordinating the offloading decisions of the devices, given the optimal policy of the operator for allocating wireless and cloud computing resources.

The rest of the paper is organized as follows. Section 2 describes the system model. In Section 3 we present the optimal resource allocation policy and an efficient algorithm for computing equilibrium offloading decisions of devices. In Section 4 we present numerical results, and Section 5 concludes the paper.

## 2 SYSTEM MODEL AND PROBLEM FORMULATION

We consider a mobile edge computing (MEC) system that consists of $N$ devices, $C$ mobile edge clouds, and $A$ access points (APs). We denote by $\mathcal{N} = \{1, 2, \ldots, N\}$, $\mathcal{C} = \{1, 2, \ldots, C\}$ and $\mathcal{A} = \{1, 2, \ldots, A\}$ the set of devices, mobile edge clouds and APs, respectively. We denote by $\mathcal{A}_c \subseteq \mathcal{A}$ the set of APs through which devices can communicate with cloud $c \in \mathcal{C}$. We consider that each device $i \in \mathcal{N}$ generates computationally intensive tasks, and we characterize device $i$'s task by two parameters, the mean size $D_i$ of the input data and the mean number $L_i$ of CPU cycles required to perform the computation. Previous works have shown that the number $X$ of CPU cycles required per data bit can be modeled by a Gamma distributed random variable [16, 17], and hence we can express $L_i = D_i E[X]$, based on the empirical mean $E[X]$.

To facilitate the analysis, we make the common assumption that the set of devices changes slowly [7, 18, 19]. Considering the potential autonomy of the devices [10, 11], every device $i \in \mathcal{N}$ is allowed to make the offloading decision by itself, i.e., it can decide to which cloud $c \in \mathcal{C}$ to offload its task and for the chosen cloud $c$ through which of the APs $a \in \mathcal{A}_c$ to transmit $D_i$ amount of data pertaining to its task. Therefore, the set of feasible decisions for device $i$ is $\mathfrak{D}_i = \{(c, a) | c \in \mathcal{C}, a \in \mathcal{A}_c\}$. We refer to decision $d_i \in \mathfrak{D}_i$ of device $i$ as its strategy, and we refer to the collection $\mathbf{d} = (d_i)_{i \in \mathcal{N}}$ as a strategy profile, i.e., $\mathbf{d} \in \times_{i \in \mathcal{N}} \mathfrak{D}_i$.

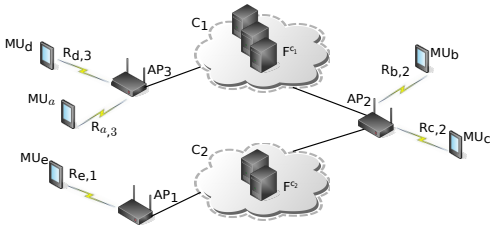


**Figure 1:** *An example of a MEC system that consists of $N = 5$ devices, $C = 2$ edge clouds and $A = 3$ APs.*

For a strategy profile $\mathbf{d}$, we define the set $O_a(\mathbf{d}) \triangleq \{i | d_i = (\cdot, a)\}$ of offloaders for AP $a$ and the set $O_c(\mathbf{d}) \triangleq \cup_{a \in \mathcal{A}_c} O_a(\mathbf{d})$ of offloaders for cloud $c$. We use $n_a(\mathbf{d}) \triangleq |O_a(\mathbf{d})|$ to denote the number of offloaders that offload the task through AP $a$, and $n_c(\mathbf{d}) \triangleq |O_c(\mathbf{d})|$ to denote the number of offloaders that offload the task to cloud $c$.

## 2.1 Wireless and Computing Resources

In what follows we introduce our model of sharing wireless and cloud computing resources in the MEC computing system presented above.

*2.1.1 Wireless resource management.* We denote by $R_{i,a}$ the PHY rate that is achievable to device $i$ when it offloads its task via AP $a$, and we consider that $R_{i,a}$ depends on the physical layer signal characteristics and the wireless channel state. Furthermore, we consider that the actual rate at which device $i$ can offload its data via AP $a$ is determined by the rate allocation policy of AP $a$. To define the policy, we introduce the uplink access provisioning coefficient $b_{i,a}(\mathbf{d})$ for every device $i \in O_a(\mathbf{d})$ and we denote by $\mathbf{b}_a = (b_{i,a})_{i \in O_a(\mathbf{d})}$ the collection of uplink access provisioning coefficients for all devices $i \in O_a(\mathbf{d})$. Given $\mathbf{b}_a$, the uplink rate $\omega_{i,a}(\mathbf{d}, \mathbf{b}_a)$ of device $i$ at AP $a$ can be expressed as

$$\omega_{i,a}(\mathbf{d}, \mathbf{b}_a) = R_{i,a} / \big(b_{i,a}(\mathbf{d}) n_a(\mathbf{d})\big). \tag{1}$$

Observe that if $b_{i,a}(\mathbf{d}) = 1$ for all $i \in O_a(\mathbf{d})$, (1) can be used to model throughput sharing mechanisms in TDMA and OFDMA based MAC protocols [20].

