This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

Joint Resource Management and Pricing for Task Offloading in Serverless Edge Computing

Feridun Tütüncüoğlu and György Dán
Division of Network and Systems Engineering
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology, Stockholm, Sweden
Email: {feridun, gyuri}@kth.se

***Abstract*—We consider the problem of resource allocation, pricing and application caching for latency sensitive task offloading in serverless edge computing. We model the interaction between a profit-maximizing operator and cost-minimizing *Wireless Devices* (WDs) as a Stackelberg game where the operator is the leader and decides the price, resource allocation and set of applications to cache, while the WDs are the followers and decide whether to offload their tasks. We first show that the game has a *Subgame Perfect Equilibrium* (SPE), but computing it, is NP-hard. Importantly, we show that an SPE, which maximizes the operator's revenue, results in minimal energy consumption among the WDs. For computing an approximate SPE, we propose a linear time approximation algorithm with bounded approximation ratio for resource allocation and pricing, and we propose an efficient heuristic based on the utility density of individual applications for the joint optimization of caching, resource allocation and pricing. Our results show that the proposed algorithm outperforms state-of-the-art methods by up to an order of magnitude both in terms of revenue and total energy savings and has small computational overhead. An interesting feature of our results is that the utility of the operator is maximized by a solution that maximizes the WDs' energy savings through computation offloading, which makes it a promising candidate for energy efficient edge cloud deployments.**

***Index Terms*—Edge computing, Function as a service, Stackelberg game, Combinatorial optimization, Convex optimization**

I. Introduction

Serverless computing (also called *Function as a Service* (FaaS)) is transforming the cloud computing landscape by offering a paradigm shift in the way applications are developed and deployed [1]–[3]. It eliminates the need for users to manage the server infrastructure, enabling them to focus solely on writing code to implement business logic [4], [5]. Its ease of use combined with the pay-as-you-go billing model make serverless computing a particularly appealing service model from a user perspective. At the same time, it allows the cloud operator more freedom in managing its communication and computing resources for serving the user demand.

Serverless computing at the edge could provide low-latency access to computing resources on-demand to mobile *Wireless Devices* (WDs), enabling task offloading for computationally intensive applications without advance reservation of resources, thereby saving battery power [6]–[8]. Nonetheless, from the operator's perspective, the inherent capacity constraints in edge computing make the adoption of serverless computing at the edge challenging compared to centralized clouds [9], as communication, storage and computing resources have to be orchestrated for meeting application latency requirements, and at the same time, the operator's financial interests have to be catered for.

The orchestration of wireless and compute resources at the edge have been extensively studied in recent years [10]–[14]. Nonetheless, the management of storage and the availability of executable code at the edge servers, which are prerequisites for serverless computing, received less attention. Existing works focus mainly on minimizing the total cost of the WDs in terms of delay, energy consumption, or their combination [15]–[18], but they do not consider the financial interests of the edge operator: the operator's objective is arguably the maximization of its profit, while the minimization of the total cost of the WDs and their latency constraints should rather be considered a possibly conflicting secondary objective or a constraint.

Indeed, storage, computing and communication resource allocation and pricing are mutually dependent. The availability of code determines whether a WD is able to offload, the compute and communication resources determine whether offloading would meet the latency requirements, and the price determines whether it is worthwhile to offload. The decisions of the WDs in turn determine the revenue of the operator and hence its decision what applications to make available. Optimal pricing and resource management is thus inherently challenging and at the same time fundamental for realizing a serverless edge ecosystem.

In this work, we address this challenging problem. We model the interaction between a profit-maximizing operator that performs storage management, resource allocation and pricing, and cost-minimizing autonomous WDs that can offload their computation subject to code availability and latency requirements as a Stackelberg game. Based on the model, we propose a pricing scheme that maximizes the operator's revenue and simultaneously incentivizes the WDs to make energy optimal decisions. Our main contributions are as follows;

- we propose a Stackelberg game to model the interaction between the operator and latency sensitive WDs,
- we show that a *Subgame Perfect Equilibrium* (SPE) exists in the proposed Stackelberg game,
- we show that the joint optimization of pricing and the allocation of wireless and computational resources is a convex problem for given set of offloading WDs, and

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

the solution results in an equilibrium that minimizes the energy consumption the WDs,

- we show that computing an optimal set of offloading WDs is NP-hard, and we propose a linear complexity approximation algorithm,
- we show that computing the optimal set of applications to cache is NP-hard and we propose an efficient algorithm for computing an approximate solution
- we provide numerical results based on simulations that show that our proposed algorithm is efficient for the joint optimization of caching, resource allocation and pricing, and it outperforms state-of-the-art algorithms.

The rest of the paper is organized as follows. We present the system model and the problem formulation in Section II. We show the best response of the WDs and the the existence of equilibria in Section III. We address optimal resource allocation and pricing for a fixed set of offloaders in Section IV and we propose an approximation algorithm to compute a near optimal set of offloaders in Section V. We address the problem of caching in Section VI, and we show numerical results in Section VII. We discuss related work in Section VIII, and Section IX concludes the paper.

II. System Model and Problem Formulation

We consider a multi-access edge computing system that consists of an edge server with storage capacity S managed by an operator, and a set $\mathcal{N}=\{1,2,\dots,N\}$ of WDs that can offload their computational tasks for execution at the edge server through a wireless link. WD $i\in\mathcal{N}$ wants to execute a task of type $\phi_i\in\mathcal{J}$, where $\mathcal{J}$ denotes the set of applications (i.e. set of task types). The applications are the software images required for the execution of the tasks. The computational task of WD i is characterized by the size D_i of the input data in terms of bytes, by the expected number L_j of *instructions* (I) per byte required to perform the task for $j=\phi_i$, and by the completion time requirement $\bar{\tau}_i^l$ defined by the WD.

The edge operator can decide to cache a subset $\mathcal{X}\subseteq\mathcal{J}$ of applications subject to its storage capacity constraint

$$\sum_{j\in\mathcal{X}} s_j \leq S, \tag{1}$$

where s_j is the memory size of the application image. If the application that WD intends to use is cached by the operator $(\phi_i\in\mathcal{X})$ then WD i can decide whether to offload the computation to the edge server. We denote by a_i the offloading decision of WD i; $a_i=1$ corresponds to offloading and $a_i=0$ to local computing. If WD i offloads then it is charged a price of $\pi_i\geq 0$ and is given a portion of the finite processing capacity F and finite bandwidth capacity W by the edge operator. The price and the resource allocation are decided by the operator before the WDs decide whether or not to offload. Fig. 1 illustrates a system with $|\mathcal{J}|=6$ applications and $N=7$ WDs. We next present our model of local computing and computation offloading, followed by the problem formulation.

A. Local Computing

If WD i chooses not to offload, the task needs to be executed using local computing resources (i.e., local CPU). We denote by f_i^l the local processing power (measured in *Giga Instructions per Second* (GIPS)) of WD i and we use it to express the local processing time as

$$\tau_i^l=\frac{L_{\phi_i}D_i}{f_i^l}. \tag{2}$$

We consider that f_i^l is chosen such that local computing completes upon the task completion deadline τ_i^l of the task of WD i, i.e., $\tau_i^l=\bar{\tau}_i^l$. Thus, the task completion deadline $\bar{\tau}_i^l$ will influence the decision of the WD whether or not to offload. This assumption is reasonable, as dynamic voltage and frequency scaling is widely used for reducing the energy consumption of battery powered WDs while meeting performance needs [19]–[21].

B. Computation Offloading

If WD i decides to offload, it has to transmit D_i amount of data over the wireless channel to the edge server via an *Access Point* (AP), and then processing is performed at the edge server. We denote by w_i the bandwidth allocated to WD i by the edge operator and, we make the common assumption of a Gaussian channel [11]. We can then express the upload time of WD i using the Shannon formula [22],

$$\tau_i^u(p_i,w_i)=\frac{D_i}{w_i\log_2(1+\frac{p_ih_i}{\bar{\sigma}_i^2})}, \tag{3}$$

where h_i is the channel coefficient from WD i to the AP, p_i is the transmit power of the WD i, and $\bar{\sigma}_i^2$ is the noise power at the AP. We consider that the transmit power is bounded by the maximum transmit power $\bar{p}_i$, i.e. $p_i\leq\bar{p}_i$. This model of the transmission rate corresponds to *Orthogonol Frequency Division Multiple Access* (OFDMA), adopted in 5G and WiFi6 [23], [24], which allocates resource blocks (also called resource units) to WDs for data transmission and avoids intra-cell interference despite simultaneous transmissions from multiple WDs by using non-overlapping subcarriers.[1] [25]. Similar to previous works [15], [26], [27], we make the common assumption that the time needed to transmit the results of the computation from the edge server to the WD is negligible, because for many applications (e.g., object detection, recognition and tracking) the size of the output is significantly smaller than the size of the input data.

We denote by f_i the allocated computing power of the edge server (measured in GIPS) and we express the processing time at the edge server as

$$\tau_i^e(f_i)=\frac{L_{\phi_i}D_i}{f_i}. \tag{4}$$

[1] The communication model could be extended to account for interference among WDs. Doing so would make the analysis more involved, but would not affect the validity of Lemma 1 and Theorem 1.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

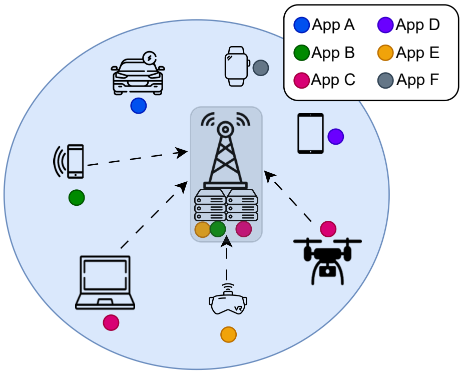

Fig. 1: Figure shows a system with $\mathcal{J} = \{A, B, C, D, E, F\}$ and $|\mathcal{N}| = 7$. The operator chooses applications $\mathcal{X} = \{B, C, E\}$ to cache. The dotted arrows show that WDs choose to offload to the edge server to reduce their computation costs. The WDs without arrow choose to execute their task locally.

C. WD Cost Model

Computation and offloading incur energy consumption and monetary cost at the WDs. In case of local computing the cost is the energy consumed by the WD for executing the task,

$$C_i^0 = \tau_i^l (f_i^l)^2 \kappa_i^l \gamma_i, \tag{5}$$

where κ_i^l is the energy efficiency parameter of the WD with unit J per Hz per GI^2 and γ_i is the unit local energy cost with unit of \$ per J. γ_i is determined by the cost of electricity and by the cost of charging the battery of WD i, e.g., in terms of time, etc. and serves as the conversion factor from energy consumption to its monetary cost. We make the reasonable assumption that γ_i is known to WD i and thus C_i^0 can be computed.

In case of offloading, we define the offloading cost as the sum of the energy consumption cost for transmitting the input data and the price that is to be paid, i.e.,

$$C_i^1(p_i, \boldsymbol{\rho}_i, a_{-i}) = \tau_i^u(p_i, w_i) p_i \beta_i \gamma_i + \pi_i, \tag{6}$$

where a_{-i} denotes the offloading decisions of WDs $i' \in \mathcal{N} \setminus \{i\}$, β_i denotes the transmit antenna power efficiency parameter of WD i. Let us define the vector $\boldsymbol{\rho}_i = [f_i\,, w_i]$ of resources allocated to WD i, then the cost of WD i is

$$C_i(a_i, p_i, \boldsymbol{\rho}_i, a_{-i}) = (1 - a_i) C_i^0 + a_i C_i^1(p_i, \boldsymbol{\rho}_i, a_{-i}). \tag{7}$$

We consider that the WDs have a preference for saving the state of charge of their batteries, thus in case of a tie between local computing cost and offloading cost the WD would choose to offload. The local cost of a WD represents its valuation of task execution, and its formulation is consistent with the modeling approach used formerly in cloud computing [28], [29]. In economic terms, this valuation corresponds to the *reservation price*, which is the highest price that a customer would pay for a particular product or service [30], [31].

