Joint Offloading Decision and Resource Allocation for Multi-user Multi-task Mobile Cloud

Meng-Hsi Chen[†], Ben Liang[†], Min Dong[‡]

[†]Dept. of Electrical and Computer Engineering, University of Toronto, Canada
[‡]Dept. of Electrical, Computer and Software Engineering, University of Ontario Institute of Technology, Canada

***Abstract*—We consider a general multi-user mobile cloud computing system where each mobile user has multiple independent tasks. These mobile users share the communication resource while offloading tasks to the cloud. We aim to jointly optimize the offloading decisions of all users as well as the allocation of communication resource, to minimize the overall cost of energy, computation, and delay for all users. The optimization problem is formulated as a non-convex quadratically constrained quadratic program, which is NP-hard in general. An efficient approximate solution is proposed by using separable semidefinite relaxation, followed by recovery of the binary offloading decision and optimal allocation of the communication resource. For performance benchmark, we further propose a numerical lower bound of the minimum system cost. By comparison with this lower bound, our simulation results show that the proposed algorithm gives nearly optimal performance under various parameter settings.**

I. Introduction

Mobile cloud computing extends the capabilities of mobile devices and improves the user experience with the help of abundant cloud resources [1] [2]. By offloading tasks to the cloud, mobile users aim to reduce its own energy consumption. However, the quality of service (QoS) of those offloaded tasks may be affected since there are additional costs such as the communication delay between mobile users and the cloud [3].

To reduce transmission latency, the authors of [4] proposed an architecture replacing the remote cloud with nearby cloudlets. Authors in [3] [5] studied the energy saving and performance tradeoff when a single user offloading its entire application to the cloud. Multi-user scenarios with a single application or task per user were addressed in [6] [7] [8]. Different from the whole-application offloading in above studies, the authors of [9]–[12] considered application partitioning for a *single user*. In [9], the authors proposed a system named MAUI to efficiently process the partitioned tasks. Clonecloud [10] and Thinkair [11] were proposed with further improvements. In [12], a multi-radio interface scenario was considered. In all cases, the partitioning problem results in difficult integer programs. In [13], we also studied the single user scenario with multiple independent tasks and a computing access point. The offloading decisions were make by considering the worst-case offloading delay.

In this work, we further study the interaction between *multiple users* and the cloud. The multi-user scenario adds substantial challenge to system design, since we need to jointly consider both the offloading decisions and the sharing of communication resource among all users as they compete to reach the cloud through a wireless link with limited capacity. We aim to conserve energy and maintain the QoS for all users. In particular, the transmission delays of the offloaded tasks of a user will be affected by its assigned communication resource. Furthermore, optimal offloading decision and resource allocation must take into consideration the computation and communication energies, system utility cost, and communication and processing delays at all local user devices and the remote cloud.

We focus on jointly optimizing the offloading decision and the communication resource allocation of all tasks, to minimize a weighted sum of the costs of energy, computation, and the delay for all users. The resultant mixed integer programming problem can be reformulated as a non-convex quadratically constrained quadratic program (QCQP) [14], which is NP-hard in general. To solve this challenging problem, we propose an efficient heuristic algorithm based on separable semidefinite relaxation (SDR) [15], with recovery of the binary offloading decision and subsequent optimal allocation of the communication resource. We also provide the lower bound of the minimum cost as the benchmark for performance comparison. Simulation results show that the proposed algorithm gives nearly optimal performance under various parameter settings. Furthermore, our method is scalable to a large number of users and tasks where computational complexity of exhaustive search becomes prohibitive.

The rest of this paper is organized as follows. In Section II, we describe the system model and present the problem formulation. In Section III, we provide details of the proposed algorithm and the lower bound of minimum system cost. We study the numerical results in Section IV and conclude in Section V.

II. System Model and Problem Formulation

A. Mobile Cloud Offloading with Multiple Users and Tasks

Consider a general cloud access network consisting of one cloud server, one access point (AP), and N mobile users, each having M independent tasks, as shown in Fig. 1. Note that our

This work has been funded in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant STPGP 447497-13.

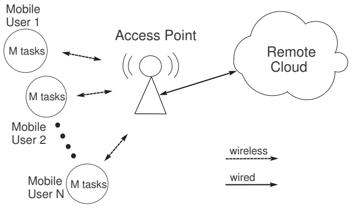

Fig. 1. System model

system model can be easily extended to the case where each mobile user has a different number of tasks. The connections between mobile users and the AP are wireless, while a wired connection is used between the AP and the cloud. Each mobile user can process its tasks locally or offload some of them to the cloud for processing through the AP. Let x_{ij} denote the offloading decision for task j of user i, given by

$$x_{ij} = \begin{cases} 0 & \text{process task } j \text{ of user } i \text{ locally;} \\ 1 & \text{offload task } j \text{ of user } i \text{ to the cloud.} \end{cases}$$

B. Cost of Local Processing

The input and output data sizes and the application type of task j of user i are denoted by $D_{\text{in}}(ij)$, $D_{\text{out}}(ij)$, and $\text{App}(ij)$, respectively. In this work, $\text{App}(ij)$ measures the number of processing cycles per input data. For task j being locally processed by user i, the corresponding energy consumed for processing is denoted by $E_{l_{ij}}$ and the processing time is denoted by $T_{l_{ij}}$.