*2.1.2 Computing resource management.* After the data are transmitted via AP $a \in \mathcal{A}_c$, the task is performed in the chosen cloud $c$. We denote by $F^c$ the computing capability of cloud $c$. We consider that the computing capability of cloud $c$ is actively allocated among devices that use it, and to define the allocation policy we define the computing power provisioning coefficient $p_{i,c}(\mathbf{d})$ for every device $i \in O_c(\mathbf{d})$ and we denote by $\mathbf{p}_c = (p_{i,c})_{i \in O_c(\mathbf{d})}$ the collection of computing power provisioning coefficients for all devices $i \in O_c(\mathbf{d})$. Given $\mathbf{p}_c$, the computing capability $F_i^c(\mathbf{d}, \mathbf{p}_c)$ allocated to device $i$ by cloud $c$ can be expressed as

$$F_i^c(\mathbf{d}, \mathbf{p}_c) = F^c / \big(p_{i,c}(\mathbf{d}) n_c(\mathbf{d})\big). \tag{2}$$

Fig. 1 shows an example of a MEC system that consists of 5 devices, 2 edge clouds and 3 APs. Devices $a, d$ and $b$ offload their tasks to cloud $c_1$ (devices $a$ and $d$ through $AP_3$ and device $b$ through $AP_2$), and devices $c$ and $e$ offload their tasks to cloud $c_2$ through $AP_2$ and $AP_1$, respectively.

## 2.2 Cost Model

We define the cost of device $i$ as the completion time of its task, which consists of two parts. The first part is the time needed to transmit $D_i$ amount of data, and the second part is the time needed to perform device $i$'s task at the cloud server. Thus, in the case of offloading to cloud $c$ through AP $a \in \mathcal{A}_c$, the cost $C_{i,a}^c(\mathbf{d}, \mathbf{b}_a, \mathbf{p}_c)$ of device $i$ can be expressed as

$$C_{i,a}^c(\mathbf{d}, \mathbf{b}_a, \mathbf{p}_c) = D_i / \omega_{i,a}(\mathbf{d}, \mathbf{b}_a) + L_i / F_i^c(\mathbf{d}, \mathbf{p}_c). \tag{3}$$

In (3) we made the common assumption that the time needed to transmit the results of the computation from the cloud to the device can be neglected [12, 21, 22], as for typical applications (e.g., face and speech recognition), the size of the result of the computation is much smaller than $D_i$.

Using the above notation, the cost $C_i(\mathbf{d}, \mathbf{b}_a, \mathbf{p}_c)$ of device $i$ can be expressed as

$$C_i(\mathbf{d}, \mathbf{b}_a, \mathbf{p}_c) = \sum_{c \in C} \sum_{a \in \mathcal{A}_c} I(d_i, (c,a)) C_{i,a}^c(\mathbf{d}, \mathbf{b}_a, \mathbf{p}_c), \tag{4}$$

where $I(d_i, d) = 1$ if $d_i = d$ and $I(d_i, d) = 0$ otherwise.

We use the shorthand notation $\mathbf{b} \triangleq (\mathbf{b}_a)_{a \in \mathcal{A}}$ and $\mathbf{p} \triangleq (\mathbf{p}_c)_{c \in C}$, and we define the system cost $C(\mathbf{d}, \mathbf{b}, \mathbf{p})$ as

$$C(\mathbf{d}, \mathbf{b}, \mathbf{p}) = \sum_{i \in \mathcal{N}} C_i(\mathbf{d}, \mathbf{b}_a, \mathbf{p}_c). \tag{5}$$

## 2.3 Problem Formulation

We consider that the autonomous devices compete for the wireless and cloud computing resources of the MEC system, managed by an operator.

The objective of the operator is to jointly optimize the allocation of the wireless and cloud computing resources, given the offloading decisions of the devices. Given a strategy profile $\mathbf{d}$ chosen by the devices, the operator optimizes the allocation of the wireless and cloud computing resources through applying the optimal provisioning coefficients $\mathbf{b}^*$ and $\mathbf{p}^*$ computed by solving

$$\min_{\mathbf{b}, \mathbf{p} \geq 0} C(\mathbf{d}, \mathbf{b}, \mathbf{p}) \tag{6}$$

$$\text{s.t.} \quad \sum_{j \in O_a(\mathbf{d})} \frac{1}{b_{j,a}(\mathbf{d})} = n_a(\mathbf{d}), \quad \forall a \in \mathcal{A} \tag{7}$$

$$\sum_{j \in O_c(\mathbf{d})} \frac{1}{p_{j,c}(\mathbf{d})} = n_c(\mathbf{d}), \quad \forall c \in C \tag{8}$$

where constraints (7) and (8) ensure the feasibility of sharing the wireless and computing resources, respectively.

The objective of every device is to minimize its own cost (4), given the allocation policy of the operator, that is, each device aims at solving

$$\min_{d_i \in \mathfrak{D}_i} C_i(d_i, d_{-i}, \mathbf{b}_a^*, \mathbf{p}_c^*), \tag{9}$$

where we use $d_{-i}$ to denote the strategies of all devices except device $i$.

