Wireless and Computing Resource Allocation for Selfish Computation Offloading in Edge Computing

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

***Abstract*—We consider the problem of allocating wireless and computing resources to a set of autonomous wireless devices in an edge computing system. Devices in the system can decide whether or not to use edge computing resources for offloading computing tasks so as to minimize their completion time, while the edge cloud operator can allocate wireless and computing resources to the devices. We model the interaction between devices and the operator as a Stackelberg game, prove the existence of Stackelberg equilibria, and propose an efficient decentralized algorithm for computing equilibria. We provide a bound on the price of anarchy of the game, which also serves as an approximation ratio bound for the proposed algorithm. Our simulation results show that the joint allocation of wireless and computing resources by the operator can halve the completion times compared to a system with static resource allocation. At the same time, the convergence time of the proposed algorithm is approximately linear in the number of devices, and thus it could be effectively implemented for edge computing resource management.**

I. Introduction

The technological evolution of handheld devices has been followed by a rapid increase of user demand for a variety of mobile applications such as mobile augmented reality, face and object recognition, and real-time voice and video [1], [2]. Yet, the computational capabilities of today's devices are not sufficient to meet the delay and computational requirements of these applications.

A promising approach to meet the requirements of these emerging applications is mobile edge computing (MEC) [3]. The key idea of MEC is to allow devices to offload their computations through a wireless network to cloud resources located at the network edge. Owing to the proximity of the edge cloud to the end users, MEC can provide significantly lower response times for individual devices than conventional centralized clouds such as Microsoft Azure or Amazon [4]. However, edge clouds are not as computationally powerful as centralized clouds, which together with the limited wireless resources may adversely affect the response times when many devices attempt to offload computations simultaneously [5], [6]. Therefore, in order to fully exploit the potential of MEC, wireless and computing resources have to be jointly managed.

The joint management of wireless and computing resources requires one to take into account the characteristics of the devices and of the infrastructure. First, devices are heterogeneous in terms of the computing capabilities and in terms of the characteristics of the computing tasks they generate. The tasks have diverse delay requirements, different amounts of input data and varying computational complexities. Second, devices in edge computing systems are likely to be autonomous entities [7], [8]. Third, edge computing systems may consist of multiple heterogeneous wireless access points and edge clouds. Consequently, the offloading decisions of the devices should be coordinated such that the resources are efficiently utilized while taking into account the interests of the individual devices, the heterogeneity of their tasks, and the interaction with the resource allocation policies of the edge cloud providers. This makes the joint management of wireless and computing resources for edge computing inherently challenging.

In this paper we propose a novel approach to address this challenge by considering the interaction between an operator that manages the allocation of wireless and computing resources, and devices that decide autonomously whether or not to use shared resources for offloading computing tasks so as to minimize their own completion times. We model the problem as a multiple-leader common-follower Stackelberg game, in which devices are leaders and the operator is the follower. We provide a closed form solution for the optimal resource allocation policy of the operator and we show that under the optimal policy the original player-specific weighted congestion game played by devices can be transformed into a weighted congestion game. We prove that Stackelberg equilibria exist, and we propose an efficient decentralized algorithm for computing an equilibrium. By establishing an upper bound on the price of anarchy of the game, we show that the proposed algorithm achieves a constant factor approximation. Finally, we use simulations to show that the completion times achieved by the proposed algorithm under the optimal policy are significantly lower than the completion times achieved in a system with static resource allocation.

The rest of the paper is organized as follows. We present the system model in Section II. We present the optimal resource allocation policy and prove the existence of Stackelberg equilibria in Section III. We provide a bound on the price of anarchy in Section IV and present numerical results in Section V. We discuss related work in Section VI and conclude the paper in Section VII.

II. System Model and Problem Formulation

We consider an edge computing system that consists of a set $\mathcal{N}=\{1,2,\dots,N\}$ of wireless devices (WDs), a set $\mathcal{A}=\{1,2,\dots,A\}$ of access points (APs), a set $\mathcal{C}=\{1,2,\dots,C\}$ of edge clouds (ECs), and an operator that manages the allocation of the wireless and computing resources. We denote by $\mathcal{A}_i\subseteq\mathcal{A}$ the set of APs through which WD $i\in\mathcal{N}$ can communicate with the ECs. Each WD $i\in\mathcal{N}$ generates computationally intensive tasks, which can be characterized by two parameters, the size D_i of the input data and the expected number L_i of CPU cycles required to perform the computation (e.g., in bits). As shown by recent works, the number X of CPU cycles required per data bit can be approximated by a Gamma distribution [9], [10]. Hence, based on the empirical mean $E[X]$, the relationship between

The work was partly funded by the Swedish Research Council through project 621-2014-6.

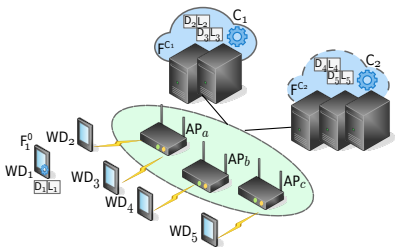

Figure 1. Example of an edge computing system with $N = 5$ WDs, $C = 2$ ECs and $A = 3$ APs. Transmission rates and cloud computing power may be actively managed by the operator.

L_i and D_i can be expressed as $L_i = D_i E[X]$. To make the analysis tractable, we make the common assumption that the set of WDs is known (e.g., through signaling) [11] [12].

Each WD $i \in \mathcal{N}$ can decide whether to perform the computation locally or to offload the computation to one of the ECs $c \in \mathcal{C}$ through one of the APs $a \in \mathcal{A}_i$. Thus, the set of feasible decisions for WD i is $\mathfrak{D}_i = \{i\} \cup \{(a,c) | a \in \mathcal{A}_i, c \in \mathcal{C}\}$, where i corresponds to local computing and (a,c) to offloading through AP a to EC c. We refer to decision $d_i \in \mathfrak{D}_i$ of WD i as its strategy, and we refer to the collection $\mathbf{d} = (d_i)_{i \in \mathcal{N}}$ as a strategy profile, i.e., $\mathbf{d} \in \times_{i \in \mathcal{N}} \mathfrak{D}_i = \mathfrak{D}$. For a strategy profile $\mathbf{d} \in \mathfrak{D}$, we define the set $O_a(\mathbf{d}) \triangleq \{i | d_i = (a, \cdot)\}$ of WDs that offload their tasks through AP a and we denote by $n_a(\mathbf{d}) \triangleq |O_a(\mathbf{d})|$ the number of WDs that offload their tasks through AP a. Similarly, we define the set $O_c(\mathbf{d}) \triangleq \{i | d_i = (\cdot, c)\}$ of WDs that offload their tasks to EC c and we denote by $n_c(\mathbf{d}) \triangleq |O_c(\mathbf{d})|$ the number of WDs that offload their tasks to EC c. Finally, we define the set $O(\mathbf{d}) \triangleq \cup_{c \in \mathcal{C}} O_c(\mathbf{d})$ of all WDs that offload their tasks.

Fig. 1 shows an example of a MEC system that consists of $N = 5$ WDs, $C = 2$ ECs and $A = 3$ APs. WD 1 performs the computation locally, WDs 2 and 3 offload their tasks to EC c_1 through AP a, WDs 4 and 5 offload their tasks to EC c_2 through APs b and c, respectively. In what follows we discuss our models of computing and wireless resource management.

A. Computing Resource Management

A WD that chooses local computing performs its task using its local computing resources. We denote by F_i^l the computational capability of WD $i \in \mathcal{N}$ (e.g., CPU cycles/second). A WD that chooses offloading has to transmit the data through an AP a, after which the task is performed in an EC c. We denote by F^c the computing capability of EC c. We consider that the computing capability allocated to WDs $i \in O_c(\mathbf{d})$ is determined by the operator's *computing resource allocation policy* $\mathcal{P}_c : \mathfrak{D} \to \mathbb{R}_{\geq 0}^{|\mathcal{C}| x |\mathcal{N}|}$. The policy sets for every strategy profile $\mathbf{d} \in \mathfrak{D}$ the computing power provisioning coefficients $(p_{i,c})_{i \in \mathcal{N}, c \in \mathcal{C}}$, akin to the weight of a job in generalized processor sharing (GPS). Using the shorthand notation $\mathbf{p}_c = (p_{i,c})_{i \in \mathcal{N}}$, we can express the computing capability allocated to WD i by EC c as

$$F_i^c(\mathbf{d}, \mathbf{p}_c) = F^c \frac{p_{i,c}}{\sum_{j \in O_c(\mathbf{d})} p_{j,c}}. \quad (1)$$

Observe that for a policy that sets $p_{i,c} = 1$, $\forall i \in O_c(\mathbf{d}), \forall \mathbf{d} \in \mathfrak{D}$, the computing power is shared equally. While GPS is an ideal scheduler, several process schedulers exist to approximate it in practice, e.g., DWRR [13].

