Joint Management of Wireless and Computing Resources for Computation Offloading in Mobile Edge Clouds

Slađana Jošilo and György Dán
Division 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 computation offloading problem in an edge computing system in which an operator jointly manages wireless and computing resources across devices that make their offloading decisions autonomously with the objective to minimize their own completion times. We develop a game theoretical model of the interaction between the devices and an operator that can implement one of two resource allocation policies, a cost minimizing or a time fair resource allocation policy. We express the optimal cost minimizing resource allocation policy in closed form and prove the existence of Stackelberg equilibria for both resource allocation policies. We propose two efficient decentralized algorithms that devices can use for computing equilibria of offloading decisions under the cost minimizing and the time fair resource allocation policies. We establish bounds on the price of anarchy of the games played by the devices and by doing so we show that the proposed algorithms have bounded approximation ratios. Our simulation results show that the cost minimizing resource allocation policy can achieve significantly lower completion times than the time fair allocation policy. At the same time, the convergence time of the proposed algorithms is approximately linear in the number of devices, and thus they could be effectively implemented for edge computing resource management.**

***Index terms*—** edge computing, resource management, computation offloading, game theory, decentralized algorithms

I. Introduction

The evolution of wireless access and the Internet of Things are driving the development of a variety of mobile applications such as face and object recognition, mobile augmented reality, and cognitive assistance [1], [2], [3]. These emerging human-in-the-loop applications have delay and computational requirements that often surpass the capabilities of handheld devices [4].

A promising approach to support these emerging applications is mobile edge computing (MEC) [5]. The key idea of MEC is to move cloud resources towards the network edge so as to overcome the issue of high end-to-end transmission delays, which are inherent to computation offloading to remote centralized clouds such as Microsoft Azure or Amazon EC2 [6]. Owing to the proximity of computing resources to the end users, MEC has the potential to significantly reduce response times for individual devices by allowing them to offload the computationally intensive tasks through a wireless network to nearby edge clouds. However, computation offloading to edge clouds imposes a huge load on limited wireless and computing resources, and thus the response times might be adversely affected by the contention for MEC resources.

In order to keep response times as low as possible, it is thus essential to jointly manage the wireless and the computing resources. Nonetheless, joint resource management in a MEC system is inherently challenging for various reasons. First, it requires one to take into consideration the heterogeneity of the devices and their workloads. For example, the devices can differ in terms of their computing capabilities, the amount of data they need to offload and the delay and the computational requirements of the tasks they generate. Second, devices in MEC systems are likely to be autonomous entities, and thus they may be interested in maximizing their own performance [7], [8]. Finally, MEC systems may consist of multiple heterogeneous communication and computing resources, e.g., wireless access points with different bandwidths and edge clouds with different computing capabilities. Therefore, the joint management of wireless and computing resources in MEC systems should be performed in accordance with the individual interest of the heterogeneous devices, the characteristics of their tasks and the heterogeneity of the infrastructure.

In this paper we consider devices that aim at minimizing the completion times of their own tasks, and we address the corresponding computation offloading problem by considering the interaction between an operator that jointly manages the wireless and computing resources, and devices that decide autonomously whether or not to offload the computations and in the case of offloading which of multiple heterogeneous wireless and computing resources to use. We model the problem as a multiple-leader common-follower Stackelberg game, in which devices are leaders and the operator is the follower. We consider two resource allocation policies for the operator, called the cost minimizing and the time fair resource allocation policy. We show that the resulting games played by the devices can be transformed into a weighted congestion game and into a player-specific congestion game under the cost minimizing and the time fair policy, respectively. We provide a closed form solution for the optimal cost minimizing resource allocation policy and we prove that Stackelberg equilibria exist for both policies. Based on our constructive equilibrium existence proofs, we propose two efficient decentralized algorithms that devices can use for computing offloading decisions under the cost minimizing and the time fair policy of the operator, respectively. We provide upper bounds on the price of anarchy of the games played by the devices, and thus we show that our proposed algorithms serve as approximation algorithms for the completion time minimization problems defined for the cost minimizing and the time fair resource allocation policies. Our analytical results show that the cost minimizing policy can guarantee better performance in terms of the worst case system cost and that the time fair policy can guarantee better performance in terms of the worst case computational complexity. Finally, we use simulations to show that the completion times achieved under the cost minimizing policy are significantly lower than the completion times achieved

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

TABLE I
SUMMARY OF KEY NOTATIONS

Notation	Description
$\mathcal{N}$	Set of N WDs
$\mathcal{A}$	Set of A APs
$\mathcal{A}_i$	Set of APs available for offloading to WD i
$\mathcal{C}$	Set of C ECs
$\mathcal{P}_c$	Operator's computing resource allocation policy
$\mathcal{P}_r$	Operator's rate allocation policy
D_i	Mean size of the input data for WD i
L_i	Mean task complexity for WD i
F_i^l	Computational capability of WD i
C_i^l	Local computing cost for WD i
$R_{i,a}$	Uplink PHY rate of WD i towards AP a
$u_{i,a}$	Uplink access provisioning coefficient, $(i,a)\in\mathcal{N}\times\mathcal{A}_i$
F^c	Computing capability of EC c
$p_{i,c}$	Computing power provisioning coefficient,$(i,c)\in\mathcal{N}\times\mathcal{C}$
$\mathfrak{D}_i$	Set of feasible decisions for WD i
d_i	Decision of WD i, $d_i\in\mathfrak{D}_i$
$\mathbf{d}$	Strategy profile
$O_a(\mathbf{d})$	Set of $n_a(\mathbf{d})$ WDs offloading through AP a in $\mathbf{d}$
$O_c(\mathbf{d})$	Set of $n_c(\mathbf{d})$ WDs offloading to EC c in $\mathbf{d}$
$O_{(a,c)}(\mathbf{d})$	Set of WDs offloading through AP a to EC c in $\mathbf{d}$
$O(\mathbf{d})$	Set of all WDs that offload their tasks in $\mathbf{d}$
$\omega_{i,a}(\mathbf{d},\mathbf{u}_a)$	Uplink rate of WD $i\in O_a(\mathbf{d})$ for $\mathbf{u}_a$
$F_i^c(\mathbf{d},\mathbf{p}_c)$	Computing capability of WD $i\in O_c(\mathbf{d})$ for $\mathbf{p}_c$
$C_{i,a}^c(\mathbf{d},\mathbf{u}_a,\mathbf{p}_c)$	Offloading cost of WD $i, d_i=(a,c)$ for $\mathbf{u}_a$ and $\mathbf{p}_c$ in $\mathbf{d}$
$C_i(\mathbf{d},\mathbf{u},\mathbf{p})$	Cost of WD i for $\mathbf{u}$ and $\mathbf{p}$ in $\mathbf{d}$
$C(\mathbf{d},\mathbf{u},\mathbf{p})$	Total cost in the system for $\mathbf{u}$ and $\mathbf{p}$ in $\mathbf{d}$

under the time fair policy and that the complexity of computing an equilibrium is on average almost linear in the number of devices for both policies.

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

II. System Model

We consider an edge computing system that consists of a set $\mathcal{N}=\{1,2,\ldots,N\}$ of wireless devices (WDs), a set $\mathcal{A}=\{1,2,\ldots,A\}$ of access points (APs), a set $\mathcal{C}=\{1,2,\ldots,C\}$ 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. For ease of reference, the key notations used in the paper are summarized in Table I.

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 L_i and D_i can be expressed as $L_i=D_iE[X]$. To make the analysis

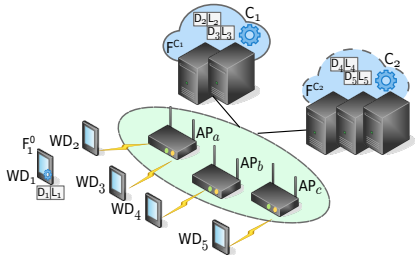

Fig. 1. Example of an edge computing system with $N=5$ WDs, $C=2$ ECs and $A=3$ APs. Transmission rates and cloud computing power may be actively managed by the operator.