We can model (6)-(8) and (9) together as a multi-leader-common-follower Stackelberg game, played by the devices as leaders and the operator as follower. We refer to the problem as the *mobile edge computation offloading game* (MEC-OG). Our objective is to answer the fundamental question whether the MEC-OG has a subgame perfect equilibrium, i.e., a combination of computation offloading strategy profile and allocation policy from which neither the devices nor the operator have an incentive to deviate.

**Definition 1** (SPE)**.** Let $(\mathbf{b}^*, \mathbf{p}^*)$ be a solution of (6)-(8), and $d_i^*$ be a solution of (9). Then the point $(\mathbf{d}^*, \mathbf{b}^*, \mathbf{p}^*)$ is a subgame perfect

equilibrium (SPE) of the MEC-OG if for any feasible $(\mathbf{d}, \mathbf{b}, \mathbf{p})$ point the following holds

$$C(\mathbf{d}^*, \mathbf{b}^*, \mathbf{p}^*) \leq C(\mathbf{d}^*, \mathbf{b}, \mathbf{p}),$$
$$C_i(d_i^*, d_{-i}^*, \mathbf{b}_a^*, \mathbf{p}_c^*) \leq C_i(d_i, d_{-i}^*, \mathbf{b}_a^*, \mathbf{p}_c^*), \forall d_i \in \mathfrak{D}_i, \forall i \in \mathcal{N}.$$

Second, we want to understand whether such an equilibrium can be computed efficiently, in terms of computational complexity and signaling.

# 3 EXISTENCE OF STACKELBERG EQUILIBRIA

We start the analysis by considering the problem (6)-(8) of the operator, and then we consider the problem to be solved by the devices.

## 3.1 Optimal Resource Allocation Policy

The optimal allocation policy of the operator, i.e., the solution to (6)-(8), is its best response to the strategy profile $\mathbf{d}$ chosen by the devices. The following result shows that the optimal policy can be expressed in closed form.

Theorem 1. *Consider a strategy profile $\boldsymbol{d}$. The optimal uplink access provisioning coefficients $b_{i,a}^*(\boldsymbol{d})$ and the optimal computing power provisioning coefficients $p_{i,c}^*(\boldsymbol{d})$ are given by*

$$b_{i,a}^*(\boldsymbol{d}) = \frac{\sum_{j \in O_a(\boldsymbol{d})} \sqrt{D_j/R_{j,a}}}{n_a(\boldsymbol{d})\sqrt{D_i/R_{i,a}}}, \forall i \in O_a(\boldsymbol{d}), \forall a \in \mathcal{A}, \tag{10}$$

$$p_{i,c}^*(\boldsymbol{d}) = \frac{\sum_{j \in O_c(\boldsymbol{d})} \sqrt{L_j/F^c}}{n_c(\boldsymbol{d})\sqrt{L_i/F^c}}, \forall i \in O_c(\boldsymbol{d}), \forall c \in C. \tag{11}$$

Proof. Let us substitute (1)-(2) into (3) to obtain

$$C(\mathbf{d}, \mathbf{b}, \mathbf{p}) = \sum_{j \in \mathcal{N}} \sum_{c \in C} \sum_{a \in \mathcal{A}_c} \Big( \frac{D_j}{R_{j,a}} b_{j,a}(\mathbf{d}) n_a(\mathbf{d}) + \frac{L_j}{F^c} p_{j,c}(\mathbf{d}) n_c(\mathbf{d}) \Big).$$

Thus, the problem (6)-(8) is a convex optimization problem, and the optimal solution must satisfy the Karush–Kuhn–Tucker (KKT) conditions. In order to formulate the dual of the problem, we express the Lagrangian associated with (6)-(8) as

$$\mathcal{L}(\mathbf{d}, \mathbf{b}, \mathbf{p}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}) = C(\mathbf{d}, \mathbf{b}, \mathbf{p}) +$$
$$\sum_{a \in \mathcal{A}} \alpha_a \Big( \sum_{j \in O_a(\mathbf{d})} \frac{1}{b_{j,a}(\mathbf{d})} - n_a(\mathbf{d}) \Big) - \sum_{a \in \mathcal{A}} \sum_{j \in O_a(\mathbf{d})} \gamma_{j,a} b_{j,a}(\mathbf{d}) +$$
$$\sum_{c \in C} \beta_c \Big( \sum_{j \in O_c(\mathbf{d})} \frac{1}{p_{j,c}(\mathbf{d})} - n_c(\mathbf{d}) \Big) - \sum_{c \in C} \sum_{j \in O_c(\mathbf{d})} \delta_{j,c} p_{j,c}(\mathbf{d}),$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are dual variables associated with constraints (7) and (8) and $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ are non-negative dual variables associated with constraints $\mathbf{b} \geq 0$ and $\mathbf{p} \geq 0$.