D. Problem Formulation

We consider that the WDs and the operator are rational, strategic entities. The objective of WD i is to minimize its cost subject to its completion time requirement, the constraint on the maximum transmission power, and the caching decision of the operator. Thus, WD i aims to solve

$$\min_{p_i,\, a_i} \quad C_i(a_i, p_i, \boldsymbol{\rho}_i, a_{-i}), \tag{8}$$

$$s.t. \quad a_i(\tau_i^u(p_i, w_i) + \tau_i^e(f_i)) \le \tau_i^l, \tag{9}$$

$$a_i = 0, \phi_i \notin \mathcal{X}, \tag{10}$$

$$p_i \le \bar{p}_i, \tag{11}$$

where the first constraint ensures that WD i does not offload if $\tau_i^u(p_i, w_i) + \tau_i^e(f_i) > \tau_i^l$, i.e., if the completion time when offloading, exceeds the task completion deadline, the second constraint ensures that the WD can only offload if its application is cached by the operator, thus offloading will not result in cold start of the application, and the last constraint ensures that the transmit power remains within the limit of the maximum transmit power. We refer $a_i \in \mathcal{A}_i = \{0, 1\}$ as the action set of WD i denoting local computing and offloading respectively.

Aligned with FaaS pricing models used today, we consider that the income of the operator depends on the price it sets for offloading and on whether or not WDs offload. Thus the operator's utility from the offloading WDs is

$$U_{\mathcal{X}}(\boldsymbol{a}, \boldsymbol{\rho}, \boldsymbol{\pi}) = \sum_{i \in \mathcal{N}} a_i \mathbb{1}_{\phi_i \in \mathcal{X}} \pi_i. \tag{12}$$

where $\mathbb{1}_{\phi_i \in \mathcal{X}}$ is the indicator function, and we refer to the collection $\boldsymbol{a} = (a_i)_{i \in \mathcal{N}}$ as the offloading decision of the WDs. We refer to the collection $\boldsymbol{\rho} = (\boldsymbol{\rho}_i)_{i \in \mathcal{N}}$ as the resource allocation decision, and to the collection $\boldsymbol{\pi} = (\pi_i)_{i \in \mathcal{N}} \in [0, \bar{\pi}]^N$ as the pricing decision, where $\bar{\pi} \in \mathbb{R}$ is a sufficiently large constant that serves as an upper bound on the price.

We consider that the operator aims at maximizing its utility, by choosing resource allocation $\boldsymbol{\rho}$, prices $\boldsymbol{\pi}$, and caching decision $\mathcal{X}$, i.e., the operator wants to solve

$$\max_{\boldsymbol{\pi}, \boldsymbol{\rho}, \mathcal{X} \subseteq \mathcal{J}} U_{\mathcal{X}}(\boldsymbol{a}, \boldsymbol{\rho}, \boldsymbol{\pi}), \tag{13}$$

$$s.t. \sum_{i \in \mathcal{N}} f_i \le F, \sum_{i \in \mathcal{N}} w_i \le W, \sum_{j \in \mathcal{X}} s_j \le S. \tag{14}$$

The resulting problem is a multiple-follower single leader Stackelberg game, where the operator is the leader and the WDs are the followers. We refer to the problem as the *Joint Pricing, Caching and Resource Allocation Game* (PICRA). We are interested in the existence of Stackelberg equilibria and the complexity of computing equilibria, under complete information, i.e., the system parameters and utilities are known. While this assumption may seem strong, it enables us to analyze the structure of the game and formulate an approach for computing an equilibrium. Moreover, analyzing the complete information case serves as an initial step for subsequent analysis under incomplete information [10], [12], [32], [33]. We start with solving the problem faced by the WDs, and we then turn to solving the problem faced by the operator.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

$\mathcal{N}$	Set of WDs
$\mathcal{J}$	Set of applications
j_i	Application used by WD i
D_i	Size of input data [MB]
L_j	Expected complexity of app j [I/B]
$\bar{\tau}_i^l$	Completion time requirement of WD i [s]
τ_i^l	Local computing time of WD i [s]
a_i	Offloading decision of WD i
$\mathcal{X}$	Set of cached applications
s_j	Memory size of application j [GB]
S	Storage capacity of the operator [GB]
F	Computing capacity of the operator [GIPS]
W	Bandwidth capacity of the operator [Hz]
π_i	Price of the EC service [$]
p_i	Transmit power of the WD i [W]
$\bar{p}_i$	Maximum transmit power of WD i [W]
h_i	Channel coefficient from WD i to AP
$\bar{\sigma}_i^2$	Noise power at the AP [W]
f_i	Allocated computing capacity to WD i by MEC [GIPS]
γ_i	Cost of electricity [$ / J]
κ_i^l	Energy efficiency parameter of WD i [J/Hz/GI^2]
β_i	Transmit antenna power efficiency of WD i

TABLE I: Summary of frequently used notations.

III. WD Best Response Characterization and Existence of Equilibria

We start the analysis with characterizing the best response of the WDs for given caching decision $\mathcal{X}$, pricing $\boldsymbol{\pi}$ and resource allocation $\boldsymbol{\rho}$, announced by the operator. For caching decision $\mathcal{X}$, we denote by $\mathcal{N}_{\mathcal{X}} = \{\forall i \in \mathcal{N} | \phi_i \in \mathcal{X}\}$ the set of WDs whose applications are cached by the operator, i.e., the potential offloaders, and we define $N_{\mathcal{X}} = |\mathcal{N}_{\mathcal{X}}|$. We first show that the best response of the WDs has a threshold structure and can be computed efficiently.

Lemma 1. *Consider a WD $i \in \mathcal{N}_{\mathcal{X}}$. If $\tau_i^e(f_i) > \tau_i^l$ then $a_i^* = 0$. Otherwise let p_i^* be such that $\tau_i^u(p_i, w_i) + \tau_i^e(f_i) = \tau_i^l$. Then if $p_i^* > \bar{p}_i$, $a_i^* = 0$, otherwise*

$$a_i^* = \begin{cases} 1, & \pi_i \leq T_i, \\ 0, & else, \end{cases} \tag{15}$$

where $T_i = L_{\phi_i} D_i \gamma_i (f_i^l \kappa_i^l - p_i^* \beta_i (\frac{1}{f_i^l} - \frac{1}{f_i}))$.

Proof. Observe that if $\tau_i^e(f_i) > \tau_i^l$, then WD i cannot complete the task on time if it offloads, thus to complete the task before the deadline it has to perform local computing, i.e., the optimal offloading decision is $a_i^* = 0$. Otherwise, WD i should choose a transmit power that minimizes its cost while ensuring timely completion. Observe that the uploading time $\tau_i^u(p_i, w_i)$ is a strictly monotonically decreasing function of p_i, and $C_i^1(p_i, \boldsymbol{\rho}_i, a_{-i})$ is a strictly monotonically increasing function of p_i. Thus, i minimizes its cost by choosing a transmit power p_i^* that yields $\tau_i^u(p_i^*, w_i) + \tau_i^e(f_i) = \tau_i^l$. Now, if $p_i^* > \bar{p}_i$ then offloading is not feasible. Otherwise, if $p_i^* \leq \bar{p}_i$ then the optimal decision is to offload if and only if $C_i^1(a_i, \boldsymbol{\rho}_i, a_{-i}) \leq C_i^0$, i.e.,

$$a_i^* = \begin{cases} 1, & \tau_i^u(p_i^*, w_i) p_i^* \beta_i \gamma_i + \pi_i \leq \tau_i^l (f_i^l)^2 \kappa_i^l \gamma_i, \\ 0, & else. \end{cases} \tag{16}$$

Since, the optimal transmit power yields $\tau_i^u(p_i^*, w_i) + \tau_i^e(f_i) = \tau_i^l$, we can substitute $\tau_i^u(p_i^*, w_i) = \tau_i^l - \tau_i^e(f_i)$, (2) and (4) into (16), and obtain (15), which proves the result. □

We know by Lemma 1 that the operator can compute the WDs' best replies for a given strategy $\boldsymbol{\rho}, \boldsymbol{\pi}, \mathcal{X}$. Given the best response of the WDs, we next show the existence of a SPE, defined as follows.

Definition 1 (Subgame Perfect Equilibrium (SPE) [34])**.** *Let $(\boldsymbol{\rho}^*, \boldsymbol{\pi}^*, \mathcal{X}^*)$ be a solution of (13)-(14), and let $\boldsymbol{a}^*$ be a solution of (8)-(11). Then the point $(\boldsymbol{\rho}^*, \boldsymbol{\pi}^*, \mathcal{X}^*, \boldsymbol{a}^*)$ is an SPE of the PICRA game if for any feasible $(\boldsymbol{\rho}, \boldsymbol{\pi}, \mathcal{X}, \boldsymbol{a})$ point the following holds,*

$$U_{\mathcal{X}^*}(\boldsymbol{a}^*, \boldsymbol{\rho}^*, \boldsymbol{\pi}^*) \geq U_{\mathcal{X}}(\boldsymbol{a}^*, \boldsymbol{\rho}, \boldsymbol{\pi}), \tag{17}$$

$$C_i(a_i^*, p_i^*, \boldsymbol{\rho}_i^*, a_{-i}^*) \leq C_i(a_i, p_i, \boldsymbol{\rho}_i^*, a_{-i}^*), \tag{18}$$
$$\forall \{a_i, p_i\} \in \mathcal{A}_i \times [0, \bar{p}_i], \forall i \in \mathcal{N}.$$

We now prove the existence of SPE.

Theorem 1. *The PICRA game possesses a SPE.*

Proof. By Lemma 1, for given $(\boldsymbol{\rho}, \boldsymbol{\pi}, \mathcal{X})$, the best response $\boldsymbol{a}^*$ is unique and can be computed efficiently. Then, by the extreme value theorem [35], there exists a solution to problem (13)-(14), and this solution is by definition an SPE. This proves the result. □

Observe that the operator could use Theorem 1 for computing an SPE. Thus, we turn to the analysis of the complexity of computing an SPE.

IV. Optimal Resource Allocation and Pricing for a Fixed Set of Offloaders

We start by considering a feasible caching decision $\mathcal{X}$ of the operator, i.e, $\sum_{j \in \mathcal{X}} s_j \leq S$, and a set $\mathcal{N}_{\mathcal{X}}^o = \{i \in \mathcal{N}_{\mathcal{X}} \,|\, a_i = 1\}$ of offloaders. We are interested in computing the optimal resource allocation and pricing, i.e., one that results in the optimal utility $U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o}$ for given caching decision $\mathcal{X}$ and set of offloaders $\mathcal{N}_{\mathcal{X}}^o$.

For given set of caching decision $\mathcal{X}$ and set of offloaders $\mathcal{N}_{\mathcal{X}}^o$, the optimal utility $U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o}$ of the operator is the solution to

$$\max_{(\pi_i, \boldsymbol{\rho}_i)_{i \in \mathcal{N}_{\mathcal{X}}^o}} \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} \pi_i, \tag{19}$$

$$s.t. \quad \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} f_i \leq F, \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} w_i \leq W, \tag{20}$$

$$f_i \geq f_i^l, w_i \geq 0, \ \forall i \in \mathcal{N}_{\mathcal{X}}^o, \tag{21}$$

$$p_i^* \leq \bar{p}_i, \forall i \in \mathcal{N}_{\mathcal{X}}^o, \tag{22}$$

$$\tau_i^u(p_i^*, w_i) p_i^* \beta_i \gamma_i + \pi_i \leq C_i^0, \forall i \in \mathcal{N}_{\mathcal{X}}^o, \tag{23}$$

where (21), (22), (23) are necessary constraints for WDs to be able to offload, consistent with (8)-(11).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

Our first result characterizes the optimal pricing strategy $\boldsymbol{\pi}^*$ of the operator, i.e., $\boldsymbol{\pi}^* = (\pi_i^*)_{i \in \mathcal{N}_{\mathcal{X}}}$.