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 $O_{(a,c)}(\mathbf{d})\triangleq O_a(\mathbf{d})\cap O_c(\mathbf{d})$ of WDs that offload their tasks through AP a to EC c and 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* $\mathcal{P}_r : \mathfrak{D} \rightarrow \mathbb{R}_{\geq 0}^{|\mathcal{A}|x|\mathcal{N}|}$. The policy sets for every strategy profile the uplink access provisioning coefficients $(u_{i,a})_{i\in\mathcal{N},a\in\mathcal{A}}$, 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 [15], [11], [16], [17], 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

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

D. Operator Policies and 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 rate 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. We consider two variants of the game, which differ in the set of operator policies. In the first game the set of feasible decisions for the operator is $\mathfrak{A}_c = \{(\mathbf{u},\mathbf{p})|\mathbf{u} \in \mathbb{R}_{\geq 0}^{|\mathcal{A}|x|\mathcal{N}|}, \mathbf{p} \in \mathbb{R}_{\geq 0}^{|\mathcal{C}|x|\mathcal{N}|}\}$; we refer to this as the *cost minimizing* (CM) operator. In the second game the set of feasible decisions for the operator is $\mathfrak{A}_t = \{(\mathbf{u},\mathbf{p})|u_{i,a} = 1, p_{i,c} = 1, \forall i \in \mathcal{N}, a \in \mathcal{A}, c \in \mathcal{C}\}$; we refer to this as the *time fair* (TF) operator.

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 $(\mathbf{u}^*,\mathbf{p}^*) \in \mathfrak{A}_o$, $o \in \{c,t\}$ to $\mathbf{d}$ through solving

$$\min_{(\mathbf{u},\mathbf{p})\in\mathfrak{A}_o} C(\mathbf{d},\mathbf{u},\mathbf{p}). \quad (7)$$

We denote by $(\mathcal{P}_r^{c,*},\mathcal{P}_c^{c,*})$ the optimal policy of the CM operator, i.e., the collection of best responses of the CM operator for every $\mathbf{d} \in \mathfrak{D}$, and we denote by $(\mathcal{P}_r^{t,*},\mathcal{P}_c^{t,*})$ the optimal policy of the TF operator, i.e., the collection of best responses of the TF operator 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

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

where we use d_{-i} to denote the strategies of all WDs except WD i. We refer to the game played between the WDs and the CM operator as the *cost minimizing computation offloading game* (CM-COG) and to the game played between the WDs and the TF operator as the *time fair computation offloading game* (TF-COG).

In this paper we address three fundamental questions for these games. First, we address whether 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 $(\mathcal{P}_r^*,\mathcal{P}_c^*)$ 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 game $\Gamma \in \{\text{CM-COG}, \text{TF-COG}\}$ 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 game $\Gamma \in \{\text{CM-COG}, \text{TF-COG}\}$ 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. Before we answer these questions we recall the following definition from game theory.

Definition 2. (**Pure NE and Best reply (BR)**) 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 $d = (d_i, d_{-i})$, a better reply of WD i is a strategy d_i' such that $C_i(d_i', d_{-i}) < C_i(d_i, d_{-i})$, and a best reply of WD i is a better reply d_i^* such that $C_i(d_i^*, d_{-i}) \leq C_i(d_i, d_{-i}), \forall d_i \in \mathfrak{D}_i$.

III. Equilibria Under the Cost Minimizing Operator

We start the analysis by considering problem (7) solved by the CM operator, i.e.,

$$\min_{(\mathbf{u},\mathbf{p})\in\mathfrak{A}_c} C(\mathbf{d}, \mathbf{u}, \mathbf{p}), \tag{9}$$

followed by problem (8) solved by the WDs.

A. Optimal Resource Allocation Policy of the CM Operator

Recall that an optimal resource allocation policy is essentially a collection of best responses $(\mathbf{u}^*, \mathbf{p}^*) \in \mathfrak{A}_c$ of the CM 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 CM 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 $(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ of the CM operator assigns to $\boldsymbol{d}$ the uplink access provisioning and computing power provisioning coefficients*

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

and

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

Proof. The proof is given in Appendix A. □

It is important to note that following the optimal resource allocation policy, the CM 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 CM operator the game can be transformed into a weighted congestion game.

Theorem 2. *Consider that the CM operator uses the optimal policy $(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$, i.e., $\boldsymbol{u}^*$ and $\boldsymbol{p}^*$ are the collections of the optimal provisioning coefficients given by (10) and (11), 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*

$$\bar{C}_i(\boldsymbol{d}) = \sum_{(a,c)\in\mathcal{A}_i\times\mathcal{C}} I_{d_i,(a,c)}\Big(w_{i,a}w_a(\boldsymbol{d}) + w_{i,c}w_c(\boldsymbol{d})\Big) + I_{d_i,i}C_i^l, \tag{12}$$

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

Proof. Let us first substitute (10) and (11) into (3) in order to obtain the offloading cost of WD i through AP a to EC c under the optimal resource allocation policy $(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$,

$$\bar{C}^c{}_{i,a}(\mathbf{d}) = \sqrt{\frac{D_i}{R_{i,a}}}\sum_{j\in O_a(\mathbf{d})}\sqrt{\frac{D_j}{R_{j,a}}} + \sqrt{\frac{L_i}{F^c}}\sum_{j\in O_c(\mathbf{d})}\sqrt{\frac{L_j}{F^c}}. \tag{13}$$

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 (13) 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 as $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*}) =< \mathcal{N}, (\mathfrak{D}_i)_{i\in\mathcal{N}}, (\bar{C}_i)_{i\in\mathcal{N}} >$, in which the players are WDs with the objective to minimize their costs given by (12). Observe that the game $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ is the CM-COG expressed in strategic form and thus if $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ has a NE then the CM-COG has an SPE. Hence, in what follows we focus on the existence and computability of pure NE for $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$.

Before we formulate our next result let us recall the definition of an exact potential function from [18].

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

$$\bar{C}i(d_i', d_{-i}) - \bar{C}_i(d_i, d_{-i}) = \Phi(d_i', d_{-i}) - \Phi(d_i, d_{-i}). \tag{14}$$

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

$$w_a^{\leq i}(\mathbf{d}) = \sum_{\{j\in O_a(\mathbf{d})|j\leq i\}} w_{j,a}, \qquad w_a^{>i}(\mathbf{d}) = \sum_{\{j\in O_a(\mathbf{d})|j>i\}} w_{j,a},$$

and

$$w_c^{\leq i}(\mathbf{d}) = \sum_{\{j\in O_c(\mathbf{d})|j\leq i\}} w_{j,c}, \qquad w_c^{>i}(\mathbf{d}) = \sum_{\{j\in O_c(\mathbf{d})|j>i\}} w_{j,c}.$$

Theorem 3. *The game $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ has the exact potential function*

$$\Phi(\boldsymbol{d}) = \sum_{i\in\mathcal{N}}\left(\sum_{a\in\mathcal{A}}\Phi_{i,a}(\boldsymbol{d}) + \sum_{c\in\mathcal{C}}\Phi_{i,c}(\boldsymbol{d}) + \Phi_{i,i}(\boldsymbol{d})\right), \tag{15}$$

where $\Phi_{i,a}(\boldsymbol{d}) = I_{d_i,(a,\cdot)}w_{i,a}w_a^{\leq i}(\boldsymbol{d})$, $\Phi_{i,c}(\boldsymbol{d}) = I_{d_i,(\cdot,c)}w_{i,c}w_c^{\leq i}(\boldsymbol{d})$, 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

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

We hence obtain the equality

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

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 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

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

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 $\Phi_{i,a}(\mathbf{d}) - \Phi_{i,a}(\mathbf{d}') = w_{i,a}w_{k,a}$. Similarly, 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

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

We hence obtain the equality

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

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 $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ has a pure NE [18]. We can thus formulate the following result.

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

AU($\mathbf{d}$)

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

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

There are a variety of algorithms that are known to converge to an equilibrium for exact potential games, such as fictitious play [18], joint strategy fictitious play [19], and the best and better reply dynamics [18]. Nonetheless, they have exponential worst case complexity in general [20], [21]. Thus, the second fundamental question we address in this paper is whether a NE of $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ (and thus an SPE of the CM-COG) 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 assume that a best reply of WD $i \in \mathcal{N}'$ is offloading to an EC c, i.e., for any EC $c' \in \mathcal{C} \setminus \{c\}$ the following holds

$$\Big(w_c(\mathbf{d}) + w_{i,c}\Big)w_{i,c} < \Big(w_{c'}(\mathbf{d}) + w_{i,c'}\Big)w_{i,c'}. \quad (16)$$

Let us next assume that upon WD i performs its best reply, a WD $j \in O_c(\mathbf{d})$ can decrease its offloading cost by changing its strategy from $(\cdot, c)$ to $(\cdot, c')$, i.e., that the following holds

$$\Big(w_c(\mathbf{d}) + w_{i,c}\Big)w_{j,c} > \Big(w_{c'}(\mathbf{d}) + w_{j,c'}\Big)w_{j,c'}. \quad (17)$$

In order to have (16) and (17) satisfied $\sqrt{L_i} > \sqrt{L_j}$ must hold, 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.