The Lagrangian dual problem is then defined as $\max_{\boldsymbol{\alpha} \in \mathbb{R}^A, \boldsymbol{\beta} \in \mathbb{R}^C, \boldsymbol{\gamma}, \boldsymbol{\delta} \geq 0} \min_{\mathbf{b}, \mathbf{p} \geq 0} \mathcal{L}(\mathbf{d}, \mathbf{b}, \mathbf{p}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta})$, and the KKT conditions can be written as follows

$$\text{Stationarity:} \quad \frac{\partial \mathcal{L}(\mathbf{d}, \mathbf{b}, \mathbf{p}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta})}{\partial b_{i,a}(\mathbf{d})} = 0, \forall a \in \mathcal{A}, \forall i \in O_a(\mathbf{d}),$$
$$\frac{\partial \mathcal{L}(\mathbf{d}, \mathbf{b}, \mathbf{p}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta})}{\partial p_{i,c}(\mathbf{d})} = 0, \forall c \in C, \forall i \in O_c(\mathbf{d}), \tag{12}$$

$$\text{Primal feasibility:} \quad \sum_{j \in O_a(\mathbf{d})} \frac{1}{b_{j,a}(\mathbf{d})} = n_a(\mathbf{d}), \forall a \in \mathcal{A},$$
$$\sum_{j \in O_c(\mathbf{d})} \frac{1}{p_{j,c}(\mathbf{d})} = n_c(\mathbf{d}), \forall c \in C, \tag{13}$$

$$\text{Dual feasibility:} \quad \gamma_{i,a}, \delta_{i,c} \geq 0, \forall i \in \mathcal{N}, \forall a \in \mathcal{A}, \forall c \in C, \tag{14}$$

$$\text{Complementary slackness:} \quad -\gamma_{i,a} b_{i,a}(\mathbf{d}) = 0, \forall a \in \mathcal{A}, \forall i \in O_a(\mathbf{d}),$$
$$-\delta_{i,c} p_{i,c}(\mathbf{d}) = 0, \forall c \in C, \forall i \in O_c(\mathbf{d}). \tag{15}$$

From (12), we have

$$b_{i,a}(\mathbf{d}) = \sqrt{\frac{\alpha_a}{n_a(\mathbf{d}) D_i / R_{i,a} + \gamma_{i,a}}}, \forall a \in \mathcal{A}, \forall i \in O_a(\mathbf{d}), \tag{16}$$

$$p_{i,c}(\mathbf{d}) = \sqrt{\frac{\beta}{n_c(\mathbf{d}) L_i / F^c + \delta_{i,c}}}, \forall c \in C, \forall i \in O_c(\mathbf{d}). \tag{17}$$

First, observe that $b_{i,a}(\mathbf{d}) > 0$ and $p_{i,c}(\mathbf{d}) > 0$ must hold in order to have (13) satisfied. Consequently, $\gamma_{i,a} = 0$ and $\delta_{i,c} = 0$ must hold in order to have (15) satisfied. Substituting $b_{i,a}(\mathbf{d}) = \sqrt{\frac{\alpha_a R_{i,a}}{n_a(\mathbf{d}) D_i}}$ and $p_{i,c}(\mathbf{d}) = \sqrt{\frac{\beta F^c}{n_c(\mathbf{d}) L_i}}$ to equations in (13) we have $\sqrt{\alpha_a} = \frac{\sum_{j \in O_a(\mathbf{d})} \sqrt{D_j/R_{j,a}}}{\sqrt{n_a(\mathbf{d})}}$ and $\sqrt{\beta} = \frac{\sum_{j \in O_c(\mathbf{d})} \sqrt{L_j/F^c}}{\sqrt{n_c(\mathbf{d})}}$, respectively. Hence, $b_{i,a}(\mathbf{d}) = \frac{\sum_{j \in O_a(\mathbf{d})} \sqrt{D_j/R_{j,a}}}{n_a(\mathbf{d})\sqrt{D_i/R_{i,a}}}$ and $p_{i,c}(\mathbf{d}) = \frac{\sum_{j \in O_c(\mathbf{d})} \sqrt{L_j/F^c}}{n_c(\mathbf{d})\sqrt{L_i/F^c}}$ follow, which proves the theorem. □

Theorem 1 provides a closed form optimal allocation policy for a strategy profile $\mathbf{d}$ as a function of the characteristics of the tasks of the devices that share a wireless or computing resource, and can be made known to the devices a priori.

## 3.2 Computing an Equilibrium Strategy Profile

We continue with modeling the interaction among the autonomous devices, given that the operator implements its optimal resource allocation policy.

Theorem 2. *Given the optimal resource allocation policy of the operator, the strategic interaction of the devices can be modeled as a congestion game with resource dependent weights.*

Proof. Let us define the link dependent weights $w_{i,a} \triangleq \sqrt{D_i/R_{i,a}}$ for each $i \in O_a(\mathbf{d})$ and the cloud dependent weights $w_{i,c} \triangleq \sqrt{L_i/F^c}$ for each $i \in O_c(\mathbf{d})$. Using this notation, and substituting (10) and (11) into (3), the offloading cost of device $i$ to cloud $c$ through AP $a$ can be rewritten as

$$C_{i,a}^c(\mathbf{d}) = w_{i,a} \sum\nolimits_{j \in O_a(\mathbf{d})} w_{j,a} + w_{i,c} \sum\nolimits_{j \in O_c(\mathbf{d})} w_{j,c}. \tag{18}$$

Since the offloading cost (18) depends on the total weight $w_a(\mathbf{d}) = \sum_{j \in O_a(\mathbf{d})} w_{j,a}$ of all devices sharing AP $a$ and the total weight $w_c(\mathbf{d}) = \sum_{j \in O_c(\mathbf{d})} w_{j,c}$ of all devices sharing cloud $c$, the interaction between the devices can be modeled as a *weighted congestion game with resource-dependent weights*, which proves the theorem. □