Proposition 1. *Consider that problem (19)-(23) is feasible for $\mathcal{N}_{\mathcal{X}}^o$, i.e., there is a resource allocation $\boldsymbol{\rho}$ and price $\boldsymbol{\pi}$ such that WDs $i \in \mathcal{N}_{\mathcal{X}}^o$ can offload. Then the operator's optimal pricing strategy is $\pi_i^* = C_i^0 - \tau_i^u(p_i^*, w_i^*)p_i^*\beta_i\gamma_i$ for $i \in \mathcal{N}_{\mathcal{X}}^o$ where $(f_i^*, w_i^*)_{i \in \mathcal{N}_{\mathcal{X}}^o} = \boldsymbol{\rho}^*(\mathcal{N}_{\mathcal{X}}^o)$ is the solution of*

$$\min_{(\boldsymbol{\rho}_i)_{i \in \mathcal{N}_{\mathcal{X}}^o}} \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} \tau_i^u(p_i^*, w_i)p_i^*\beta_i\gamma_i, \tag{24}$$

$$s.t. \quad \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} f_i \le F, \ \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} w_i \le W, \tag{25}$$

$$f_i \ge f_i^l, w_i \ge 0, \forall i \in \mathcal{N}_{\mathcal{X}}^o, \tag{26}$$

$$p_i^* \le \bar{p}_i, \forall i \in \mathcal{N}_{\mathcal{X}}^o, \tag{27}$$

$$\tau_i^u(p_i^*, w_i)p_i^*\beta_i\gamma_i \le C_i^0, \forall i \in \mathcal{N}_{\mathcal{X}}^o. \tag{28}$$

Proof. Observe that in the optimal solution of problem (19) – (23) , the constraint (23) is always active. Hence, the optimal price satisfies $\pi_i^* = C_i^0 - \tau_i^u(p_i^*, w_i^*)p_i^*\beta_i\gamma_i$. Next, observe that the original problem (19)-(23) is equivalent to

$$\max_{(\boldsymbol{\rho}_i)_{i \in \mathcal{N}_{\mathcal{X}}^o}} \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} C_i^0 - \tau_i^u(p_i^*, w_i)p_i^*\beta_i\gamma_i, \tag{29}$$

$$s.t. \quad \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} f_i \le F, \ \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} w_i \le W, \tag{30}$$

$$f_i \ge f_i^l, w_i \ge 0, \forall i \in \mathcal{N}_{\mathcal{X}}^o, \tag{31}$$

$$p_i^* \le \bar{p}_i, \forall i \in \mathcal{N}_{\mathcal{X}}^o, \tag{32}$$

$$\tau_i^u(p_i^*, w_i)p_i^*\beta_i\gamma_i \le C_i^0, \forall i \in \mathcal{N}_{\mathcal{X}}^o. \tag{33}$$

Observe that $\sum_{i \in \mathcal{N}_{\mathcal{X}}^o} C_i^0$ is constant and hence, the solution set of (29)-(33) is the same as that of (24)-(28), which proves the result. □

Importantly, Proposition 1 implies that computing the optimal price requires computation of the resource allocation $(f_i^*, w_i^*)_{i \in \mathcal{N}_{\mathcal{X}}^o}$ that minimizes the total transmission energy cost. We next show that problem (24) – (28) is convex, thus an optimal strategy can be computed using numerical solvers for a given set of offloaders [36].

Theorem 2. *Problem (24)-(28) is a convex problem.*

Before providing the proof of the theorem, we present two auxiliary results.

Lemma 2. *The optimal transmit power is $p_i^* = \frac{\bar{\sigma}_i^2}{h_i}(2^{(L_{\phi_i} w_i(\frac{1}{f_i^l} - \frac{1}{f_i}))^{-1}} - 1)$. p_i^* is a convex and monotonically decreasing function of (f_i, w_i) for $W \ge w_i \ge 0$ and $F \ge f_i \ge f_i^l$.*

Proof. We provide the proof in the Appendix. □

Lemma 2 shows that the optimal transmit power of the WD is a convex function of the allocated resources. We now use this result to show that the offloading energy cost is a convex function of the computing and wireless capacity allocation.

Lemma 3. *$\tau_i^u(p_i^*, w_i)p_i^*$ is a convex monotonically decreasing function of (f_i, w_i) for $W \ge w_i \ge 0$ and $F \ge f_i \ge f_i^l$.*

Proof. We provide the proof in the Appendix. □

Proof of Theorem 2. The convexity of constraints (27), (28) and of the objective function (24) follow from Lemma 2 and Lemma 3. The capacity constraints in (20) are convex and compact. This proves the convexity of the problem. □

Thus, if (24) – (28) is feasible then it can be solved in polynomial time, e.g., using interior point methods [36]. Using numerical solvers is, however, computationally not feasible if decisions are to be taken in real time. In what follows, we thus propose a closed-form approximate solution that can obtain a good solution at minimal computational effort.

Proposition 2. *Let $\mathcal{N}_j^o$ be the set of offloaders that execute application $j \in \mathcal{J}$, and let $L_{\phi_i} = L$. Assume that $Lw_i\Big(\frac{1}{f_i^l} - \frac{1}{f_i}\Big) \to \infty$ and $f_i > f_i^l$, and consider that the constraints (27) and (28) are not binding, corresponding to the high capacity case. Then the optimal solution is*

$$w_i^* = W\frac{\sqrt{H_i}}{\sum_{k \in \mathcal{N}_j^o}\sqrt{H_k}} \quad f_i^* = F\frac{f_i^l}{\sum_{k \in \mathcal{N}_j^o} f_k^l}, \forall i \in \mathcal{N}_j^o \tag{34}$$

where $H_i = \frac{\bar{\sigma}_i^2 D_i \beta_i \gamma_i \log(2)}{h_i}$.

Proof. Since the problem (24)-(28) is convex, any feasible allocation $\boldsymbol{\rho}^*$ that satisfies the *Karush-Kuhn-Tucker* (KKT) conditions will be optimal if Slater's condition holds. To obtain the KKT conditions, consider the Lagrangian dual [36]

$$\begin{aligned}\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\lambda}) = &\sum_{k=1}^{|\mathcal{N}_j^o|} \tau_k^u(p_k^*, w_k)p_k^*\beta_k\gamma_k + \lambda_1(\sum_{i \in \mathcal{N}_j^o} f_i - F) \\ &+ \lambda_2(\sum_{i \in \mathcal{N}_j^o} w_i - W) + \sum_{i=1}^{|\mathcal{N}_j^o|} \lambda_{i+2}(\tau_i^u(p_i^*, w_i)p_i^*\beta_i\gamma_i - C_i^0) \\ &+ \sum_{i=1}^{|\mathcal{N}_j^o|} \lambda_{i+|\mathcal{N}_j^o|+2}(p_i^* - \bar{p}_i),\end{aligned} \tag{35}$$

and denote by $\boldsymbol{\lambda}^*$ the KKT multipliers in the optimal solution. Recall from Lemma 3 that the objective function is monotonically decreasing in $(f_i, w_i), \forall i \in \mathcal{N}_j^o$, thereby the capacity constraints in (20) will always be binding in the optimal solution, i.e. $f_i^* = F - \sum_{i' \in \mathcal{N}_j^o \setminus \{i\}} f_{i'}^l$, $w_i^* = W - \sum_{i' \in \mathcal{N}_j^o \setminus \{i\}} w_{i'}$, $\lambda_1^*, \lambda_2^* > 0$. Since we consider the high capacity case where constraints (27) and (28) are not binding, the KKT multipliers $\lambda_k^* = 0$, for $k > 2$ in order to satisfy complementary slackness conditions. Next, we show that the stationary conditions can be expressed as

$$\frac{\partial \tau_i^u(p_i^*, w_i^*)p_i^*\beta_i\gamma_i}{\partial w_i} + \lambda_2^* = 0, \frac{\partial \tau_i^u(p_i^*, w_i^*)p_i^*\beta_i\gamma_i}{\partial f_i} + \lambda_1^* = 0. \tag{36}$$

Observe that the term A in (70) is approximately 1 in the high capacity case, and the expression $A - 1 - \frac{\log(2)A}{Lw_iK}$ is approximately $-(Lw_iK)^{-2}$ in (71). These approximations are indeed valid for $Lw_iK >> 1$ and $f_i > f_i^l$. We then obtain the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

expressions in (34), which satisfy the primal feasibility conditions, where we use the notation $H_i = \frac{\bar{\sigma}_i^2 D_i \log(2)\beta_i\gamma_i}{h_i}$. Observe that $\lambda_2^* = -\frac{\partial \tau_1^u(p_1^*, w_1^*)p_1^*}{\partial w_1} \geq 0$ and $\lambda_1^* = -\frac{\partial \tau_1^u(p_1^*, w_1^*)p_1^*}{\partial f_1} \geq 0$ by Lemma 3, hence satisfying dual feasibility. Finally, complementary slackness conditions are satisfied since $\lambda_1^* \geq 0, \lambda_2^* \geq 0$ and $\lambda_k^* = 0, k > 2$. Thus, we found the primal and dual optimal points $\boldsymbol{\rho}^*, \boldsymbol{\lambda}^*$ that satisfy the KKT conditions. □

Proposition 2 shows that under high capacity conditions a closed form approximate solution could be used to further decrease the computation time. Unfortunately, (24)−(28) need not be feasible in general, i.e., there may be a set $\mathcal{N}_{\mathcal{X}}^o$ such that some $i \in \mathcal{N}_{\mathcal{X}}^o$ cannot offload, in which case the optimal utility $U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o} = 0$. We thus turn to computing the optimal set of offloaders.

V. Choosing an Optimal Set of Offloaders

In this section we show how to choose the set of offloaders that maximizes the operator's utility for given caching decision, i.e., we address the problem

$$\max_{\mathcal{N}_{\mathcal{X}}^o \subseteq \mathcal{N}_{\mathcal{X}}} \max_{(\boldsymbol{\rho}_i)_{i\in\mathcal{N}_{\mathcal{X}}^o}} \sum_{i\in\mathcal{N}_{\mathcal{X}}^o} C_i^0 - \tau_i^u(p_i^*, w_i)p_i^*\beta_i\gamma_i, \tag{37}$$

$$s.t. \quad \sum_{i\in\mathcal{N}_{\mathcal{X}}^o} f_i \leq F, \sum_{i\in\mathcal{N}_{\mathcal{X}}^o} w_i \leq W, \tag{38}$$

$$f_i \geq f_i^l, w_i \geq 0, \forall i \in \mathcal{N}_{\mathcal{X}}^o, \tag{39}$$

$$p_i^* \leq \bar{p}_i, \forall i \in \mathcal{N}_{\mathcal{X}}^o, \tag{40}$$

$$\tau_i^u(p_i^*, w_i)p_i^*\beta_i\gamma_i \leq C_i^0, \forall i \in \mathcal{N}_{\mathcal{X}}^o. \tag{41}$$

Recall that for given $\mathcal{N}_{\mathcal{X}}^o$, the inner maximization problem is computable by Theorem 2 and Proposition 1. Thus the optimization problem (37)-(41) is a set function maximization problem over the ground set $\mathcal{N}_{\mathcal{X}}$, and can be equivalently written as,

$$U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^*} = \max_{\mathcal{N}_{\mathcal{X}}^o \subseteq \mathcal{N}_{\mathcal{X}}} U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o}. \tag{42}$$

A. Complexity Analysis

Our first result in this section shows that problem (37)-(41) is NP-hard.

Proposition 3. *Problem (37)-(41) is NP-hard.*

Proof. Before providing the proof we first introduce some notation. We write the optimal utility as a difference of the total local cost of the set of offloaders $\mathcal{N}_{\mathcal{X}}^o$ and the total energy consumption cost due to transmitting data to the edge server,

$$U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o} = \sum_{i\in\mathcal{N}_{\mathcal{X}}^o} C_i^0 - \sum_{i\in\mathcal{N}_{\mathcal{X}}^o} E_i^{\mathcal{N}_{\mathcal{X}}^o}, \tag{43}$$

where we denote by $E_i^{\mathcal{N}_{\mathcal{X}}^o} = \tau_i^u(p_i^*, w_i^*)p_i^*\beta_i\gamma_i$ the transmission energy consumption cost at the optimal resource allocation $(f_i^*, w_i^*) = \boldsymbol{\rho}_i^*(\mathcal{N}_{\mathcal{X}}^o)$, and we denote by $E(\mathcal{N}_{\mathcal{X}}^o) = \sum_{i\in\mathcal{N}_{\mathcal{X}}^o} E_i^{\mathcal{N}_{\mathcal{X}}^o}$ the total transmission energy cost for the set $\mathcal{N}_{\mathcal{X}}^o$ of offloaders.

In what follows we prove the result through reduction from the partition problem, which is known to be NP-hard.