C. Cost of Remote Processing

For task j of user i being offloaded to the cloud through the AP, we denote the energy consumed for transmitting and receiving data between the mobile user and the AP by $E_{t_{ij}}$ and $E_{r_{ij}}$, respectively. For the wireless connections between mobile users and the AP, we denote the uplink and downlink transmission times by $T_{t_{ij}} = D_{\text{in}}(ij)/c_{u_i}$ and $T_{r_{ij}} = D_{\text{out}}(ij)/c_{d_i}$, respectively, where c_{u_i} and c_{d_i} are uplink and downlink bandwidths allocated to user i for data transmission. Their values are constrained by the capacities of the corresponding wireless links, denoted by C_{UL} and C_{DL}, respectively, as well as the number of tasks of other users offloaded to the cloud through the AP.

Since the AP has to further offload the task to the cloud, we denote the required transmission time between the AP and the cloud by $T_{ac_{ij}} = (D_{\text{in}}(ij) + D_{\text{out}}(ij))/R_{ac}$, and the cloud processing time by $T_{c_{ij}} = D_{\text{in}}(ij)\text{App}(ij)/f_C$. We assume the wired transmission rate R_{ac} between the AP and the cloud and the cloud processing rate f_C *for each user* are pre-fixed values. Thus, $T_{ac_{ij}}$ and $T_{c_{ij}}$ only depend on the size of each task itself. Finally, the system utility cost of processing user i's task j at the cloud is denoted by $C_{c_{ij}}$. The above notations are summarized in Table I.

D. Problem Formulation

We aim at reducing mobile users' energy consumption and maintain the QoS of processing their tasks, measured by the delays incurred due to transmission and/or processing. For this goal, we define the total system cost as the weighted sum of total energy consumption, the costs to offload and process all tasks, and the corresponding transmission and processing delays for all users. Our objective is to minimize the total system cost by jointly optimizing the task offloading decisions x_{ij} and the communication bandwidth resource allocation $\mathbf{r}_i = (c_{u_i},\ c_{d_i})$. This optimization problem is formulated as follows:

TABLE I
NOTATIONS AND THEIR CORRESPONDING DESCRIPTIONS.

Notation	Description
$E_{l_{ij}}$	local processing energy of user i's task j
$E_{t_{ij}}$, $E_{r_{ij}}$	uplink transmitting energy and downlink receiving energy of user i's task j between the mobile user and the AP
$T_{l_{ij}}$, $T_{c_{ij}}$	local processing time and cloud processing time of user i's task j
$T_{t_{ij}}$, $T_{r_{ij}}$	uplink transmission time and downlink transmission time of user i's task j between the mobile user and the AP
$T_{ac_{ij}}$	transmission time of user i's task j between the AP and the cloud
C_{UL}, C_{DL}	uplink transmission capacity and downlink transmission capacity between mobile users and the AP
c_{u_i}, c_{d_i}	uplink transmission rate and downlink transmission rate assigned to user i's tasks
$C_{c_{ij}}$	system utility cost of user i's task j
R_{ac}	transmission rate for each user between the AP and the cloud
f_C	cloud processing rate for each user
β	weight of the system utility cost
ρ_i	weight of the delay of processing user i's tasks

$$\min_{\{x_{ij}\},\{\mathbf{r}_i\}} \quad \sum_{i=1}^{N} \left[\sum_{j=1}^{M} (E_{l_{ij}}(1 - x_{ij}) + E_{C_{ij}} x_{ij}) + \rho_i \max\{T_{L_i}, T_{C_i}\} \right] \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} c_{u_i} \leq C_{UL}, \tag{2}$$

$$\sum_{i=1}^{N} c_{d_i} \leq C_{DL}, \tag{3}$$

$$c_{u_i}, c_{d_i}, \geq 0, \forall i, \tag{4}$$

$$x_{ij} \in \{0, 1\}, \forall i, j, \tag{5}$$

where $E_{C_{ij}} \triangleq (E_{t_{ij}} + E_{r_{ij}} + \beta C_{c_{ij}})$ is the energy cost of offloading and processing task j of user i to the cloud; it is a weighted sum of transmission energy and system utility cost, with β being the relative weight; In addition, $T_{L_i} \triangleq \sum_{j=1}^{M} T_{l_{ij}}(1 - x_{ij})$ is the processing delay of tasks processed by the mobile user i itself, T_{C_i} is the overall transmission and remote-processing delay for tasks of mobile user i processed at the cloud, and ρ_i is the weight on the task processing delay relative to energy consumption in the total system cost. Depending on the performance requirement, the value of ρ_i can be adjusted to impose different emphasis on delay and energy consumption. Constraints (2) and (3) are the uplink and downlink bandwidth resource constraints.