We refer to the resulting weighted congestion game $\Gamma = < \mathcal{N}, (\mathfrak{D}_i)_i, (C_i)_i >$ as the *devices computation offloading game* (DC-OG), in which the players are devices, and the cost of each device $i$ is given by

$$C_i(\mathbf{d}) = \sum\nolimits_{c \in C} \sum\nolimits_{a \in \mathcal{A}_c} I(d_i, (c, a)) C_{i,a}^c(\mathbf{d}), \tag{19}$$

where $C_{i,a}^c(\mathbf{d})$ is given by (18). Note that $\Gamma$ is a strategic game.

In what follows we answer the fundamental question whether the DC-OG has a pure strategy Nash equilibrium.

**Definition 2.** A pure strategy Nash equilibrium (NE) is a strategy profile $\mathbf{d}^*$ in which all players play their best replies to each others' strategies, that is,

$$C_i(d_i^*, d_{-i}^*) \leq C_i(d_i, d_{-i}^*), \forall d_i \in \mathfrak{D}_i, \forall i \in \mathcal{N}.$$

Given a strategy profile $d = (d_i', d_{-i})$, an *improvement step* of device $i$ is a strategy $d_i'$ such that $C_i(d_i', d_{-i}) < C_i(d_i, d_{-i})$. A *best improvement step* is an improvement step that is a best reply.

Before we formulate our next result let us recall the definition of an exact potential function from [23].

**Definition 3.** A function $\Phi : \times_i(\mathfrak{D}_i) \rightarrow \mathbb{R}$ is an exact potential for a finite strategic game $\Gamma =< \mathcal{N}, (\mathfrak{D}_i)_i, (C_i)_i >$ if for any arbitrary strategy profile $(d_i, d_{-i})$ and for any improvement step $d_i'$ the following holds

$$C_i(d_i', d_{-i}) - C_i(d_i, d_{-i}) = \Phi(d_i', d_{-i}) - \Phi(d_i, d_{-i}). \tag{20}$$

We continue by introducing the following shorthand notation,

$$w_r^{\leq i}(\mathbf{d}) = \sum\nolimits_{\{j \in \mathcal{N} | j \leq i, r \in d_j, r \in d_i, r \in \mathcal{A} \cup C\}} w_{j,r}.$$

Let us now recall the following result about *weighted congestion games with resource-dependent weights.*

Lemma 3. *[24] Let $\Gamma$ be a weighted congestion game with resource-dependent weights in which the players compete over a finite set $\mathcal{R}$ of resources. Then, $\Gamma$ has an exact potential function $\Phi(\boldsymbol{d})$ if and only if the cost $c_r$ of sharing every resource $r \in \mathcal{R}$ is an affine function of the congestion on resource $r \in \mathcal{R}$, i.e, $c_r(\boldsymbol{d}) = a_r \sum_{\{j \in \mathcal{N} | r \in d_j\}} w_{j,r} + b_r$, $a_r, b_r \in \mathbb{R}$. The potential function $\Phi(\boldsymbol{d})$ is given by*

$$\Phi(\boldsymbol{d}) = \sum\nolimits_{i \in \mathcal{N}} \sum\nolimits_{r \in d_i} c_r^{\leq i}(\boldsymbol{d}) w_{i,r}, \tag{21}$$

*where $c_r^{\leq i}(\boldsymbol{d}) = a_r w_r^{\leq i}(\boldsymbol{d}) + b_r$.*

We can now formulate the following result on equilibrium existence.

Theorem 4. *The DC-OG is an exact potential game with the exact potential function as given in (21), and hence it possesses a pure strategy Nash equilibrium.*

Proof. Observe that the cost $c_a(\mathbf{d}) = \sum_{j \in O_a(\mathbf{d})} w_{j,a}$ of sharing AP $a$ and the cost $c_c(\mathbf{d}) = \sum_{j \in O_c(\mathbf{d})} w_{j,c}$ of sharing cloud $c$ are identity functions of the congestion on AP $a$ and the congestion on cloud $c$, respectively. Since the identity function is affine, Lemma 3 applies, which proves the theorem. □

The existence of an exact potential function allows us to compute a NE of the DC-OG using the *ImproveOffloading* (IO) algorithm, shown in Fig. 2. The algorithm adds devices one at a time, and lets them perform their best improvement steps given the other devices' strategies.