B. Wireless Resource Management

The wireless medium of AP a is shared by the WDs that choose to offload through AP a. We denote by $R_{i,a}$ the achievable PHY rate of WD i through AP a, which is determined by the physical characteristics of the wireless medium, distance, etc. The actual rate at which WD i can offload its data through AP a is determined by the operator's *rate allocation policy* $\mathcal{P}_r : \mathfrak{D} \to \mathbb{R}_{\geq 0}^{|\mathcal{A}| x |\mathcal{N}|}$. The policy sets for every strategy profile the uplink access provisioning coefficients $(u_{i,a})_{i \in \mathcal{N}, c \in \mathcal{A}}$, akin to the weight of a flow in GPS. Using the shorthand notation $\mathbf{u}_a = (u_{i,a})_{i \in \mathcal{N}}$, we can express the uplink rate assigned to WD i at AP a as

$$\omega_{i,a}(\mathbf{d}, \mathbf{u}_a) = R_{i,a} \frac{u_{i,a}}{\sum_{j \in O_a(\mathbf{d})} u_{j,a}}. \quad (2)$$

Observe that for a policy that sets $u_{i,a}(\mathbf{d}) = 1, \forall i \in O_a(\mathbf{d})$ we obtain the model that describes the time-fair throughput sharing mechanisms in TDMA and OFDMA based MAC protocols [14].

C. Cost Model

We define the cost of a WD as the completion time of its task. In what follows we introduce our cost model in the case of computation offloading and in the case of local computing.

Computation offloading: In the case of computation offloading the completion time of WD i's task 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 L_i CPU cycles at the cloud server. Thus, if in strategy profile $\mathbf{d}$ WD i offloads to EC $c \in \mathcal{C}$ through AP $a \in \mathcal{A}_i$ then its cost can be expressed as

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

In (3) we made the common assumption that the time needed to transmit the results from the cloud to the device can be neglected [5], [11], [15], [16], as for typical applications (e.g., face and object recognition), the size of the result of the computation is much smaller than D_i.

Local computing: In the case of local computing the completion time of WD i's task is determined by the number L_i of CPU cycles pertaining to the task and by the computing capability F_i^l. Thus, the local computing cost can be expressed as

$$C_i^l = L_i / F_i^l. \quad (4)$$

Total cost: To define the total cost, we first define the shorthand notation $\mathbf{u} \triangleq (\mathbf{u}_a)_{a \in \mathcal{A}}$ and $\mathbf{p} \triangleq (\mathbf{p}_c)_{c \in \mathcal{C}}$, and express the cost of WD i

$$C_i(\mathbf{d}, \mathbf{u}, \mathbf{p}) = \sum_{(a,c) \in \mathcal{A}_i \times \mathcal{C}} I_{d_i,(a,c)} C_{i,a}^c(\mathbf{d}, \mathbf{u}_a, \mathbf{p}_c) + I_{d_i,i} C_i^l, \quad (5)$$

where $I_{d_i,r} = 1$ if $d_i = r$ and $I_{d_i,r} = 0$ otherwise. Finally, we define the system cost $C(\mathbf{d}, \mathbf{u}, \mathbf{p})$ as

$$C(\mathbf{d}, \mathbf{u}, \mathbf{p}) = \sum_{i \in \mathcal{N}} I_{d_i,i} C_i^l + \sum_{i \in \mathcal{N}} \sum_{(a,c) \in \mathcal{A}_i \times \mathcal{C}} I_{d_i,(a,c)} C_{i,a}^c(\mathbf{d}, \mathbf{u}_a, \mathbf{p}_c). \quad (6)$$

D. Problem Formulation

We consider that in the edge computing system each WD is allowed to make an offloading decision so as to minimize its own cost. On the one hand, this assumption is motivated by the potential autonomy of WDs in edge computing systems [7], [8]. On the other hand, the obtained decentralized algorithms can serve as a good approximation for the optimal solution. Nonetheless, the decisions of the WDs interact with the computing resource and allocation policies of the operator, and hence we model the problem as a multiple-leader common-follower Stackelberg game, in which WDs are leaders and the operator is the follower.

Given a strategy profile $\mathbf{d}$ chosen by the WDs, the objective of the operator is to minimize the system cost by jointly optimizing the allocation of wireless and computing resources. It does so by computing a best response $(\mathbf{u}^*, \mathbf{p}^*)$ to $\mathbf{d}$ through solving

$$\min_{\mathbf{u},\mathbf{p}\succeq 0} C(\mathbf{d},\mathbf{u},\mathbf{p}) \tag{7}$$

We denote by $(\mathcal{P}_r^*, \mathcal{P}_c^*)$ an optimal policy, i.e., a collection of best responses for every $\mathbf{d} \in \mathfrak{D}$.

The objective of every WD is to minimize its own completion time (5), given the announced allocation policy $(\mathcal{P}_r, \mathcal{P}_c)$ of the operator through solving

$$\min_{d_i \in \mathfrak{D}_i} C_i(d_i, d_{-i}, \mathcal{P}_r(d_i, d_{-i}), \mathcal{P}_c(d_i, d_{-i})), \tag{8}$$

where we use d_{-i} to denote the strategies of all WDs except WD i. We refer to the problem as the *mobile edge computation offloading game* (MEC-OG).

The fundamental questions we address in this paper are threefold. First, we address wheter there is a combination of computation offloading strategy profile and allocation policy from which neither the WDs nor the operator have an incentive to deviate, i.e., a subgame perfect equilibrium of the Stackelberg game.

Definition 1 (SPE)**.** Let $(\mathcal{P}_c^*, \mathcal{P}_r^*)$ be a solution of (7), and d_i^* be a solution of (8). Then the point $(\mathbf{d}^*, \mathcal{P}_r^*, \mathcal{P}_c^*)$ is a subgame perfect equilibrium (SPE) of the MEC-OG if for any feasible $(\mathbf{d}, \mathcal{P}_r, \mathcal{P}_c)$ point the following holds

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

If the MEC-OG admits an SPE, the second question is whether an SPE can be computed efficiently. Third, we address whether the system cost in an SPE is efficient compared to a centrally optimized system.

III. Existence of Stackelberg Equilibria

We start the analysis by considering problem (7) solved by the operator, followed by problem (8) solved by the WDs.

A. Optimal Resource Allocation Policy

Recall that an optimal resource allocation policy is essentially a collection of best responses of the operator to the strategy profiles $\mathbf{d} \in \mathfrak{D}$ played by the WDs. In what follows we show that a best response of the operator to a strategy profile $\mathbf{d}$ is unique up to a scale factor and can be expressed in closed form.

Theorem 1. *Let $\boldsymbol{d}$ be a strategy profile played by the WDs. The optimal allocation policy $(\mathcal{P}_r^*, \mathcal{P}_c^*)$ of the operator assigns to $\boldsymbol{d}$ the uplink access provisioning and computing power provisioning coefficients*

$$u_{i,a}^* = \frac{\sqrt{D_i/R_{i,a}}}{\sum_{j\in O_a(\boldsymbol{d})}\sqrt{D_j/R_{j,a}}}, \forall a \in \mathcal{A}, \forall i \in O_a(\boldsymbol{d}), \text{and} \tag{9}$$

$$p_{i,c}^* = \frac{\sqrt{L_i/F^c}}{\sum_{j\in O_c(\boldsymbol{d})}\sqrt{L_j/F^c}}, \forall c \in \mathcal{C}, \forall i \in O_c(\boldsymbol{d}). \tag{10}$$

Proof. By inspecting the leading minors of the Hessian matrix of (3) it is easy to show that the matrix is neither positive nor negative semidefinite already for the case when there are only two WDs sharing a resource. Hence, the problem (7) is neither convex nor concave in $\mathbf{u}$ and $\mathbf{p}$. Furthermore, it is easy to see from expressions (1) and (2) that the optimal solution of (7) cannot be unique, since any non-zero scalar multiple of feasible policies $(\mathcal{P}_r, \mathcal{P}_c)$ yields the same objective value, and hence if there is an optimal solution then there is a continuum of optimal solutions.

