Uplink Resource Allocation for Multiple Access Computational Offloading

Mahsa Salmani and Timothy N. Davidson

Abstract—The opportunity to offload computational tasks that is provided by the mobile edge computing framework enables mobile users to broaden the range of tasks that they can execute. When multiple users with different requirements seek to offload their tasks, the available communication and computation resources need to be efficiently allocated. The nature of the computational tasks, whether they are divisible or indivisible, and the choice of the multiple access scheme employed by the system have a fundamental impact on the total energy consumption of the offloading users. In this paper, we show that using the full capabilities of the multiple access channel can significantly reduce the energy consumption, and that the required resource allocation can be efficiently computed. In particular, we provide a closed-form optimal solution of the energy minimization problem when a set of users with different latency constraints are completely offloading their computational tasks, and a tailored greedy search algorithm for a good set of users. We also consider “data-partitionable” computational tasks and develop a low-complexity iterative algorithm to find a stationary solution to the energy minimization problem in that case. In addition, we develop low-complexity optimal algorithms for the energy minimization problem under the Time Division Multiple Access (TDMA) scheme in the binary offloading and partial offloading scenarios. Our numerical experiments show that the proposed algorithms outperform existing algorithms in terms of energy consumption and computational cost.

Index Terms—Mobile edge computing, mobile cloud computing, computation offloading, resource allocation.

I. INTRODUCTION

THE rapid development of mobile device technology and wireless communication networks is bringing the vision of ubiquitous computing to fruition, at least for tasks of modest complexity. However, as the demand for ubiquity in computationally-intensive and latency-sensitive tasks increases, the limited computation, memory and energy resources of mobile and other small scale devices present significant impediments to progress. The Mobile Edge Computing (MEC) framework seeks to address these impediments by offering the devices the opportunity to offload (a portion of) their computational tasks to a local shared computational resource. This offloading option enables the users to execute more computationally complex applications within a certain deadline, and can also prolong the battery lifetime of the devices [11–3]. Some early prototype systems include MAUI [4], CloneCloud [5], and ThinkAir [6].

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under grant RGPIN-2015-06631. The authors are with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada. Email: {salmam, davidson}@mcmaster.ca. A preliminary version of portions of this paper will appear in Conf. Rec. 52nd Asilomar Conf. Signals, Syst. Comput., Oct 2018.

In order to effectively exploit the opportunities provided by the MEC framework, a computational offloading system must address a number of challenges, including the energy that each user would expend to offload (a portion of) its computational task to the access point [7], the latency requirements of the tasks [8], contention for the limited communication resources of the networks [1], and, in some cases, contention for the limited shared computation resources at the access point [9]. Accordingly, the problem of selecting the mobile devices that offload their computational tasks and the allocation of resources to those users is a joint optimization problem over the available communication and computation resources [10–18]. Often, that problem targets the energy consumption of the offloading users, while ensuring that the latency constraints of the computational tasks are met.

The nature of the computational tasks that the users wish to complete has a fundamental impact on the formulation of this resource allocation problem [1]. If the components of the tasks are tightly coupled, the tasks must be either completely offloaded or executed locally. On the other hand, if the computational tasks have independent or loosely coupled components, the mobile devices can benefit from parallelism between the access point and the mobile device by offloading a portion of their computational tasks while the remainder is executed locally. When there are multiple devices that are seeking access to the computing resources, the architecture of the envisioned system will determine whether the offloading decisions are to be made centrally, or in a distributed fashion. For such a multi-user offloading system, the choice of the multiple access scheme can have a significant impact, especially when the latency constraints of the users are tight.

The main focus of this work is to find the minimum total user energy consumption of a centralized offloading system with $K$ users, each of which wishes to complete its computational task within its own specific deadline, either locally or by offloading (a portion of) the task to an access point that has substantial computation resources. In most of the previous work on such systems, the multiple access schemes employed by the system are restricted. For example, the channels between the users and the access point are sometimes assumed to be orthogonal in time or frequency [19–21], or it is assumed that the access point performs independent decoding [10]. These schemes limit the range of the achievable rates and powers by which the users can operate, and hence the optimal energy consumption cannot be obtained.

In this paper, we will provide efficient algorithms for optimizing the communication resource allocations for the multiple access scheme that exploits the full capabilities of the
multiple access channel. We will consider energy minimization problems for indivisible computational tasks, and for “data-partitionable” tasks [22], in which a simple-to-describe operation is applied, independently, to different blocks of data. For the indivisible case, the combinatorial structure of the “binary offloading” problem of deciding which users will offload their tasks and which will complete them locally suggests a natural decomposition into an outer search strategy for the offloading decisions and the inner optimization of the communication resources for given offloading decisions. That inner subproblem will be referred to as the “complete offloading” problem. For the case of “partial offloading” of data-partitionable tasks, the fraction of each task to be offloaded will be optimized jointly with the communication resource allocation.

Our strategy for solving the resource allocation problems is based on the insights developed in our previous work on the two-user case [14], [17], which suggests elegant decompositions of the problem. We will also exploit the polymatroid structure of the capacity region of the multiple access channel (see [23]) to obtain closed-form optimal solutions for the powers in terms of the transmission rates in both the complete offloading and partial offloading cases. In the complete offloading case, we will also find the closed-form optimal solutions for the transmission rates of the users, and hence the optimal energy consumption is obtained. Those solutions also form the core of a tailored greedy search algorithm for good solutions to the binary offloading problem. In the partial offloading case, in addition to the closed-form optimal solutions for the transmission powers, we will find the closed-form optimal solutions for the fraction of bits offloaded by each user [17]. Accordingly, the energy minimization problem is reduced to a $K$-variable optimization problem in terms of the transmission rates for which a simple coordinate descent algorithm is guaranteed to find a stationary solution.

The energy minimization problem can also be addressed for sub-optimal multiple access schemes, such as Time Division Multiple Access (TDMA) schemes. In this paper, we will show that under the TDMA scheme, that problem can be written as a jointly convex optimization problem in the $K$ transmission rates. In our simulation results, we will show that although there are scenarios in which TDMA provides good performance, there are others in which exploiting the full capabilities of the multiple access channel enables a substantial reduction in the energy consumption.

A special case of our total energy minimization problem for a $K$-user offloading system with the full multiple access scheme appears in [23]. In [23] it was assumed that the latency constraints of all the users are the same, while we consider the more general case in which different users have different latency requirements. In [23], the solutions for the transmission rates, transmission powers, and the fraction of offloaded bits in the partial offloading scenario are obtained iterating using a variant of the ellipsoidal algorithm. For a $K$-user system, that algorithm imposes a computational cost of $O(K^3)$ operations per iteration. In contrast, the closed-form optimal solutions for the transmission powers and the fraction of offloaded bits provided by the decomposition-based approach developed herein result in an algorithm whose computational cost is only $O(K \log K)$. Since the class of scenarios for which the proposed algorithm is developed includes the scenario of equal latencies for which the algorithm in [23] was developed, our algorithm has the same performance as that in [23] in the equal latency case. However, our numerical results, and those in [23], show that the number of iterations required by the corresponding algorithm in [23] can be quite large. As a result, in the case of a single-antenna access point the proposed algorithm has a significant computational advantage.

An analogous equal-latency assumption has also been considered for the TDMA scheme in [21]. An additional difference of the problem in [21] and the one proposed herein is that we have considered the dynamic voltage scaling approach [22] for computation energy management in the mobile devices. This approach guarantees the minimum local energy consumption in the users subject to the latency constraints. In our numerical results, we will show that the energy consumption of the problem formulation proposed for the TDMA case in this paper is significantly lower than the energy consumption of the problem formulation in [21].

II. SYSTEM MODEL

We will consider a system consisting of $K$ single-antenna users, each of which has a computational task that is to be executed within that user’s latency constraint, and an access point that is equipped with sufficiently large computational resources so that the users have the option of offloading (a portion of) their tasks. The offloading users are served over a single time slot by the single-antenna (coherent) receiver at the access point, and the channels between the users and the access point are assumed to be frequency-flat and quasi-static, with complex gain $h_k$ and additive circular zero mean white Gaussian noise of variance $\sigma^2$. If the computational tasks to be executed are indivisible (i.e., if the components of the tasks are tightly coupled), each user must either offload its task to the access point completely or execute the whole task locally [10], [25], [26]. If the tasks have independent components, some of those components can be offloaded while the remainder are executed locally [22], [27].