Corollary 1. *The IO algorithm converges to an equilibrium allocation $\boldsymbol{d}^*$ in a finite number of improvement steps.*

Proof. First, observe that since the number of devices is finite, the number of induction steps in which a new device is added by the IO algorithm is finite too. Second, since the DC-OG is an exact potential game, it follows from (20) that the decrease in a device's cost due to its best improvement update step results in exactly the same amount of decrease in the potential function $\Phi$ defined by (21). Since $\times_i(\mathfrak{D}_i)$ is a finite set, $\Phi$ cannot decrease indefinitely and thus the IO algorithm must terminate in an equilibrium allocation upon every induction step, which proves the result. □

**d** = IO($\mathcal{N}$)

$\mathcal{N}' \leftarrow \emptyset$
**for** $N = 1, \ldots, |\mathcal{N}|$ **do**
  $i \leftarrow N$, $\mathcal{N}' = \mathcal{N}' \cup \{i\}$
  $d_i^*(N) = \arg\min_{i \in \mathfrak{D}_i} C_i(d_i, \mathbf{d}^*(N-1))$
  $\mathbf{d} = (d_i^*(N), \mathbf{d}^*(N-1))$
  **while** $\exists j \in \{\mathcal{N}' | d_j \neq \arg\min_{d_j' \in \mathfrak{D}_j} C_j(d_j', d_{-j})\}$ **do**
    $d_j^* = \arg\min_{d_j' \in \mathfrak{D}_j} C_j(d_j', d_{-j})$
    $\mathbf{d} = (d_j^*, d_{-j})$
  **end while**
  $\mathbf{d}^*(N) \leftarrow \mathbf{d}$
**end for**
**return d**

**Figure 2:** *Pseudo code of the IO algorithm.*

Finally, we can formulate our main result concerning the existence of a SPE of the MEC-OG.

Proposition 1. *The SPE for the MEC-OG is the point $(\boldsymbol{d}^*, \boldsymbol{b}^*, \boldsymbol{p}^*)$, where $\boldsymbol{d}^*$ is computed by the IO algorithm, and $\boldsymbol{b}^*$ and $\boldsymbol{p}^*$ are defined according to (10) and (11), respectively.*

Proof. The proof of the proposition follows from the proofs of Theorem 1 and Corollary 1, respectively. □

### 3.3 Implementation considerations

In practice, computing the SPE can be implemented as follows. First, with the obtained information about the resource allocation mechanism implemented at the operator side, the devices implement a NE in a decentralized manner by performing improvement steps one at a time according to the IO algorithm. Each device reports its offloading decision to the operator, which sends the information about the congestion on the wireless and cloud computing resources back to the devices. Based on the received information, devices may update their offloading decisions one at a time. According to Corollary 1, the devices will stop updating their offloading decisions within a finite number of improvement steps, and thus they will converge to a NE. Second, given the computed NE, the operator allocates the wireless and cloud computing resources to the devices optimally according to (10) and (11). After the resource provisioning is done, neither the devices nor the operator would have an incentive to deviate, and thus the system reaches the SPE.

## 4 NUMERICAL RESULTS

We use extensive simulations to evaluate the cost performance of the system in which the operator allocates the wireless and cloud computing resources according to the *optimal allocation* (OA) policy, and the devices implement their offloading decisions according to the IO algorithm. We consider that the devices and clouds are placed uniformly at random over a square area of $1km \times 1km$, while the APs are placed at random on a *regular grid* with $A^2$ points defined over the area. We consider that the channel gain of device $i$ to an AP $a$ is proportional to $d_{i,a}^{-\alpha}$, where $d_{i,a}$ is the distance between device $i$ and AP $a$, and $\alpha$ is the path loss exponent, which we set to 4 according to the path loss model in urban and suburban areas [25]. For simplicity we assign a bandwidth of $B_a = 5$ $MHz$ to every AP $a$. We set the data transmit power $P_i^t$ of every device $i$ to 0.4 $W$ according to [26] and given the noise power $P_n$ we calculate the transmission rate $R_{i,a}$ achievable to device $i$ for the communication

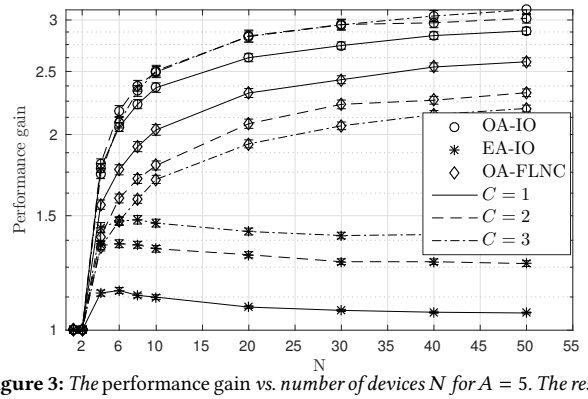


**Figure 3:** *The* performance gain *vs. number of devices N for $A = 5$. The results shown are the averages of 1000 simulations, together with 95% confidence intervals.*

with AP $a$ as $R_{i,a} = B_a log(1+P_i^t d_{i,a}^{\alpha}/P_n)$. Unless otherwise noted, we consider a system with $A = 5$ APs and clouds with the same computing capability $F^c = 64GHz$ [27]. The input data size $D_i$ is uniformly distributed on $[0.2, 4]$ $Mb$ and we calculate the number $L_i$ of CPU cycles required to perform the computation as $L_i = D_i X$, where the number $X$ of CPU cycles required per data is a Gamma distributed random variable with the shape parameter $k = 0.5$ and scale parameter $\theta = 1.6$.