To make the solution unique with respect to scalar multiplication, let us introduce normalization constraints on the sums of the provisioning coefficients, and obtain

$$\min_{\mathbf{u},\mathbf{p}\succeq 0} C(\mathbf{d},\mathbf{u},\mathbf{p}) \tag{11}$$

$$\text{s.t.} \quad \textstyle\sum_{j\in O_a(\mathbf{d})} u_{j,a} = 1, \quad \forall a \in \mathcal{A} \tag{12}$$

$$\textstyle\sum_{j\in O_c(\mathbf{d})} p_{j,c} = 1. \quad \forall c \in \mathcal{C} \tag{13}$$

Observe that due to the normalization constrainst the cost function $C(\mathbf{d},\mathbf{u},\mathbf{p})$ can be rewritten as

$$C'(\mathbf{d},\mathbf{u},\mathbf{p}) = \sum_{a\in\mathcal{A}}\sum_{i\in O_a(\mathbf{d})}\frac{D_i}{R_{i,a}u_{i,a}} + \sum_{c\in\mathcal{C}}\sum_{i\in O_c(\mathbf{d})}\frac{L_i}{F^c p_{i,c}} + \sum_{i\in\mathcal{N}\setminus O(\mathbf{d})} C_i^l$$

Since the Hessian matrix of (3) is positive semidefinite on the domain defined by (12) and (13) the problem (11)-(13) is a convex optimization problem, and thus its optimal solution must satisfy the Karush–Kuhn–Tucker (KKT) conditions. To define the Lagrangian dual of (11)-(13), we denote by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ the dual variables associated with constraints (12) and (13) and by $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ the non-negative dual variables associated with constraints $\mathbf{u}\succeq 0$ and $\mathbf{p}\succeq 0$. Using this notation, we express the Lagrangian associated with (11)-(13) as

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

Finally, we define the Lagrangian dual problem as $\max_{\boldsymbol{\alpha}\in\mathbb{R}^A,\boldsymbol{\beta}\in\mathbb{R}^C,\boldsymbol{\gamma},\boldsymbol{\delta}\succeq 0}\min_{\mathbf{u},\mathbf{p}\succeq 0}\mathcal{L}(\mathbf{d},\mathbf{u},\mathbf{p},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta})$, and we formulate the following KKT conditions.

Stationarity:	$\frac{\partial\mathcal{L}(\mathbf{d},\mathbf{u},\mathbf{p},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta})}{\partial u_{i,a}} = 0, \forall a\in\mathcal{A}, \forall i \in O_a(\mathbf{d})$ $\frac{\partial\mathcal{L}(\mathbf{d},\mathbf{u},\mathbf{p},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta})}{\partial p_{i,c}} = 0, \forall c\in\mathcal{C}, \forall i \in O_c(\mathbf{d})$
Primal feasibility:	$\sum_{j\in O_a(\mathbf{d})} u_{j,a} = 1, \forall a\in\mathcal{A}$ $\sum_{j\in O_c(\mathbf{d})} p_{j,c} = 1, \forall c\in\mathcal{C}$
Dual feasibility:	$\gamma_{i,a}, \delta_{i,c} \geq 0, \forall i\in\mathcal{N}, \forall a\in\mathcal{A}, \forall c\in\mathcal{C}$
Complementary slackness:	$-\gamma_{i,a}u_{i,a} = 0, \forall a\in\mathcal{A}, \forall i\in O_a(\mathbf{d})$ $-\delta_{i,c}p_{i,c} = 0, \forall c\in\mathcal{C}, \forall i\in O_c(\mathbf{d})$

Observe that $u_{i,a} = 0$ and $p_{i,c} = 0$ would lead to an infinite completion time for WD i's task, and thus $u_{i,a} > 0$ and $p_{i,c} > 0$ must hold. Therefore, $\gamma_{i,a} = 0$ and $\delta_{i,c} = 0$ must hold in order to have the complementary slackness conditions satisfied. Finally, from the stationarity conditions we can express $u_{i,a}$ and $p_{i,c}$ as

$$u_{i,a} = \sqrt{D_i/\alpha_a R_{i,a}}, \forall a\in\mathcal{A}, \forall i\in O_a(\mathbf{d}), \text{and} \tag{14}$$

$$p_{i,c} = \sqrt{L_i/\beta_c F^c}, \forall c\in\mathcal{C}, \forall i\in O_c(\mathbf{d}). \tag{15}$$

By substituting (14) and (15) in the primal feasibility equations we can obtain the expressions for α_a and β_c, and we can rewrite equations (14) and (15) as $u_{i,a} = \frac{\sqrt{D_i/R_{i,a}}}{\sum_{j\in O_a(\mathbf{d})}\sqrt{D_j/R_{j,a}}}$ and $p_{i,c} = \frac{\sqrt{L_i/F^c}}{\sum_{j\in O_c(\mathbf{d})}\sqrt{L_j/F^c}}$, which proves the theorem. □

It is important to note that following the optimal resource allocation policy the operator allocates resources to the WDs depending on the characteristics of their tasks (i.e., D_i and L_i). Furthermore, the resource allocation policy of the operator can be made known a priori to the WDs,

which allows us to analyze the computation offloading problem of the WDs.

B. Computing Equilibrium Offloading Decisions

Observe that for an arbitrary resource allocation policy $(\mathcal{P}_r, \mathcal{P}_c)$ the interaction between the WDs can be modeled by a player-specific weighted congestion game $\Gamma(\mathcal{P}_r, \mathcal{P}_c) =< \mathcal{N}, (\mathfrak{D}_i)_{i\in\mathcal{N}}, (C_i)_{i\in\mathcal{N}} >$, as (5) is both a function of the WDs' parameters and of the resource provisioning coefficients. Unfortunately, for this class of games general equilibrium existence results are not available. In what follows we show that under the optimal resource allocation policy of the operator the game is transformed into a weighted congestion game.

Theorem 2. *Consider that the operator uses the optimal policy $(\mathcal{P}_r^*, \mathcal{P}_c^*)$, i.e., $\boldsymbol{u}^*$ and $\boldsymbol{p}^*$ are the collections of the optimal provisioning coefficients given by (9) and (10), respectively. Then, the strategic interaction of the WDs can be modeled as a congestion game with resource-dependent weights $w_{i,r}, \forall (i,r) \in \mathcal{N}\times\{\mathcal{A}_i \cup \mathcal{C}\}$, in which the cost of WD i is given by*

$$\bar{C}_i(\boldsymbol{d}) = \sum_{(a,c)\in\mathcal{A}_i\times\mathcal{C}} I_{d_i,(a,c)}\Big(w_{i,a}w_a(\boldsymbol{d}) + w_{i,c}w_c(\boldsymbol{d})\Big) + I_{d_i,i}C_i^l, \quad (16)$$

where $w_r(\boldsymbol{d}) = \sum_{j\in O_r(\boldsymbol{d})} w_{j,r}$.

Proof. Let us first substitute (9) and (10) into (3) in order to obtain the offloading cost of WD i through AP a to EC c under the opimal resource allocation policy $(\mathcal{P}_r^*, \mathcal{P}_c^*)$,

$$\bar{C}^c{}_{i,a}(\mathbf{d}) = \sqrt{\frac{D_i}{R_{i,a}}}\sum_{j\in O_a(\mathbf{d})}\sqrt{\frac{D_j}{R_{j,a}}} + \sqrt{\frac{L_i}{F^c}}\sum_{j\in O_c(\mathbf{d})}\sqrt{\frac{L_j}{F^c}}. \quad (17)$$

Second, let us define the weight $w_{i,a} \triangleq \sqrt{D_i/R_{i,a}}$ for each tuple $(i,a) \in \mathcal{N}\times\mathcal{A}_i$ and the weight $w_{i,c} \triangleq \sqrt{L_i/F^c}$ for each tuple $(i,c) \in \mathcal{N}\times\mathcal{C}$. Observe that the offloading cost (17) in strategy profile $\mathbf{d}$ depends on the total weight $w_a(\mathbf{d}) = \sum_{j\in O_a(\mathbf{d})} w_{j,a}$ associated to AP a and on the total weight $w_c(\mathbf{d}) = \sum_{j\in O_c(\mathbf{d})} w_{j,c}$ associated to EC c. Thus, the interaction between the WDs can be modeled as a *weighted congestion game with resource-dependent weights*. This proves the theorem. □

We refer to the resulting strategic game $\Gamma(\mathcal{P}_r^*, \mathcal{P}_c^*) =< \mathcal{N}, (\mathfrak{D}_i)_{i\in\mathcal{N}}, (\bar{C}_i)_{i\in\mathcal{N}} >$ as the *optimal allocation computation offloading game* (OA-COG), in which the players are WDs with the objective to minimize their costs given by (16). Clearly, if the OA-COG has a pure strategy Nash equilibrium (NE) then the MEC-OG has an SPE. Hence, in what follows we focus on the existence and computability of pure NE for the OA-COG.

Definition 2. (**Pure NE and Best reply**) 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,

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

Given a strategy profile $d = (d_i', d_{-i})$, a better reply of WD i is a strategy d_i' such that $\bar{C}_i(d_i', d_{-i}) < \bar{C}_i(d_i, d_{-i})$, and a best reply of WD i is a better reply d_i^* such that $\bar{C}_i(d_i^*, d_{-i}) \le \bar{C}_i(d_i, d_{-i}), \forall d_i \in \mathfrak{D}_i$.

Before we formulate our next result let us recall the definition of an exact potential function from [17].