In Section III we will tackle the problem of minimizing the energy consumption of the users in the case of indivisible tasks. In this “binary offloading” problem the energy consumption of each user is either the energy expended in local computation or the energy expended in offloading the task. Since the latter depends on the subset of users that are offloading, the problem admits a natural decomposition into an inner problem of energy minimization for “complete offloading” for a given subset of users, and an outer search for the subset of users that should offload their tasks. In Section IV we will address the energy minimization problem in the case where the computational tasks of the users are “data-partitionable” [22]. In this “partial offloading” scenario, the total energy consumption of each user is the summation of the energy consumed in transmitting the offloaded portion of task to the access point and the energy consumed in executing the remainder of the task locally.

Our goal is to find the allocations of the communication resources to the users, in the binary offloading and partial
offloading scenarios, that minimize the total energy consumption of the users while ensuring that the latency constraints of the tasks are met. Accordingly, we will find allocations of the transmission rate and transmission power of each user and the optimal fraction of bits offloaded by each user (in partial offloading case) under the constraint of the achievable rate region of the chosen multiple access scheme.

In order to formulate that problem, let $R_k$ and $P_k$ denote the transmission rate and transmission power (in units per channel use) of the $k^{th}$ user, respectively. If $B_k$ denotes the total number of bits describing the computational task of user $k$, then $\gamma_k B_k$ defines the number of bits offloaded by the $k^{th}$ user to the access point, where $\gamma_k \in \{0, 1\}$ for the binary offloading case, and $\gamma_k \in [0, 1]$ for the partial offloading case. (In the partial offloading case we will model the tasks as being infinitesimally divisible.) Accordingly, the time it takes for user $k$ to offload (the portion of) its task to the access point is $t_{UL_k} = T_s \frac{\gamma_k B_k}{R_k}$, where $1/T_s$ is the symbol rate of the channel.

The energy that it expends in doing so is $T_s \frac{\gamma_k B_k}{R_k} P_k$.

The structure of data-partitionable tasks is such that the time that it takes for the access point to process the portion of the task that is offloaded can be modeled as a simple multiple of the size of that portion [27],

$$t_{exec} = \delta_c \gamma_k B_k,$$

where $\delta_c$ is the time it takes to process one bit at the access point. While data-partitionable tasks can also be completed using binary offloading, the class of “binary-offloadable” tasks is essentially unrestricted. To enable us to capture the richness of that class using a common notation, we observe that in the binary offloading case $\gamma_k \in \{0, 1\}$, and we can employ the expression in (1) if we scale $\delta_c$ so that $\delta_c B_k$ is equal to the time that it would take to complete the task at the access point.

If the time it takes for the access point to send the results back to the $k^{th}$ user is denoted by $t_{DL_k}$, then the latency constraint of that offloading user can be written as

$$t_{UL_k} + t_{exec} + t_{DL_k} \leq L_k,$$

in which $L_k$ denotes the maximum allowable latency for user $k$. As the description length of the results of the (partially) offloaded task are often considerably shorter than those of the task itself, it will be assumed that $t_{DL_k}$ is a (possibly different) constant for each user, independent of the number of users and the computational load imposed by the other users.

If the users employ a conventional local computational architecture, the local execution time takes a similar form to that in (1), leading to a local latency constraint for data-partitionable tasks that takes the form $t_{loc_k} = \delta_k (1 - \gamma_k) B_k \leq L_k$, where $\delta_k$ is the time it takes for the $k^{th}$ user to process one bit. An analogous scaling can be used for the binary offloading case.

The energy that user $k$ expends on local computation depends on the number of operations that the local processor must perform. For data-partitionable tasks, the number of operations can be modeled as being proportional to the fraction of the description that user $k$ retains [22]. Hence, we will denote the local computational energy consumption of user $k$ as the function $F_k ((1 - \gamma_k) B_k)$. Consistent with our approach to notation for the execution time, we will employ the same notation for the local computation energy for the binary offloading case, with the understanding that in that case the two relevant energies are $F_k (B_k)$, which is the energy required to complete the task locally, and $F_k (0) = 0$.

In terms of the constraints on the data rate, we observe that since the system that we consider serves the users over a single time slot, each user operates at a single power-rate pair $(P_k, R_k)$. Therefore, the rate region constraint can be expressed as $(R_k) \in \mathcal{R} \{(P_k)\}$, where $\mathcal{R} \{(P_k)\}$ is the $K$-user achievable rate region for the chosen multiple access scheme; see [14] for the two-user case. In expressing that region, we will assume that the data blocks are long enough for the asymptotic characterizations to be valid; e.g., [28]. Extensions to characterizations of the rate region for finite block lengths (e.g., [29], [30]) and to offloading systems with multiple time slots [17], [51] will be guided by the insight developed in the asymptotic single time slot case that we will consider.

For the offloading system that we have described, the general problem of minimizing the total energy consumption of a system with $K$ offloading users can be formulated as

$$\min_{\{R_k\}, \{P_k\}} \sum_k T_s \frac{\gamma_k B_k}{R_k} P_k + F_k ((1 - \gamma_k) B_k)$$

subject to

$$0 \leq \gamma_k \leq 1, \quad \forall k,$$

$$T_s \left( \frac{\gamma_k B_k}{R_k} + \delta_c \gamma_k B_k + t_{DL_k} \right) \leq L_k, \quad \forall k,$$

$$\delta_k (1 - \gamma_k) B_k \leq L_k, \quad \forall k,$$

$$0 \leq P_k, \quad \forall k,$$

$$\{R_k\} \in \mathcal{R} \{(P_k)\}.$$  

Although the conventional local computational architecture enables the use of commodity hardware, the energy expended on local computation can be reduced by adopting the dynamic voltage scaling architecture [22], which enables the user to adjust the local CPU cycle frequency. The dynamic voltage scaling architecture enables the user to minimize the energy required to complete (the local portion of) the task within the specified latency constraint. In particular, if optimized dynamic voltage scaling is employed in partial offloading of a data-partitionable task, then the local computational energy, which is denoted by $F_k ((1 - \gamma_k) B_k)$ in the objective function in (3a), is minimized, and can be expressed in the form [22]

$$\frac{M_k}{T_k} ((1 - \gamma_k) B_k)^3,$$

where the coefficient $M_k$ depends on the characteristics of the chip of user $k$. Under this architecture, the local latency constraint in (3c) is implicitly satisfied, and hence it can be removed from (3). In the case of binary offloading of a generic task with optimized dynamic voltage scaling, that constraint is also implicitly satisfied, and we will denote the minimum local computational energy by $E_{loc_k}$.  

In Sections [III] and [IV] we will focus on the development of algorithms for users that employ dynamic voltage scaling in the binary and partial offloading scenarios, respectively. However, with simple modifications the proposed algorithms can be applied to users with conventional computation architectures. The required modifications in the binary case are discussed
at the end of Section III and the modifications for the case of partial offloading were illustrated for a two-user system in [32]. In our numerical results in Section V we will illustrate that dynamic voltage scaling approach provides significant energy savings.

As mentioned in the Introduction, the problem in [3] is different from those in [21] and [24]. We allow the latency constraints of the users to be different, \( L_k \), which enables the users with larger latencies to benefit from their own available time to transmit. In [21] and [24] the latency constraints of the users are assumed to be the same, which forces the system to work with the minimum latency constraint among the users. If the latencies are different, doing so will increase the total energy consumption of the users. In addition, for the partial offloading case, the formulations in [21] and [24] assume that \( \delta_k \) is small enough that the dependence of the execution time at the access point, \( t_{\text{exec}} \), on the fraction of the task that is offloaded, \( \gamma_k \), can be neglected. We do not make that assumption in our formulations; see [1]. Finally, in contrast to [21], in our formulation we assume that the users can employ dynamic voltage scaling \([22],[27]\) to minimize the energy that they expend in local computation.

III. Binary Offloading

In this section we will consider minimizing the total energy consumption of the \( K \)-user system when the computational tasks of the users are indivisible, i.e., the task of each user must be either totally offloaded to the access point or executed by the user. The problem of finding the optimal selection of offloading users that results in the minimum total energy consumption of the system is a combinatorial problem in the number of users. As a result, the joint offloading decision-resource allocation problem is typically partitioned, with the optimal resource allocation being found for given offloading decisions and a combinatorial search strategy being used to make the optimal offloading decisions. Accordingly, in this section we first seek the optimal solution of energy minimization problem for the complete offloading case in which a subset of users is scheduled to offload their tasks to the access point. Then we will develop a low-complexity pruned greedy search technique that is tailored to the characteristics of the problem to find a set of offloading users that typically results in close-to-optimal energy consumption.