As a baseline for resource sharing we use an equal resource allocation policy, according to which the devices receive the same amount of the shared resource. We refer to this policy as the *equal allocation* (EA) policy. Given that the operator implements the EA policy, the devices are playing a *player-specific congestion game* on a network made of two parallel networks connected in series. It follows from Theorem 2 in [28] that the resulting game has a pure strategy Nash equilibrium and from Theorem 1 in [28] that the equilibrium offloading strategy profile $\mathbf{d}^*$ can be computed using the IO algorithm.

In order to evaluate the performance of the proposed IO algorithm, we use a simple *fastest-link nearest-cloud* (FLNC) algorithm, according to which devices offload their computation through the AP that provides the highest achievable transmission rate and to the cloud with the shortest distance to the chosen AP.

For a resource allocation policy $P$ and the offloading strategy profile $\mathbf{d}^A$ computed by an algorithm $A$ we define the *performance gain* as

$$\frac{C^{EA}(\mathbf{d}^{FLNC})}{C^P(\mathbf{d}^A)},$$

where $C^P(\mathbf{d}^A)$ is the system cost reached when the operator implements the resource allocation policy $P$, and the devices compute their offloading decisions according to the algorithm $A$.

### 4.1 User focused performance analysis

We start with evaluating the *performance gain* as a function of number $N$ devices. Fig. 3 shows the *performance gain* achieved by the IO algorithm under the OA policy, the performance of the IO algorithm under the EA policy, and the performance of the FLNC algorithm under the OA policy in a system with $C = 1$, $C = 2$ and $C = 3$ clouds.

The results show that the performance gain increases with a decreasing marginal gain in $N$, showing that the achievable *performance gain* is limited by the number of devices, i.e., by the congestion on the wireless and cloud computing resources. We observe that the *performance gain* is largest when the operator implements the OA policy, and the devices compute their equilibrium offloading decisions using the IO algorithm. Furthermore, we observe that the *performance gain* achieved by the FLNC algorithm under the OA policy is significantly higher than the *performance gain* achieved by the IO algorithm under the EA policy. This suggests that the choice of the resource allocation policy has a large impact on the system performance.

We also observe that when the offloading decisions are calculated using the IO algorithm, the *performance gain* increases with the number of clouds for both resource allocation policies. This is due to that for larger number of clouds the offloading strategy profile computed by the IO algorithm and the offloading strategy profile computed by the FLNC algorithm might differ significantly. Consequently, the *performance gain* becomes more dominated by the offloading strategy profile than by the resource allocation policy. On the contrary, when the offloading decisions are calculated using the FLNC algorithm and the operator implements the OA policy, the *performance gain* decreases as the number of clouds increases. This is due to that the *performance gain* is determined only by the implemented resource allocation policy (the offloading decisions are calculated using the FLNC algorithm as in the case of the used baseline), and since the impact of the resource allocation policy decreases as the number of clouds increases, the *performance gain* decreases too.

### 4.2 Cloud focused performance analysis

We look at the system from a cloud perspective by considering the number $n_c(\mathbf{d})$ of offloaders per cloud and by considering the cost per cloud, which we define for a strategy profile $\mathbf{d}$ as $C^c(\mathbf{d}) = \sum_{i \in O_c(\mathbf{d})} C_i(\mathbf{d})$. In the following we consider a system with three clouds with computing capabilities $100GHz$, $64GHz$, and $32GHz$, respectively.

Figures 4 and 5 show $n_c(\mathbf{d})$ and $C^c(\mathbf{d})$, respectively, for each of the clouds as a function of $N$. The results are shown for all possible combinations of the introduced resource allocation policies and the introduced algorithms for computing the offloading strategy profile. Fig. 4 shows that when the offloading strategy profile is computed using the FLNC algorithm, the devices are almost equally distributed among the clouds. This is due to that the FLNC algorithm chooses the cloud that is nearest to the fastest link, and since the devices and the clouds are placed uniformly at random over the region, all clouds will experience the same congestion on average. Consequently, the corresponding cost per cloud, shown in Fig. 5, is larger for clouds with less computing capability.

Fig. 4 also shows that when the offloading decisions are calculated using the IO algorithm and the operator implements the EA policy, the congestion of a cloud increases with its computing capability. This is due to that the resources are allocated equally among the devices, and thus independently of the tasks' complexities. Consequently, the cost per cloud is highest for the most congested (and the most powerful) cloud as shown in Fig. 5. On the contrary, when the offloading decisions are calculated using the IO algorithm and

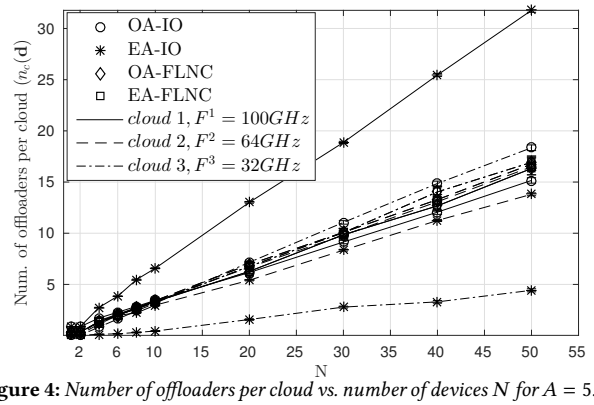


**Figure 4:** *Number of offloaders per cloud vs. number of devices N for A = 5. The results shown are the averages of 1000 simulations, together with 95% confidence intervals.*