Problem 1 (Partition Problem). *Given positive integers $b_1, b_2, \ldots, b_k$ is there a vector $z = [z_1, z_2, \ldots, z_k]$ with $z_i = \{0, 1\}, \forall i,\ 1 \leq i \leq k$ such that $\sum_{i=1}^k b_i z_i = A$ where $\sum_{i=1}^k b_i = 2A$?*

Given an instance of the partition problem, we let $N_{\mathcal{X}} = k$, and let the corresponding set of WDs be $\mathcal{N}_{\mathcal{X}}$. Let us set $f_i^l = b_i$, $L_{\phi_i} = D_i = \bar{\sigma}_i^2 = h_i = \gamma_i = \beta_i = \kappa_i^l = 1$, this implies $C_i^0 = b_i$. We set $F = A + 0.99$, which allows a set of offloaders $\mathcal{N}_{\mathcal{X}}^o$ with $\sum_{i\in\mathcal{N}_{\mathcal{X}}^o} f_i^l \leq A + 0.99$. Since b_i's are positive integers, the set of offloaders have at most $\sum_{i\in\mathcal{N}_{\mathcal{X}}^o} f_i^l = A$. From (34) we calculate $w_i^* = \frac{W}{|\mathcal{N}_{\mathcal{X}}^o|}$ for any $\mathcal{N}_{\mathcal{X}}^o \subseteq \mathcal{N}_{\mathcal{X}}$. Similarly, from (34), $f_i^* = \frac{(A+0.99)b_i}{\sum_{i\in\mathcal{N}_{\mathcal{X}}^o} b_i}$. For the reduction to work, we need to ensure that the total energy consumption $E(\mathcal{N}_{\mathcal{X}}^o) < 1$ for any feasible $\mathcal{N}_{\mathcal{X}}^o$. With this, operator would always choose $\mathcal{N}_{\mathcal{X}}^o$ with maximal $\sum_{i\in\mathcal{N}_{\mathcal{X}}^o} C_i^0$ from (43) since $b_i \geq 1$ by the partition problem up to $\sum_{i\in\mathcal{N}_{\mathcal{X}}^o} f_i^l = A$. We first set a suboptimal resource allocation for the WDs and satisfy $E(\mathcal{N}_{\mathcal{X}}^o) < 1$. If a suboptimal allocation satisfies this inequality, so does the optimal one. We set $w_i' = \frac{W}{k} \leq w_i^*$ and $f_i' = \frac{(A+0.99)b_i}{A} \leq f_i^*$ since $\sum_{i\in\mathcal{N}_{\mathcal{X}}^o} b_i \leq A$ in any feasible solution of any $\mathcal{N}_{\mathcal{X}}^o \subseteq \mathcal{N}_{\mathcal{X}}$. Next, we set the transmission power as

$$p_i' = (2^{\frac{1}{w_i'(\frac{1}{b_i} - \frac{1}{f_i'})}} - 1),\ \forall i \in \mathcal{N}_{\mathcal{X}}, \tag{44}$$

thus the transmission energy cost becomes

$$E_i' = (\frac{1}{b_i} - \frac{1}{f_i'})(2^{\frac{1}{w_i'(\frac{1}{b_i} - \frac{1}{f_i'})}} - 1). \tag{45}$$

Assume a hypothetical case that all $b_i = 1$ and $k = 2A$. Then there has to be a set $\mathcal{N}_{\mathcal{X}}^o$ with $\sum_{i\in\mathcal{N}_{\mathcal{X}}^o} C_i^0 = A$. In this scenario, f_i' will be the lowest by its definition, thus the energy consumption cost will be the highest from Lemma 3. Observe that the only unknown variable in (45) is W, thus one can find W such that $E'(\mathcal{N}_{\mathcal{X}}^\dagger) = \sum_{i\in\mathcal{N}_{\mathcal{X}}^\dagger} E_i' < 1$ and $|\mathcal{N}_{\mathcal{X}}^\dagger| = A$. This ensures that any set of offloaders in the $\mathcal{N}_{\mathcal{X}}^o$ has $E'(\mathcal{N}_{\mathcal{X}}^o) < 1$. After setting W, we calculate p_i', $\forall i \in \mathcal{N}_{\mathcal{X}}$ and set $\bar{p} > \max_i p_i'$. Observe that by construction, if the answer to the partition problem is YES, then the solution set of our problem is $\mathcal{N}_{\mathcal{X}}^*$ and gives $\sum_{i\in\mathcal{N}_{\mathcal{X}}^*} b_i = A$ and $z_i^* = b_i$ such that $i \in \mathcal{N}_{\mathcal{X}}^*$, if the answer is NO then our problem has solution $\sum_{i\in\mathcal{N}_{\mathcal{X}}^*} b_i < A$. This concludes the proof.

□

The NP-hardness of the problem implies that an optimal set of offloaders cannot be computed efficiently. Thus, we are interested in designing an approximation algorithms that can compute a near optimal solution efficiently.

B. Singleton Greedy Maximization

Before we describe our proposed algorithm, let us recall the definition of monotonicity and submodularity of set functions. These two properties of set functions are widely relied upon in the design of approximation algorithms.

Definition 2 (Monotonicity). *Let Ω be a finite set and $V : \Omega \to \mathbb{R}$ a set function. V is monotone if for any $\Omega^\dagger \subset \Omega$ and*

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

$i \in \Omega \setminus \Omega^\dagger$ we have $V(\Omega^\dagger \cup \{i\}) \geq V(\Omega^\dagger)$. That is, adding a new element to any feasible input of the function does not decrease its value.

Definition 3 (Submodularity)**.** *Let Ω be a finite set. The set function $V : 2^\Omega \to \mathbb{R}$, where 2^Ω denotes the power set of Ω, is submodular if for every $\Omega^\dagger \subseteq \Omega'$ and $\omega \in \Omega \setminus \Omega'$ it satisfies*

$$V(\Omega^\dagger \cup \{\omega\}) - V(\Omega^\dagger) \geq V(\Omega' \cup \{\omega\}) - V(\Omega'). \tag{46}$$

Monotonicity and submodularity are known to allow efficient approximation algorithms [37], but the utility of the operator is neither monotone (see Proposition 6 in the Appendix) nor submodular in general (see Proposition 7 in the Appendix). Nonetheless, as we show next, the operator's utility is submodular in the high capacity region.

Lemma 4. *Under high capacity conditions, the utility function $U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o}$ is submodular, and can be expressed as*

$$U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o} = \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} C_i^0 - \tau_i^u\Big(p_i^*, \frac{W\sqrt{H_i}}{\sum_{j \in \mathcal{N}_{\mathcal{X}}^o} \sqrt{H_j}}\Big) p_i^* \beta_i \gamma_i, \tag{47}$$

where $p_i^* = \Big(2^{\frac{\sum_{j \in \mathcal{N}_{\mathcal{X}}^o} \sqrt{H_j}}{LW\sqrt{H_i}(\frac{1}{f_i^l} - \frac{\sum_{j \in \mathcal{N}_{\mathcal{X}}^o} f_j^l}{F f_i^l})}} - 1\Big) \frac{\bar{\sigma}_i^2}{h_i}$.

Proof. As shown in Proposition 1, for all $i \in \mathcal{N}_{\mathcal{X}}^o$, the optimal price is $\pi_i^* = C_i^0 - \tau_i^u(p_i^*, w_i^*) p_i^* \beta_i \gamma_i$, the optimal utility will be

$$U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o} = \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} C_i^0 - \tau_i^u(p_i^*, w_i^*) p_i^* \beta_i \gamma_i = \sum_{i \in \mathcal{N}_{\mathcal{X}}^o} C_i^0 - \\ LD_i\Big(\frac{1}{f_i^l} - \frac{1}{f_i^*}\Big)\Big(2^{(Lw_i^*(\frac{1}{f_i^l} - \frac{1}{f_i^*}))^{-1}} - 1\Big) \frac{\bar{\sigma}_i^2 \beta_i \gamma_i}{h_i}. \tag{48}$$

To show submodularity we need to show that

$$U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o \cup \{i'\}} + U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o \cup \{j\}} \geq U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o \cup \{i', j\}} + U_{\mathcal{X}}^{\mathcal{N}_{\mathcal{X}}^o}. \tag{49}$$

We show (49) holds, by showing that inequality holds for the each individual WD using (48). For WD i' and j local cost at the both sides of the inequality (49) cancel out. Observe from (34) that w_i^* and f_i^* decreases as number of offloaders increases, hence, for WD i' and WD j, $E_{i'}^{\mathcal{N}_{\mathcal{X}}^o \cup i'} \leq E_{i'}^{\mathcal{N}_{\mathcal{X}}^o \cup \{i', j\}}$ and $E_j^{\mathcal{N}_{\mathcal{X}}^o \cup i'} \leq E_j^{\mathcal{N}_{\mathcal{X}}^o \cup \{i', j\}}$ hold since transmission energy cost is monotonically decreasing function of (f_i, w_i) from Lemma 3. Thus for only WD i' and WD j inequality (49) holds.

We next define $F^l = \sum_{k \in \mathcal{N}_{\mathcal{X}}^o} f_k^l$, and $W^H = \sum_{k \in \mathcal{N}_{\mathcal{X}}^o} \sqrt{H_k}$ for notational simplicity. Next, we need to show that the inequality (49) holds for any WD $i \in \mathcal{N}_{\mathcal{X}}^o$. For a WD i, (49) can be expressed as

$$\Big(F - (F^l + f_{i'}^l)\Big)\Big(2^{\frac{F f_i^l (W^H + \sqrt{H_{i'}})}{LW\sqrt{H_i}(F - (F^l + f_{i'}^l))}}\Big) + \\ \Big(F - (F^l + f_j^l)\Big)\Big(2^{\frac{F f_i^l (W^H + \sqrt{H_j})}{LW\sqrt{H_i}(F - (F^l + f_j^l))}}\Big) \\ \leq \Big(F - (F^l + f_j^l + f_{i'}^l)\Big)\Big(2^{\frac{F f_i^l (W^H + \sqrt{H_j} + \sqrt{H_{i'}})}{LW\sqrt{H_i}(F - (F^l + f_j^l + f_{i'}^l))}}\Big) + \\ \Big(F - F^l\Big)\Big(2^{\frac{F f_i^l W^H}{LW\sqrt{H_i}(F - F^l)}}\Big), \tag{50}$$

after substituting f_i^*, w_i^* from (34) and applying algebraic manipulations.

Next we define $g(x) = (F - x) 2^{\frac{F f_i^l W^H}{LW\sqrt{H_i}(F - x)}}$, which is convex for $x \leq F$. By the definition of convexity we know that

$$g(x' + z) + g(x' + y) \leq g(x' + y + z) + g(x'), x', y, z > 0, \tag{51}$$

thus

$$g(F^l + f_{i'}^l) + g(F^l + f_j^l) \leq g(F^l + f_{i'}^l + f_j^l) + g(F^l), \tag{52}$$

holds. To conclude the proof, let us define the functions $v(x, y) = (F - x) 2^{\frac{F f_i^l (W^H + y)}{LW\sqrt{H_i}(F - x)}}$ and,

$$\chi(y, z) = v(F^l + f_{i'}^l + f_j^l, y + z) + \\ v(F^l, 0) - v(F^l + f_{i'}^l, y) - v(F^l + f_j^l, z). \tag{53}$$

Notice that showing $\chi(y, z) \geq 0$ is equivalent to showing (50) holds. Thus if we show that $\frac{\partial \chi(y,z)}{\partial y} \geq 0$ and $\frac{\partial \chi(y,z)}{\partial z} \geq 0$ for any $y, z \geq 0$, this would imply that $\chi(y, z) \geq 0$ since $\chi(0, 0) \geq 0$ from (52). The partial derivatives are

$$\frac{\partial \chi(y, z)}{\partial y} = \frac{F f_i^l \log(2)}{LW\sqrt{H_i}} \Big(2^{\frac{F f_i^l (W^H + y + z)}{LW\sqrt{H_i}(F - (F^l + f_{i'}^l + f_j^l))}} \\ - 2^{\frac{F f_i^l (W^H + y)}{LW\sqrt{H_i}(F - (F^l + f_{i'}^l))}}\Big), \tag{54}$$

$$\frac{\partial \chi(y, z)}{\partial z} = \frac{F f_i^l \log(2)}{LW\sqrt{H_i}} \Big(2^{\frac{F f_i^l (W^H + y + z)}{LW\sqrt{H_i}(F - (F^l + f_{i'}^l + f_j^l))}} \\ - 2^{\frac{F f_i^l (W^H + z)}{LW\sqrt{H_i}(F - (F^l + f_j^l))}}\Big). \tag{55}$$

Observe that $\frac{\partial \chi(y,z)}{\partial y} \geq 0, \frac{\partial \chi(y,z)}{\partial z} \geq 0$ for any $y, z \geq 0$, hence (50) holds for any $i \in \mathcal{N}_{\mathcal{X}}^o$. We already showed that inequality holds for WD i' and j. Thus, (49) holds as well. This concludes the proof. □

Lemma 4 shows that the utility function is submodular and monotone under certain conditions, and thus existing approximation algorithms for monotone submodular functions can guarantee an approximation ratio bound of $\frac{1}{2}$, e.g., by always adding an element based on marginal gain [38] with $\mathcal{O}(\mathcal{N}_{\mathcal{X}}^2)$ time complexity. In what follows we propose an approximation algorithm with lower computational complexity called *Singleton Greedy Maximization* (SGM). The algorithm greedily adds WD i^* to the set of offloaders with the highest singleton utility, i.e., obtained when only WD i^* offloads. Then since the utility is not monotone with respect to set of offloaders, the algorithm checks for the increase of the utility after the addition of WD i^* in Line 4. It then removes the WD i^* from the ground set and keeps iterating until the ground set becomes empty. A flow chart of the proposed SGM algorithm is shown in Fig. 2. The flow chart also marks the steps in the proposed algorithm that have been made possible by our analytical results.