The above mixed-integer programming problem is difficult to solve in general. Moreover, we note that the overall delay for remote processing, T_{C_i}, is challenging to calculate exactly. This is because, when there are multiple tasks offloaded by a users, the transmission times and processing times may overlap in an unpredictable manner, which depends on the offloading decision, communication resource allocation, and task scheduling order. In fact, since T_{C_i} consists of the uplink transmission times, remote-processing time, and downlink transmissions times of all tasks, it may be viewed as the output of a multi-machine flowshop schedule, which remains an open research problem [16]. Since T_{C_i} is not precisely tractable, we will use both upper and lower bounds of T_{C_i} in our proposed solution and performance benchmarking. They are shown to give total system costs that are close to each other.

III. MULTI-USER MULTI-TASK OFFLOADING SOLUTION

The joint optimization problem (1) is a mixed-integer non-convex programming problem, which is NP-hard in general. To find an efficient solution to the original problem (1), in the following, we first propose both upper bound and lower bound formulations of T_{C_i}, then transform the optimization problem (1) into a separable QCQP, and finally propose a separable SDR approach to obtain the binary offloading decisions $\{x_{ij}\}$ and the communication resource allocation $\{\mathbf{r}_i\}$.

A. Bounds of Remote-Processing Delay

When a mobile user offloads more than one task to the cloud, there will be overlaps in the communication and processing times as mentioned above, making it difficult to exactly characterize the overall delay T_{C_i}. Next, we first provide a upper bound of T_{C_i} as the *worst case delay* formulation:

$$T^U_{C_i} \triangleq \sum_{j=1}^{M}(\frac{D_{\text{in}}(ij)}{c_{u_i}} + \frac{D_{\text{out}}(ij)}{c_{d_i}} + T_{ac_{ij}} + T_{c_{ij}})x_{ij}, \forall i. \quad (6)$$

Since the worst case delay sums the transmission delays and processing delays together without any overlap, it will always be greater than the real delay given the same offloading decision and resource allocation. On the other hand, we separate the offloading delays of all mobile users into several components and only consider the largest one as the lower bound of T_{C_i}:

$$T^L_{C_i} \triangleq \max\{T_{u_i}, T_{d_i}, T_{uac_i}, T_{dac_i}, T_{C'_i}\}, \forall i, \quad (7)$$

where $T_{u_i} \triangleq \sum_{j=1}^{M} D_{\text{in}}(ij)x_{ij}/c_{u_i}$ and $T_{d_i} \triangleq \sum_{j=1}^{M} D_{\text{out}}(ij)x_{ij}/c_{d_i}$ are total uplink and downlink transmission time between the user and the AP for user i, respectively, $T_{uac_i} \triangleq \sum_{j=1}^{M} D_{\text{in}}(ij)x_{ij}/R_{ac}$ and $T_{dac_i} \triangleq \sum_{j=1}^{M} D_{\text{out}}(ij)x_{ij}/R_{ac}$ are total uplink and downlink transmission time between the AP and the cloud for user i, respectively, and $T_{C'_i} \triangleq \sum_{j=1}^{M} D_{\text{in}}(ij)\text{App}(ij)x_{ij}/f_C$ is the total cloud processing time for user i.

In the following, we will use the worst case delay $T^U_{C_i}$ in optimization problem (1) to obtain an approximate solution, which can provide an upper bound to the total system cost. We then use $T^L_{C_i}$ similarly, to obtain a lower bound of the total system cost, for performance benchmarking. In Section VI, by comparing both cases, we show that the proposed algorithm based on the worst case formulation gives nearly optimal performance.