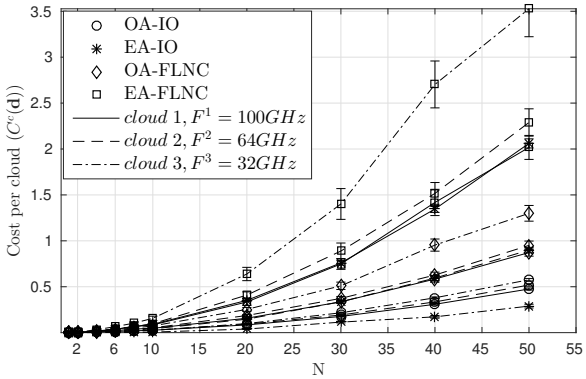


**Figure 5:** *Cost per cloud vs. number of devices N for A = 5. The results shown are the averages of 1000 simulations, together with 95% confidence intervals.*

the operator implements the OA policy, the results in Fig. 4 show that the devices are fairly distributed among the clouds, which is due to that the OA policy takes into the consideration the complexity of devices' tasks. For the same reason, the corresponding cost per cloud, shown in Fig. 5, is almost the same for all clouds.

## 5 CONCLUSION

We have provided a game theoretical analysis of the interaction between an operator that jointly manages wireless and cloud computing resources and autonomous devices that aim at minimizing the completion times of their offloaded tasks. Our model of the interaction as a Stackelberg game allowed us to identify the optimal policy of the operator, and given the optimal policy of the operator, we showed that devices can implement an equilibrium allocation of their offloading decisions in a decentralized manner efficiently. Our numerical results show that the joint optimization of wireless and cloud computing resources can reduce the completion time of the tasks significantly compared to a system where resource allocation is not optimized.

## REFERENCES


[1] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing: A key technology towards 5G," Sep. 2015.

[2] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE IoT-J*, pp. 450–465, 2018.

[3] T. Y.-H. Chen, L. Ravindranath, S. Deng, P. Bahl, and H. Balakrishnan, "Glimpse: Continuous, real-time object recognition on mobile devices," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015, pp. 155–168.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, pp. 1645–1660, 2013.

[5] Z. Yin, F. R. Yu, S. Bu, and Z. Han, "Joint cloud and wireless networks operations in mobile cloud computing environments with telecom operator cloud," *IEEE Transactions on Wireless Communications*, vol. 14, no. 7, pp. 4020–4033, 2015.

[6] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload? The bandwidth and energy costs of mobile cloud computing," in *Proc. of IEEE INFOCOM*, April 2013, pp. 1285–1293.

[7] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE T-SIPN*, vol. 1, no. 2, pp. 89–103, 2015.

[8] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, pp. 1397–1411, 2017.

[9] A. Al-Shuwaili, O. Simeone, A. Bagheri, and G. Scutari, "Joint uplink/downlink optimization for backhaul-limited mobile cloud computing with user scheduling," *IEEE Transactions on Signal and Information Processing over Networks*, pp. 787–802, 2017.

[10] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM CCR*, vol. 44, no. 5, pp. 27–32, 2014.

[11] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, "Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 120–128, 2016.

[12] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2015.

[13] V. Cardellini, V. D. N. Personé, V. Di Valerio, F. Facchinei, V. Grassi, F. L. Presti, and V. Piccialli, "A game-theoretic approach to computation offloading in mobile cloud computing," *Mathematical Programming*, vol. 157, no. 2, pp. 421–449, 2016.

[14] S. Jošilo and G. Dán, "A game theoretic analysis of selfish mobile computation offloading," in *Proc. of IEEE INFOCOM*, 2017, pp. 1–9.

[15] ——, "Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks," *IEEE Transactions on Mobile Computing*, 2018.

[16] J. R. Lorch and A. J. Smith, "Improving dynamic voltage scaling algorithms with pace," in *ACM SIGMETRICS*, 2001, pp. 50–61.

[17] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. of Usenix HotCloud*, 2010.

[18] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in *Proc. of IEEE INFOCOM*, March 2012, pp. 2716–2720.

[19] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 4, pp. 23–32, Apr. 2013.

[20] T. Joshi, A. Mukherjee, Y. Yoo, and D. P. Agrawal, "Airtime fairness for ieee 802.11 multirate networks," *IEEE TMC*, pp. 513–527, 2008.

[21] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[22] K. Kumar and Y. H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *IEEE Computer Mag.*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[23] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[24] T. Harks, M. Klimm, and R. H. Möhring, "Characterizing the existence of potential functions in weighted congestion games," *Theory of Computing Systems*, pp. 46–70, 2011.

[25] A. Aragon-Zavala, *Antennas and propagation for wireless communication systems*. John Wiley & Sons, 2008.

[26] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proc. of the 9th ACM SIGCOMM conference on Internet measurement*, 2009, pp. 280–293.

[27] M. Satyanarayanan, "A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets," *GetMobile: Mobile Computing and Communications*, pp. 19–23, 2015.

[28] I. Milchtaich, "Congestion games with player-specific payoff functions," *Games and economic behavior*, pp. 111–124, 1996.