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

$$\bar{C}i(d_i', d_{-i}) - \bar{C}_i(d_i, d_{-i}) = \Phi(d_i', d_{-i}) - \Phi(d_i, d_{-i}). \quad (18)$$

Given an arbitrary ordering of WDs, let us introduce the following shorthand notation,

$$w_a^{\le i}(\mathbf{d}) = \sum_{\{j\in O_a(\mathbf{d})|j\le i\}} w_{j,a}, \quad w_a^{>i}(\mathbf{d}) = \sum_{\{j\in O_a(\mathbf{d})|j> i\}} w_{j,a},$$
$$w_c^{\le i}(\mathbf{d}) = \sum_{\{j\in O_c(\mathbf{d})|j\le i\}} w_{j,c}, \quad w_c^{>i}(\mathbf{d}) = \sum_{\{j\in O_c(\mathbf{d})|j> i\}} w_{j,c}.$$

Theorem 3. *The OA-COG has the exact potential function*

$$\Phi(\boldsymbol{d}) = \sum_{i\in\mathcal{N}}\left(\sum_{a\in\mathcal{A}}\Phi_{i,a}(\boldsymbol{d}) + \sum_{c\in\mathcal{C}}\Phi_{i,c}(\boldsymbol{d}) + \Phi_{i,i}(\boldsymbol{d})\right), \quad (19)$$

where $\Phi_{i,a}(\boldsymbol{d}) = I_{d_i,(a,\cdot)}w_{i,a}w_a^{\le i}(\boldsymbol{d})$, $\Phi_{i,c}(\boldsymbol{d}) = I_{d_i,(\cdot,c)}w_{i,c}w_c^{\le i}(\boldsymbol{d})$, and $\Phi_{i,i}(\boldsymbol{d}) = I_{d_i,i}C_i^l$.

Proof. Let us define function $\Phi_i(\mathbf{d}) = \sum_{a\in\mathcal{A}}\Phi_{i,a}(\mathbf{d}) + \sum_{c\in\mathcal{C}}\Phi_{i,c}(\mathbf{d}) + \Phi_{i,i}(\mathbf{d})$, and rewrite $\Phi(\mathbf{d}) = \sum_{i\in\mathcal{N}}\Phi_i(\mathbf{d})$. To prove that $\Phi(\mathbf{d})$ is an exact potential function, let us consider strategy profiles $\mathbf{d}$ and $\mathbf{d}'$ such that $\mathbf{d}=(d_k, d_{-k})$ and $\mathbf{d}'=(d_k', d_{-k})$, and consider the following two cases.

Case 1: Changing offloading strategy: We start with considering the case when WD k offloads its task in both strategy profiles $\mathbf{d}$ and $\mathbf{d}'$. Let us denote by $d_k = (a,c)$ and $d_k' = (a',c')$ the offloading decisions of WD k in $\mathbf{d}$ and $\mathbf{d}'$, respectively. If $a\neq a'$ and $c\neq c'$ then the difference between the cost of WD k in $\mathbf{d}$ and that in $\mathbf{d}'$ is given by

$$\bar{C}_k(\mathbf{d}) - \bar{C}_k(\mathbf{d}') = w_{k,a}w_a(\mathbf{d}) + w_{k,c}w_c(\mathbf{d}) - \\ - w_{k,a'}w_{a'}(\mathbf{d}) - w_{k,c'}w_{c'}(\mathbf{d}).$$

To compute the change of the potential, observe that $\Phi_{i,i}(\mathbf{d}) = \Phi_{i,i}(\mathbf{d}')$ for all WDs $i \in \mathcal{N}$, since the set of WDs that perform the computation locally is the same in $\mathbf{d}$ and $\mathbf{d}'$. We also have that $\Phi_{i,r}(\mathbf{d}) = \Phi_{i,r}(\mathbf{d}')$ for every resource $r \in \mathcal{A}\cup\mathcal{C}\setminus\{a,a',c,c'\}$ since $O_r(\mathbf{d}) = O_r(\mathbf{d}')$. Furthermore, we observe that $\Phi_i(\mathbf{d}) = \Phi_i(\mathbf{d}')$ for all WDs $i<k$. For WDs $i>k$ that offload their tasks through APs a and a' we have that $\Phi_{i,a}(\mathbf{d}) - \Phi_{i,a}(\mathbf{d}') = w_{i,a}w_{k,a}$ and $\Phi_{i,a'}(\mathbf{d}) - \Phi_{i,a'}(\mathbf{d}') = -w_{i,a'}w_{k,a'}$, respectively. Similarly, for WDs $i > k$ that offload their tasks to ECs c and c' we have that $\Phi_{i,c}(\mathbf{d}) - \Phi_{i,c}(\mathbf{d}') = w_{i,c}w_{k,c}$ and $\Phi_{i,c'}(\mathbf{d}) - \Phi_{i,c'}(\mathbf{d}') = -w_{i,c'}w_{k,c'}$, respectively. For WD k we have the following

$$\Phi_k(\mathbf{d}) - \Phi_k(\mathbf{d}') = \quad w_{k,a}w_a^{\le k}(\mathbf{d}) + w_{k,c}w_c^{\le k}(\mathbf{d}) - \\ - w_{k,a'}w_{a'}^{\le k}(\mathbf{d}) - w_{k,c'}w_{c'}^{\le k}(\mathbf{d}).$$

We hence obtain the equality

$$\Phi(\mathbf{d}) - \Phi(\mathbf{d}') = w_{k,a}w_a^{>k}(\mathbf{d}) + w_{k,c}w_c^{>k}(\mathbf{d}) - w_{k,a'}w_{a'}^{>k}(\mathbf{d}) - \\ w_{k,c'}w_{c'}^{>k}(\mathbf{d}) + w_{k,a}w_a^{\le k}(\mathbf{d}) + w_{k,c}w_c^{\le k}(\mathbf{d}) - w_{k,a'}w_{a'}^{\le k}(\mathbf{d}) - \\ w_{k,c'}w_{c'}^{\le k}(\mathbf{d}) = w_{k,a}w_a(\mathbf{d}) + w_{k,c}w_c(\mathbf{d}) - w_{k,a'}w_{a'}(\mathbf{d}) - \\ w_{k,c'}w_{c'}(\mathbf{d}) = \bar{C}_k(\mathbf{d}) - \bar{C}_k(\mathbf{d}').$$

Similarly, we can show that $\Phi(\mathbf{d}) - \Phi(\mathbf{d}') = \bar{C}_k(\mathbf{d}) - \bar{C}_k(\mathbf{d}')$ if WD k changes only the AP, i.e., if $d_k = (a,c)$ and $d_k' = (a',c)$, $a \neq a'$ or if WD k changes only the EC, i.e., if $d_k = (a,c)$ and $d_k' = (a,c')$, $c \neq c'$.

Case 2: Changing between offloading and local computing: We continue with considering the case when WD k offloads its task in one of the strategy profiles $\mathbf{d}$ and $\mathbf{d}'$ and

AU($\mathbf{d}$)

1: **while** $\exists$ WD j s.t. $d_j \neq \arg\min_{d'_j \in \mathfrak{D}_j} \bar{C}_j(d'_j, d_{-j})$ **do**
2: $\quad d^*_j = \arg\min_{d'_j \in \mathfrak{D}_j} \bar{C}_j(d'_j, d_{-j})$
3: $\quad \mathbf{d} = (d^*_j, d_{-j})$
4: **end while**

Figure 2. Pseudo code of the *AsynchronousUpdates* (AU) algorithm.

it performs the computation locally in the other strategy profile. Let us first consider that WD k offloads its task in strategy profile $\mathbf{d}$, and denote by $d_k = (a, c)$ its offloading decision, and that WD k performs the computation locally in strategy profile $\mathbf{d}'$, i.e., $d'_k = 0$. Then the difference between the cost of WD k in $\mathbf{d}$ and that in $\mathbf{d}'$ is given by

$$\bar{C}_k(\mathbf{d}) - \bar{C}_k(\mathbf{d}') = w_{k,a} w_a(\mathbf{d}) + w_{k,c} w_c(\mathbf{d}) - C^l_k.$$

For the potential, we know that $\Phi_{i,i}(\mathbf{d}) = \Phi_{i,i}(\mathbf{d}')$ for all WDs $i \in \mathcal{N} \setminus \{k\}$ and we also have that $\Phi_{i,r}(\mathbf{d}) = \Phi_{i,r}(\mathbf{d}')$ for every resource $r \in \mathcal{A} \cup \mathcal{C} \setminus \{a, c\}$. Furthermore, we observe that $\Phi_i(\mathbf{d}) = \Phi_i(\mathbf{d}')$ for all $i < k$. For WDs $i > k$ that offload their tasks through AP a we have that $\Phi_{i,a}(\mathbf{d}) - \Phi_{i,a}(\mathbf{d}') = w_{i,a} w_{k,a}$. Similarily, for WDs $i > k$ that offload their tasks to EC c we have that $\Phi_{i,c}(\mathbf{d}) - \Phi_{i,c}(\mathbf{d}') = w_{i,c} w_{k,c}$. Finally, for WD k we have

$$\Phi_k(\mathbf{d}) - \Phi_k(\mathbf{d}') = w_{k,a} w^{\leq k}_a(\mathbf{d}) + w_{k,c} w^{\leq k}_c(\mathbf{d}) - C^l_k.$$

We hence obtain the equality

$$\Phi(\mathbf{d}) - \Phi(\mathbf{d}') = w_{k,a} w^{>k}_a(\mathbf{d}) + w_{k,c} w^{>k}_c(\mathbf{d}) + w_{k,a} w^{\leq k}_a(\mathbf{d}) + w_{k,c} w^{\leq k}_c(\mathbf{d}) - C^l_k = w_{k,a} w_a(\mathbf{d}) + w_{k,c} w_c(\mathbf{d}) - C^l_k = \bar{C}_k(\mathbf{d}) - \bar{C}_k(\mathbf{d}').$$

Similarily, we can show that $\Phi(\mathbf{d}) - \Phi(\mathbf{d}') = \bar{C}_k(\mathbf{d}) - \bar{C}_k(\mathbf{d}')$ if WD k changes its strategy from local computing in $\mathbf{d}$ to offloading to EC c through AP a in $\mathbf{d}'$, i.e., if $d_k = 0$ and $d'_k = (a, c)$, which proves the theorem. □

The existence of an exact potential function implies that the OA-COG has a pure NE [17]. We can thus formulate the following result.