B. QCQP Transformation and Semidefinite Relaxation

We transform the optimization problem (1) with $T^U_{C_i}$ into a separable QCQP. We first rewrite the integer constraint (5) as

$$x_{ij}(x_{ij} - 1) = 0, \forall i, j. \quad (8)$$

Furthermore, we introduce additional auxiliary variables t_i for $\max\{T_{L_i}, T^U_{C_i}\}$, D_{u_i} for $\sum_{j=1}^{M} D_{\text{in}}(ij)x_{ij}/c_{u_i}$, and D_{d_i} for $\sum_{j=1}^{M} D_{\text{out}}(ij)x_{ij}/c_{d_i}$ in the optimization problem (1). Let $\mathbf{d}_i = (D_{u_i}, D_{d_i})$. The problem (1) is now transformed into the following equivalent problem

$$\begin{aligned}
\min_{\{x_{ij}\},\{\mathbf{r}_i,\mathbf{d}_i,t_i\}} \quad & \sum_{i=1}^{N}\left[\sum_{j=1}^{M}(E_{l_{ij}}(1-x_{ij}) + E_{C_{ij}}x_{ij}) + \rho_i t_i\right] \quad (9)\\
\text{s.t.} \quad & \sum_{j=1}^{M} T_{l_{ij}}(1-x_{ij}) \le t_i, \forall i\\
& D_{u_i} + D_{d_i} + \sum_{j=1}^{M}(T_{ac_{ij}} + T_{c_{ij}})x_{ij} \le t_i, \forall i\\
& \sum_{j=1}^{M} D_{\text{in}}(ij)x_{ij} - c_{u_i}D_{u_i} \le 0, \forall i\\
& \sum_{j=1}^{M} D_{\text{out}}(ij)x_{ij} - c_{d_i}D_{d_i} \le 0, \forall i\\
& (2) - (4), \text{ and } (8).
\end{aligned}$$

Define $\mathbf{w}_i \triangleq [x_{i1}, \ldots, x_{iM}, c_{u_i}, D_{u_i}, c_{d_i}, D_{d_i}, t_i]^T$, $\forall i$, and $\mathbf{e}_i$ as the $(M+5)\times 1$ standard unit vector with the ith entry being 1. The optimization problem (9) can now be further transformed into the following equivalent separable QCQP formulation

$$\begin{aligned}
\min_{\{\mathbf{w}_i\}} \quad & \sum_{i=1}^{N}\mathbf{b}_i^T\mathbf{w}_i + \sum_{i=1}^{N}\sum_{j=1}^{M} E_{l_{ij}} \quad (10)\\
\text{s.t.} \quad & \mathbf{b}_{l_i}^T\mathbf{w}_i \le -\sum_{j=1}^{M} T_{l_{ij}}, \quad \mathbf{b}_{c_i}^T\mathbf{w}_i \le 0, \ \forall i\\
& \mathbf{w}_i^T\mathbf{A}_{u_i}\mathbf{w}_i + \mathbf{b}_{u_i}^T\mathbf{w}_i \le 0, \ \forall i\\
& \mathbf{w}_i^T\mathbf{A}_{d_i}\mathbf{w}_i + \mathbf{b}_{d_i}^T\mathbf{w}_i \le 0, \ \forall i\\
& \sum_{i=1}^{N}\mathbf{b}_{U_i}^T\mathbf{w}_i = C_{\text{UL}}, \quad \sum_{i=1}^{N}\mathbf{b}_{D_i}^T\mathbf{w}_i = C_{\text{DL}}\\
& \mathbf{w}_i^T\text{diag}(\mathbf{e}_j)\mathbf{w}_i - \mathbf{e}_j^T\mathbf{w}_i = 0, \ \forall i, j
\end{aligned}$$

$$\mathbf{w}_i \geq \mathbf{0}, \forall i$$

where

$$\mathbf{A}_{u_i} \triangleq \begin{bmatrix} \mathbf{0}_{M\times M} & \mathbf{0}_{M\times 2} & \mathbf{0}_{M\times 3} \\ \mathbf{0}_{2\times M} & \mathbf{A}'_{u_i} & \mathbf{0}_{2\times 3} \\ \mathbf{0}_{3\times M} & \mathbf{0}_{3\times 2} & \mathbf{0}_{3\times 3} \end{bmatrix},$$

$$\mathbf{A}_{d_i} \triangleq= \begin{bmatrix} \mathbf{0}_{(M+2)\times(M+2)} & \mathbf{0}_{(M+2)\times 2} & \mathbf{0}_{(M+2)\times 1} \\ \mathbf{0}_{2\times(M+2)} & \mathbf{A}'_{d_i} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times(M+2)} & \mathbf{0}_{1\times 2} & 0 \end{bmatrix},$$

$$\mathbf{A}'_{s_i} \triangleq -0.5 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \text{for } s = u, d,$$

$$\mathbf{b}_i \triangleq [(E_{C_{i1}} - E_{l_{i1}}), \ldots,\ (E_{C_{iM}} - E_{l_{iM}}),\ \mathbf{0}_{1\times 4},\ \rho_i]^T,$$

$$\mathbf{b}_{l_i} \triangleq -[T_{l_{i1}}, \ldots,\ T_{l_{iM}},\ \mathbf{0}_{1\times 4},\ 1]^T,$$