IV. Equilibria Under the Time Fair Operator

We have so far shown that the the game played by the WDs under the resource allocation policy $(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ of the CM operator can be transformed into a weighted congestion game, and we have proven that the CM-COG has an SPE. In what follows we show that under the resource allocation policy $(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ of the TF operator the game played by the WDs can be transformed into a player-specific congestion game.

Proposition 2. *Consider that the TF operator uses the time fair resource allocation policy $(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$. Then, the strategic interaction of the WDs can be modeled as a player-specific congestion game, in which the cost of WD i is given by*

$$\widetilde{C}_i(\boldsymbol{d}) = \sum_{(a,c)\in\mathcal{A}\times\mathcal{C}} I_{d_i,(a,c)}\Big(\frac{D_i}{R_{i,a}} n_a(\boldsymbol{d}) + \frac{L_i}{F^c} n_c(\boldsymbol{d})\Big) + I_{d_i,i} C_i^l \quad (18)$$

Proof. Given the equal sharing of resources, it follows from (1) and (2) that the offloading cost (3) of WD i through AP a to EC c can be expressed as

$$\widetilde{C}_{i,a}(\mathbf{d}) = \frac{D_i}{R_{i,a}} n_a(\mathbf{d}) + \frac{L_i}{F^c} n_c(\mathbf{d}). \quad (19)$$

Observe that the offloading cost (19) depends on the total number $n_a(\mathbf{d})$ of WDs sharing the AP a, the total number $n_c(\mathbf{d})$ of WDs sharing the EC c, and on the characteristics of WD i's task. Thus, the interaction between the WDs can be modeled as a *player-specific congestion game*. This proves the result. □

A. Computing Equilibrium Offloading Decisions

We refer to the resulting strategic game as $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*}) = < \mathcal{N}, (\mathfrak{D}_i)_{i\in\mathcal{N}}, (\widetilde{C}_i)_{i\in\mathcal{N}} >$, in which the players are WDs with the objective to minimize their cost given by (18). Observe that the game $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ is the TF-COG expressed in strategic form and thus if $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ has a NE then the TF-COG has an SPE. Hence, in what follows we focus on the existence and computability of pure NE for $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$. In what follows we prove our result concerning the existence of a pure strategy NE under the TF operator. Our proof is based on the *JoinAndPlayAsynchronousUpdates* (JPAU) algorithm, which we propose for computing a NE of the game $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$. The pseudo code of the JPAU algorithm is shown in Fig. 3. The algorithm adds WDs one at a time, and lets them play their best replies given the other WDs' strategies, and thus the following result is based on an induction in the number N of WDs.

Theorem 4. *The game $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ has a pure strategy NE.*

Proof. The proof is given in Appendix B. □

Even though the proof of Theorem 4 is fairly involved, the JPAU algorithm itself is computationally efficient, as we show next.

Proposition 3. *When a new WD enters the game $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ in a NE $\boldsymbol{d}^*(n-1)$, a new NE can be computed in $O((A-2)|\mathcal{N}_{n-1}|^2 - (A-3)|\mathcal{N}_{n-1}|)$ time.*

Proof. The proof is given in Appendix C. □

$\mathbf{d}^* = JPAU(\mathcal{N}, \mathcal{A}, \mathcal{C})$

/*First WD enters the game*/
Let $\mathbf{d} \leftarrow \emptyset, i \leftarrow 1$
$d_i^*(1) = \arg\min_{d_i\in\mathfrak{D}_i} C_i(d_i, d_{-i})$
$\mathbf{d}^*(1) = d_i^*(1)$
for $n = 2 : N$ **do**
  /*Corresponds to induction phase*/
  Let $i \leftarrow n$
  $d_i^*(n) = \arg\min_{d_i\in\mathfrak{D}_i} C_i(d_i, d^*_{-i}(n-1))$
  $\mathbf{d}(n) = (d_i^*(n), \mathbf{d}^*(n-1))$
  if $d_i^*(n) = (a, c)$
    /*Corresponds to update phase*/
    if $\exists j \in O_{(a,c)}(\boldsymbol{d}(n))$ *for which a BR is local computing*
      /*Corresponds to case (i)*/
      $\mathbf{d}'(n) = (j, d_{-j}(n))$
    end
    else if $\exists j \in O_{(a,c')}(\boldsymbol{d}(n)), c' \neq c$ *for which a BR is local computing*
      /*Corresponds to case (ii)*/
      $\mathbf{d}'(n) = (j, d_{-j}(n))$
      $k \leftarrow O_c(\mathbf{d}'(n))$
      $\mathbf{d}'(n) = ((\cdot, c'), d'_{-k}(n))$
    else if $\exists j \in O_a(\boldsymbol{d}(n)), a' \neq a$ *for which a BR is changing to AP* a'
      /*Corresponds to case (iii)*/
      $\mathbf{d}'(n) = ((a', \cdot), d_{-j}(n))$
      while $\exists j \in O(\boldsymbol{d}'(n))$ *that can decrease its offloading cost*
        $d_j^*(n) = \arg\min_{d'_j\in\mathfrak{D}_j} C_j(d'_j, d'_{-j}(n))$
        $\mathbf{d}'(n) = (d_j^*(n), d'_{-j}(n))$
      end
    else
      $\mathbf{d}'(n) = \mathbf{d}(n)$
    end
    $a'' \leftarrow a$ for which $n_a(\mathbf{d}'(n)) = n_a(\mathbf{d}^*(n-1)) + 1$
    $c \leftarrow c'$ for which $n_{c'}(\mathbf{d}'(n)) = n_{c'}(\mathbf{d}^*(n-1)) + 1$
    if $\exists j \in O_{(a',c)}(\boldsymbol{d}'(n)), a' \neq a''$ *for which a BR is local computing*
      /*Corresponds to case (iv)*/
      $\mathbf{d}'(n) = (j, d'_{-j}(n))$
      if $\exists j \in O_{a''}(\boldsymbol{d}'(n))$ *for which a BR is changing to AP* a'
        $k \leftarrow O_{a''}(\mathbf{d}'(n))$
        $\mathbf{d}'(n) = ((a', \cdot), d'_{-k}(n))$
      else
        while $\exists j \in O(\boldsymbol{d}'(n))$ *that can decrease its offloading cost*
          $d_j^*(n) = \arg\min_{d'_j\in\mathfrak{D}_j} C_j(d'_j, d'_{-j}(n))$
          $\mathbf{d}'(n) = (d_j^*(n), d'_{-j}(n))$
        end
        if $\exists k \in \mathcal{N}_n \setminus O(\boldsymbol{d}'(n))$ *for which a BR is* (a', c)
          $\mathbf{d}'(n) = ((a', c), d'_{-k}(n))$
        end
        if $\exists j \in O_{(a',c)}(\boldsymbol{d}'(n)), a' \neq a$ *for which a BR is local computing*
          go to 35
        end
      end
    end
    $\mathbf{d}^*(n) = \mathbf{d}'(n)$
end
return $\boldsymbol{d}^*(N)$

Fig. 3. Pseudo code of the $JPAU$ algorithm.

By adding WDs one at a time, it follows that the JPAU algorithm computes a NE of the game $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ in polynomial time.

Corollary 2. *The JPAU algorithm terminates in a NE of the game $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ in $O((A-2)N^3 - (A-3)N^2)$ time.*

Finally, since $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ is the TF-COG expressed in strategic form, we can formulate the following result.

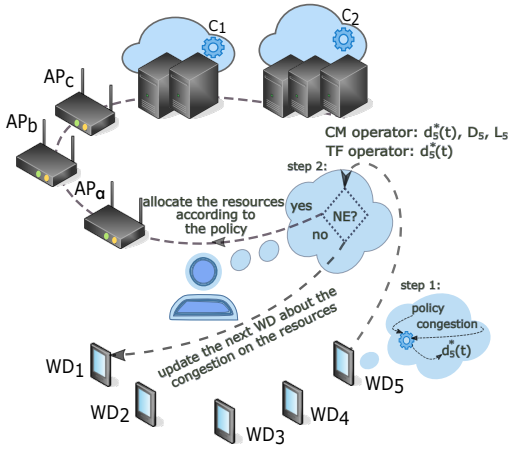

Fig. 4. Example of the information exchange between the operator and WD_1 and WD_5.