A. Complete Computation Offloading

Let \( S = \{1, 2, \ldots, K\} \) denote the set of all \( K \) users in the system and let \( S' \subseteq S \) denote the subset of users scheduled to fully offload their computational tasks, i.e., \( \gamma_k = 1, \forall k \in S' \) and \( \gamma_k = 0, \forall k \notin S' \). In that case, the total energy consumption of the system consists of the total transmission energy consumption of the users in subset \( S' \) and the summation of the local energy consumption of the remaining users, namely,

\[
E_{\text{total}} = T_s \sum_{k \in S'} \frac{P_k}{R_k} + \sum_{j \in S \setminus S'} E_j(B_j).
\] (5)

In order to minimize the total energy consumption of the system, both the communication resource allocation to the offloading users and the local energy consumption of the users which are executing their tasks locally must be optimal, while the latency constraints are met in each case; see (3) and (4).

We will assume that the users are able to employ dynamic voltage scaling \([22],[27]\) to minimize the local computational energy consumption of each user, while ensuring that its latency constraint is satisfied; see Section II.

The problem of minimizing the total energy consumption of a set of users, \( S' \), that have been scheduled to offload their computational tasks can be formulated as

\[
\begin{align*}
\min_{\{R_k\},\{P_k\}} & \quad T_s \sum_{k \in S'} \frac{P_k}{R_k} \\
\text{s.t.} & \quad T_s \left( \frac{P_k}{R_k} \right) + \frac{L_k}{\gamma_k} \leq L_k, \quad \forall k \in S', \\
& \quad 0 \leq P_k, \quad \forall k \in S', \\
& \quad \{R_k\} \in \mathcal{R}(\{P_k\}), \quad \forall k \in S'. 
\end{align*}
\] (6a)

As discussed in Section III the achievable rate region, \( \mathcal{R} \), depends on the chosen multiple access scheme. In the following sections, we will first find optimal energy consumption of an offloading system that employs an optimal multiple access scheme, i.e., a scheme that can operate at any point in the capacity region of the multiple access channel. We will then consider the sub-optimal TDMA scheme and find the optimal resource allocation for that scheme.

B. Full Multiple Access Scheme

Let us consider a system that employs a multiple access scheme that exploits of the full capabilities of the multiple access channel. Then for any subset of those users, \( \mathcal{N} \subseteq S' \), there is a rate region constraint of the form, e.g., \([33]\),

\[
\sum_{i \in \mathcal{N}} R_i \leq \log(1 + \sum_{i \in \mathcal{N}} \alpha_i P_i),
\] (7)

in which \( \alpha_i = \frac{\log^2 \gamma_i}{P_i} \). If we let \( K' \) denote the number of users in \( S' \), then the rate region is bounded by \( 2K' - 1 \) such constraints, and \( K' \) constraints of the form \( R_k \geq 0 \). Therefore, for a system that fully exploits the capabilities of the multiple access channel, the problem in (6) becomes

\[
\begin{align*}
\min_{\{R_k\},\{P_k\}} & \quad T_s \sum_{k \in S'} \frac{P_k}{R_k} \\
\text{s.t.} & \quad T_s \left( \frac{P_k}{R_k} \right) \leq \tilde{L}_k, \quad \forall k \in S', \\
& \quad 0 \leq P_k, \quad \forall k \in S', \\
& \quad 2\sum_{i \in \mathcal{N}} R_i \leq 1 + \sum_{i \in \mathcal{N}} \alpha_i P_i, \quad \forall \mathcal{N} \subseteq S', 
\end{align*}
\] (8a)

where \( \tilde{L}_k = L_k - \delta_k \gamma_k B_k - t_{\text{DL}}. \)

As the first step toward solving the problem in (8) we decompose the problem into an inner optimization over the transmission powers and an outer optimization over the rates:

\[
\begin{align*}
\min_{\{R_k\}} & \quad T_s \sum_{k \in S'} \frac{P_k}{R_k} \\
\text{s.t.} & \quad \{P_k\}, \quad \text{s.t. (8b)} - (8d).
\end{align*}
\] (9)

For a given set of transmission rates \( \{R_k\} \), the inner optimization problem is a linear programme in \( \{P_k\} \) and the feasibility region for the transmission powers is a polyhedron. Hence, in the search for an optimal solution it is sufficient to restrict attention to the vertices of the feasibility region. Each vertex is described by the simultaneous satisfaction of \( K' \) of the linear
inequality constraints in (8d) with equality. As we show in the next section, by exploiting the polymatroid structure of the constraints in (8d) (e.g., (23)), we can significantly reduce the number of the candidate vertices. In fact, we will show that we can find a closed-form optimal solution for the powers.

1) Closed-form optimal solutions for the powers: To begin, let us group the rate region constraints in (8d) into \( K' = |S'| \) classes, where a constraint is assigned to class-\( \ell \) if it involves the powers and rates of \( \ell \) users; i.e., the constraint is assigned to class \( \ell \) if \( |\mathcal{N}| = \ell \). In Appendix A we show that the vertices of the rate region that are candidates for optimality arise from the simultaneous satisfaction of at most one constraint from each of the classes. Since such vertices involve the simultaneous satisfaction of \( K' \) constraints, that implies that at optimality one constraint from each class holds with equality; see also [31].

Since class-\( K' \) contains only one constraint, that implies that at optimality

\[
2 \sum_{i=1}^{K'} R_i = 1 + \sum_{i=1}^{K'} \alpha_i P_i. \tag{10}
\]

Accordingly, the power of any arbitrary user, say user \( n \), can be written in terms of the powers of the other users as

\[
\alpha_n P_n = 2 \sum_{i \neq n} R_i - \sum_{i \neq n} \alpha_i P_i - 1. \tag{11}
\]

By substituting the closed-form expression (11) in the objective function (8a) and the constraints (8d), the inner optimization problem in (9) remains a linear programming problem, but now with \( K' - 1 \) variables, namely,

\[
\min_{\{P_k\}} T_s \sum_{k \in S\setminus \{n\}} (\rho_k - \rho_n) \alpha_k P_k \tag{12a}
\]

subject to

\[
2 \sum_{i \in \mathcal{N}} R_i - 1 \leq \sum_{i \in \mathcal{N}} \alpha_i P_i \leq 2 R_n \left(2 \sum_{i \in \mathcal{N}} R_i - 1\right), \quad \forall \mathcal{N} \subseteq S\setminus \{n\}, \tag{12b}
\]

in which

\[
\rho_k = \frac{B_k}{\alpha_k R_k}. \tag{13}
\]

It can be seen that the constraints of the problem in (12a) have a polymatroid structure and hence, the optimal solution of (12a) results from simultaneous satisfaction of \( (K' - 1) \) constraints with at most one constraint from each class. For positive coefficients of the powers in the objective function in (12a) it can be shown that, analogous to (10), at optimality the single lower bound constraint in class-\( (K' - 1) \) is satisfied with equality. Accordingly, we can obtain a closed-form solution for the transmission power of another arbitrary user by using an expression analogous to (11).

Considering the above explanation, we can obtain a sequence of closed-form solutions for all transmission powers for a given set of transmission rates if we can guarantee that in each step all the coefficients \( \rho_k - \rho_n \) are positive. We can do that if we determine the permutation \( \pi \) such that

\[
\rho_{\pi(K')} \leq \rho_{\pi(K' - 1)} \leq \cdots \leq \rho_{\pi(1)}, \tag{14}
\]

and choose the sequence of values of \( n \) to be \( \pi(K'), \pi(K' - 1), \cdots, \pi(1) \). Once the ordering in (14) has been determined, the first step of the algorithm is to obtain the closed-form solution for \( P_{\pi(K)} \) by substituting the expression in (11) with \( n = \pi(K') \) into (10); that is,

\[
\rho_{\pi(K')} P_{\pi(K')} = 2 \sum_{i=1}^{K'} R_{\pi(i)} - \sum_{i=1}^{K' - 1} \alpha_{\pi(i)} P_{\pi(i)} - 1. \tag{15}
\]

The same procedure can then be applied in a sequential manner to find closed-form solutions for all the powers. In the last step, the closed-form solution for the power of the user corresponding to \( \rho_{\pi(1)} \) is

\[
P_{\pi(1)} = (2 R_{\pi(1)} - 1)/\rho_{\pi(1)}. \tag{16}
\]

It can be seen that the optimal transmission power of this user is only a function of its transmission rate and its channel, and does not depend on the transmission power of the other users. Now, by retracing our steps, a closed-form solution for the optimal power of each user can be written in terms of the transmission rates rather than the powers of other users; i.e.,

\[
P_{\pi(k)} = \frac{2 R_{\pi(k)} - 1}{\alpha_{\pi(k)}} 2 \sum_{j=1}^{K' - 1} R_{\pi(j)}. \tag{17}
\]

We observe that the ordering in (14) not only ensures that the terms \( (\rho_{k} - \rho_n) \) in (12a), and the corresponding terms in the subsequent instances of (14) are positive, it also determines the (optimal) decoding order that enables the rates that will be chosen in (19) below to be achieved by successive decoding. (Since these rates correspond to vertices of the capacity region, no time sharing is required.) In particular, it can be seen from (16) that user \( \pi(1) \) is transmitting at its maximum achievable rate while the interference from other offloading users in the system has been cancelled. That implies that the message from user \( \pi(1) \) is being decoded after the messages from all other offloading users have been decoded and the corresponding interference canceled. Similarly, the expression in (15) reveals that the message from user \( \pi(K') \) is the first message to be decoded, with the interference from the messages from the other users being treated as noise.