SGM is computationally very efficient, and at the same time it allows an approximation ratio bound. As we show in Proposition 6 and Proposition 7, the considered problem is neither submodular nor monotone in general. An approximation

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

Algorithm 1 SGM

```
Require: $\mathcal{X}, \mathcal{N}_\mathcal{X}$ return $U_\mathcal{X}^{SGM}, \mathcal{N}_\mathcal{X}^{SGM}$
1: $\mathcal{N}_\mathcal{X}^{SGM} = \emptyset$
2: for $1 : N_\mathcal{X}$ do
3:     $i^* = \text{argmax}_{i \in \mathcal{N}_\mathcal{X}} U_\mathcal{X}^i$
4:     $\mathcal{N}_\mathcal{X}^{SGM} = \text{argmax}_{\mathcal{N}_\mathcal{X}^o \in \{\mathcal{N}_\mathcal{X}^{SGM}, \mathcal{N}_\mathcal{X}^{SGM} \cup \{i^*\}\}} U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^o}$
5:     $\mathcal{N}_\mathcal{X} = \mathcal{N}_\mathcal{X} \setminus \{i^*\}$
6: end for
7: $U_\mathcal{X}^{SGM} = U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^{SGM}}$
```

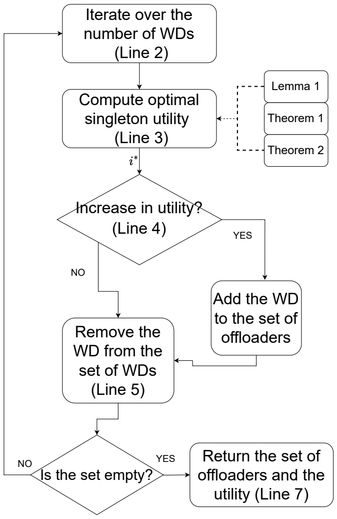
Iterate over the number of WDs (Line 2)
Compute optimal singleton utility (Line 3)
Lemma 1
Theorem 1
Theorem 2
i^*
Increase in utility? (Line 4)
YES
NO
Add the WD to the set of offloaders
Remove the WD from the set of WDs (Line 5)
NO
Is the set empty?
YES
Return the set of offloaders and the utility (Line 7)

Fig. 2: Flow chart of the proposed SGM algorithm.

ratio bound for this kind of set maximization problems is not known in general. Nonetheless, in what follows, we develop an approximation ratio bound based on two properties of the total transmission energy cost function $E(.)$, introduced in the next two lemmas.

Lemma 5. *Let $\mathcal{N}_\mathcal{X}^o$ be such that Problem (24) − (28) has a feasible solution. Then*

$$E(\mathcal{N}_\mathcal{X}^o) \geq \sum_{i \in \mathcal{N}_\mathcal{X}^o} E(\{i\}). \tag{56}$$

Proof. Observe that for any $\mathcal{N}_\mathcal{X}^o \ni i$ the transmission energy consumption cost $E_i^{\mathcal{N}_\mathcal{X}^o}$ is minimal when $\mathcal{N}_\mathcal{X}^o = \{i\}$, since in this case the optimal solution for the operator is to allocate $f_i^* = F, w_i^* = W$, i.e. WD i gets the full capacity. The statement then follows from $E(\mathcal{N}_\mathcal{X}^o) = \sum_{i \in \mathcal{N}_\mathcal{X}^o} E_i^{\mathcal{N}_\mathcal{X}^o}$, which concludes the proof. □

Lemma 6. *Let $\mathcal{N}_\mathcal{X}^o \subseteq \mathcal{N}_\mathcal{X}$, then*

$$U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^o} \leq \sum_{i \in \mathcal{N}_\mathcal{X}^o} U_\mathcal{X}^i. \tag{57}$$

Proof. We will first use the decomposition of the utility in the form of (43). Then we write out (57) as,

$$\sum_{i \in \mathcal{N}_\mathcal{X}^o} C_i^0 - \sum_{i \in \mathcal{N}_\mathcal{X}^o} E_i^{\mathcal{N}_\mathcal{X}^o} \leq \sum_{i \in \mathcal{N}_\mathcal{X}^o} C_i^0 - \sum_{i \in \mathcal{N}_\mathcal{X}^o} E(i), \tag{58}$$

$$\sum_{i \in \mathcal{N}_\mathcal{X}^o} E(i) \leq \sum_{i \in \mathcal{N}_\mathcal{X}^o} E_i^{\mathcal{N}_\mathcal{X}^o} = E(\mathcal{N}_\mathcal{X}^o), \tag{59}$$

holds from Lemma 5 if $\mathcal{N}_\mathcal{X}^o$ admits a feasible solution. If $\mathcal{N}_\mathcal{X}^o$ does not admit a feasible solution then $U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^o} = 0$, and (57) holds trivially since $U_\mathcal{X}^i \geq 0, \forall i \in \mathcal{N}_\mathcal{X}$. This concludes the proof. □

Lemma 6 shows the intuitive result that the sum of the utility of WDs offloading individually is higher than when all WDs offload simultaneously. Based on this result, we are now ready to derive a bound on the approximation ratio of SGM.

Theorem 3. *Let $\mathcal{N}_\mathcal{X}^*$ be the optimal set of offloaders for given caching decision $\mathcal{X}$, and let $\mathcal{N}_\mathcal{X}^{SGM}$ be the set of offloaders computed by SGM. Then, $\frac{U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^*}}{N_\mathcal{X}^*} \leq U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^{SGM}}$, i.e.; SGM is a $\frac{1}{N_\mathcal{X}^*}$-approximation algorithm.*

Proof. By using Lemma 6 we write the upper bound of the optimal utility,

$$U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^*} \leq \sum_{i \in \mathcal{N}_\mathcal{X}^*} U_\mathcal{X}^i \leq N_\mathcal{X}^* U_\mathcal{X}^{i^*}, \tag{60}$$

$$\frac{U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^*}}{N_\mathcal{X}^*} \leq U_\mathcal{X}^{i^*} \leq U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^{SGM}}, \tag{61}$$

where $i^* = \text{argmax}_{i \in \mathcal{N}_\mathcal{X}} U_\mathcal{X}^i$. In the first iteration of the algorithm, the WD with maximal singleton utility i.e., i^*, is chosen by the algorithm. Thus, $i^* \in \mathcal{N}_\mathcal{X}^{SGM}$ provided that there is $i \in \mathcal{N}_\mathcal{X}$ such that $U_\mathcal{X}^i > 0$. In the rest of the iterations if the algorithm chooses a new WD $j \neq i^*$ that implies $U_\mathcal{X}^{i^*} \leq U_\mathcal{X}^{i^*,j}$ thanks to Line 4. Thus, justifies (61). This concludes the proof. □

Importantly, Theorem 3 provides a bound on the worst case performance of the proposed SGM algorithm.

VI. Optimal Caching Policy

In this section, we address the problem of choosing an optimal set of applications to cache, i.e.,

$$\max_{\mathcal{X} \subseteq \mathcal{J}} \quad U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^*} \quad s.t. \quad \sum_{j \in \mathcal{X}} s_j \leq S. \tag{62}$$

Choosing an optimal set of applications is NP-hard, a result that can be shown using the same approach as in Proposition 3 by setting $\phi_i \notin \cup_{i' \in \mathcal{N}_\mathcal{X} \setminus \{i\}} \phi_{i'}, \forall i \in \mathcal{N}_\mathcal{X}$ i.e., all WDs want to execute different applications, and by setting $s_i = b_i$, and setting F and W high enough such that $E(\mathcal{N}) < 1$. We next show that despite the non-monotonicity of the utility function with respect to the addition of new WDs to the set of offloaders, the utility of the operator is a monotone function with respect to the addition of new applications to the cached set.

Proposition 4. *Let $\mathcal{X} \subseteq \mathcal{J}$ and $j \in \mathcal{J} \setminus \mathcal{X}$. Then*

$$U_{\mathcal{X} \cup \{j\}}^{\mathcal{N}_{\mathcal{X} \cup \{j\}}^*} \geq U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^*}. \tag{63}$$

Proof. We will prove the statement by contradiction. Let $U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^*}$ be the optimal utility for caching decision $\mathcal{X}$. Let $U_{\mathcal{X} \cup \{x\}}^{\mathcal{N}_{\mathcal{X} \cup \{x\}}^*}$ be the optimal utility for caching decision $\mathcal{X} \cup \{x\}$. Assume that $U_\mathcal{X}^{\mathcal{N}_\mathcal{X}^*} > U_{\mathcal{X} \cup \{x\}}^{\mathcal{N}_{\mathcal{X} \cup \{x\}}^*}$, thus $\mathcal{N}_\mathcal{X}^* \neq \mathcal{N}_{\mathcal{X} \cup \{x\}}^*$. It is clear that

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

Algorithm 2 SRM

```
Require: $\mathcal{J}, \mathcal{N}$ return $U^{SRM}, \mathcal{N}^*_{\mathcal{X}}$
1: $\mathcal{X} = \emptyset$
2: while $\sum_{j\in\mathcal{X}} s_j \leq S \wedge \mathcal{J} \neq \emptyset$ do
3:     $j^* = \text{argmax}_{j\in\mathcal{J}} \frac{U_j^{SGM}}{s_j}$
4:     if $S \geq \sum_{j\in\mathcal{X}\cup\{j^*\}} s_j$ then
5:         $\mathcal{X} = \mathcal{X} \cup \{j^*\}$
6:     end if
7:     $\mathcal{J} = \mathcal{J} \setminus \{j^*\}$
8: end while
9: $U^{SRM} = U_{\mathcal{X}}^{SGM}$                ▷ Compute using SGM
```

$\mathcal{N}^*_{\mathcal{X}} \subseteq \mathcal{N}_{\mathcal{X}} \subseteq \mathcal{N}_{\mathcal{X}\cup\{x\}}$. If $\mathcal{N}^*_{\mathcal{X}}$ gives higher utility compared to $\mathcal{N}^*_{\mathcal{X}\cup\{x\}}$, then the operator would choose $\mathcal{N}^*_{\mathcal{X}}$ instead of $\mathcal{N}^*_{\mathcal{X}\cup\{x\}}$ when set $\mathcal{X}\cup\{x\}$ is cached. Hence, $U_{\mathcal{X}\cup\{x\}}^{\mathcal{N}^*_{\mathcal{X}\cup\{x\}}}$ cannot be optimal, which is a contradiction. This concludes the proof. □

At the same time, the operator's utility with respect to the addition of a new application need not be submodular in general.

Proposition 5. *Let $\mathcal{X} \subseteq \mathcal{X}'$ and $j \in \mathcal{J} \setminus \mathcal{X}'$, then*

$$U^*_{\mathcal{X}\cup\{j\}} - U^*_{\mathcal{X}} \lesseqgtr U^*_{\mathcal{X}'\cup\{j\}} - U^*_{\mathcal{X}'}, \tag{64}$$

i.e., the optimal utility $U^*_{\mathcal{X}}$ need not be submodular with respect to the set of cached applications.