Corollary 3. *The game $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ has a pure strategy NE $\boldsymbol{d}^*$. Hence, an SPE $(\boldsymbol{d}^*, \mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ for the TF-COG exists.*

B. 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 and JPAU algorithms in the case of the CM-COG and TF-COG, respectively. 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, the operator allocates wireless and computing resources according to the adopted resource allocation policy. By Corollary 1 and Corollary 3 the resulting state is an SPE of the CM-COG and the TF-COG, respectively. Fig. 4 illustrates the information exchange between the operator and the WDs for the edge computing system shown in Fig. 1.

Observe that the WDs need to report only their offloading decisions in the case of the time fair operator and apart from the offloading decisions they need to reveal the characteristics of their tasks (i.e., the size D_i of the input data and the expected complexity L_i) in the case of the cost minimizing operator. Therefore, the implementation of the SPE of the TF-COG requires less information about the WDs' tasks than the implementation of the SPE of the CM-COG, and thus the time fair resource allocation policy may be a better choice in systems in which privacy and confidentiality are of major concern.

V. Price of Anarchy

We have so far analyzed the interaction between the WDs under the cost minimizing and the time fair resource allocation policies of the operator and we proposed the ILC and the JPAU algorithms for computing an equilibrium of offloading decisions of the WDs under these two policies, respectively. Furthermore, we showed that the computational complexity of the ILC algorithm (which is exponential in the worst case) can be reduced by letting WDs start to offload in non-increasing order of their task complexities, and we proved that the worst case complexity of the JPAU algorithm is polynomial in N. In this section we quantify the worst case ratio between the system performance in an SPE and the optimal performance using the price of anarchy (PoA). We do so by providing an upper bound on the PoA of the CM-COG (denoted by $PoA_{\text{CM-COG}}$) and the TF-COG (denoted by $PoA_{\text{TF-COG}}$), respectively. Let us recall that the games $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ and $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ are strategic representations of the CM-COG and the TF-COG games, respectively. Therefore, we have $PoA_{\text{CM-COG}} = PoA(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ and $PoA_{\text{TF-COG}} = PoA(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$.

We start with the definition of the $PoA(\mathcal{P}_r, \mathcal{P}_c)$ 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

$$PoA(\mathcal{P}_r, \mathcal{P}_c) = \frac{\max_{\mathbf{d}^* \in \mathfrak{D}^*} \sum_{i \in \mathcal{N}} C_i(\mathbf{d}^*, \mathcal{P}_r, \mathcal{P}_c)}{\min_{\mathbf{d} \in \mathfrak{D}} \sum_{i \in \mathcal{N}} C_i(\mathbf{d}, \mathcal{P}_r, \mathcal{P}_c)}, \quad (20)$$

where $\mathfrak{D}^*$ is the set of equilibria of offloading decisions under $(\mathcal{P}_r, \mathcal{P}_c)$.

A. Price of Anarchy of the CM-COG

In order to provide an upper bound on $PoA_{\text{CM-COG}}$, we provide an upper bound on $PoA(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ of the strategic game $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$.

Theorem 5. $PoA_{\text{CM-COG}} = PoA(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*}) \leq \frac{3+\sqrt{5}}{2}$.

Proof. Our proof is inspired by Theorem 3.1 in [22], which provides a PoA bound for normalized weighted congestion games. Our proof extends the PoA bound to the game $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$, 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 $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$, 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 $C(\mathbf{d}, \mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ for $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ in a strategy profile $\mathbf{d}$ as

$$C(\mathbf{d}, \mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*}) = \sum_{r \in \mathcal{R}} \sum_{i \in O_r(\mathbf{d})} w_r(\mathbf{d}) w_{i,r} = \sum_{r \in \mathcal{R}} w_r^2(\mathbf{d}). \quad (21)$$

Furthermore, from the definition of a NE we obtain

$$\sum_{r \in \mathcal{R}_{d_i^*}} w_r(\mathbf{d}^*) w_{i,r} \leq \sum_{r \in \mathcal{R}_{d_i^*} \cap \mathcal{R}_{\hat{d}_i}} w_r(\mathbf{d}^*) w_{i,r} + \quad (22)$$

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

First, by summing inequality (22) over all WDs i we obtain

$$\sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_{d_i^*}} w_r(\mathbf{d}^*) w_{i,r} \leq \sum_{i \in \mathcal{N}} \sum_{r \in \mathcal{R}_{\hat{d}_i}} \big(w_r(\mathbf{d}^*) + w_{i,r}\big) w_{i,r}. \quad (23)$$

Second, by reordering the summations, (23) can be rewritten as

$$\sum_{r\in\mathcal{R}}\sum_{i\in O_r(\mathbf{d}^*)} w_r(\mathbf{d}^*)w_{i,r} \leq \sum_{r\in\mathcal{R}}\sum_{i\in O_r(\hat{\mathbf{d}})} \big(w_r(\mathbf{d}^*)w_{i,r}+w_{i,r}^2\big). \quad (24)$$

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

$$\sum_{r\in\mathcal{R}} w_r^2(\mathbf{d}^*) \leq \sum_{r\in\mathcal{R}} w_r(\mathbf{d}^*)w_r(\hat{\mathbf{d}}) + \sum_{r\in\mathcal{R}} w_r^2(\hat{\mathbf{d}}). \quad (25)$$

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

$$\sum_{r\in\mathcal{R}} w_r^2(\mathbf{d}^*) \leq \sqrt{\sum_{r\in\mathcal{R}} w_r^2(\mathbf{d}^*) \sum_{r\in\mathcal{R}} w_r^2(\hat{\mathbf{d}})} + \sum_{r\in\mathcal{R}} w_r^2(\hat{\mathbf{d}}). \quad (26)$$

If we divide the right and the left side of inequality (26) by $\sum_{r\in\mathcal{R}} w_r^2(\hat{\mathbf{d}}) > 0$ we can rewrite it using (21) as

$$\frac{C(\mathbf{d}^*, \mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})}{C(\hat{\mathbf{d}}, \mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})} \leq \sqrt{\frac{C(\mathbf{d}^*, \mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})}{C(\hat{\mathbf{d}}, \mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})}} + 1. \quad (27)$$

Since (27) holds for any NE of the game $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$, it holds for the worst case NE too, and thus we have

$$PoA(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*}) \leq \sqrt{PoA(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})} + 1. \quad (28)$$

By solving (28) we obtain that $PoA_{\text{CM-COG}} = PoA(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*}) \leq \frac{3+\sqrt{5}}{2}$, which proves the theorem. □

B. Price of Anarchy of the TF-COG

Next, using a similar approach to the one presented in the proof of Theorem 5, in what follows we provide an upper bound on $PoA_{\text{TF-COG}}$ by providing an upper bound on the $PoA(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ of the strategic game $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$.

Theorem 6. *$PoA_{TF\text{-}COG} = PoA(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*}) \leq N + 1.$*

Proof. We start with the definition of the weights in $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ for all resources $\mathcal{R}{=}\mathcal{N}\cup\mathcal{A}\cup\mathcal{C}$ available in the system

$$\widetilde{w}_{i,i} \triangleq \frac{L_i}{F_i^l}, \widetilde{w}_{i,c} \triangleq \frac{L_i}{F^c}, \widetilde{w}_{i,a} \triangleq \frac{D_i}{R_{i,a}}.$$

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

$$\widetilde{C}(\mathbf{d}, \mathcal{P}_r^{eq}, \mathcal{P}_c^{eq}) = \sum_{r\in\mathcal{R}}\sum_{i\in O_r(\mathbf{d})} n_r(\mathbf{d})\widetilde{w}_{i,r} = \sum_{r\in\mathcal{R}} n_r(\mathbf{d})\widetilde{w}_r(\mathbf{d}), \quad (29)$$

where $\widetilde{w}_r(\mathbf{d}) \triangleq \sum_{i\in O_r(\mathbf{d})} \widetilde{w}_{i,r}$.

Furthermore, let us use $\mathbf{d}^*$ and $\hat{\mathbf{d}}$ to denote a NE and an optimal solution of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$, respectively.