Corollary 1. *The OA-COG has a pure strategy NE $\mathbf{d}^*$. Hence, an SPE $(\mathbf{d}^*, \mathcal{P}^*_r, \mathcal{P}^*_c)$ for the MEC-OG exists.*

There are a variety of algorithms that are known to converge to an equilibrium for exact potential games, such as fictitious play [17], joint strategy fictitious play [18], and the best and better reply dynamics [17]. Nonetheless, they have exponential worst case complexity in general [19], [20]. Thus, the second fundamental question we address in this paper is whether a NE of the OA-COG (and thus an SPE of the MEC-OG) can be computed efficiently.

In what follows we propose the *ImproveLocalComputing* (ILC) algorithm to address this important question. The ILC algorithm starts from a strategy profile in which all WDs perform computation locally. Let us first denote by $\mathcal{N}'$ the set of WDs that have never changed their strategy from local computing to computation offloading (note that at the beginning $\mathcal{N}' = \mathcal{N}$). The ILC algorithm consists of two phases that are executed alternatingly. In the first phase, among all WDs $i \in \mathcal{N}'$ that can decrease their cost by starting to offload, a WD with the maximum task complexity L_i is allowed to perform a best reply. In the second phase, which we refer to as the update phase, WDs $i \in \mathcal{N} \setminus \mathcal{N}'$ are allowed to update their best replies according to the AU algorithm shown in Fig. 2.

In what follows we show that by letting WDs to start to offload in non-increasing order of their task complexities, the ILC algorithm reduces the number of iterations compared to the best reply dynamic that lets WDs to start using cloud resources in an arbitrary order.

Proposition 1. *Let us consider a strategy profile $\mathbf{d}$ in which all WDs $j \in \mathcal{N} \setminus \mathcal{N}'$ perform best replies and let us assume that there is a WD $i \in \mathcal{N}'$ that can decrease its cost by starting to offload to one of the ECs. Then upon WD i performs its best reply, WDs $j \in O(\mathbf{d})$ will not have an incentive to change between ECs.*

Proof. Let us consider two WDs, $i \in \mathcal{N}'$ and $k \in O_c(\mathbf{d})$, and let us assume that WD i can decrease its cost by starting to offload. Furthermore, let us assume that a best reply of WD i is offloading to an EC c, i.e., for any EC $c' \in \mathcal{C} \setminus \{c\}$ the following holds

$$\Big(w_c(\mathbf{d}) + w_{i,c}\Big) w_{i,c} < \Big(w_{c'}(\mathbf{d}) + w_{i,c'}\Big) w_{i,c'}. \tag{20}$$

Let us denote by $\mathbf{d}'$ the resulting strategy profile in which $O_c(\mathbf{d}') = O_c(\mathbf{d}) \cup \{i\}$ and $O_{c'}(\mathbf{d}') = O_{c'}(\mathbf{d})$ for $c' \neq c$. Let us next assume that WD $k \in O_c(\mathbf{d}')$ can decrease its offloading cost by changing its strategy from $(\cdot, c)$ to $(\cdot, c')$, i.e.,

$$w_c(\mathbf{d}') w_{k,c} > \Big(w_{c'}(\mathbf{d}') + w_{k,c'}\Big) w_{k,c'}. \tag{21}$$

Since $O_{c'}(\mathbf{d}') = O_{c'}(\mathbf{d})$ for $c' \neq c$, we have that $w_{c'}(\mathbf{d}') = w_{c'}(\mathbf{d})$. By applying $w_{c'}(\mathbf{d}') = w_{c'}(\mathbf{d})$ in (21) and by combining inequalities (20) and (21) we obtain that $\sqrt{L_i} > \sqrt{L_k}$ holds, which contradicts the fact that the ILC algorithm allows WDs $i \in \mathcal{N}'$ to start to offload in non-increasing order of their task complexities L_i. This proves the result. □

Note that WDs can change between ECs only if the congestion in an EC decreases, i.e., if one of the WDs changes its strategy from offloading to local computing. This is, however, rarely the case, and as we show later, the number of iterations needed to compute an equilibrium allocation of offloading decisions using the ILC algorithm is on average almost linear in the number of WDs.

C. Implementation considerations

In what follows we discuss how the SPE can be implemented in practice. Given the information about the resource allocation policy adopted by the operator, WDs perform best replies one at a time according to the ILC algorithm. Upon its turn, a WD computes the set of its best replies based on the information about the congestion on resources, as provided by the operator. If it can improve its current offloading decision then it reports one of its best replies to the operator, otherwise it reports its current offloading decision. The operator then sends the updated information about the congestion on the resources to the next WD that is supposed to update its offloading decision. Upon convergence, given the equilibrium offloading decisions of WDs computed by the ILC algorithm, the operator allocates wireless and computing resources optimally according to (9) and (10). By Corollary 1 the resulting state is an SPE.

IV. Price of Anarchy

We have so far shown that the OA-COG has a pure strategy NE, and hence the MEC-OG has an SPE. Furthermore, both can be computed efficiently. In what follows we provide a bound on the suboptimality of the computed SPE, with respect to a solution that minimizes the system cost. We do so by bounding the price of anarchy (PoA)

of the SPE (denoted by $PoA_{\text{MEC-OG}}$), which is defined as the ratio of the cost in the worst case SPE $(\mathbf{d}^*, \mathcal{P}_r^*, \mathcal{P}_c^*)$ and the cost in an optimal solution $(\hat{\mathbf{d}}, \hat{\mathcal{P}}_r, \hat{\mathcal{P}}_c)$.

To compute such a bound, let us denote by $PoA(\mathcal{P}_r, \mathcal{P}_c)$ the PoA of the strategic game played by the WDs for a policy $(\mathcal{P}_r, \mathcal{P}_c)$ of the operator for which an equilibrium allocation $\mathbf{d}^*$ of offloading decisions exists. The $PoA(\mathcal{P}_r, \mathcal{P}_c)$ is the ratio of the worst case NE cost and the optimal offloading cost, i.e.,

$$PoA(\mathcal{P}_r, \mathcal{P}_c) = \frac{\max_{\mathbf{d}^* \in \mathfrak{D}^*} \sum_{i \in \mathcal{N}} C_i(\mathbf{d}^*, \mathcal{P}_r, \mathcal{P}_c)}{\min_{\mathbf{d} \in \mathfrak{D}} \sum_{i \in \mathcal{N}} C_i(\mathbf{d}, \mathcal{P}_r, \mathcal{P}_c)}, \quad (22)$$

where $\mathfrak{D}^*$ is the set of equilibria of offloading decisions under $(\mathcal{P}_r, \mathcal{P}_c)$. In what follows we provide an upper bound on $PoA_{\text{OA-COG}} = PoA(\mathcal{P}_r^*, \mathcal{P}_c^*)$ of the strategic game OA-COG.

Theorem 4. $PoA_{\textit{OA-COG}} \leq \frac{3+\sqrt{5}}{2}$.

Proof. Our proof is inspired by Theorem 3.1 in [21], which provides a PoA bound for normalized weighted congestion games. Our proof extends the PoA bound to the OA-COG, which is not a normalized weighted congestion game.

We start with defining the set $\mathcal{R} = \mathcal{N} \cup \mathcal{A} \cup \mathcal{C}$ of all resources available in the system. Furthermore, we denote by $\mathcal{R}_{d_i}$ the set of resources that WD i uses in strategy profile $\mathbf{d}$, and we use $\mathbf{d}^*$ and $\hat{\mathbf{d}}$ to denote a NE and an optimal strategy profile of the OA-COG, respectively. Let us define the local computing weight $w_{i,i} \triangleq \sqrt{L_i / F_i^l}$ for each WD $i \in \mathcal{N}$, and the set of WDs using local computing link i $O_i(\mathbf{d}) = \{i | d_i = i\}$. Observe that either $O_i(\mathbf{d}) = \emptyset$ or $O_i(\mathbf{d}) = \{i\}$ holds since the local computing resources are not shared among WDs. We can thus express the total weight $w_i(\mathbf{d}) = \sum_{i \in O_i(\mathbf{d})} w_{i,i}$ associated with local computing link i, which is either $w_i(\mathbf{d}) = 0$ or $w_i(\mathbf{d}) = w_{i,i}$.