Proof. The proof is based on a counterexample and is given in the Appendix. □

Thus, the problem (62) is a monotone non-submodular set function maximization problem subject to a knapsack constraint, imposed by the cache capacity constraint. Recently proposed solutions for such problems provide approximation guarantees, but at the price of high computational cost [39], we thus propose a fast heuristic called *Singleton Revenue Maximization* (SRM), which uses SGM for pricing and resource allocation. The algorithm first calculates the utility of each individual application $j \in \mathcal{J}$ using SGM. Then, it selects the application j^* with the highest utility to storage size ratio (Line 3) and adds it to the to caching decision set if storage capacity allows (Line 4-6). The algorithm then removes the application j^* from the ground set $\mathcal{J}$ (Line 7). The algorithm stops when all applications have been considered or if the storage capacity is exceeded. Fig. 3 shows the flow chart of the proposed SRM algorithm, including the interaction between SGM and SRM.

VII. Numerical Results

We used extensive simulations to evaluate the performance of the proposed algorithm in terms of operator utility, simulation time, total energy saving and consumption through task offloading, the number of offloaders in SPE and provide a sensitivity analysis of the proposed algorithm to faults in wireless communication.

For the evaluation we consider a system with up to $N = 200$ WDs, and up to $|\mathcal{J}| = 50$ applications. The storage capacity is $S = 10$ GB, and the computational complexity L_j is drawn from a uniform distribution on $[100, 500]$ I/B and size of the application s_j is drawn from a uniform distribution on $[1, 2.5]$ GB. The computational capacity of the edge server is $F = 200$ GIPS, and total channel bandwidth $W = 200$ MHz. The task type ϕ_i of WD i is chosen uniform at random from the set $\mathcal{J}$. The maximum transmission power $\bar{p}_i$ is drawn from a uniform distribution on $[100, 1000]$ mW, f_i^l is drawn from a uniform distribution on $[0.5, 3]$ GIPS, D_i is drawn from a uniform distribution on $[5, 50]$ MB. The channel noise variance $\bar{\sigma}_i^2$ and the channel gain h_i is uniformly distributed on $[0.1, 1]$ and $[0.1, 1]$, respectively. The energy efficiency parameter κ_i^l and the unit energy cost parameter β_i are drawn from a uniform distribution on $[10^{-22}, 10^{-19}]$ J/Hz/GI^2 , and on $[10^{-3}, 1]$, respectively. We set $\gamma_i = 0.1$ \$/J, $\forall i \in \mathcal{N}$. These choices of parameters are similar to those used in works [17], [32]. The results shown are the averages of at least 200 simulations, together with 95% confidence intervals, which are within 1% of the mean. The simulations were conducted using Matlab on a desktop computer with Intel i9 and on a server with AMD EPYC 7543P CPU.

We consider five baselines for the evaluation. The first baseline is exhaustive search over the set of offloaders and computes the optimal resource allocation and pricing. The second baseline is called *Random Greedy Selection* (RGS); it randomly chooses a set of WDs, calculates the optimal

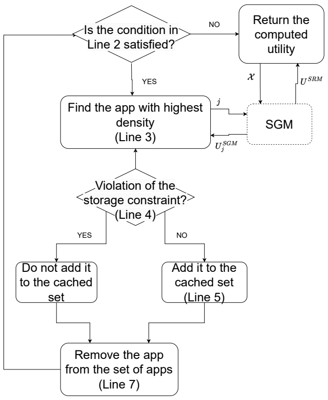
Is the condition in Line 2 satisfied?
NO
Return the computed utility
YES
$\mathcal{X}$
U^{SRM}
Find the app with highest density (Line 3)
j
SGM
U_j^{SGM}
Violation of the storage constraint? (Line 4)
YES
NO
Do not add it to the cached set
Add it to the cached set (Line 5)
Remove the app from the set of apps (Line 7)

Fig. 3: Flow chart of the proposed SRM algorithm.

L_j	$Unif(100, 500)$ I/B
s_j	$Unif(1, 2.5)$ GB
F	200 GIPS
W	200 MHz
S	10 GB
$\bar{p}_i$	$Unif(100, 1000)$ mW
f_i^l	$Unif(0.5, 3)$ GIPS
D_i	$Unif(5, 50)$ MB
$\bar{\sigma}_i^2$	$Unif(0.1, 1)$
h_i	$Unif(0.1, 1)$
κ_i^l	$Unif(10^{-22}, 10^{-19})$ J/Hz/GI^2
β_i	$Unif(10^{-3}, 1)$
γ_i	0.1 \$/J

TABLE II: Overview of the simulation parameters.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

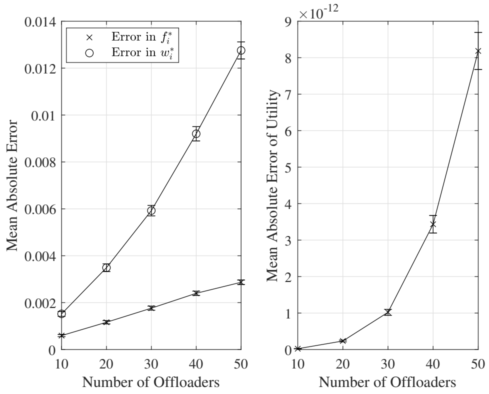

Fig. 4: Relative error in resource allocation vector and utility for 200 randomly generated problem instances, obtained using a numerical method and using the proposed approximation.

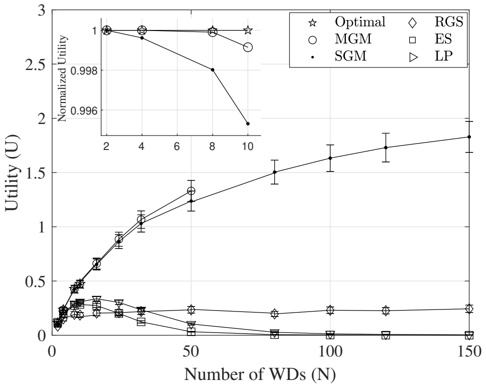

Fig. 5: Utility vs. number of WDs, $|\mathcal{J}| = 1$.

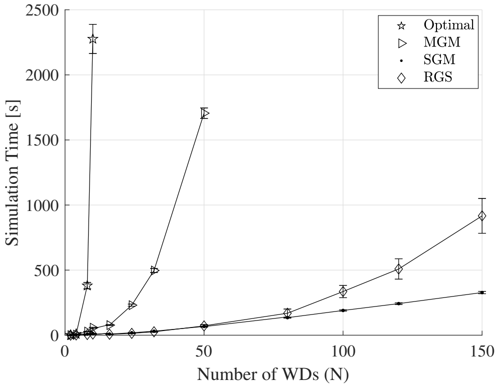

Fig. 6: Simulation time [s] vs. number of WDs, $|\mathcal{J}| = 1$.

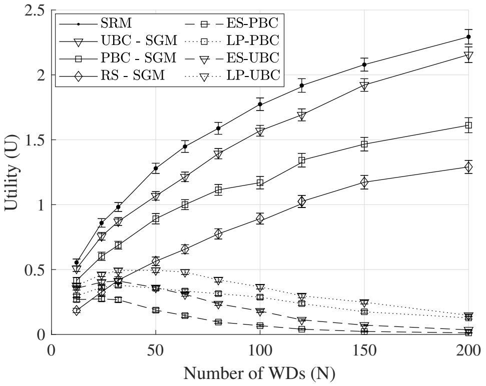

Fig. 7: Utility vs. number of WDs, $|\mathcal{J}| = 20$.

utility and returns it if it is positive, else chooses another set of WDs. The third baseline is called *Marginal Greedy Maximization* (MGM), and is based on the greedy algorithm proposed in [40]. MGM computes the marginal utility of each WD, and adds the WD with the highest marginal gain to the set of offloaders if doing so increases the utility. The last two baselines are approaches that are widely used in previous works [10], [32]. The first, *Equal Sharing* (ES) allocates resources equally among offloading WDs. The second, *Load Proportional* (LP) allocation, allocates resources to all WDs proportionally to their task complexity $L_{\phi_i} D_i$.

For computing the optimal set of cached applications we use three baselines. The first baseline is *Popularity Based Caching* (PBC), which selects the set of applications with the highest number of WDs $|\{\phi_i \in \mathcal{X}|\}|$ while satisfying the storage constraint. The second baseline is *Utility Based Caching* (UBC), which chooses the set of applications with highest $\sum_{i \in \mathcal{N}_{\mathcal{X}}} L_{\phi_i} D_i$. The third baseline is *Random Selection* (RS), which chooses a random set of applications satisfying the storage constraint.

A. Validation of the Approximate Resource Allocation

We first evaluate the accuracy of the proposed approximation in (34). Fig. 4 shows the mean absolute relative error of the resource allocation vector and of the utility computed using the proposed approximation and using the interior point method. To create high capacity conditions, we set W to 2 GHz, $\bar{p}_i$ is drawn uniformly from $[1, 2]$ W, β_i is drawn uniformly from $[10^{-8}, 10^{-6}]$, L_{ϕ_i} is drawn uniformly from $[1500, 2000]$ and κ_i^l is drawn uniformly from $[10^{-20}, 10^{-19}]$ J/Hz/GI^2 so that the constraints (27) and (28) are not binding for all WDs. The figure shows that under these conditions the approximation is very accurate.

B. Choosing Optimal Set of Offloaders

Fig. 5 and Fig. 6 show the utility as a function of the number of WDs and the simulation time as a function of number of the WDs for a single cached application, respectively. The figures show that for a small number of WDs ($N \leq 10$), the utilized approximation algorithms for joint pricing and resource management, namely MGM and SGM, performs close to the optimal solution with a much lower simulation time. In contrast, for $N > 10$, the proposed SGM performs close to MGM at a much lower computational cost. For high number of users, MGM becomes practically infeasible as the operator would have to solve problem (13)-(14) in real time. This highlights the significance of the proposed SGM algorithm. As the number of WDs increases the only algorithm that provides high utility at low computational cost is SGM. It is interesting to note that RGS is computationally

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

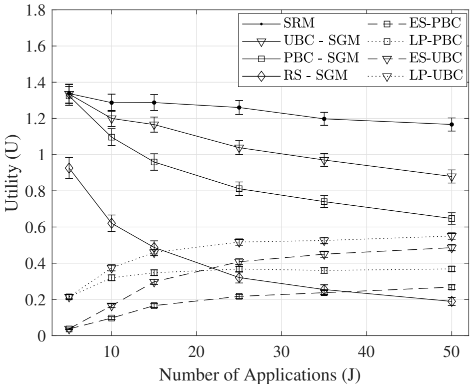

Fig. 8: Utility vs. number of applications, $N = 50$.

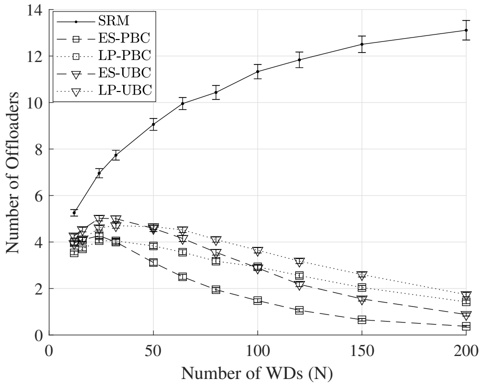

Fig. 9: Number of offloaders vs. number of WDs, $|\mathcal{J}| = 20$.

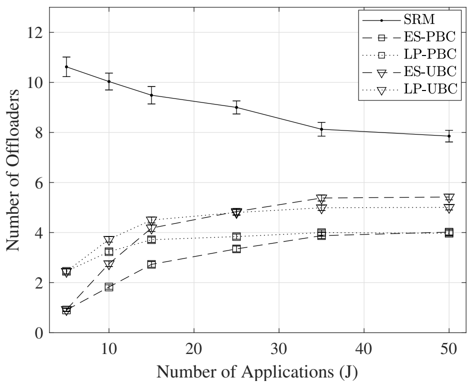

Fig. 10: Number of offloaders vs. number of applications, $N = 50$.