Now, from the definition of a NE we obtain

$$\sum_{r\in\mathcal{R}_{d_i^*}} n_r(\mathbf{d}^*)\widetilde{w}_{i,r} \leq \sum_{r\in\mathcal{R}_{d_i^*}\cap\mathcal{R}_{\hat{d}_i}} n_r(\mathbf{d}^*)\widetilde{w}_{i,r} + \sum_{r\in\mathcal{R}_{d_i^*}\setminus\mathcal{R}_{\hat{d}_i}} \big(n_r(\mathbf{d}^*)+1\big)\widetilde{w}_{i,r} \leq \sum_{r\in\mathcal{R}_{\hat{d}_i}} \big(n_r(\mathbf{d}^*)+1\big)\widetilde{w}_{i,r}. \quad (30)$$

First, by summing inequality (30) over all WDs i we obtain

$$\sum_{i\in\mathcal{N}}\sum_{r\in\mathcal{R}_{d_i^*}} n_r(\mathbf{d}^*)\widetilde{w}_{i,r} \leq \sum_{i\in\mathcal{N}}\sum_{r\in\mathcal{R}_{\hat{d}_i}} \big(n_r(\mathbf{d}^*)+1\big)\widetilde{w}_{i,r}. \quad (31)$$

Second, by reordering the summations (31) can be rewritten as

$$\sum_{r\in\mathcal{R}}\sum_{i\in O_r(\mathbf{d}^*)} n_r(\mathbf{d}^*)\widetilde{w}_{i,r} \leq \sum_{r\in\mathcal{R}}\sum_{i\in O_r(\hat{\mathbf{d}})} \big(n_r(\mathbf{d}^*)\widetilde{w}_{i,r} + \widetilde{w}_{i,r}\big). \quad (32)$$

Using the definition of the total weight $\widetilde{w}_r(\mathbf{d}) \triangleq \sum_{i\in O_r(\mathbf{d})} \widetilde{w}_{i,r}$ associated with resource r we can rewrite (32) as

$$\sum_{r\in\mathcal{R}} n_r(\mathbf{d}^*)\widetilde{w}_r(\mathbf{d}^*) \leq \sum_{r\in\mathcal{R}} n_r(\mathbf{d}^*)\widetilde{w}_r(\hat{\mathbf{d}}) + \sum_{r\in\mathcal{R}} \widetilde{w}_r(\hat{\mathbf{d}}). \quad (33)$$

Next, observe that $n_r(\mathbf{d}) \leq N$ must hold for any feasible strategy profile $\mathbf{d}$ and for every resource $r \in \mathcal{R}$, and that $|O_r(\mathbf{d})| \geq 1$ implies $n_r(\mathbf{d}) \geq 1$. Therefore, we have that $\sum_{r\in\mathcal{R}} n_r(\mathbf{d}^*)\widetilde{w}_r(\hat{\mathbf{d}}) \leq N\sum_{r\in\mathcal{R}} n_r(\hat{\mathbf{d}})\widetilde{w}_r(\hat{\mathbf{d}})$ and $\sum_{r\in\mathcal{R}} \widetilde{w}_r(\hat{\mathbf{d}}) \leq \sum_{r\in\mathcal{R}} n_r(\hat{\mathbf{d}})\widetilde{w}_r(\hat{\mathbf{d}})$. By using these observations in (33) we obtain the following inequality

$$\sum_{r\in\mathcal{R}} n_r(\mathbf{d}^*)\widetilde{w}_r(\mathbf{d}^*) \leq (N+1)\sum_{r\in\mathcal{R}} n_r(\hat{\mathbf{d}})\widetilde{w}_r(\hat{\mathbf{d}}). \quad (34)$$

Finally, since $\sum_{r\in\mathcal{R}} n_r(\hat{\mathbf{d}})\widetilde{w}_r(\hat{\mathbf{d}}) > 0$ must hold, we can divide the right and the left side of inequality (34) by $\sum_{r\in\mathcal{R}} n_r(\hat{\mathbf{d}})\widetilde{w}_r(\hat{\mathbf{d}})$ to obtain

$$\frac{\sum_{r\in\mathcal{R}} n_r(\mathbf{d}^*)\widetilde{w}_r(\mathbf{d}^*)}{\sum_{r\in\mathcal{R}} n_r(\hat{\mathbf{d}})\widetilde{w}_r(\hat{\mathbf{d}})} \leq N+1. \quad (35)$$

Since (35) holds for any NE of the game $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$, it also holds for the worst case NE, and thus using (29) we obtain

$$PoA_{\text{TF-COG}} = PoA(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*}) \leq N+1, \quad (36)$$

which proves the theorem. □

Observe that the $PoA(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ and $PoA(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ are in fact bounds on the approximation ratio of the ILC and JPAU algorithms used for computing a NE of the games $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ and $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$, respectively. Therefore, the ILC algorithm outperforms the JPAU algorithm in terms of the worst case system performance and the JPAU algorithm outperforms the ILC algorithm in terms of the worst case complexity. Consequently, the cost minimizing resource allocation policy might be a better choice than the time fair resource allocation policy in systems in which a guarantee on the worst case system performance is more important than a guarantee on the worst case computational efficiency, and vice versa.

VI. 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 [23]. For simplicity we assign a bandwidth of $B_{i,a}{=}5MHz$ 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 $[0.05, 0.18]W$

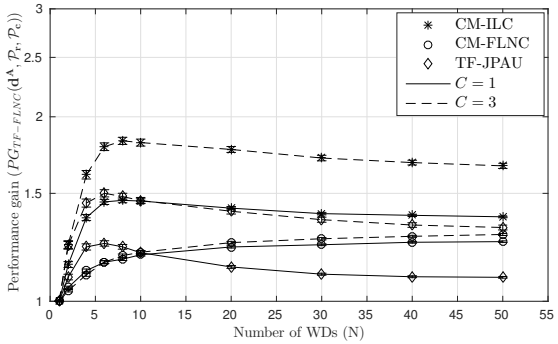

Fig. 5. *Performance gain* vs. the number of WDs N for $A = 5$ APs. *Homogeneous ECs*, $F^{c,tot} = 192GHz$.

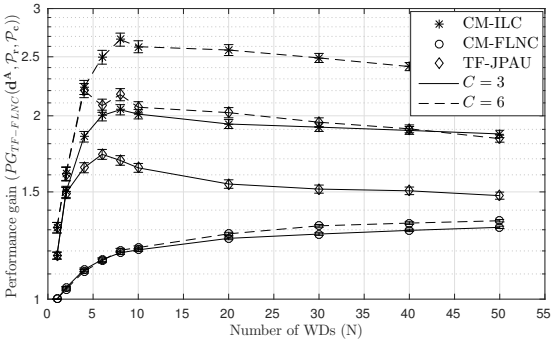

Fig. 6. *Performance gain* vs. the number of WDs N for $A = 5$ APs. *Heterogeneous ECs*, $F^{c,tot} = 192GHz$.

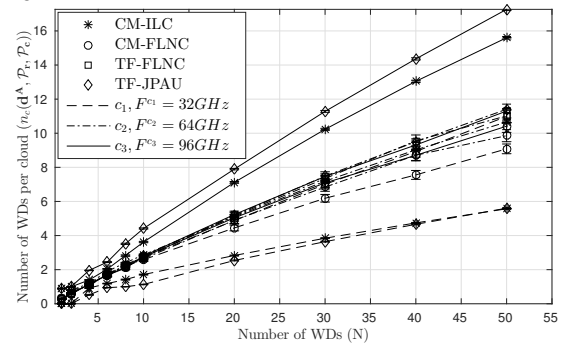

Fig. 7. Congestion per EC vs. the number of WDs N for $A = 5$ APs, $C = 3$ ECs. *Heterogeneous ECs*, $F^{c,tot} = 192GHz$.

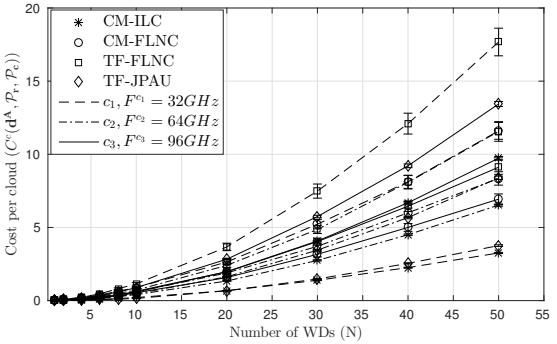

Fig. 8. Cost per EC vs. the number of WDs N for $A = 5$ APs, $C = 3$ ECs. *Heterogeneous ECs*, $F^{c,tot} = 192GHz$.

according to [24]. 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 $(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ as the CM policy. Under the CM policy the WDs use the ILC algorithm for computing a NE of the game $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$, as shown in Section III. As a baseline for comparison, we consider the TF policy $(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$, under which the WDs use the JPAU algorithm for computing a NE of the game $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$, as shown in Section IV.