Using the above notation we can express the system cost $C(\mathbf{d}, \mathcal{P}_r^*, \mathcal{P}_c^*)$ for the OA-COG in a strategy profile $\mathbf{d}$ as

$$C(\mathbf{d}, \mathcal{P}_r^*, \mathcal{P}_c^*) = \sum_{r \in \mathcal{R}} \sum_{i \in O_r(\mathbf{d})} w_r(\mathbf{d}) w_{i,r} = \sum_{r \in \mathcal{R}} w_r^2(\mathbf{d}). \quad (23)$$

Furthermore, from the definition of a NE we obtain

$$\sum_{r \in \mathcal{R}_{d_i^*}} w_r(\mathbf{d}^*) w_{i,r} \leq \sum_{r \in \mathcal{R}_{d_i^*} \cap \mathcal{R}_{\hat{d}_i}} w_r(\mathbf{d}^*) w_{i,r} + \quad (24)$$

$$\sum_{r \in \mathcal{R}_{d_i^*} \setminus \mathcal{R}_{\hat{d}_i}} \big(w_r(\mathbf{d}^*) + w_{i,r}\big) w_{i,r} \leq \sum_{r \in \mathcal{R}_{\hat{d}_i}} \big(w_r(\mathbf{d}^*) + w_{i,r}\big) w_{i,r}.$$

First, by summing inequality (24) over all WDs i we obtain

$$\sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_{d_i^*}} w_r(\mathbf{d}^*) w_{i,r} \leq \sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_{\hat{d}_i}} \big(w_r(\mathbf{d}^*) + w_{i,r}\big) w_{i,r}. \quad (25)$$

Second, by reordering summations, we can rewrite (25) as

$$\sum_{r \in \mathcal{R}} \sum_{i \in O_r(\mathbf{d}^*)} w_r(\mathbf{d}^*) w_{i,r} \leq \sum_{r \in \mathcal{R}} \sum_{i \in O_r(\hat{\mathbf{d}})} \big(w_r(\mathbf{d}^*) w_{i,r} + w_{i,r}^2\big). \quad (26)$$

Next, from the definition of the total weight $w_r(\mathbf{d}) = \sum_{i \in O_r(\mathbf{d})} w_{i,r}$ associated with resource r and from $\sum_{i \in O_r(\mathbf{d})} w_{i,r}^2 \leq w_r^2(\mathbf{d})$ we obtain

$$\sum_{r \in \mathcal{R}} w_r^2(\mathbf{d}^*) \leq \sum_{r \in \mathcal{R}} w_r(\mathbf{d}^*) w_r(\hat{\mathbf{d}}) + \sum_{r \in \mathcal{R}} w_r^2(\hat{\mathbf{d}}). \quad (27)$$

We can now use the Cauchy-Schwartz inequality $(\sum_{r \in \mathcal{R}} a_r b_r \leq \sqrt{\sum_{r \in \mathcal{R}} a_r^2 \sum_{r \in \mathcal{R}} b_r^2})$ to obtain

$$\sum_{r \in \mathcal{R}} w_r^2(\mathbf{d}^*) \leq \sqrt{\sum_{r \in \mathcal{R}} w_r^2(\mathbf{d}^*) \sum_{r \in \mathcal{R}} w_r^2(\hat{\mathbf{d}})} + \sum_{r \in \mathcal{R}} w_r^2(\hat{\mathbf{d}}). \quad (28)$$

If we divide the right and the left side of inequality (28) by $\sum_{r \in \mathcal{R}} w_r^2(\hat{\mathbf{d}}) > 0$ we can rewrite it using (23) as

$$\frac{C(\mathbf{d}^*, \mathcal{P}_r^*, \mathcal{P}_c^*)}{C(\hat{\mathbf{d}}, \mathcal{P}_r^*, \mathcal{P}_c^*)} \leq \sqrt{\frac{C(\mathbf{d}^*, \mathcal{P}_r^*, \mathcal{P}_c^*)}{C(\hat{\mathbf{d}}, \mathcal{P}_r^*, \mathcal{P}_c^*)}} + 1. \quad (29)$$

Since (29) holds for any NE of the OA-COG, it holds for the worst case NE too, and thus we have

$$PoA_{\text{OA-COG}} \leq \sqrt{PoA_{\text{OA-COG}}} + 1. \quad (30)$$

By solving (30) we obtain that $PoA_{\text{OA-COG}} \leq \frac{3+\sqrt{5}}{2}$, which proves the theorem. □

Given the PoA bound for the OA-COG we are now ready to provide a PoA bound for the MEC-OG.

Theorem 5. $PoA_{\textit{MEC-OG}} \leq \frac{3+\sqrt{5}}{2}$.

Proof. Let $(\mathbf{d}^*, \mathcal{P}_r^*, \mathcal{P}_c^*)$ be an SPE of the MEC-OG and let $(\hat{\mathbf{d}}, \hat{\mathcal{P}}_r, \hat{\mathcal{P}}_c)$ be an optimal solution. Clearly, by Theorem 1 we have that $C(\hat{\mathbf{d}}, \hat{\mathcal{P}}_r, \hat{\mathcal{P}}_c) = C(\hat{\mathbf{d}}, \mathcal{P}_r^*, \mathcal{P}_c^*)$, as $(\mathcal{P}_r^*, \mathcal{P}_c^*)$ is an optimal policy. The result then follows from the definition of the PoA and from Theorem 4. □

V. Numerical Results

In the following we show results from extensive simulations to evaluate the system performance from the perspective of the operator of the WDs.

For the simulations we placed ECs and WDs uniformly at random over a square area of $1km \times 1km$, and we placed 5 APs at random on a *regular grid* with 25 points defined over the area. This uniform deployment corresponds to a dense urban area. We consider that the channel gain of WD i in the case of offloading through the same AP a depends on its distance $d_{i,a}$ from the AP and on the path loss exponent α. We use $\alpha = 4$ according to the path loss model in urban and suburban areas [22]. For simplicity we assign a bandwidth of $B_{i,a} = 5$ *MHz* to each communication link $(i, a) \in \mathcal{N} \times \mathcal{A}_i$. The transmit power $P_{i,a}^t$ at which WD i offloads the data through AP a is drawn from a continuous uniform distribution on $[0.05, 0.18]$ *W* according to [23]. Given the noise power P_n we calculate the transmission rate $R_{i,a}$ achievable to WD i for offloading to AP a as $R_{i,a} = B_{i,a} log(1 + d_{i,a}^{-\alpha} \frac{P_{i,a}^t}{P_n})$. The input data size D_i is drawn from a uniform distribution on $[0.2, 4]$ *Mb*, and the number X of CPU cycles required per data bit is a Gamma distributed random variable with the shape $k = 0.5$ and scale $\theta = 1.6$. Given D_i and X, we calculate the complexity of a task as $L_i = D_i X$.

We consider two operator policies in the evaluation. We refer to $(\mathcal{P}_r^*, \mathcal{P}_c^*)$ as the *optimal allocation* (OA) policy. Under the OA policy the WDs can use the ILC algorithm for computing an SPE, as shown before. As a baseline for comparison, we consider the policy $(\mathcal{P}_r^{ea}, \mathcal{P}_c^{ea})$ that shares the resources equally among WDs. We refer to this policy as the *equal allocation* (EA) policy, and to the resulting strategic game played by WDs as the *equal allocation computation offloading game* (EA-COG). As shown in [24] equilibria under the policy $(\mathcal{P}_r^{ea}, \mathcal{P}_c^{ea})$ exist for $C = 1$, and the algorithm proposed in [24] can be extended such that it computes a NE also for $C > 1$, as we discuss next. Observe that under the EA policy and for $C \geq 1$, WDs offloading their computation to a particular EC receive an equal share fraction of the EC's computing capability, and thus the best reponse EC is the same for all WDs. We can thus use

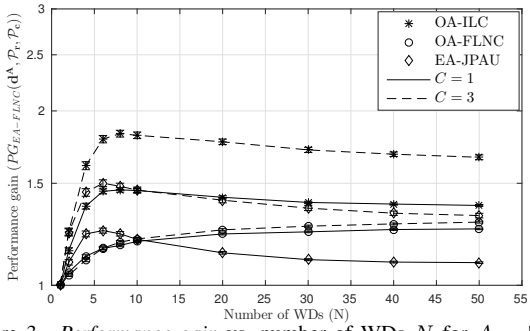

Figure 3. *Performance gain* vs. number of WDs N for $A = 5$ APs. *Homogeneous ECs*, $F^{c,tot} = 192GHz$.

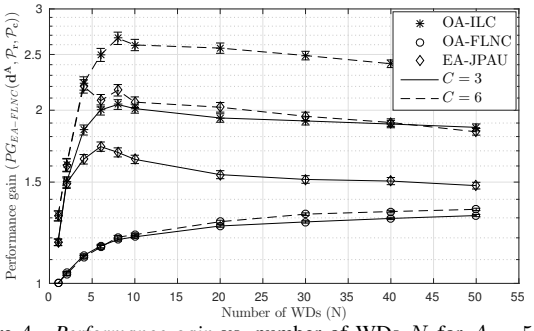

Figure 4. *Performance gain* vs. number of WDs N for $A = 5$ APs. *Heterogeneous ECs*, $F^{c,tot} = 192GHz$.