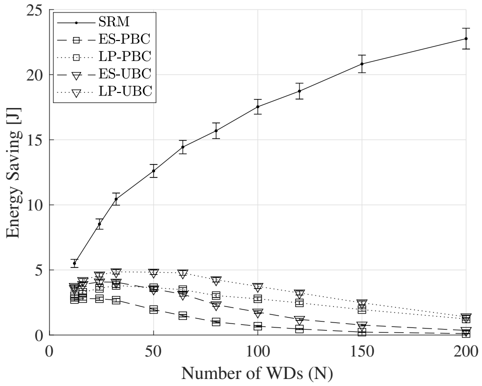

Fig. 11: Energy saving vs. number of WDs, $|\mathcal{J}| = 20$.

more expensive than SGM, as finding a set of offloaders that gives a positive utility becomes harder as the number of WDs increase. As RGS could be considered as one of the most light weight solution, SGM can still be computed faster and the computation time difference increases above 80 WDs, which shows that our proposed approach strikes a good balance time complexity and the achieved utility. Lastly, the poor performance of ES and LP justifies the importance of joint optimization of resource allocation and pricing for maximizing the utility.

C. Operator's Profit

Fig. 7 shows the operator's utility as a function of the number of WDs for $|\mathcal{J}| = 20$. The figure shows that SRM, which jointly optimizes caching, resource allocation and pricing, outperforms ES-UBC, LP-UBC and ES-PBC, LP-PBC, by up to an order of magnitude particularly for high number of WDs. More importantly, SRM outperforms UBC-SGM and PBC-SGM, i.e. the baselines that use the proposed SGM to compute optimal resource allocation and pricing, showing that joint caching and pricing provides significant benefits compared to pricing-unaware caching.

Fig. 8 shows the utility as a function of the number of applications for $N = 50$. The utility obtained by the algorithms that use *SGM* for pricing and resource allocation decreases monotonically in J, as the number of WDs per application decreases. On the contrary, the utility obtained by the algorithms that use *ES* or *LP* for resource allocation increases, as for a few WDs per application they get closer to the optimal allocation. Importantly, the proposed SRM algorithm is almost insensitive to the increase in the number of applications and its performance advantage increases as with the number of applications.

D. Energy Optimal Resource Allocation, Pricing and Number of Offloaders in SPE

Fig. 9 and Fig. 10 show the number of offloaders as a function of number of WDs and number of applications, respectively. The figures show the superior utility of SRM compared to the baselines is correlated with that it allows more WDs to offload, owing to that it computes the optimal resource allocation and pricing. Finally, Fig. 11 and Fig. 12 show the total energy saving, defined as $\sum_{i\in\mathcal{N}^o_{\mathcal{X}}} \frac{C_i^0}{\gamma_i} - \tau_i^u(p_i^*, w_i)p_i^*\beta_i$, through task offloading as a function of the number of WDs and as a function of the number of applications. The figures show that SRM achieves the highest total energy saving. Consequently, the objective of the operator to maximize its utility combined with the objective of the WDs to minimize their cost leads to an energy efficient solution for WDs.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

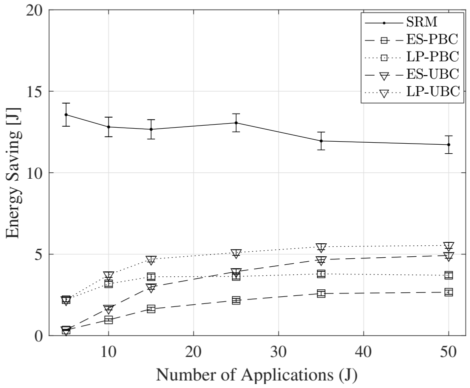

Fig. 12: Energy saving vs. number of applications, $N = 50$.

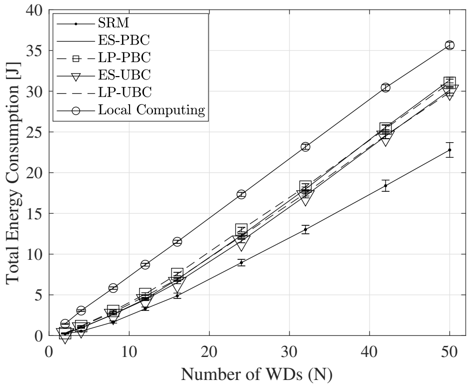

Fig. 13: Total energy consumption vs. number of WDs, $|\mathcal{J}| = 20$.

E. Energy Consumption of the WDs

We also evaluate the energy efficiency of the proposed solution from the WDs' perspective. Fig. 13 shows the total energy cost of the WDs as a function of the number of WDs. The results in the figure are aligned with those in Fig. 11, showing that the operator's optimal strategy for maximizing utility indeed leads to lower energy cost for the WDs. The figure also shows that as the number of WDs increases, the difference in terms of energy consumption between SRM and the baselines increases, showing the superiority of the proposed approach.

Fig. 14 shows the empirical CDF of the energy consumption of the WDs for $N = 20$ and $N = 40$, with $|\mathcal{J}| = 20$, across 500 randomly generated problem instances. Aligned with the results shown in Fig. 13, the proposed SRM outperforms the state-of-the-art methods, and the performance difference becomes more pronounced as the number of WDs increases (c.f. the subfigures in Fig. 14). The figure also shows that compared to local computing, SRM decreases the 99^{th} percentile of the energy consumption by 50% and by 33% and its median by 53% and by 39% for $N = 20$ and for $N = 40$, respectively. We show corresponding results for SGM in the Appendix.

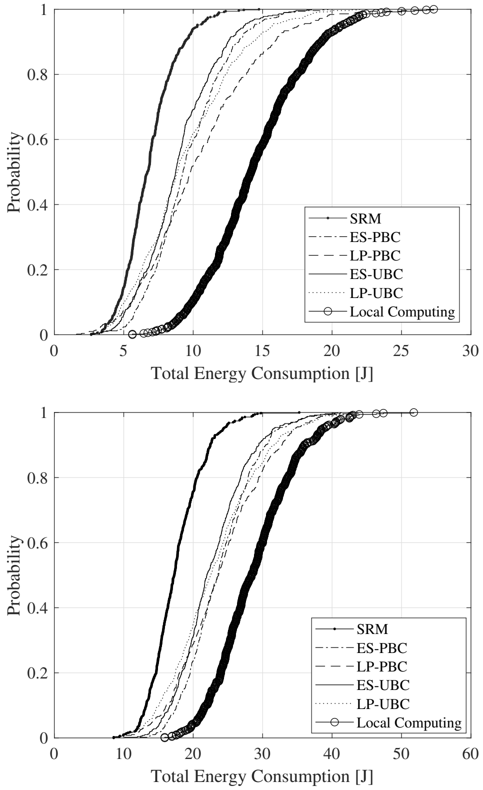

Fig. 14: Emprical CDF of total energy consumption for $N = 20$ (top) and $N = 40$ (bottom) for $|\mathcal{J}| = 20$.

F. Sensitivity to Communication Failure

Finally, we evaluate the sensitivity of the proposed solution to communication failure due to channel outage. Channel outages are inherent to wireless communication [41], [42], and are often due to imperfect knowledge of the channel state information by the transmitter due to, e.g., fading and mobility [43]–[45]. An outage happens when the intended transmission rate exceeds the instantaneous channel capacity [46], and its probability can be up to 1% [47]. In what follows we show results considering that WDs may experience channel outage, and cannot perform offloading even if they would like to, but this information is not known to the operator during the optimization, leading to loss of revenue. We consider two models of outage based on the model presented in [46]; in the first model the outage probability of a WD is proportional to its transmission rate, i.e. $w_i^* \log_2(1 + \frac{p_i^* h_i}{\bar{\sigma}_i^2})$ [48]. In the second model the outage probability is proportional to the amount of transmitted data D_i [49].

Figure 15 shows the utility as a function of the average outage probability among offloading WDs for the proposed SRM algorithm and for the baselines. The figure shows that the utility of the operator decreases approximately linearly with the average outage probability under both outage models, both for SRM and for the baselines. We can observe a slight difference between the shapes of the curves due to the correlation between the outage of a WD and the revenue it would give to the operator under different pricing and resource allocation schemes, but based on the results we can

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

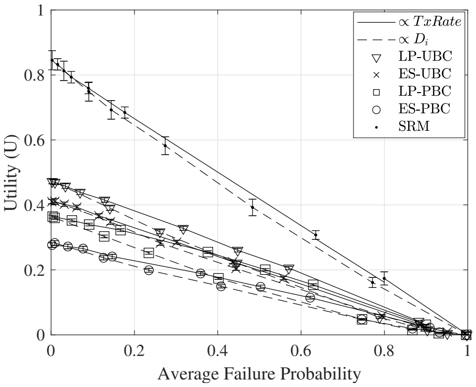

Fig. 15: Utility vs. average outage probability for $|\mathcal{N}| = 24$, $|\mathcal{J}| = 20$.

conclude that all algorithms exhibit graceful degradation under communication failure.

VIII. Related Work

A number of recent works deal with energy efficient computation offloading for a single mobile user and show the energy reduction obtained by computation offloading [50]–[54]. [50] introduces a system that facilitates energy-aware offloading to the infrastructure. [51] conducted an investigation into cloud computing with a focus on the utilization of bandwidth and energy consumption. They presented the results obtained from an experimental platform, specifically Amazon EC2. The findings of the study indicate that cloud offloading can be considered sustainable in terms of energy consumption. The authors also propose an algorithm that aims to maximize energy savings while minimizing the computational burden associated with offloading. [52] proposes CPU frequency scaling and transmission power adaptation to optimize the energy consumption of the computation of a task. [53] presents a dynamic offloading algorithm in order to achieve energy savings under time constraints. In [54], experimental results are used to show that battery power savings can be achieved using computation offloading. Inspired by these works that show the potential energy savings through offloading, we consider a system level optimization problem of computation offloading with an emphasis on the interaction between the WDs and the operator.

Going beyond offloading by a single device, a number of works consider computation offloading for multiple WDs [55]–[57]. [55] considers a model in which tasks arrive simultaneously to the cloud through a single wireless link and proposes a non-cooperative game among users that minimize their own energy use. [56] considers a hierarchical MEC network, where mobile users can make offloading decisions, and can decide the uplink transmission power, perform cloud selection, and route the tasks. A distributed offloading approach is developed based on game theory, in which user equipment collaborates with each other to minimize the network cost in terms of energy consumption and latency. [57] models the load-balancing problem as a stochastic congestion game in which each users aims to minimize its task execution time. Unlike these works that focus on the WDs' costs only, our problem formulation accounts for the financial incentives, i.e., the pricing of the operator and for service caching together with the optimization problem faced by WDs, resulting in a Stackelberg game formulation.

Related to our work are recent works that address the pricing problem in edge computing [11], [58]–[60]. Authors in [11] consider a Bayesian Stackelberg game in which the operator is the leader, and the WDs are followers. The objective of the operator is to maximize its revenue through pricing storage. In contrast, the WDs minimize a combination of the price paid and the delay. Different from ours, this work does not consider the optimization of communication and computing resources. Authors in [58] examine various models to optimize pricing for the task offloading problem, but they do not optimize pricing and resource allocation jointly; instead, they allocate compute resources to WDs proportional to their payment, and do not take into account communication resources. Authors in [59] proposed an auction for resource allocation and offloading. Resource allocation is based on bids from the WDs for a portion of the available edge resources, but joint optimization of pricing and resource allocation is not considered. Authors in [60] consider the problem of offloading, pricing and risk awareness in edge computing, modeled by a Stackelberg game played between the WDs and the edge servers. Compared to our work, the model does not consider the optimization of the edge resources.

Most related to ours are recent works that consider application caching and computation offloading [15]–[18]. In [15] authors consider a computation offloading and service caching problem with the objective of minimizing the total system cost defined as the weighted sum of energy consumption and completion time. Different from our work, they do not consider the joint optimization of bandwidth and computing resource allocation, as they do not consider bandwidth in the proposed optimization problem. In [16], authors consider computation offloading, resource allocation (wireless and computation resources) and service caching. They formulate the problem of total delay minimization subject to the capacity of the operator without considering the energy consumption of the WDs. Similarly, in [17], authors consider computation offloading, wireless and computation resource allocation and service caching and they formulate the problem of minimizing the total weighted sum of the delay and the computation energy cost of the WDs. The model was extended in [18] to consider maximization of the users' quality of service focusing on a multi-edge server scenario, and a decentralized solution was proposed.