2) Closed-form optimal solutions for the rates: Now that we have the closed-form solutions for the transmission powers in (17), the outer optimization problem in (9) becomes

\[
\min_{\{R_k\}} T_s \sum_{k \in S'} B_{\pi(k)} \left(\frac{2 R_{\pi(k)} - 1}{\alpha_{\pi(k)}}\right) 2 \sum_{j=1}^{K' - 1} R_{\pi(j)} \tag{18a}
\]

subject to

\[
\frac{T_s B_k}{L_k} \leq R_k, \quad \forall k \in S'. \tag{18b}
\]

It can be shown that the objective function in (18) is an increasing function with respect to each transmission rate and that the constraints on the transmission rates are separable. Hence, the optimal rate for each user is the minimum feasible rate according to its latency constraint,

\[
R_k = \frac{T_s B_k}{L_k}. \tag{19}
\]

Now by using the closed-form optimal values of the transmission rates, which depend only on the parameters of the problem, we can obtain the \( \rho_k \)'s in (13). Once those \( \rho_k \)'s have been sorted, the optimal solutions for the transmission powers can be found using (17). These steps are summarized in Algorithm 1. The computational efficiency of the algorithm is apparent from the observation that the number of operations required is dominated by the sorting procedure in Step 3, which requires \( O(K' \log K') \) operations.
Algorithm 1: The optimal solution to (8)

Input data: \( S' \), \( \{B_k \} \), \( \{\hat{L}_k \} \), \( \{\alpha_k \} \), and \( T_s \).

Step 1: Calculate the optimal rates \( \{R_k \} \) using (19).

Step 2: Calculate the values \( \{p_k \} \) using (13).

Step 3: Order \( \{\rho_k \} \) according to (14) to find the optimal permutation \( \pi \).

Step 4: Calculate the optimal powers using (17).

C. Time Division Multiple Access

In this section we will tackle the total energy minimization problem of a system with \( K' \) (completely) offloading users when TDMA is selected as the transmission scheme. In the TDMA scheme there is only one user offloading at a time, and hence the communication resources can be fully assigned to that user. However, the latency constraint of the \( k \)th user must include the transmission time of all the users that already offloaded their tasks, as well as the transmission time of user \( k \) itself. If we order the users so that \( L_1 \leq L_2 \leq \cdots \leq L_K' \), the problem of minimizing the total energy consumption in the TDMA case can be written as

\[
\min_{\{R_k \}, \{p_k \}} \left( \sum_{k \in S'} \frac{R_k}{R_{\text{Th}}} P_k \right) \tag{20a}
\]

s.t. \[
\sum_{i=1}^{k} T_s \left( \frac{B_i}{R_i} \right) \leq \hat{L}_k, \quad \forall k \in S', \tag{20b}
\]

\[
0 \leq p_k, \quad \forall k \in S', \tag{20c}
\]

\[
0 \leq R_k \leq \log_2 (1 + \alpha_k p_k), \quad \forall k \in S'. \tag{20d}
\]

Since for a given set of rates \( \{R_k \} \), the objective function in (20) is increasing in terms of the transmission powers \( \{p_k \} \), and the constraints on the powers are separable, the closed-form optimal solution for the transmission powers is

\[
P_k = \frac{\alpha_k - 1}{\alpha_k} R_{\text{Th}}. \tag{21}
\]

The remaining problem can be written in terms of the transmission rates as follows

\[
\min_{\{R_k \}} T_s \left( \sum_{k \in S'} \frac{B_k}{\alpha_k} (\frac{\alpha_k - 1}{R_{\text{Th}}}) \right) \tag{22a}
\]

s.t. \[
\sum_{i=1}^{k} T_s \left( \frac{B_i}{R_i} \right) \leq \hat{L}_k, \quad \forall k \in S', \tag{22b}
\]

\[
0 \leq R_k, \quad \forall k \in S'. \tag{22c}
\]

It can be shown that the objective function in (22) is jointly convex in terms of the transmission rates and hence, the optimal solution to (22) can be efficiently obtained. The optimal solution to (20) is then the concatenation of these rates and the corresponding powers in (21).

D. Binary Computational Offloading

Now that we have obtained a closed-form optimal resource allocation for a given set of offloading users in the case of the full multiple access scheme, and a quasi-closed-form solution based on a convex optimization problem with \( K' \) variables in the case of TDMA, we can tackle the “outer” problem of finding an optimal set of offloading users. This is a combinatorial problem, with a search space of \( 2^K \) possibilities, but it admits a tree structure. Therefore, in addition to the branch-and-bound algorithm for finding an optimal set of offloading users, the problem is amenable to a wide variety of lower-complexity tree-search algorithms that typically provide offloading sets with low energy consumption. As an example, we will develop a customized greedy search technique in which the search tree is (deterministically) pruned at each iteration.

1) Greedy search algorithm: To describe the proposed algorithm, we let \( S' \) denote the set of users that have already been chosen for offloading, and let \( \mathcal{U} \) denote the set of users for which a decision as to whether or not to offload has yet to be made. We initialize the algorithm with all the users in \( \mathcal{U} \) and none in \( S' \). The key steps in each iteration of the algorithm are an exploratory step, a deterministic pruning step, and a greedy user selection step that selects the “best” user to add to the offloading set (if any remain after the pruning step). These steps are summarized in steps 3, 4, and 6 in Algorithm 2. In the exploration step, for each user in \( \mathcal{U} \) we obtain the energy consumption of the system if that user were to be added to the set of offloading users. In the case of the full multiple access scheme that can be computed using the closed-form expression in Algorithm 1 and in the case of TDMA it can be found by solving the convex optimization problem in (22) and using the expression in (21). In the pruning step we remove from \( \mathcal{U} \) all those users for whom the exploration step revealed that (at this iteration) offloading would incur more energy consumption than local computation. These users can be “safely” removed, because in subsequent iterations there will be more users offloading and hence the energy required by any individual user to offload their task does not decrease as the iterations progress. In the greedy user selection for offloading step we select the user for which offloading offers the greatest reduction in the energy consumption of the system.

To analyze the computational effort required by the algorithm, let \( Q(i) \) denote the cardinality of the set \( \mathcal{U} \) at the beginning of the \( i \)th iteration; i.e., at Step 3. At each iteration of the algorithm, the exploration step involves the solution of \( Q(i) \) complete offloading problems (Algorithm 1) for full multiple access scheme or (22) then (21) for TDMA scheme). The combination of the pruning and greedy selection steps requires \( Q(i) \) comparisons. At iteration \( i \), there are \( i \) users in \( S' \) and hence, in the full multiple access case the cost of each complete offloading problem in Step 3 is \( O(i \log i) \). Hence, the computational cost of Algorithm 2 in the full multiple access case is dominated by a term that is \( O(\sum_i Q(i) i \log i) \). In the worst case, no users are pruned in Step 4, so \( Q(i) \leq K - i + 1 \) and hence the computation cost is at most \( O(\sum_i (K - i + 1) i \log i) \). A loose upper bound for the argument of that expression is \( K^3 \log K \). In our numerical results in Section VI we will show that the proposed search strategy produces solutions that typically provide near optimal energy consumption and that it does so at a low computational cost.

We remark that an alternative greedy-based algorithm to solve the energy minimization problem of a binary offloading system was developed in (24) for systems in which all the users have the same latency constraint. The greedy choice at each iteration in that algorithm is similar to that in Algorithm 2.
i.e., at each iteration a user which results in the maximum reduction of the total energy consumption is added to the set of offloading users. However, as we will illustrate in Section V the computational cost of the greedy algorithm in [24] is significantly higher than that of Algorithm 2. This is mainly due to the fact that each component of the equivalent to Step 2 of Algorithm 2 involves solving a problem using the ellipsoid algorithm. Using analysis similar to that in the previous paragraph, that results in a computational cost that is \(O(K^5)\). In contrast, Algorithm 2 solves the corresponding problems using the closed-form expressions in Algorithm 1. The analogous analysis for Algorithm 2 yields a computational cost that is \(O(K^3 \log K)\).