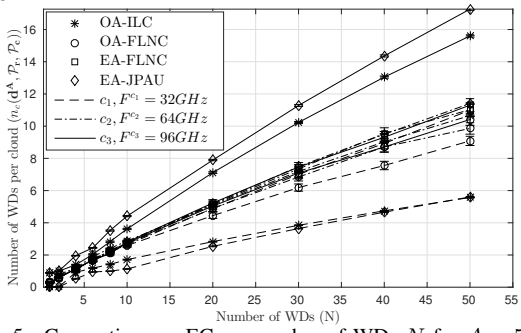

Figure 5. Congestion per EC vs. number of WDs N for $A = 5$ APs, $C = 3$ ECs. *Heterogeneous ECs*, $F^{c,tot} = 192GHz$.

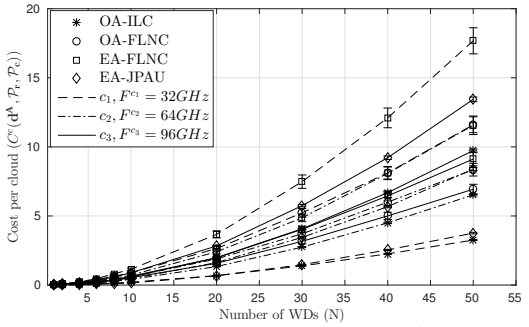

Figure 6. Cost per EC vs. number of WDs N for $A = 5$ APs, $C = 3$ ECs. *Heterogeneous ECs*, $F^{c,tot} = 192GHz$.

the algorithm proposed in [24] for computing a NE of the EA-COG for $C \geq 1$. We refer to the resulting algorithm as the *JoinAndPlayAsynchronousUpdates* (JPAU) algorithm.

As a baseline for the ILC and JPAU algorithms proposed for computing an equilibrium of the OA-COG and EA-COG, respectively, we use the *FastestLinkNearestCloud* (FLNC) algorithm. According to the FLNC algorithm WDs offload the computation through the AP with the highest achievable transmission rate and to the EC closest to the chosen AP. Observe that FLNC can be used with both operator policies. The results shown are the averages of 1000 simulations, together with 95% confidence intervals.

A. User-oriented performance

We start with considering the system performance from the point of view of the WDs. We define the *performance gain* $PG_{\text{EA-FLNC}}(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c)$ (w.r.t. the EA-FLNC) for a strategy profile $\mathbf{d}^A$ computed by algorithm $A \in \{\text{ILC}, \text{JPAU}, \text{FLNC}\}$ under a resource allocation policy $(\mathcal{P}_r, \mathcal{P}_c) \in \{(\mathcal{P}_r^*, \mathcal{P}_c^*), (\mathcal{P}_r^{ea}, \mathcal{P}_c^{ea})\}$ as

$$PG_{\text{EA-FLNC}}(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c) = \frac{C(\mathbf{d}^{FLNC}, \mathcal{P}_r^{ea}, \mathcal{P}_c^{ea})}{C(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c)}.$$

Fig. 3 shows the *performance gain* as a function of the number N of WDs for two MEC systems, one with C=1 (F^{c_1}=192 *GHz*) and one with C=3 (F^{c_i}=64 *GHz*), i.e., ECs are homogeneous. The figure shows that the *performance gain* is largest when the operator uses the OA policy and WDs offload according to an equilibrium computed by the ILC algorithm. Interestingly, even OA-FLNC outperforms EA-JPAU for C=1 ECs and N>10 WDs. These results indicate that the operator's resource allocation policy has a large impact on the user-perceived performance. Overall, we can observe that the *performance gain* increases with a decreasing marginal gain in N, which suggests that the achievable *performance gain* is limited by the congestion on the APs and ECs.

Fig. 4 shows the corresponding *performance gain* for heterogeneous ECs for two MEC systems, one with C= 3 ECs and one with C=6 ECs. The total cloud computing capability $F^{c,tot}$=192GHz of the system is distributed among the ECs such that F^{c_1}=32 *GHz* and F^{c_i}=$F^{c_{i-1}}$+ 32 *GHz*, $i > 1$, for $C = 3$ ECs, and $F^{c_1} = 12$ *GHz* and F^{c_i}=$F^{c_{i-1}}$+8 *GHz*, i>1, for C=6 ECs. As in Fig. 3, the results in Fig. 4 show a decreasing marginal gain in N and confirm that the largest *performance gain* is achieved by the OA-ILC. Nonetheless, a comparison of Fig. 3 and Fig. 4 reveals that the *performance gain* is affected by the number of ECs in the system and the way the total cloud computing capability is shared among the ECs. One the one hand, the *performance gain* increases with C. On the other hand, the *performance gain* for C=3 ECs is greater in the case of heterogeneous ECs than that in the case of homogeneous ECs. Thus, OA-ILC is most beneficial when edge cloud resources are heterogeneous. The improved performance is partly due to that the WDs in the baseline strategy profile (computed by the FLNC) offload their tasks through the fastest link to the EC that is closest to the chosen AP, and since WDs, APs and ECs are randomly placed over the area, the number of WDs per EC is not proportional to its computing capability, as we will see later.

B. Infrastructure-oriented performance

In order to evaluate the system performance from operator's perspective, we investigate how the choice of the resource allocation policy and the algorithm for computing the offloading decisions of WDs affects the number $n_c(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c)$ of WDs per EC and the cost $C^c(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c) = \sum_{i \in O_c(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c)} C_i(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c)$ per EC. For consistency, we show results for a system with heterogeneous cloud resources, i.e., $F^{c,tot} = 192$ *GHz* divided among three ECs such that $F^{c_1} = 32$ *GHz* and $F^{c_i} = F^{c_{i-1}} + 32$ *GHz*, for $i > 1$.

Fig. 5 and Fig. 6 show $n_c(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c)$ and $C^c(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c)$ for each of the ECs as a function of the number N of WDs, respectively. The results are shown for the ILC, JPAU and FLNC algorithms under both the OA and EA resource allocation policies. By looking at $n_c(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c)$ for all ECs for a fixed N, we observe from Fig. 5 that the ratio of

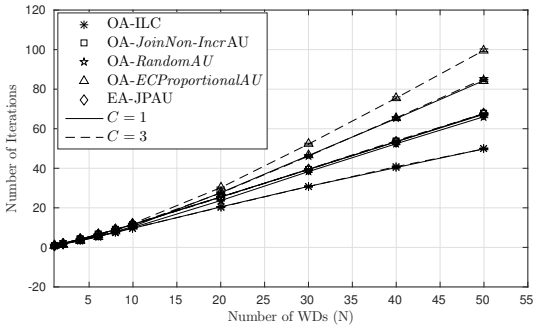

Figure 7. Number of iterations vs. number of WDs N for $A = 5$. *Homogeneous ECs*, $F^{c,tot} = 192GHz$.

the WDs that offload their tasks decreases as N increases. This happens because the number of WDs that cannot benefit from offloading due to high congestion on the shared resources increases with N. Fig. 5 also shows that the difference in the congestion experienced by the ECs is smallest when the offloading decisions of the WDs are computed by the FLNC algorithm. This is due to that in the strategy profile computed by the FLNC algorithm WDs offload their tasks to the EC that is closest to the fastest AP, and since the WDs, APs, and ECs are placed uniformly at random over the region, all ECs experience the same congestion on average. Consequently, the corresponding cost per EC, shown in Fig. 6, is inverse proportional to the computing capability of the EC.

On the contrary, in the case of equilibria computed by ILC and by JPAU (i.e. equilibria under the OA and EA policies, respectively) the congestion and the cost per EC are proportional to the computing capability of the EC as shown in Fig. 5 and Fig. 6, respectively. We also observe that the total number of WDs that offload their tasks and the total offloading cost are higher in an equilibrium computed by the JPAU algorithm than in an equilibrium computed by the ILC algorithm. This is due to that the cloud computing resources are shared among WDs independently of their tasks' complexities in the case of the EA policy, and consequently the WDs overuse the ECs.

C. Computational complexity

We characterize the computational complexity of an algorithm as the number of iterations needed to compute a computation offloading strategy profile. Since the OA-COG is a potential game, we use the AU algorithm (c.f. Fig. 2) as a baseline for comparison, as it is guaranteed to converge from an arbitrary initial strategy profile [17]. For the AU algorithm we consider three initial strategy profiles: a randomly chosen initial strategy profile (*Random*AU), an initial strategy profile in which all WDs offload their tasks such that the number of WDs offloading the computation to an EC is proportional to its computing capability (*ECProportional*AU), and an empty strategy profile where the WDs enter the game in non-increasing order of their task complexities (*JoinNon-Incr*AU). Furthermore, we consider the complexity of computing an equilibrium of the EA-COG using the JPAU algorithm.