Different from the above works, our paper is the first to jointly consider service caching, wireless and computation resource allocation, as well as the financial incentives of the edge operator, specifically pricing. While previous works have focused on minimization of the total cost, with various cost definitions, they have not taken into account the operator's financial incentives in conjunction. On the contrary, our game-theoretic model considers the interaction between WDs and the profit maximizing operator in the form of a Stackelberg game. Our results confirm that caching, pricing and the strategic

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

interactions of the WDs need to be jointly considered for maximizing the operator's utility and for minimizing the WDs' cost.

IX. Conclusion

In this work we have provided a game theoretic analysis of pricing, caching, wireless and computation resource allocation for edge computing. We modeled the interaction between WDs and the operator as a Stackelberg game. We showed that the operator's utility maximization problem is NP-hard and we proposed an efficient approximation based on a decomposition of the problem and by characterizing the subproblems. Our numerical results show that joint optimization of caching, pricing and resource allocation provides significant advantages compared to non-joint optimization, and our proposed algorithm can indeed find a near optimal solution, outperforming state of the art methods.

There are a number of interesting avenues of future work concerning pricing and resource allocation in edge computing. One example is the case of incomplete information, where the operator has to learn the applications' and the WDs' utilities and resource requirements in real time. Another direction is the case of a dynamic population of users, where pricing and resource allocation have to anticipate the effect of future arrivals of WDs.

References

[1] T. Elgamal, A. Sandur, K. Nahrstedt, and G. Agha, "Costless: Optimizing cost of serverless computing through function fusion and placement," in *Proc. IEEE/ACM Symposium on Edge Computing (SEC)*, 2018, pp. 300–312.

[2] P. Castro, V. Ishakian, V. Muthusamy, and A. Slominski, "The rise of serverless computing," *Communications of the ACM*, vol. 62, no. 12, p. 44–54, 2019.

[3] L. Baresi and D. F. Mendonça, "Towards a serverless platform for edge computing," in *Proc. IEEE International Conference on Fog Computing (ICFC)*, 2019, pp. 1–10.

[4] G. McGrath and P. R. Brenner, "Serverless computing: Design, implementation, and performance," in *In Proc. IEEE International Conference on Distributed Computing Systems Workshops*, 2017, pp. 405–410.

[5] H. Shafiei, A. Khonsari, and P. Mousavi, "Serverless computing: A survey of opportunities, challenges, and applications," *ACM Computing Surveys*, vol. 54, no. 11s, pp. 1–32, 2022.

[6] C. Xu, G. Zheng, and X. Zhao, "Energy-minimization task offloading and resource allocation for mobile edge computing in noma heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16 001–16 016, 2020.

[7] C.-F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *Proc. IEEE Globecom Workshops*, 2017, pp. 1–7.

[8] F. Wang, J. Xu, and S. Cui, "Optimal energy allocation and task offloading policy for wireless powered mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2443–2459, 2020.

[9] S. Safavat, N. N. Sapavath, and D. B. Rawat, "Recent advances in mobile edge computing and content caching," *Digital Communications and Networks*, vol. 6, no. 2, pp. 189–194, 2020.

[10] S. Jošilo and G. Dán, "Wireless and computing resource allocation for selfish computation offloading in edge computing," in *Proc. IEEE INFOCOM*, 2019, pp. 2467–2475.

[11] J. Yan, S. Bi, L. Duan, and Y.-J. A. Zhang, "Pricing-driven service caching and task offloading in mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4495–4512, 2021.

[12] F. Tütüncüoğlu and G. Dán, "Optimal service caching and pricing in edge computing: a Bayesian Gaussian process bandit approach," *IEEE Transactions on Mobile Computing*, pp. 1–15, 2022, to be published.

[13] S. Jošilo and G. Dán, "Joint management of wireless and computing resources for computation offloading in mobile edge clouds," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1507–1520, 2021.

[14] F. Tütüncüoğlu, S. Jošilo, and G. Dán, "Online learning for rate-adaptive task offloading under latency constraints in serverless edge computing," *IEEE Transactions on Networking*, vol. 31, no. 2, pp. 695–709, 2023.

[15] H. Zhou, Z. Zhang, D. Li, and Z. Su, "Joint optimization of computing offloading and service caching in edge computing-based smart grid," *IEEE Transactions on Cloud Computing*, vol. 11, no. 2, pp. 1122–1132, 2023.

[16] G. Zhang, S. Zhang, W. Zhang, Z. Shen, and L. Wang, "Joint service caching, computation offloading and resource allocation in mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 8, pp. 5288–5300, 2021.

[17] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, B. Hu, and V. C. M. Leung, "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4283–4294, 2019.

[18] W. Chu, X. Jia, Z. Yu, J. C. Lui, and Y. Lin, "Joint service caching, resource allocation and task offloading for MEC-based networks: A multi-layer optimization approach," *IEEE Transactions on Mobile Computing*, pp. 1–17, 2023, to be published.

[19] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: Task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1634–1644, 2021.

[20] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, 2017.

[21] A. Younis, T. X. Tran, and D. Pompili, "Energy-latency-aware task offloading and approximate computing at the mobile edge," in *Proc. International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2019, pp. 299–307.

[22] V. K. Garg, *An Overview of Digital Communication and Transmission*. Morgan Kaufmann, 2007.

[23] T. Joshi, A. Mukherjee, Y. Yoo, and D. P. Agrawal, "Airtime fairness for IEEE 802.11 multirate networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 513–527, 2008.

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

[25] L. Tan, Z. Kuang, L. Zhao, and A. Liu, "Energy-efficient joint task offloading and resource allocation in ofdma-based collaborative edge computing," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1960–1972, 2022.

[26] S. Jošilo and G. Dán, "Joint wireless and edge computing resource management with dynamic network slice selection," *IEEE Transactions on Networking*, vol. 30, no. 4, pp. 1865–1878, 2022.

[27] L. N. T. Huynh, Q. V. Pham, T. D. T. Nguyen, M. D. Hossain, Y.-R. Shin, and E.-N. Huh, "Joint computational offloading and data-content caching in NOMA-MEC networks," *IEEE Access*, vol. 9, pp. 12 943–12 954, 2021.

[28] Z. Zhang, Z. Li, and C. Wu, "Optimal posted prices for online cloud resource allocation," in *Proc. the ACM on Measurement and Analysis of Computing Systems*, 2017.

[29] H. Qiu and T. Li, "Auction method to prevent bid-rigging strategies in mobile blockchain edge computing resource allocation," *Future Generation Computer Systems*, vol. 128, pp. 1–15, 2022.

[30] H. R. Varian, *Microeconomic analysis*, 3rd ed. Norton, New York, 1992.

[31] T. Wang, R. Venkatesh, and R. Chatterjee, "Reservation price as a range: An incentive-compatible measurement approach," *Journal of Marketing Research*, vol. 44, no. 2, pp. 200–213, 2007.

[32] F. Tütüncüoğlu and G. Dán, "Optimal pricing for service caching and task offloading in edge computing," in *Proc. Wireless On-Demand Network Systems and Services Conference*, 2022, pp. 1–8.

[33] J. Moura and D. Hutchison, "Game theory for multi-access edge computing: Survey, use cases, and future trends," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 260–288, 2019.

[34] S. Jošilo and G. Dán, "Joint allocation of computing and wireless resources to autonomous devices in mobile edge computing," in *Proc. of Workshop on Mobile Edge Communications*, 2018, p. 13–18.

[35] S. Ghorpade and B. V. Limaye, *A course in multivariable calculus and analysis*. Springer, 2010.

[36] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3334914

[37] U. Feige, V. S. Mirrokni, and J. Vondrák, “Maximizing non-monotone submodular functions,” *Journal on Computing*, vol. 40, no. 4, pp. 1133–1153, 2011.

[38] N. Buchbinder and M. Feldman, “Deterministic algorithms for submodular maximization problems,” *ACM Transaction on Algorithms*, vol. 14, no. 3, 2018.

[39] Z. Zhang, B. Liu, Y. Wang, D. Xu, and D. Zhang, “Maximizing a monotone non-submodular function under a knapsack constraint,” *Journal of Combinatorial Optimization*, vol. 43, no. 5, p. 1125–1148, 2022.

[40] A. A. Bian, J. M. Buhmann, A. Krause, and S. Tschiatschek, “Guarantees for greedy maximization of non-submodular functions with applications,” in *Proc. International Conference on Machine Learning*, 2017, pp. 498–507.

[41] J. Cao, Q. Zhang, and W. Shi, *Edge Computing: A Primer*. Springer, 2017.

[42] S. A. Huda and S. Moh, “Survey on computation offloading in uav-enabled mobile edge computing,” *Journal of Network and Computer Applications*, vol. 201, pp. 103 341–103 367, 2022.

[43] C. Isheden and G. P. Fettweis, “Energy-efficient link adaptation with transmitter CSI,” in *Proc. IEEE Wireless Communications and Networking Conference*, 2011, pp. 1381–1386.

[44] C. Pan, H. Ren, K. Wang, W. Xu, M. Elkashlan, A. Nallanathan, and L. Hanzo, “Multicell MIMO communications relying on intelligent reflecting surfaces,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5218–5233, 2020.

[45] J. Zhang, J. Liu, S. Ma, C.-K. Wen, and S. Jin, “Transmitter design for large intelligent surface-assisted MIMO wireless communication with statistical CSI,” in *Proc. IEEE International Conference on Communications*, 2020, pp. 1–5.

[46] J. G. Proakis and M. Salehi, *Digital communications*. McGraw-hill New York, 2001.

[47] A. Goldsmith, *Wireless Communications*. New York: Cambridge University Press, 2005.

[48] G. Yang, H. Zhang, Z. Shi, S. Ma, and H. Wang, “Asymptotic outage analysis of spatially correlated rayleigh MIMO channels,” *IEEE Transactions on Broadcasting*, vol. 67, no. 1, pp. 263–278, 2021.

[49] A. O. Yilmaz, “Calculating outage probability of block fading channels based on moment generating functions,” *IEEE Transactions on Communications*, vol. 59, no. 11, pp. 2945–2950, 2011.

[50] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “MAUI: making smartphones last longer with code offload,” in *Proc. International Conference on Mobile Systems, Applications, and Services*, 2010, pp. 49–62.

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

[52] 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*, 2012, pp. 2716–2720.

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

[54] A. Rudenko, P. Reiher, G. Popek, and G. Kuenning, “Saving portable computer battery power through remote process execution,” *Mobile Computing and Communications Review*, vol. 2, 1998.

[55] E. Meskar, T. D. Todd, D. Zhao, and G. Karakostas, “Energy efficient offloading for competing users on a shared communication channel,” in *Proc. IEEE International Conference on Communications*, 2015, pp. 3192–3197.

[56] B. Wu, J. Zeng, L. Ge, X. Su, and Y. Tang, “Energy-latency aware offloading for hierarchical mobile edge computing,” *IEEE Access*, vol. 7, pp. 121 982–121 997, 2019.

[57] F. Zhang and M. M. Wang, “Stochastic congestion game for load balancing in mobile-edge computing,” *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 778–790, 2021.

[58] B. Baek, J. Lee, Y. Peng, and S. Park, “Three dynamic pricing schemes for resource allocation of edge computing for IoT environment,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4292–4303, 2020.

[59] T. Bahreini, H. Badri, and D. Grosu, “Mechanisms for resource allocation and pricing in mobile edge computing systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 667–682, 2022.

[60] G. Mitsis, E. E. Tsiropoulou, and S. Papavassiliou, “Price and risk awareness for data offloading decision-making in edge computing systems,” *IEEE Systems Journal*, vol. 16, no. 4, pp. 6546–6557, 2022.

[61] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.

Feridun Tütüncüoğlu is a Ph.D. student at the Division of Network and Systems Engineering in KTH Royal Institute of Technology, Stockholm, Sweden. He received M.Sc in Electrical & Electronics Engineering from Bilkent University, Turkey in 2019. He worked as a research engineer at the department of Electrical & Electronics Engineering, Bilkent University from 2017 to 2019. His research interests include design and analysis of online learning algorithms and game theoretical models of edge computing resource management, allocation and pricing.

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 served as area editor of Computer Communications 2014-2021, and as editor of IEEE Transactions on Mobile Computing 2019-2023. 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.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/