Algorithm 2 can be modified for the case of conventional local communication architecture by replacing each \(E_{\text{loc}}\) by the energy required to complete the task locally using the conventional architecture, and by adjusting the initialization of the offloading set \(S'\) and the undecided set \(U\). The set \(S'\) is initialized with those users for which the task cannot be completed locally by the deadline, and \(U\) is initialized as \(\{1, 2, \ldots, K\} \setminus S'\). The initial offloading energy \(E_{\text{off}}(0)\) is set to be the optimal total energy consumption of the users in the initial set \(S'\).

2) Rounding-based algorithm: The authors in [24] also proposed a binary offloading algorithm for systems with equal latencies that is based on choosing the set of offloading users by rounding the solution to the corresponding partial offloading problem, and then solving the complete offloading problem for that set. As we will illustrate in Section V that rounding approach extends naturally to the approach that we have considered. Furthermore, since the rounding process selects users for offloading by rounding the optimal value of the offloading fraction in the partial offloading problem, \(\gamma^*_k \in [0, 1]\), to a value in \(\{0, 1\}\), there is a natural extension to randomized rounding (cf. [24]). In that case, multiple candidate sets of offloading users are selected according to independent Bernoulli distributions with the probability of offloading for user \(k\) being \(\gamma^*_k\). The hybrid scheme of deterministic and randomized rounding also arises naturally. Our numerical results in Section V will show that although the incorporation of randomized rounding offers better performance than deterministic rounding of the partial offloading solution, the proposed greedy search over the tree of full offloading problems offers significant reductions in the energy required to complete the tasks, at a computational cost that is similar to that of the deterministic rounding approach.

IV. PARTIAL OFFLOADING

Up until this point, we have considered computational tasks with tightly coupled components which can be either totally offloaded or executed locally. If the computational tasks of the users are divisible, the total energy consumption of the system can be reduced by taking advantage of the parallelism between the access point and the mobile devices. In that case each user offloads a portion of its computational task to the access point and executes the remaining portion locally. In this section we will focus on the class of “data-partitionable” tasks [22].

Algorithm 2: Binary Offloading Solution

**Input data:** values of \(\{B_k\}, \{\bar{L}_k\}, \{\alpha_k\}, \{E_{\text{loc}}\}, T_s.

**Step 1:** Set \(U = \{1, 2, \ldots, K\}, \ S' = \emptyset, \ E_{\text{off}}(0) = 0, \ i = 0.\)

**Step 2:** Set \(V = \emptyset\) and \(i \leftarrow i + 1.\)

for each \(k \in U\) do

Obtain the energy consumption of the system when user \(k\) is added to the set of offloading users, \(E_{\text{off}}(i)\); i.e., perform Alg. 1 or solve (22) then (21), for \(S' \cup \{k\}\).

if \(E_{\text{off}}(i) + E_{\text{loc}} - E_{\text{total}}(i)\) then

Add user \(k\) to the set of users to be pruned; i.e., \(V \leftarrow V \cup \{k\}\).

end if

end for

**Step 3:** Prune the selected users from the tree; i.e., \(U \leftarrow U \setminus V\).

**Step 4:** If \(U = \emptyset\), terminate the algorithm.

**Step 5:** Select the “best” user by choosing \(k^* = \arg \max_{k \in U} \left( E_{\text{off}}(i) + E_{\text{loc}} - E_{\text{total}}(i) \right)\).

**Step 6:** Update the offloading set and the undecided set; i.e., \(S' \leftarrow S' \cup \{k^*\}\) and \(U \leftarrow U \setminus \{k^*\}\).

**Step 7:** Update the offloading energy of the system; i.e., \(E_{\text{off}}(i)\).

**Step 8:** If \(U = \emptyset\), stop. If not go to Step 3.

Such tasks involve a relatively simple-to-describe action being applied, independently, to multiple blocks of data. As such, the number of operations required to complete a fraction of the task can be modeled as being a function of the description length [23], [24], [27], cf. (1) and (4).

**A. Full Multiple Access**

In this section we will address the energy minimization problem for a \(K\)-user partial offloading system that employs the full multiple access scheme. The users are assumed to adopt dynamic voltage scaling so that they can minimize their local computation energy consumption. (In this setting the local latency constraint is satisfied implicitly and \(F_k(\cdot)\) takes the form in (4).) The optimization problem is formulated as

\[
\min_{\{R_k\}, \{P_k\}} \sum_{k} \frac{2^{2R_k}}{H_k} P_k + M_k \left(1 - \gamma_k\right) B_k \right)^3 \tag{23a}
\]

s.t. \(T_s \left(\frac{2^{2R_k}}{H_k}\right) + \delta \gamma_k B_k \leq \bar{L}_k, \ \forall k, \tag{23b}\)

\(0 \leq \gamma_k \leq 1, \ \forall k, \tag{23c}\)

\(0 \leq P_k, \ \forall k, \tag{23d}\)

\(\{R_k\} \in \mathcal{R}(\{P_k\}), \ \forall k, \tag{23e}\)

where \(\bar{L}_k = L_k - t_{\text{loc}}\). The achievable rate region of the full multiple access scheme was described in Section III-B.

Using the insights provided for a two-user case in [17], it can be shown that optimal solution of the problem in (23) is obtained when each user utilizes its maximum allowable latency, i.e., the constraints in (23b) hold with equality. Accordingly, the closed-form solution for the optimal fraction of bits offloaded by the \(k\)th user is

\[
\gamma_k = \frac{\bar{L}_k R_k}{H_k (P_k + \gamma_k B_k)}, \tag{24}\]
and the problem in (23) can be reduced to
\[
\min_{\{R_k\}, \{P_k\}} \sum_k \frac{T_k L_k}{T_k + \delta_k R_k} P_k + \frac{M_k}{L_k} (B_k - \frac{L_k R_k}{T_k + \delta_k R_k})^3 \quad (25a)
\]
s.t. \[0 \leq \frac{L_k R_k}{B_k (T_k + \delta_k R_k)} \leq 1, \quad \forall k, \quad (25b)\]
\[\text{23d}, \quad 23e.\]
where (25b) results from the constraints on the \(\gamma_k\)'s in (23c).

The problem in (25) can be decomposed as
\[
\min_{\{R_k\}} \min_{\{P_k\}} \sum_k \frac{T_k L_k}{T_k + \delta_k R_k} P_k \quad (26)
\]
s.t. \[23c, \quad 25b.\]
\[\text{s.t.} \quad 23d, \quad 23e.\]

It can be seen that for a given set of transmission rates, the objective function in (26) has a structure that is analogous to that of the objective function in (9). Hence, if the permutation \(\pi\) is defined so that
\[
\rho'_{\pi(K)} \leq \rho'_{\pi(K-1)} \leq \cdots \leq \rho'_{\pi(1)},
\]
where \[
\rho_k = \frac{T_k L_k}{T_k + \delta_k R_k},
\]
the following closed-form optimal solution for the transmission powers can be obtained
\[
P_{\pi(k)} = \left(\frac{\rho_{\pi(k)}-1}{\alpha_{\pi(k)}}\right)2^{\sum_{j=1}^{k-1} R_{\pi(j)}}. \quad (27)
\]

As in the full offloading case, the ordering in (27) also specifies the decoding order that enables the rates that will be found in (30) to be achieved using successive decoding.

Now the outer optimization problem in (26) becomes
\[
\min_{\{R_k\}} \sum_k \frac{T_k L_k}{T_k + \delta_k R_k} \left(\frac{\rho_{\pi(k)}-1}{\alpha_{\pi(k)}}\right)2^{\sum_{j=1}^{k-1} R_{\pi(j)}}
\]
\[+ \sum_k \frac{M_k}{L_k} (B_k - \frac{L_k R_k}{T_k + \delta_k R_k})^3 \quad (30a)\]
s.t. \[23b.\]
\[0 \leq \frac{L_k R_k}{B_k (T_k + \delta_k R_k)} \leq 1, \quad \forall k. \quad (30b)\]

We have shown in Appendix B that the objective function in (30) is quasi-convex in terms of each \(R_k\) when the other transmission rates are given. In addition, the constraints on the transmission rates are separable. Therefore, the coordinate descent algorithm can be employed to find a stationary solution for the transmission rates in (30), e.g., [35] Theorem 1].

Using the obtained solutions for the transmission rates, we can update the values of the \(\rho_k\)'s in (28) and consequently the optimal values of the transmission powers in (29). By substituting the updated transmission powers into the problem in (30), updated solutions for the transmission rates can be achieved. The resulting iterative algorithm is summarized in Algorithm 3. While the development of a formal convergence analysis of Algorithm 3 remains a work in progress, in our numerical experience, some of which is reported in Section V, the algorithm always converged.