$$\mathbf{b}_{c_i} \triangleq [(T_{ac_{i1}} + T_{c_{i1}}), \ldots,\ (T_{ac_{iM}} + T_{c_{iM}}), 0, 1, 0, 1, -1]^T,$$

$$\mathbf{b}_{u_i} \triangleq [\mathbf{D}_{\text{in}}(i1), \ldots,\ \mathbf{D}_{\text{in}}(iM),\ \mathbf{0}_{1\times 5}]^T,$$

$$\mathbf{b}_{d_i} \triangleq [\mathbf{D}_{\text{out}}(i1), \ldots,\ \mathbf{D}_{\text{out}}(iM),\ \mathbf{0}_{1\times 5}]^T,$$

$$\mathbf{b}_{U_i} \triangleq [\mathbf{0}_{1\times M},\ 1,\ \mathbf{0}_{1\times 4}]^T,$$

$$\mathbf{b}_{D_i} \triangleq [\mathbf{0}_{1\times M+2},\ 1,\ \mathbf{0}_{1\times 2}]^T.$$

Note that all constraints in the optimization problems (9) and (10) have one-to-one correspondence. By further defining $\mathbf{z}_i \triangleq [\mathbf{w}_i^T, 1]^T$ and dropping the constant term $\sum_{i=1}^N \sum_{j=1}^M E_{l_{ij}}$ from the objective function in (10), we can homogenize the optimization problem (10) to the following problem

$$\begin{aligned} \min_{\{\mathbf{z}_i\}} \quad & \sum_{i=1}^N \mathbf{z}_i^T \mathbf{G}_i \mathbf{z}_i \qquad (11) \\ \text{s.t.} \quad & \mathbf{z}_i^T \mathbf{G}_{l_i} \mathbf{z}_i \leq -\sum_{j=1}^M T_{l_{ij}}, \forall i, \\ & \mathbf{z}_i^T \mathbf{G}_{r_i} \mathbf{z}_i \leq 0, \forall i, \quad r = c, u, d, \\ & \sum_{i=1}^N \mathbf{z}_i^T \mathbf{G}_{U_i} \mathbf{z}_i \leq C_{\text{UL}}, \ \sum_{i=1}^N \mathbf{z}_i^T \mathbf{G}_{D_i} \mathbf{z}_i \leq C_{\text{DL}}, \\ & \mathbf{z}_i^T \mathbf{G}_{I_j} \mathbf{z}_i = 0, \forall i, j, \\ & \mathbf{z}_i \geq \mathbf{0}, \forall i, \end{aligned}$$

where

$$\mathbf{G}_i \triangleq \begin{bmatrix} \mathbf{0}_{(M+5)\times(M+5)} & \frac{1}{2}\mathbf{b}_i \\ \frac{1}{2}\mathbf{b}_i^T & 0 \end{bmatrix}, \forall i,$$

$$\mathbf{G}_{l_i} \triangleq \begin{bmatrix} \mathbf{0}_{(M+5)\times(M+5)} & \frac{1}{2}\mathbf{b}_{l_i} \\ \frac{1}{2}\mathbf{b}_{l_i}^T & 0 \end{bmatrix}, \forall i,$$

$$\mathbf{G}_{s_i} \triangleq \begin{bmatrix} \mathbf{A}_{s_i} & \frac{1}{2}\mathbf{b}_{s_i} \\ \frac{1}{2}\mathbf{b}_{s_i}^T & 0 \end{bmatrix}, \forall i, \quad s = u, d,$$

$$\mathbf{G}_{S_i} \triangleq \begin{bmatrix} \mathbf{0}_{(M+5)\times(M+5)} & \frac{1}{2}\mathbf{b}_{S_i} \\ \frac{1}{2}\mathbf{b}_{S_i}^T & 0 \end{bmatrix}, \forall i, \quad S = U, D,$$

$$\mathbf{G}_{I_j} \triangleq \begin{bmatrix} \text{diag}(\mathbf{e}_j) & -\frac{1}{2}\mathbf{e}_j \\ -\frac{1}{2}\mathbf{e}_j^T & 0 \end{bmatrix}, \forall j.$$