As a baseline for the ILC and JPAU algorithms proposed for computing an equilibrium of offloading decisions, 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* $PG_{\text{TF-FLNC}}(\mathbf{d}^A, \mathcal{P}_r, \mathcal{P}_c)$ (w.r.t. the TF-FLNC) for a strategy profile $\mathbf{d}^A$ computed by algorithm $A \in \{\text{ILC}, \text{JPAU}, \text{FLNC}\}$ under a resource allocation policy $(\mathcal{P}_r, \mathcal{P}_c) \in \{(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*}), (\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})\}$ as

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

Fig. 5 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 CM policy and WDs offload according to an equilibrium computed by the ILC algorithm. Interestingly, even CM-FLNC outperforms TF-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. 6 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. 5, the results in Fig. 6 show a decreasing marginal gain in N and confirm that the largest *performance gain* is achieved by the CM-ILC. Nonetheless, a comparison of Fig. 5 and Fig. 6 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. On 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, CM-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., $F^{c,tot} = 192GHz$ divided among three ECs such that $F^{c_1} = 32GHz$ and $F^{c_i} = F^{c_{i-1}} + 32GHz$, for $i > 1$.

Fig. 7 and Fig. 8 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 CM and TF 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. 7 that the ratio of 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. 7 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. 8, 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 CM and TF policies, respectively) the congestion and the cost per EC are proportional to the computing capability of the EC as shown in Fig. 7 and Fig. 8, 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 TF 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 $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ 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 [18]. 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 $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ using the JPAU algorithm.

Fig. 9 shows the number of iterations needed to compute an equilibrium of $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ and an equilibrium of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$, as a function of N for the same set of parameters as in Fig. 5. We observe that the number of iterations scales approximately linearly with N in all cases and that computing an equilibrium of $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$ using the ILC algorithm is more efficient than computing an equilibrium of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ using the JPAU algorithm; the difference is up to 50%.

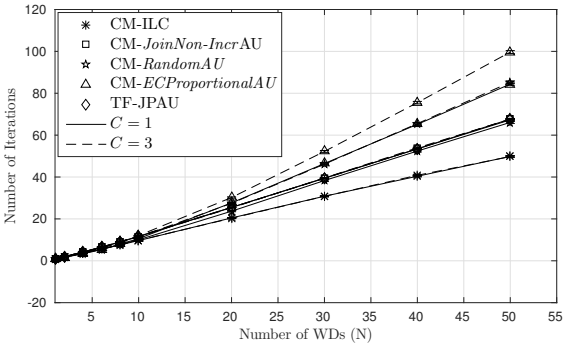

Fig. 9. Number of iterations vs. the number of WDs N for $A = 5$. *Homogeneous ECs*, $F^{c,tot} = 192GHz$.

We also observe that the choice of the initial strategy profile affects the complexity of computing an equilibrium of the game $\Gamma(\mathcal{P}_r^{c,*}, \mathcal{P}_c^{c,*})$, 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 CM-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.

VII. Related Work

There is a large body of recent works on computation offloading for mobile cloud computing [25], [26], [11], [27], [28], [29], [30], [31], [32], [33], [15], [34], [35], [36]. 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], [28], [29], [30]. [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. [28] considered the problem of joint optimization of network selection and service placement under random mobility of users and proposed an iterative algorithm that minimizes the average system delay. [29] proposed an online algorithm for distributing the workload across multiple edge clouds, which are managed by an operator that apart from the workload distribution decides about the activation status of the edge clouds, and acts as the auctioneer that solicits bids from multiple service providers. [30] considered a mobile cloud computing system in which a centralized entity located in the cloud implements an online algorithm for scheduling the transmissions between the mobile devices and the cloud so as to minimize the energy consumption of mobile devices. Unlike these works, we propose a novel approach to address the computation offloading problem by considering the interaction between an operator that manages the allocation of wireless and computing resources and devices that make their offloading decisions in a decentralized manner.

Closer related to ours are recent works that propose decentralized algorithms based on a game theoretic treatment of the computation offloading problem [31], [32], [33], [15], [34], [35], [36]. 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. Compared to [32], we consider both the cost minimizing and the time fair resource allocation policies and besides the analysis of a game in the case of the cost minimizing operator, we also prove the existence of Stackelberg equilibria in the case of the time fair operator, we establish an upper bound on the price of anarchy of the resulting Stackelberg game and we propose a polynomial complexity algorithm for computing an equilibrium of the game. [33] 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. [15], [34], [35] 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. [15] 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. [34] extended the equilibrium existence results of [15] to a dynamic environment, where devices may be active or inactive. [35] 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. [36] 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 edge clouds, which are managed by an operator that can implement one of two resource allocation policies and given the resource allocation policy of the operator we consider that devices can autonomously decide whether or not to offload the computations, and if so, to which of multiple edge clouds and through which of multiple wireless links. To the best of our knowledge, ours is the first work on computation offloading for mobile cloud computing that closes the gap between the works that propose centralized solutions and the works that propose decentralized solutions.

Closest to our work in the literature on game theory is [37], which considers the effectiveness of Stackelberg strategies for atomic congestion games. Authors in [37] 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 [38]. Thus, our work provides a novel game theoretic perspective on congestion games.

VIII. 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. We consider the cost minimizing and the time fair allocation policies of the operator and we use a Stackelberg game to model the interaction between the operator and devices. We expressed the cost minimizing resource allocation policy in closed form and proved the existence of Stackelberg equilibria for both policies. Using game theoretical tools, we developed efficient decentralized approximation algorithms for computing offloading decisions of devices under both policies of the operator. Our numerical results show that the proposed algorithms are computationally efficient and that the system performance can be significantly improved through optimally allocating wireless and computing resources in a system, while allowing the devices to make their offloading decisions autonomously.

Appendix

A. Proof of Theorem 1

By inspecting the leading minors of the Hessian matrix of (3) it is easy to show that (9) is neither convex nor concave in $\mathbf{u}$ and $\mathbf{p}$ already for the case when there are only two WDs sharing a resource. Furthermore, it is easy to see from expressions (1) and (2) that the optimal solution of

(9) 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

$$\min_{(\mathbf{u},\mathbf{p})\in\mathfrak{A}_c} C(\mathbf{d},\mathbf{u},\mathbf{p}) \tag{37}$$

$$\text{s.t.} \quad \textstyle\sum_{j\in O_a(\mathbf{d})} u_{j,a} = 1, \quad \forall a \in \mathcal{A} \tag{38}$$

$$\textstyle\sum_{j\in O_c(\mathbf{d})} p_{j,c} = 1. \quad \forall c \in \mathcal{C} \tag{39}$$

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

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

Unlike problem (9), problem (37)-(39) is a convex minimization problem, and thus its optimal solution must satisfy the Karush–Kuhn–Tucker (KKT) conditions. To define the Lagrangian dual of (37)-(39), we denote by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ the dual variables associated with constraints (38) and (39) 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 (37)-(39) as

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

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

$$u_{i,a} = \sqrt{D_i/\alpha_a R_{i,a}}, \forall a\in\mathcal{A}, \forall i\in O_a(\mathbf{d}), \tag{40}$$

and

$$p_{i,c} = \sqrt{L_i/\beta_c F^c}, \forall c\in\mathcal{C}, \forall i\in O_c(\mathbf{d}). \tag{41}$$

By substituting (40) and (41) in the primal feasibility equations we can obtain the expressions for α_a and β_c, and we can rewrite equations (40) and (41) 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.

B. Proof of Theorem 4

The JPAU algorithm starts from an empty system, adds WDs into the game one at a time in the induction phase and lets WDs to update their best replies one at a time in the update phase. We denote by $\mathcal{N}_n$ the set of WDs that participate in the game upon an induction step $1 \leq n \leq N$ and we use $\mathbf{d}(n) = (d_i(n), d_{-i}(n))$ to denote a strategy profile played by WD i and the other WDs $j \in \mathcal{N}_n \setminus \{i\}$. Observe that for $N = 1$, there is only one WD i in the game playing its best reply $d_i^*(1)$, and thus $\mathbf{d}^*(1) = d_i^*(1)$ is a NE of the game.