**B. Time Division Multiple Access**

We now tackle the total energy minimization problem for a TDMA-based \(K\)-user system with divisible computational tasks. As discussed in Section III-C, ordering the users such that \(L_1 \leq L_2 \leq \cdots \leq L_K\) simplifies the description of the latency constraints and does so without loss of generality.

With that ordering, the energy minimization problem for the optimized dynamic voltage scaling architecture is
\[
\min_{\{R_k\}, \{P_k\}} \sum_k T_k \gamma_k B_k P_k + \frac{M_k}{L_k} (1 - \gamma_k B_k)^3 \quad (31a)
\]
s.t. \[\sum_{i=1}^{k} T_i \gamma_i B_i + \delta_k \gamma_k B_k \leq \bar{L}_k, \quad \forall k, \quad (31b)\]
\[0 \leq \gamma_k \leq 1, \quad \forall k, \quad (31c)\]
\[0 \leq P_k, \quad \forall k, \quad (31d)\]
\[0 \leq R_k \leq \log_2(1 + \alpha_k P_k), \quad \forall k. \quad (31e)\]

It can be shown that for a given set of \(\{R_k\}, \{\gamma_k\}\) the objective function in (31a) is increasing in terms of \(P_k\). Since the constraints on the powers, (31d) and (31e), are separable, the optimal value of transmission power of each user is the minimum feasible value. Hence, the optimal power of user \(k\) is
\[P_k = \frac{2^{\alpha_k-1}}{\alpha_k}.\]
If we let \(B_k' = \gamma_k B_k\) and \(t_k = \frac{B_k'}{R_k}\) denote the offloaded portion of the computational task for the \(k\)th user, and the time it takes to offload that portion to the access point, respectively, the problem in (31) can then be written as
\[
\min_{\{B_k'\}, \{t_k\}} \sum_k T_k t_k \left(\frac{B_k' - B_k'^2}{\alpha_k}\right) + \frac{M_k}{L_k} (B_k - B_k'^3) \quad (32a)
\]
s.t. \[\sum_{i=1}^{k} T_i t_i + \delta_k B_k' \leq \bar{L}_k, \quad \forall k, \quad (32b)\]
\[0 \leq B_k' \leq B_k, \quad \forall k, \quad (32c)\]
\[0 \leq t_k, \quad \forall k. \quad (32d)\]

It is shown in Appendix C that this problem is jointly convex in terms of \(t_k\) and \(B_k'\) and hence, the optimal solution of the problem can be efficiently obtained.

**V. Numerical Results**

In this section, we will evaluate the performance of the proposed energy minimization algorithms in both binary offloading and partial offloading scenarios, using either the full multiple access scheme (FullMA) or TDMA. We will compare the performance and computational cost of the proposed algorithms to those in [24] and [21]. The approach in [24] is a “full multiple access” approach, but is constrained to the case in which the latencies of the users are the same. Furthermore, the algorithm in [24] does not exploit as much of the algebraic structure of the problem as our algorithm and hence its computational cost grows more quickly than that of the proposed algorithm; see the discussion in Section III-D1. The approach in [21] tackles the energy minimization problem and...
for partial offloading in the TDMA case. Like the approach in \cite{24}, it is also constrained to the case in which the latencies of all users are the same. The approach in \cite{21} is developed for conventional local computing architectures whereas the proposed approaches and those in \cite{24} are developed for the dynamic voltage scaling architecture.

We will consider a cell of radius 1,000m over which the users are uniformly distributed. The symbol interval is $T_s = 10^{-8}$s and we consider a (discrete-time) slowly fading channel model with a path-loss exponent of 3.7 and independent Rayleigh distributed small-scale fading. The receiver noise variance is set to $\sigma^2 = 10^{-13}$. The energy consumption in each experiment is averaged over 100 channel realizations. We assume that the time it takes to download the results to the mobile users is equal for all the users, $t_{\text{DL}} = 0.2s$.

As explained in Section II, we consider data-partitionable computational tasks for which the (optimal) local energy consumption can be modeled as a function of number of bits; see \cite{23}. In order to be consistent with the measurements in \cite{23}, we set the constants $M_k$ in the local energy consumption expression in \cite{23} to $M_k = 10^{-19}$ \cite{23, 27}. In order to make fair comparisons with the conventional local computational architecture considered in \cite{21} we consider problems that require 1,000 computational cycles per bit, and we set the local computing energy per cycle for each user in such a way that that user is able to complete its computational task locally within its latency constraint.

A. Binary Computation Offloading

In the first phase of our numerical experiments we will view the tasks as being indivisible, and hence the users should either offload their task or complete it locally. We will begin by considering a four-user system in which the users latencies are different, $[L_1, L_2, L_3, L_4] = [1, 2, 1.5, 1.8, 2.5]s$, and we will examine the energy consumption of FullMA and TDMA-based binary offloading systems as the (different) description lengths of the tasks grow (in proportion); $[B_1, B_2, B_3, B_4] = \zeta \times [2, 1, 3, 4] \times 10^6$ bits. We apply the proposed greedy algorithm (Algorithm 2) to both the FullMA and TDMA schemes to find a good set of offloading users and the corresponding power and rate allocation. We will compare the energy consumption of these schemes to that of schemes in which the offloading set is chosen by deterministic rounding of the solution of the corresponding partial offloading problem, and to a scheme that selects the best solution from the deterministically rounded case and $(K-1)$ randomized roundings; see Section III-D2.

In the case of FullMA we compare the performance and the computational cost of the proposed algorithm with those of the binary offloading algorithm proposed in \cite{24}.

Fig. 1 plots the average energy consumption of the four-user system as the problem sizes grow. Our first observation is that the proposed greedy search algorithm to find a set of offloading users provides close-to-optimal performance for both FullMA and TDMA, and significantly better performance than the deterministic rounding approach. In this setting, the optimized TDMA scheme performs quite well, but in other scenarios that we will consider (Figs 2-4 and 5) the optimized FullMA scheme enables a significantly larger reduction in the energy consumption.

It can be seen from Fig. 1 that utilizing the maximum available latencies of the users enables the proposed algorithm to substantially reduce the energy consumption compared to the algorithm in \cite{24}, in which the users are assumed to have the same latency constraints. The performance gap increases quite quickly as the sizes of the problems increase. In the “No Offloading” approach in Fig. 1 the users complete their tasks locally employing the dynamic voltage scaling approach, by which they can minimize the local energy consumption within their maximum available deadlines. Interestingly, the energy consumption when all users complete their tasks locally using the maximum available latency is substantially less than that of the latency-equal algorithm proposed in \cite{24} and the case in which the offloading set is chosen by deterministically rounding the solution to the partial offloading problem with different latencies.

In order to compare the computational costs of the proposed FullMA algorithm with that in \cite{24}, Table I provides the average CPU times. These times are essentially independent of the description length of the tasks. All the algorithms were coded in MATLAB, with similar diligence paid to the efficiency of the programs. The convex optimization subproblems in the method in \cite{24} were solved using SDPT3 \cite{37} through the CVX interface \cite{38}. The CPU times were evaluated on a MacBook Pro with a Core i5 processor running at 3.1GHz, and 8GB of RAM. It can be seen that the closed-form optimal solution that we have obtained for any given set of offloading users significantly reduces the computational cost of our proposed algorithm in comparison to the algorithm in \cite{24}. As discussed in Section III-D1, the main reason for such a significant computational cost reduction is that at each iteration of the proposed algorithm the optimal closed-form solution for a given set of offloading users is obtained with the cost of order $O(K \log K)$, while at each iteration of the algorithm proposed in \cite{24} an optimization problem needs to

---

![Fig. 1: Average energy consumption of a binary offloading system with four users with different latency constraints versus the parameter that defines the required number of bits to describe the users’ tasks.](image-url)
be solved by employing the ellipsoid method which involves matrix inversion with the cost of order \( O(K^3) \). (As suggested in [24], for the ellipsoid method we employed the approach in [19], and we chose a termination criterion of \( \epsilon = 10^{-3} \).)

In our next numerical experiment for the binary offloading case, we examine the total energy consumption as the number of users increases. In this experiment we consider a scenario in which all the users have equal problem sizes and the same latency constraints. In particular, we set \( B_k = 6 \times 10^6 \) bits and \( L_k = 2s \). As in the previous experiment, the “randomized rounding” scheme refers to the selection of the best solution from offloading sets that are generated by a deterministic rounding of the partial offloading solution and \( (K - 1) \) randomized roundings.