The optimization problem (11) is a non-convex separable QCQP problem [14], which is still NP-hard in general. To find an approximate solution, we apply the separable SDR approach [15], where we relax the problem into a separable semidefinite programming (SDP) problem. Specifically, define $\mathbf{Z}_i \triangleq \mathbf{z}_i \mathbf{z}_i^T$. By dropping the rank constraint rank$(\mathbf{Z}_i) = 1$, we have the following separable SDP problem:

$$\begin{aligned} \min_{\{\mathbf{Z}_i\}} \quad & \sum_{i=1}^N \text{Tr}(\mathbf{G}_i \mathbf{Z}_i) \qquad (12) \\ \text{s.t.} \quad & \text{Tr}(\mathbf{G}_{l_i} \mathbf{Z}_i) \leq -\sum_{j=1}^M T_{l_{ij}}, \forall i, \\ & \text{Tr}(\mathbf{G}_{r_i} \mathbf{Z}_i) \leq 0, \forall i, \quad r = c, u, d, \\ & \sum_{i=1}^N \text{Tr}(\mathbf{G}_{U_i} \mathbf{Z}_i) \leq C_{\text{UL}}, \\ & \sum_{i=1}^N \text{Tr}(\mathbf{G}_{D_i} \mathbf{Z}_i) \leq C_{\text{DL}}, \\ & \text{Tr}(\mathbf{G}_{I_j} \mathbf{Z}_i) = 0, \forall i, j, \\ & \mathbf{Z}_i(M+6, M+6) = 1, \forall i, \quad \mathbf{Z}_i \succeq \mathbf{0}, \forall i. \end{aligned}$$

The optimal solution $\{\mathbf{Z}_i^*\}$ to the above SDP problem can be obtained efficiently in polynomial time using standard SDP software, such as SeDuMi [17].

Once $\{\mathbf{Z}_i^*\}$ is obtained, we need to recover a rank-1 solution from $\{\mathbf{Z}_i^*\}$ for the original problem (1). In the following, we propose an algorithm to obtain the binary offloading decisions $\{x_{ij}\}$ and the corresponding optimal communication resource allocation $\{\mathbf{r}_i\}$.

C. Binary Offloading Decisions and Resource Allocation

Define $\mathbf{x} \triangleq [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$, where $\mathbf{x}_i \triangleq [x_{i1}, \ldots, x_{iM}]$, $\forall i$. Note that, first, only the upper-left $M \times M$ sub-matrix of $\mathbf{Z}_i^*$, denote by $\mathbf{Z}'^*_i$, $\forall i$, is needed to recover the solution $\mathbf{x}$; second, each diagonal entry in each $\mathbf{Z}'^*_i$ is always between 0 and 1. That is, define $\mathbf{p}_i = [p_{i1}, \ldots, p_{iM}] \triangleq$ diag$(\mathbf{Z}'^*_i)$. We have $p_{ij} \in [0, 1]$, $\forall i, j$. We recover the feasible decisions $\mathbf{x}_i^o$ using $\mathbf{p}_i$, where $x_{ij}^o = \text{round}(p_{ij})$ is the rounding result, and obtain the overall offloading decision as $\mathbf{x}^o = [\mathbf{x}_1^o, \ldots, \mathbf{x}_N^o]^T$.

Once the offloading decision $\mathbf{x}^o$ is obtained, the optimization problem (1) reduces to the optimization of communication

Algorithm 1 MUMTO Algorithm

```
Obtain optimal solution $\mathbf{Z}_i^*$'s of the separable SDP problem (12). Extract the upper-left $M \times M$ sub-matrices $\mathbf{Z'}_i^*$'s from $\mathbf{Z}_i^*$'s.
Record the values of diagonal terms in $\mathbf{Z'}_i^*$ by $\mathbf{p}_i = [p_{i1}, \ldots, p_{iM}]$.
$x_{ij}^o = \text{round}(p_{ij}), \forall i, j$.
Set $\mathbf{x}^o = [\mathbf{x}_1^o, \ldots, \mathbf{x}_N^o]^T$, where $\mathbf{x}_i^o = [x_{i1}^o, \ldots, x_{iM}^o]$.
Solve the resource allocation problem (13) based on $\mathbf{x}^o$;
Compare $\mathbf{x}^o$ with the solutions from local processing only and cloud processing only. Set the best one among them as the solution $\mathbf{x}^s$.
Output: the proposed offloading solution $\mathbf{x}^s$ and the corresponding optimal resource allocation.
```

resource allocation $\{\mathbf{r}_i\}$, which is given by

$$\min_{\{\mathbf{r}_i\}} \quad \left(\mathbf{E} + \sum_{i=1}^{N} \rho_i \max\{T_{L_i}, T_{C_i}^U\}\right) \qquad (13)$$
$$\text{s.t.} \quad (2) - (4)$$

where $\mathbf{E} \triangleq \sum_{i=1}^{N}\sum_{j=1}^{M}(E_{l_{ij}}(1-x_{ij})+E_{C_{ij}}x_{ij})$ is a constant value once $\{x_{ij}\}$ are given. This resource allocation problem (13) is convex, which can be solved optimally using standard convex optimization solvers.