For $N > 1$ let us assume that in induction step $n-1$ WDs play a NE $\mathbf{d}^*(n-1)$, and let us consider a WD $i \in \mathcal{N}\setminus\mathcal{N}_n$ added into the game by the JPAU algorithm in induction step n. If a best reply $d_i^*(n)$ of WD i is to perform the computation locally, then $\mathbf{d}^*(n) = (d_i^*(n), \mathbf{d}^*(n-1))$ is a NE of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ since the congestion on the APs and the ECs remained unchanged. Otherwise, let us assume that a best reply of WD i is offloading through AP a to EC c, i.e., $d_i^*(n) = (a, c)$. Observe that $\mathbf{d}(n) = (d_i^*(n), \mathbf{d}^*(n-1))$ may or may not be a NE of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$. If there is no WD that wants to deviate from its strategy played in $\mathbf{d}^*(n-1)$ then $\mathbf{d}(n)$ is a NE of the game. Otherwise, one or all of the following cases can happen: (i) there is a WD $j \in O_{(a,c)}(\mathbf{d}(n))$ that wants to update its best reply by changing its strategy from (a, c) to local computing, (ii) there is a WD $j \in O_{(a,c')}(\mathbf{d}(n))$ that wants to update its best reply by changing its strategy from (a, c'), $c' \neq c$ to local computing, (iii) there is a WD $j \in O_a(\mathbf{d}(n))$ that wants to update its best reply by changing its offloading strategy from $(a, \cdot)$ to $(a', \cdot)$, $a' \neq a$, (iv) there is a WD $j \in O_{(a',c)}(\mathbf{d}(n))$ that wants to update its best reply by changing its strategy from (a', c), $a' \neq a$ to local computing. Observe that WDs $j \in O_c(\mathbf{d}(n))$ cannot decrease their offloading cost by changing between the ECs, since EC c was WD i's best reply (i.e., $(n_c(\mathbf{d}^*(n-1)) + 1)/F^c \leq (n_{c'}(\mathbf{d}^*(n-1)) + 1)/F^{c'}$), and thus it is also a best reply for all WDs $j \in O_c(\mathbf{d}(n))$.

The JPAU algorithm lets WDs to update their best replies in the following order. If case (i) happens, the JPAU algorithm allows one of the WDs $j \in O_{(a,c)}(\mathbf{d}(n))$ to stop offloading. Now, in the updated strategy profile $\mathbf{d}'(n) = (j, d_{-j}(n))$ we have that $n_a(\mathbf{d}'(n)) = n_a(\mathbf{d}^*(n-1))$ and $n_c(\mathbf{d}'(n)) = n_c(\mathbf{d}^*(n-1))$ hold for every AP $a \in \mathcal{A}$ and every EC $c \in \mathcal{C}$, and thus $\mathbf{d}'(n)$ is a NE of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$. Otherwise, if case (ii) happens, the JPAU algorithm allows one of WDs $j \in O_{(a,c')}(\mathbf{d}(n))$ to stop offloading. In the updated strategy profile $\mathbf{d}'(n) = (j, d_{-j}(n))$ we have that $n_a(\mathbf{d}'(n)) = n_a(\mathbf{d}^*(n-1))$ for every AP a, $n_c(\mathbf{d}'(n)) = n_c(\mathbf{d}^*(n-1)) + 1$, $n_{c'}(\mathbf{d}'(n)) = n_{c'}(\mathbf{d}^*(n-1)) - 1$ and $n_{c''}(\mathbf{d}'(n)) = n_{c''}(\mathbf{d}^*(n-1))$ for every $c'' \in \mathcal{C} \setminus \{c, c'\}$. Since WD j was offloading its task to EC c' in a NE $\mathbf{d}^*(n-1)$ (i.e. before WD i enter the game) we have that $(n_c(\mathbf{d}^*(n-1)) + 1)/F^c > n_{c'}(\mathbf{d}^*(n-1))/F^{c'}$. Therefore WDs $k \in O_c(\mathbf{d}'(n))$ can decrease their offloading cost by changing their strategy from offloading to EC c to offloading to EC c'. Let us now consider the updated strategy profile $\mathbf{d}'(n) = ((\cdot, c'), d'_{-k}(n))$ after a WD $k \in O_c(\mathbf{d}'(n))$ performed its best reply $d_k^*(n) = (\cdot, c')$. We have that $n_a(\mathbf{d}'(n)) = n_a(\mathbf{d}^*(n-1))$ and $n_c(\mathbf{d}'(n)) = n_c(\mathbf{d}^*(n-1))$ hold for every AP $a \in \mathcal{A}$ and every EC $c \in \mathcal{C}$, and thus $\mathbf{d}'(n)$ is a NE of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$.

Now, let us assume that neither (i) nor (ii) happened. In that case, the JPAU algorithm allows a sequence of

update steps in which WDs $j \in O(\mathbf{d}(n))$ are allowed to decrease their offloading costs. Let us recall that WDs $j \in O(\mathbf{d}(n))$ cannot decrease their offloading cost by changing between ECs, and thus in the resulting sequence of update steps WDs only change between APs. Furthermore, observe that the sequence starts with an update step performed by WD $j \in O_a(\mathbf{d}(n))$, which corresponds to case (iii). It follows from [39] that this sequence of update steps is finite. Observe that after the sequence terminates in the updated strategy profile $\mathbf{d}'(n)$, we have that $n_{a''}(\mathbf{d}'(n)) = n_{a''}(\mathbf{d}^*(n-1)) + 1$ holds for AP a'' through which a WD started offloading in the last update step, and thus some of the WDs $j \in O_{a''}(\mathbf{d}'(n))$ may want to stop offloading. Observe that the case when there is a WD $j \in O_{(a'',c)}(\mathbf{d}'(n))$ that wants to stop offloading corresponds to case (i) and the case when there is a WD $j \in O_{(a'',c')}(\mathbf{d}'(n))$, $c' \neq c$ that wants to stop offloading corresponds to case (ii). In the discussion above we showed that the JPAU algorithm terminates in a NE of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ in both cases. Let us now consider that none of the previous two cases happened. Then, if there are no WDs $j \in O_{a'}(\mathbf{d}'(n))$, $a' \neq a''$ that want to stop offloading then the JPAU algorithm terminates in a NE because there are no WDs that want to start offloading either because $n_r(\mathbf{d}'(n)) \geq n_r(\mathbf{d}^*(n-1))$, $\forall r \in \mathcal{A} \cup \mathcal{C}$.

Otherwise, let us assume that a WD $j \in O_{(a',c)}(\mathbf{d}'(n))$, $a' \neq a''$ (i.e., $n_{a'}(\mathbf{d}'(n)) = n_{a'}(\mathbf{d}^*(n-1))$) wants to stop offloading because the congestion in EC c increased, i.e., because $n_c(\mathbf{d}'(n)) = n_c(\mathbf{d}^*(n-1))+1$ holds. Observe that if cases (i)-(iii) did not happen, we have that $a'' = a$, which corresponds to case (iv). After a WD $j \in O_{(a',c)}(\mathbf{d}'(n))$ stops to offload, in the updated strategy profile $\mathbf{d}'(n) = (j, d'_{-j}(n))$ we have that $n_{a''}(\mathbf{d}'(n)) = n_{a''}(\mathbf{d}^*(n-1))+1$, $n_{a'}(\mathbf{d}'(n)) = n_{a'}(\mathbf{d}^*(n-1))-1$, $n_b(\mathbf{d}'(n)) = n_b(\mathbf{d}^*(n-1))$ for APs $b \in \mathcal{A}\setminus\{a'', a'\}$ and $n_c(\mathbf{d}'(n)) = n_c(\mathbf{d}^*(n-1))$ for every EC $c \in \mathcal{C}$. The JPAU algorithm first allows one of the WDs $k \in O_{a''}(\mathbf{d}'(n))$ to decrease its offloading cost by changing its strategy from $(a'', \cdot)$ to $(a', \cdot)$. If such a WD k exists, after it performs the best reply we have that $n_a(\mathbf{d}'(n)) = n_a(\mathbf{d}^*(n-1))$ and $n_c(\mathbf{d}'(n)) = n_c(\mathbf{d}^*(n-1))$ hold for every AP $a \in \mathcal{A}$ and every EC $c \in \mathcal{C}$ in the updated strategy profile $\mathbf{d}'(n) = ((a', \cdot), d'_{-k}(n))$. Thus, there is no WD that can further decrease its cost and the JPAU algorithm terminates in a NE of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$. Otherwise, if there is no WD $k \in O_{a''}(\mathbf{d}'(n))$ that can decrease its offloading cost by changing the strategy from $(a'', \cdot)$ to $(a', \cdot)$, then the JPAU algorithm allows a sequence of the update steps in which WDs $k \in O(\mathbf{d}'(n))$ are allowed to decrease their offloading costs by changing between the APs. It follows from [39] that this sequence of the update steps is finite. Let us now consider a strategy profile $\mathbf{d}'(n)$ after the last update step in the sequence was performed. We can either have $n_a(\mathbf{d}'(n)) = n_a(\mathbf{d}^*(n-1))$ for every AP $a \in \mathcal{A}$ or $n_{a''}(\mathbf{d}'(n)) = n_{a''}(\mathbf{d}^*(n-1))+1$, $n_{a'}(\mathbf{d}'(n)) = n_{a'}(\mathbf{d}^*(n-1))-1$ for some $a' \in \mathcal{A}\setminus\{a''\}$ and $n_b(\mathbf{d}'(n)) = n_b(\mathbf{d}^*(n-1))$ for $b \in \mathcal{A} \setminus \{a'', a'\}$. In the first case the JPAU algorithm terminates in a NE of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$ since $n_c(\mathbf{d}'(n)) = n_c(\mathbf{d}^*(n-1))$ still holds for every EC $c \in \mathcal{C}$ (the WDs were not changing between ECs). In the second case we have that some of the WDs $k \in \mathcal{N}_n \setminus O(\mathbf{d}'(n))$ that are performing computation locally may want to start to offload through AP a'. If such a WD does not exist, $\mathbf{d}'(n)$ is a NE of $\Gamma(\mathcal{P}_r^{t,*}, \mathcal{P}_c^{t,*})$, since there is no WD that can further decrease its cost. Otherwise, if such a WD k exists its best reply is (a', c) since $n_c(\mathbf{d}'(n)) = n_c(\mathbf{d}^*(n-1))$ and $(n_c(\mathbf{d}^*(n-1))+1)/F^c \leq (n_{c'}(\mathbf{d}^*(n-1))+1)/F^{c'}$. Observe that in the updated strategy profile $\mathbf{d}'(n) = ((a', c), d'_{-k}(n))$ we have that $n_{a'}(\mathbf{d}'(n)) = n_{a'}(\mathbf{d}^*(n-1))$ and $n_b(\mathbf{d}'(n)) \geq n_b(\mathbf{d}^*(n-1))$ for $b \in \mathcal{A} \setminus \{a'\}$, and thus WDs $j \in O_{a'}(\mathbf{d}'(n))$ cannot decrease their offloading cost. Furthermore, WDs $j \in O(\mathbf{d}'(n)) \setminus O_{a'}(\mathbf{d}'(n))$ cannot decrease their offloading cost either since they could not do so before WD k started offloading. Finally, we observe that WDs $O_{(a'',c)}(\mathbf{d}'(n))$ do not want to stop offloading because they did not want to do so after the sequence of update steps of type (iii) terminated, and consequently only an update step of type (iv) may happen.