In Fig. 2 we present the average energy consumption of the system versus the number of users. In this setting all of the considered methods provide a significant reduction in the energy consumption over the No Offloading case. In the case that the FullMA scheme is employed, it can be seen that since the latency constraints of all the users are equal, the algorithm in [24] can achieve the same performance as our proposed algorithm, for both greedy search and rounding approaches. However, Fig. 4 indicates that the computational cost of the proposed algorithm is significantly less than that of the algorithm in [24]. Fig. 2 also shows that by using the full capabilities of the channel, the FullMA scheme together with the proposed greedy search method can reduce the total energy consumption compared to the TDMA scheme with the same greedy approach.

### Table I: Average CPU times required for the proposed algorithms and the algorithms in [24] for a four-user binary offloading system that employs the full multiple access scheme.

| Algorithm               | Average CPU time (sec) |
|-------------------------|------------------------|
| Proposed FullMA Greedy Search | 4.6 \times 10^{-5}   |
| Proposed FullMA Rounding    | 4.0 \times 10^{-5}   |
| FullMA Greedy Search in [24] | 1.7 \times 10^3     |
| FullMA Rounding in [24]     | 0.2 \times 10^3      |

In Fig. 3 we present the average CPU time required for the proposed algorithms and the algorithm in [24] for different number of users when the full multiple access scheme is employed in binary offloading case.

B. Partial Computation Offloading

In the second phase of our numerical analysis we allow the users to partition their “data-partitionable” tasks and hence they can employ partial offloading. We first examine the energy consumption of the four-user system in Section V-A in which each user has its own deadline. Fig. 4 plots the total energy consumption as the problem sizes grow. It can be seen that our proposed algorithms, which benefit from the maximum available latency of each user, achieve substantially lower energy consumption than the existing techniques. Indeed, it can be seen that in the TDMA case, the energy consumption of the proposed algorithm is lower than that of the algorithm in [21], and the performance gap increases as the number of bits increases. That is because in the proposed algorithm the users not only utilize their maximum available deadline to complete their tasks, they also employ dynamic voltage scaling which minimizes the local energy consumption. The energy consumptions in Fig. 4 and the computational costs in Table II indicate that in the FullMA case the proposed algorithm can achieve significantly lower energy consumption than the algorithm in [24], and does so at much lower computational cost. Fig. 4 also exhibits the impact of the multiple access scheme. Using the FullMA scheme substantially reduces the total energy consumption over TDMA.

In our final numerical experiment we examine the energy consumption as the number of users increases for a partial offloading system with equal problem sizes and the same latency constraints. We set \( B_k = 4 \times 10^6 \) bits and \( L_k = 2s \). Fig. 5 like Fig. 4 shows that using the full capabilities of the channel enables the users to complete their computational tasks with significantly less energy consumption compared to TDMA. In the FullMA case, it can be seen in Fig. 5 that because the latencies of the users are equal, the algorithm in [24] can achieve the same performance as our proposed algo-
Fig. 4: Average energy consumption of a four-user partial offloading system with different latency constraints versus the coefficient that defines the description length of the tasks.

TABLE II: Average CPU times for the proposed algorithm and the algorithm in [24] for a four-user FullMA partial offloading system.

| Algorithm          | Average CPU time (sec) |
|--------------------|------------------------|
| Proposed FullMA    | $4.1 \times 10^{-3}$   |
| FullMA in [24]     | $1.9 \times 10^{2}$    |

Fig. 5: Average energy consumption of a partial offloading system, in which the users’ tasks have the same latency constraints, for different number of users.

Fig. 6: Average CPU time required for the proposed algorithm and the algorithm in [24] for different number of users when the full multiple access scheme is employed in partial offloading case.

VI. CONCLUSION

In this work, we have considered the problem of optimal uplink resource allocation in a $K$-user offloading system. In the binary offloading case, we obtained the optimal energy consumption of a given set of offloading users in the full multiple access and TDMA schemes, and then we proposed a customized greedy search algorithm to find a set of offloading users with close-to-optimal energy consumption. In the partial offloading case, the energy minimization problem was tackled by proposing a low-complexity algorithm for a stationary solution in the full multiple access case and by finding the optimal solution of a convex optimization problem when TDMA scheme is employed. Our strategy to decompose the optimization problem and to find the optimal values of some variables in terms of the others enabled us to significantly reduce the computational cost of our proposed algorithms compared to the existing algorithms in this area.

While the proposed resource allocation algorithms have significant advantages over the existing algorithms, like the existing algorithms they have been based on a single time slot for communication. Recent work on the two-user case [17] suggests that a further reduction in the energy consumption can be obtained by adopting a time-slotted structure in which different groups of users transmit in each time slot. One avenue for future work is the development of efficient resource allocation algorithms for the time-slotted structure.

APPENDIX A

EXPLOITING THE POLYMATROID STRUCTURE OF THE POWER FEASIBILITY REGION

Given the definition of class-$\ell$ constraints in Section III-B1, we will show that the candidate vertices are the result of simultaneous satisfaction with equality of a set of $K$ constraints in (8d) such that there is at most one constraint from each class. In order to prove this statement, let us assume that there are two constraints in (8d) that are satisfied with equality, namely $C_1$ and $C_2$, both of which belong to class-$c$. If $M_{com}$ denotes
the set of users that are present in both $C_1$ and $C_2$ and $M_{c_1}$ and $M_{c_2}$ denote the set of users that are participating only in $C_1$ and $C_2$ respectively, we can write

$$C_1 : 2 \left( \sum_{i \in M_{\text{com}}} R_i + \sum_{j \in M_{c_1}} R_j \right) = 1 + \sum_{i \in M_{\text{com}}} \alpha_i P_i + \sum_{j \in M_{c_1}} \alpha_j P_j,$$

$$C_2 : 2 \left( \sum_{i \in M_{\text{com}}} R_i + \sum_{k \in M_{c_2}} R_k \right) = 1 + \sum_{i \in M_{\text{com}}} \alpha_i P_i + \sum_{k \in M_{c_2}} \alpha_k P_k.$$

By adding the above two equations we have that

$$\sum_{i \in M_{\text{com}}} \alpha_i P_i + \sum_{j \in M_{c_1}} \alpha_j P_j + \sum_{k \in M_{c_2}} \alpha_k P_k.$$

In addition, there is a rate region constraint that includes all the users of the set $M_{\text{com}} \cup M_{c_1} \cup M_{c_2}$ namely,

$$2 \left( \sum_{i \in M_{\text{com}}} R_i + \sum_{j \in M_{c_1}} R_j + \sum_{k \in M_{c_2}} R_k \right) \leq 1 + \sum_{i \in M_{\text{com}}} \alpha_i P_i + \sum_{j \in M_{c_1}} \alpha_j P_j + \sum_{k \in M_{c_2}} \alpha_k P_k.$$

The right hand side of Equation (34) can be replaced by its equivalent term given on the left hand side of Equation (33). That results in

$$2 \left( \sum_{i \in M_{\text{com}}} R_i + \sum_{j \in M_{c_1}} R_j + \sum_{k \in M_{c_2}} R_k \right) \leq \left(2 \sum_{i \in M_{\text{com}}} R_i \right) \left( \sum_{j \in M_{c_1}} R_j + \sum_{k \in M_{c_2}} R_k \right)$$

$$- 1 - \sum_{i \in M_{\text{com}}} \alpha_i P_i \leq \left(2 \sum_{i \in M_{\text{com}}} R_i \right) \left( \sum_{j \in M_{c_1}} R_j + \sum_{k \in M_{c_2}} R_k + 1 \right).$$

Equation (35) is obtained from the rate region constraint $2 \sum_{i \in M_{\text{com}}} R_i \leq 1 + \sum_{i \in M_{\text{com}}} \alpha_i P_i$. By factoring out the term $2 \sum_{i \in M_{\text{com}}} R_i$, we obtain

$$2 \sum_{j \in M_{c_1}} \sum_{k \in M_{c_2}} R_k \leq 2 \sum_{i \in M_{\text{com}}} R_i + 2 \sum_{k \in M_{c_2}} R_k - 1$$

$$0 \leq \left(2 \sum_{i \in M_{\text{com}}} R_i - 1 \right) \left( 1 - 2 \sum_{k \in M_{c_2}} R_k \right),$$

which is a contradiction, because of the fact that $0 \leq R_i$ and hence $0 \leq 2 \sum_{j \in M_{c_1}} R_j - 1$ for any $j$. Therefore, at an optimal vertex of the inner problem in (9) no more than one constraint from any class can hold with equality.