Note that to obtain the best offloading decision, in practice, we should compare $\mathbf{x}^o$ with local processing only and cloud processing only decisions, and select the one resulting in the minimum total system cost objective of (1) as the final offloading decision $\mathbf{x}^s$.

We name the above the multi-user multi-task offloading (MUMTO) algorithm and summarize it in Algorithm 1. Notice that the SDP problem (12) can be solved within precision ϵ by the interior point method in $O(\sqrt{MN}\log(1/\epsilon))$ iterations, where the amount of work per iteration is $O(M^6N^4)$ [18], while there are 2^{MN} choices in exhaustive search to find the optimal offloading decision. In addition, once the offloading decision is made, we may schedule the multiple tasks to be offloaded in any arbitrary order. The resultant T_{C_i} will be less than $T_{C_i}^U$. To measure the effectiveness of this solution, in the following, we introduce a lower bound of the optimal solution to the original problem (1).

D. Lower Bound on the Optimal Solution

Previously, the cost function in our original optimization problem (1) considers the worst case transmission plus processing delay (6) for all users, and we know the real cost based on MUMTO will be lower. However, we are still interested in the performance of MUMTO compared with the optimal solution. To find a lower bound of the optimal solution, we introduce a new optimization problem in which $T_{C_i}^L$ is used instead as

$$\min_{\{x_{ij}\},\{\mathbf{r}_i\}} \quad \sum_{i=1}^{N}\Bigg[\sum_{j=1}^{M}(E_{l_{ij}}(1-x_{ij})+E_{C_{ij}}x_{ij})$$
$$+ \rho_i \max\{T_{L_i}, T_{u_i}, T_{d_i}, T_{uac_i}, T_{dac_i}, T_{C_i'}\}\Bigg] \qquad (14)$$
$$\text{s.t.} \quad (2) - (5).$$

Notice that under the same offloading decisions and communication resource allocation, this new objective function will always give us a lower cost than the real cost.

Since the above optimization problem (14) is still noconvex, we formulate a separable SDR problem similar to (12), whose details are omitted due to page limitation. We note that the optimal objective of this SDR problem is smaller than the optimal objective of (14). Hence, it can serve as a lower bound of the minimum total system cost defined by the original optimization problem (1).

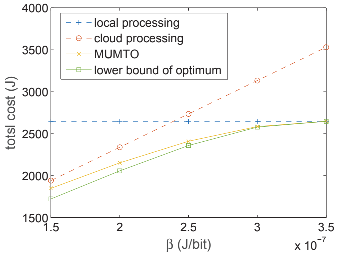

Fig. 2. The total system cost versus β (J/bit).

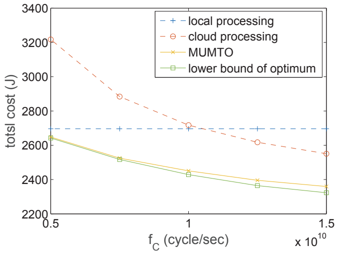

Fig. 3. The total system cost versus cloud CPU rate f_C (cycles/s).

IV. Performance Evaluation

We evaluate the performance of MUMTO using computer simulation in Matlab under different parameter settings. The following default parameter values are used unless specified otherwise later. We adopt the mobile device characteristics from [19], which is based on Nokia N900, and set the number of mobile users as $N = 5$. Each user has $M = 4$ independent tasks. From Tables 1 and 3 in [19], the mobile device is assumed to have CPU rate 500×10^6 cycles/s and processing energy consumption $\frac{1}{730\times10^6}$ J/cycle. The local computation time 4.75×10^{-7} s/bit and local processing energy consumption 3.25×10^{-7} J/bit are calculated when the x264 CBR encode application (1900 cycles/byte) is considered as App(ij) in our simulations. The input and output data sizes of each task are assumed to be uniformly distributed from 10 to 30MB and from 1 to 3MB, respectively.

In addition, both uplink and downlink transmission capacities are 150 Mbps (e.g., IEEE 802.11n) between the mobile users and the AP, and the transmission and receiving energy consumptions of the mobile user are both 1.42×10^{-7} J/bit as indicated in Table 2 in [19]. The CPU rate of each server assigned to each user at the remote cloud is 10×10^9 cycle/s. When tasks are offloaded to the cloud, the transmission rate R_{ac} is 15 Mpbs. The system utility cost $C_{c_i} = D_{\text{in}}(i) + \alpha_1/f_C + \alpha_2/C_{\text{UL}} + \alpha_3/C_{\text{DL}}$, where $\alpha_1 = 10^{18}$ bit×cycle/s and $\alpha_2 = \alpha_3 = 10^{16}$ bit×Mbps, will be a function of the input data size, cloud CPU rate, and uplink and downlink capacities. Also, $\beta = 2.5 \times 10^{-7}$ J/bit. We further set $\rho_i = 1$ J/s as the weight of the delay to process each user's task.