In the following we show that an update step of type (iv) can happen a finite number of times. First, observe that $n_{a'}(\mathbf{d}'(n)) = n_{a'}(\mathbf{d}^*(n-1))$, $a' \neq a''$ holds. Thus, a WD $j \in O_{(a',c)}(\mathbf{d}'(n))$, may want to stop offloading only because the congestion in EC c increased. Second, observe that a WD $j \in O_c(\mathbf{d}'(n))$ for which a best reply in one of the previous steps was to share EC c with $n_c(\mathbf{d}^*(n-1))$ WDs, will not have an incentive to perform an improvement step of type (iv) after a WD $k \in \mathcal{N}_n \setminus O(\mathbf{d}'(n))$ starts to offload to EC c, i.e., after the congestion in EC c increases to $n_c(\mathbf{d}^*(n-1))+1$ again. The same holds for all WDs that decide to change their strategy from local computing to offloading to EC c. Consequently, the length of the sequence of update steps of type (iv) is at most $n_c(\mathbf{d}^*(n-1))$, which proves the theorem.

C. Proof of Proposition 3

First, observe that in a sequence of update steps of type (iii) each WD $j \in O(\mathbf{d}^*(n-1))$ can deviate at most once. This is due to that when a WD $j \in O(\mathbf{d}^*(n-1))$ moves to an AP a it brings the system to a state where $n_a(\mathbf{d}'(n)) = n_a(\mathbf{d}^*(n-1))+1$ holds and since $n_a(\mathbf{d}'(n))$ can only decrease in the following improvement steps, WD j will not have an incentive to deviate again.

In the worst case scenario $A \geq 3, C = 1$, all WDs offload their tasks in a NE $\mathbf{d}^*(n-1)$ i.e., $O(\mathbf{d}^*(n-1)) = \mathcal{N}_{n-1}$ and case (iii) happens such that every WD $j \in O(\mathbf{d}^*(n-1))$ changes between APs exactly once. Furthermore, in the worst case scenario, after an $|\mathcal{N}_{n-1}|$ long sequence of update steps of type (iii), one of the WDs stops to offload due to increased congestion only in the EC. Observe that in the resulting strategy profile there is one AP a'' on which the congestion increased, one AP a' on which the congestion decreased and the congestion on the other APs has not changed compared with the congestions in $\mathbf{d}^*(n-1)$. Now, observe that in the worst case scenario each of the WDs that offload through an AP $a \in \mathcal{A} \setminus \{a''\}$ changes between the APs, and consequently in the worst case scenario $n_{a''}(\mathbf{d}^*(n-1)) = 1$. Since the remaining $|\mathcal{N}_{n-1}| - 1$ WDs do not have an incentive to move to the AP a'' because of the increased congestion, it follows from the definition of a best reply that in the resulting sequence each of $|\mathcal{N}_{n-1}| - 1$ WDs updates its strategy at most $A-2$ times. Furthermore, it follows from the proof of Theorem 4 that an update of type (iv) can happen at most $n_c(\mathbf{d}^*(n-1))$ times, where $n_c(\mathbf{d}^*(n-1)) = |\mathcal{N}_{n-1}|$ for $C = 1$ and $O(\mathbf{d}^*(n-1)) = \mathcal{N}_{n-1}$. Finally, we obtain that a NE is reached after at most $|\mathcal{N}_{n-1}| + (1 + (A-2)(|\mathcal{N}_{n-1}| - 1) + 1)|\mathcal{N}_{n-1}|$, which proves the result.

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] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 2014, pp. 68–81.

[4] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," *IEEE Wireless communications*, vol. 20, no. 3, pp. 14–22, 2013.

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

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

[16] 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.

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

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

[19] 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.

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

[21] 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.

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

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

[24] 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.

[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] G. Bin, F. L. Zhi, Zhou and, and X. Fei, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *Proc. of IEEE INFOCOM*, 2019.

[29] C. Shutong, J. Lei, W. Lin, and L. Fangming, "An online market mechanism for edge emergency demand response via cloudlet control," in *Proc. of IEEE INFOCOM*, 2019.

[30] F. Liu, P. Shu, and J. C. Lui, "Appatp: An energy conserving adaptive mobile-cloud transmission protocol," *IEEE Transactions on Computers*, vol. 64, no. 11, pp. 3051–3063, 2015.

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

[32] ——, "Wireless and computing resource allocation for selfish computation offloading in edge computing," in *Proc. of IEEE INFOCOM*, 2019.

[33] 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.

[34] 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.

[35] 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.

[36] ——, "Decentralized algorithm for randomized task allocation in fog computing systems," *IEEE/ACM Transactions on Networking*, 2018.

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

[38] I. Milchtaich, "The equilibrium existence problem in finite network congestion games," in *Proc. of WINE*, 2006, pp. 87–98.

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

Slađana Jošilo is a Ph.D. student at the Department of Network and Systems Engineering in KTH, Royal Institute of Technology. She received her M.Sc. degree in electrical engineering from the University of Novi Sad, Serbia in 2012. She worked as a research engineer at the Department of Power, Electronics and Communication Engineering, University of Novi Sad from 2013 to 2014. Her research interests are design and analysis of decentralized algorithms for exploiting resources available at the network edge using game theoretical tools.

György Dán (M'07, SM'17) is a professor at KTH Royal Institute of Technology, Stockholm, Sweden. He received the M.Sc. in computer engineering from the Budapest University of Technology and Economics, Hungary in 1999, the M.Sc. in business administration from the Corvinus University of Budapest, Hungary in 2003, and the Ph.D. in Telecommunications from KTH in 2006. He worked as a consultant in the field of access networks, streaming media and videoconferencing 1999-2001. He was a visiting researcher at the Swedish Institute of Computer Science in 2008, a Fulbright research scholar at University of Illinois at Urbana-Champaign in 2012-2013, and an invited professor at EPFL in 2014-2015. He has been an area editor of Computer Communications since 2014. His research interests include the design and analysis of content management and computing systems, game theoretical models of networked systems, and cyber-physical system security and resilience.