**APPENDIX B**

**QUASI-CONVEXITY OF THE OBJECTIVE FUNCTION IN (30)**

A function $f$ is quasi-convex if at least one of the following conditions holds [40]: (a) $f$ is non-increasing; (b) $f$ is non-decreasing; (c) there is a (turning) point, $c$, such that for any $x \leq c$ the function $f(x)$ is non-increasing and for any $x \geq c$ the function $f(x)$ is non-decreasing.

In this section we will show that for each $R_k$, when the other transmission rates are constant, the objective function in (30) will satisfy either condition (b) or condition (c). To do so, we will rewrite the objective function as

$$f_k = \lambda_k \left( \frac{R_k}{T_k^2 + \delta_k R_k} \right) + \Omega_k 2 R_k + \frac{R_k}{T_k^2 + \delta_k R_k}.$$
in $H$ is positive semidefinite; i.e., $H_{22} - H_{12}^TH_{11}^{-1}H_{12} \succeq 0$. Hence, the matrix $H$ is positive semidefinite \[40\] and the objective function is jointly convex.

REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” IEEE Commun. Surveys Tuts., vol. 19, no. 4, pp. 2322–2358, 2017.
[2] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing—A key technology towards 5G?”
[3] P. Much and Z. Bicvar, “Mobile edge computing: A survey on architecture and computation offloading,” IEEE Commun. Surveys Tuts., vol. 19, no. 3, pp. 1628–1656, 2017.
[4] 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. ACM Int. Conf. Mobile Syst. Appl. Serv., San Francisco, CA, USA, Jun. 2010, pp. 49–62.
[5] B.-G. Chun, S. Ihm, P. Muniatis, M. Naik, and A. Pattì, “CloneCloud: Elastic execution between mobile device and cloud,” in Proc. ACM Conf. Computer Syst., 2011, pp. 301–314.
[6] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in Proc. IEEE INFOCOM, Mar. 2012, pp. 945–953.
[7] M. 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 Intl. Conf. Comput. Commun., Turin, Apr. 2013, pp. 1285–1293.
[8] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, “Challenges on wireless heterogeneous networks for mobile cloud computing,” IEEE Wireless Commun., vol. 20, no. 3, pp. 34–44, 2013.
[9] F. Liu, P. Shu, H. Jia, L. Ding, J. Yu, D. Niu, and B. Li, “Gearing resource-poor mobile devices with powerful clouds: Architectures, challenges, and applications,” IEEE Wireless Commun., vol. 20, no. 3, pp. 14–22, 2013.
[10] S. Sardellitti, G. Scutari, and S. Barbarossa, “Joint optimization of radio and computational resources for multicell mobile-edge computing,” IEEE Trans. Signal Info. Process. over Network, vol. 1, no. 2, pp. 89–103, 2015.
[11] M.-H. Chen, B. Liang, and M. Dong, “A semi-definite relaxation approach to mobile cloud offloading with computing access point,” in Proc. IEEE Workshop Signal Process. Adv. Wireless Commun., Stockholm, Jun. 2015, pp. 186–190.
[12] O. Munoz, A. Pascual-Iseret, and J. Vidal, “Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading,” IEEE Trans. Veh. Technol., vol. 64, no. 10, pp. 4738–4755, 2015.
[13] O. Muñoz, A. Pascual-Iseret, J. Vidal, and M. Molina, “Energy-latency trade-off for multiuser wireless computation offloading,” in Proc. IEEE Wireless Commun. Network Conf. Workshops, Istanbul, Apr. 2014, pp. 29–33.
[14] M. Salmani and T. N. Davidson, “Multiple access computational offloading,” in Proc. IEEE Workshop Signal Process. Adv. Wireless Commun., Edinburgh, Scotland, Jul. 2016, pp. 1–6.
[15] F. Wang, J. Xu, and Z. Ding, “Optimized multiuser computation offloading with multi-antenna NOMA,” arXiv preprint arXiv:1707.02486v1, 2017.
[16] F. Wang, J. Xu, X. Wang, and S. Cui, “Joint offloading and computing optimization in wireless powered mobile-edge computing systems,” arXiv preprint arXiv:1702.00606, 2017.
[17] M. Salmani and T. N. Davidson, “Multiple access computational offloading: Communication resource allocation in the two-user case (extended version),” 2018. [Online]. Available: https://arxiv.org/abs/1805.10638
[18] Z. Ding, J. Xu, O. A. Dobre, and H. V. Poor, “Joint power and time allocation for NOMA-MEC offloading,” 2018. [Online]. Available: https://arxiv.org/abs/1807.06306
[19] M.-H. Chen, M. Dong, and B. Liang, “Joint offloading decision and resource allocation for mobile cloud with computing access point,” in Proc. IEEE Intl. Conf. Acoust. Speech, Signal Process., Shanghai, China, Mar. 2016, pp. 3516–3520.
[20] ——, “Multi-user mobile cloud offloading game with computing access point,” in Proc. IEEE Intl. Conf. Cloud Netw. (Cloudnet), Pisa, Italy, Oct. 2016, pp. 64–69.
[21] C. You, K. Huang, H. Chae, and B.-H. Kim, “Energy-efficient resource allocation for mobile-edge computation offloading,” IEEE Trans. Wireless Commun., vol. 16, no. 3, pp. 1397–1411, 2017.
[22] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, “Mobile-edge computing: Partial computation offloading using dynamic voltage scaling,” IEEE Trans. Commun., vol. 64, no. 10, pp. 4268–4282, 2016.
[23] D. N. C. Tse and S. V. Hanly, “Multiaccess fading channels. I. Polymatroid structure, optimal resource allocation and throughput capacities,” IEEE Trans. Inf. Theory, vol. 44, no. 7, pp. 2796–2815, 1998.
[24] F. Wang, J. Xu, and Z. Ding, “Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems,” 2018. [Online]. Available: https://arxiv.org/abs/1707.02486v3
[25] K. Kumar and Y.-H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?” IEEE Computer, vol. 43, no. 4, pp. 51–56, 2010.
[26] H. Wu, Q. Wang, and K. Wolter, “Tradeoff between performance improvement and energy saving in mobile cloud offloading systems,” IEEE Int. Conf. Commun. Workshops (ICC), pp. 728–732, Jun. 2013.
[27] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, “Energy-optimal mobile cloud computing under stochastic wireless channel,” IEEE Trans. Wireless Commun., vol. 12, no. 9, pp. 4569–4581, 2013.
[28] T. M. Cover and J. A. Thomas, Elements of Information Theory, 2nd ed. John Wiley & Sons, 2012.
[29] Y. Polyanskiy, H. V. Poor, and S. Verdú, “Channel coding rate in the finite blocklength regime,” IEEE Trans. Inf. Theory, vol. 56, no. 5, pp. 2307–2359, 2010.
[30] E. M. Jazi and J. N. Laneman, “Simpler achievable rate regions for multiaccess with finite blocklength,” in Proc. IEEE Intl. Symp. Information Theory, 2012, pp. 36–40.
[31] M. Salmani and T. N. Davidson, “Multiple access binary computational offloading in the $K$-user case,” May 2018, to appear in the Conf. Rec. 52nd Asilomar Conf. Signals. Syst. Comput.
[32] ——, “Multiple access partial computational offloading: Two-user case,” in 23rd Asia-Pacífic Conf. Commun., Perth, Australia, Dec. 2017.
[33] A. E. Gamal and T. M. Cover, “Multiple user information theory,” Proc. IEEE, vol. 68, no. 12, pp. 1466–1483, 1980.
[34] P. Raghavan and C. D. Tompson, “Randomized rounding: A technique for provably good algorithms and algorithmic proofs,” Combinatorica, vol. 7, no. 4, pp. 365–374, 1987.
[35] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, “A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing,” IEEE Signal Process. Mag., vol. 33, no. 1, pp. 57–77, 2016.
[36] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in Proc. USENIX Workshop Hot Topics Cloud Comput. (HotCloud), Boston, US, Jun. 2010, pp. 4–11.
[37] K.-C. Toh, M. J. Todd, and R. H. Tutuncu, “SDPT3—A MATLAB package for semidefinite programming, version 1.3,” Optim. Methods Softw., vol. 11, no. 1-4, pp. 545–581, 1999.
[38] M. Grant, S. Boyd, and Y. Ye, CVX: MATLAB software for disciplined convex programming, 2008. [Online]. Available: http://cvxr.com/cvx/
[39] S. Boyd, “Ellipsoid method,” Notes for EE364B, Stanford Univ. Stanford, CA, USA, 2018. [Online]. Available: http://web.stanford.edu/class/ee364b/lectures/ellipsoid_method_notes.pdf
[40] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge University Press, 2004.