Fig. 7 shows the number of iterations needed to compute an equilibrium of the OA-COG and an equilibrium of the EA-COG, as a function of N for the same set of parameters as in Fig. 3. We observe that the number of iterations scales approximately linearly with N in all cases and that computing an equilibrium of the OA-COG using the ILC algorithm is more efficient than computing an equilibrium of the EA-COG using the JPAU algorithm; the difference is up to 50%.

We also observe that the choice of the initial strategy profile affects the complexity of computing an equilibrium of the OA-COG, and we make three observations. First, the number of iterations required by ILC and by *JoinNon-Incr*AU is insensitive to the number of ECs, while the number of iterations required by *Random*AU and by *ECProportional*AU increases with the number of ECs. This is due to that in the case of ILC and of *JoinNon-Incr*AU the WDs start using ECs in non-increasing order of their task complexities, and thus it follows from Proposition 1 that when a new WD starts offloading, WDs will not have an incentive to change between ECs. This is not true in the case of *Random*AU and of *ECProportional*AU, since they start from a strategy profile where WDs did not start to offload in the order of the complexities of their tasks, and consequently the WDs can decrease their offloading cost not only by changing between the APs, but also by changing between the ECs. Second, the *ECProportional*AU has the highest computational complexity. This is due to that *ECProportional*AU starts from an initial strategy profile that has the highest congestion on the resources and thus when a WD updates its strategy the number of WDs affected by the update step is higher than in the case of the other initial strategy profiles. Finally, the smallest computational complexity can be achieved by the proposed ILC algorithm. On the one hand, this is because the WDs do not have to choose their initial strategy as in the case of the *JoinNon-Incr*AU. On the other hand, the WDs cannot decrease their offloading cost by changing between the ECs as in the case of the *Random*AU and *ECProportional*AU.

To summarize, the proposed OA-ILC algorithm can provide a significant reduction in terms of completion times and has low computational complexity, and could be a good candidate for coordinating the offloading decisions of WDs for edge computing.

VI. Related Work

There is a large body of recent works on computation offloading for mobile cloud computing [25], [26], [11], [27], [28], [5], [29], [24]. Many of these works assume that the offloading decisions of devices are determined by a centralized entity with the objective to meet the energy and latency constraints of the devices [25], [26], [11], [27]. [25] considered that devices offload the computation either to a computationally limited local cloud or to a computationally rich remote cloud, and proposed a policy that schedules resources in the clouds so as to meet the delay requirements of the applications. [26], [11], [27] formulated the computation offloading problem as an optimization problem that minimizes the energy consumption of the mobile devices under latency constraints. [26] considered that devices may offload their tasks to an edge cloud through a base station, and proposed a policy for managing computing and communication resources assuming that the base station has perfect knowledge about the system. [11], [27] considered a network composed of multiple cells, each equipped with an edge cloud. [11] proposed an iterative algorithm for jointly optimizing the allocation of computing and uplink bandwidth resources, and [27] proposed an iterative algorithm for jointly optimizing the allocation of computing and both uplink and downlink bandwidth resources. Unlike these works, we consider that devices make offloading decisions autonomously.

Closer related to ours are recent works that propose decentralized algorithms based on a game theoretic treatment of the computation offloading problem [28], [5], [29], [24], [30], [31]. [28] considered that devices may offload the computation to the cloud through a single wireless link if doing so minimizes their own energy consumption, and proved the existence of equilibria when devices with the same delay budget compete only for wireless resources. [5], [29], [24] considered that devices may offload their tasks to the cloud through one of multiple wireless links so as to minimize the linear combination of the delay and the energy consumption. [5] considered the congestion only on the wireless links and proved the existence of equilibria under the assumption that a device experiences the same channel gain for all wireless links. [29] extended the equilibrium existence results of [5] to a dynamic environment, where devices may be active or inactive. [24] considered that devices may offload their tasks to the cloud through one of multiple heterogeneous wireless links, modeled the congestion on both cloud and wireless links and provided a polynomial time algorithm for computing equilibria. Authors in [31] considered the interaction between devices that always offload their tasks and an operator that optimizes the allocation of wireless and computing resources. [30] considered a fog computing system where multiple devices may offload their computational tasks to each other or to an edge cloud and provided an efficient algorithm for computing a mixed strategy equilibrium in a decentralized way. Our work differs significantly from these works, as we model the congestion on multiple heterogeneous wireless links and in edge clouds, we consider devices that can autonomously decide whether to offload, and consider that the resources in the system are managed by an operator.

Closest to our work in the literature on game theory is [32], which considers the effectiveness of Stackelberg strategies for atomic congestion games. Authors in [32] consider that the leader controls a subset of non-selfish players, focus on affine latency functions and on congestion games on parallel links. On the contrary, in our model the leader manages the sharing of resources, and we consider a player-specific weighted network congestion game for which the existence of equilibria is not known in general. Thus, our work provides a novel game theoretic perspective on congestion games.

VII. Conclusion

We have provided a game theoretical analysis of selfish computation offloading in a mobile edge computing system where wireless and computing resources are jointly managed by an operator, and devices make offloading decisions autonomously so as to minimize the completion times of their tasks. Based on a Stackelberg model of the interaction between the operator and devices, we proved the existence of an equilibrium allocation policy and we proposed an efficient decentralized approximation algorithm for computing offloading decisions of devices. Our numerical results show that the proposed algorithm is computationally efficient and can significantly improve the system performance through optimally allocating wireless and computing resources to the devices, while allowing the devices to make their offloading decisions autonomously.

References

[1] M. Hakkarainen, C. Woodward, and M. Billinghurst, "Augmented assembly using a mobile phone," in *Proc. of IEEE/ACM ISMAR*, Sept 2008, pp. 167–168.

[2] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," in *Proc. of IEEE PerCom*, March 2009, pp. 1–9.

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

[4] S. R. Group, "The leading cloud providers continue to run away with the market," Tech. Rep., 2017.

[5] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM TON*, no. 5, pp. 2795–2808, 2016.

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

[7] 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.

[8] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *ACM SIGCOMM CCR*, vol. 45, no. 5, pp. 37–42, 2015.

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

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

[11] 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.

[12] 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.

[13] T. Li, D. Baumberger, and S. Hahn, "Efficient and scalable multiprocessor fair scheduling using distributed weighted round-robin," *SIGPLAN Not.*, vol. 44, no. 4, pp. 65–74, Feb. 2009.

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

[15] 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.

[16] S. Jošilo and G. Dán, "Decentralized scheduling for offloading of periodic tasks in mobile edge computing," in *Proc. of IFIP NETWORKING*, 2018.

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

[18] J. R. Marden, G. Arslan, and J. S. Shamma, "Joint strategy fictitious play with inertia for potential games," *IEEE Trans. on Automatic Control*, vol. 54, no. 2, pp. 208–220, Feb 2009.

[19] A. Fabrikant, C. Papadimitriou, and K. Talwar, "The complexity of pure nash equilibria," in *Proc. of ACM STOC*, 2004, pp. 604–612.

[20] H. Ackermann, H. Röglin, and B. Vöcking, "On the impact of combinatorial structure on congestion games," *Journal of the ACM (JACM)*, vol. 55, no. 6, p. 25, 2008.

[21] B. Awerbuch, Y. Azar, and A. Epstein, "The price of routing unsplittable flow," in *Proc. of ACM STOC*, 2005, pp. 57–66.

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

[23] E. Casilari, J. M. Cano-García, and G. Campos-Garrido, "Modeling of current consumption in 802.15. 4/zigbee sensor motes," *Sensors*, vol. 10, no. 6, pp. 5443–5468, 2010.

[24] S. Jošilo and G. Dán, "Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks," *IEEE Trans. on Mob. Comp*, vol. 18, no. 1, pp. 207–220, 2019.

[25] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing," in *IEEE GC Wkshps*, 2015, pp. 1–6.

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

[27] A. Al-Shuwaili, O. Simeone, A. Bagheri, and G. Scutari, "Joint uplink/downlink optimization for backhaul-limited mobile cloud computing with user scheduling," *IEEE T-SIPN*, pp. 787–802, 2017.

[28] E. Meskar, T. D. Todd, D. Zhao, and G. Karakostas, "Energy aware offloading for competing users on a shared communication channel," *IEEE Trans. on Mob. Comp*, vol. 16, no. 1, pp. 87–96, 2017.

[29] J. Zheng, Y. Cai, Y. Wu, and X. S. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. on Mob. Comp*, 2018.

[30] S. Jošilo and G. Dán, "Decentralized algorithm for randomized task allocation in fog computing systems," *IEEE/ACM Transactions on Networking, accepted for publication*, 2018.

[31] ——, "Joint allocation of computing and wireless resources to autonomous devices in mobile edge computing," in *Proc. of ACM SIGCOMM Mecomm'18 Workshop*, 2018.

[32] D. Fotakis, "Stackelberg strategies for atomic congestion games," *Theory of Computing Systems*, vol. 47, no. 1, pp. 218–249, 2010.