For comparison, we also consider the following methods: 1) the *local processing only* method where all tasks are processed by mobile users, 2) the *cloud processing only* method where all tasks are offloaded to the cloud, 3) the *lower bound of optimum*, which is obtained from the optimal objective value of the SDR of problem (14). Notice that in all figures the real cost under the same offloading decision and resource allocation will always between the costs of the proposed MUMTO and

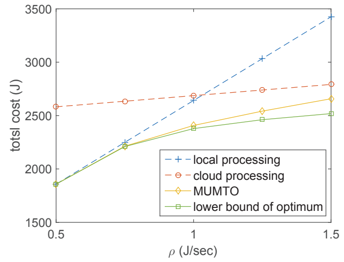

Fig. 4. The total cost under different policies versus ρ (J/s).

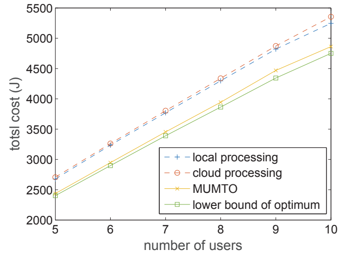

Fig. 5. The total cost under different policies versus number of users.

the lower bound of optimum. Finally, all simulation results are obtained by averaging over 100 realizations of the input and output data sizes of each task.

In Figs. 2, we show the system cost vs. the weight β on the system utility cost. When β becomes large, all tasks are more likely to be processed by mobile users themselves. Both MUMTO and the lower bound of optimum in this case converge to the local processing only method. Though the existence of the cloud can provide additional computation capacity, the processing time at the cloud depends on the cloud CPU rate f_C assigned to each user. In Fig. 3, we plot the total system cost vs. f_C. As expected, a more powerful per-user cloud CPU can dramatically increase system performance, and MUMTO coverages to the local processing only method when the per-user cloud CPU rate is too slow to help.

In Fig. 4, we study the system cost under various values of weight ρ on the delays. We observe that MUMTO substantially outperforms all other methods and is nearly optimal. Finally, we examine the scalability of MUMTO. Fig. 5 and Fig. 6 plot the total system cost vs. the number of users N and the number of tasks M per user, respectively. We see that MUMTO in both figures is close to the lower bound of optimum, indicating that it is nearly optimal for all N and M values.

V. Conclusion

A general mobile cloud computing system consisting of multiple users and one remote cloud server has been considered, in which each user has multiple independent tasks. To minimize a weighted total cost of energy, computation, and the delay of all users, we aim to find the overall optimal tasks offloading decisions and communication resource allocation. We show that the resultant optimization problem is a non-convex separable QCQP. The proposed MUMTO algorithm uses SDR and binary recovery to jointly compute the offloading decision and communication resource allocation. By comparison with a lower bound of the minimum cost, we show that MUMTO gives nearly optimal performance.

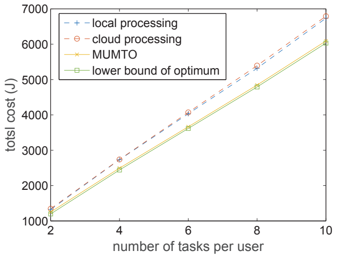

Fig. 6. The total cost under different policies versus number of tasks per user.

References

[1] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mob. Netw. Appl.*, vol. 18, pp. 129–140, 2013.

[2] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, pp. 84 – 106, 2013.

[3] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, pp. 51–56, 2010.

[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, pp. 14–23, 2009.

[5] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, pp. 4569–4581, 2013.

[6] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile cloud computing," in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2014, pp. 354–358.

[7] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, pp. 974–983, 2015.

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

[9] 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 International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2010, pp. 49–62.

[10] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud," in *Proc. ACM Conference on Computer Systems (EuroSys)*, 2011, pp. 301–314.

[11] 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 International Conference on Computer Communications (INFOCOM)*, 2012, pp. 945–953.

[12] S. Mahmoodi, K. Subbalakshmi, and V. Sagar, "Cloud offloading for multi-radio enabled mobile devices," in *Proc. IEEE International Conference on Communications (ICC)*, 2015, pp. 5473–5478.

[13] M.-H. Chen, B. Liang, and M. Dong, "A semidefinite relaxation approach to mobile cloud offloading with computing access point," in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2015, pp. 186–190.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[15] Z.-Q. Luo, W.-K. Ma, A.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, pp. 20–34, 2010.

[16] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Mathematics of Operations Research*, vol. 1, pp. 117–129, 1976.

[17] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2009. [Online]. Available: http://cvxr.com/cvx/

[18] Y. Nesterov, A. Nemirovskii, and Y. Ye, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[19] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. USENIX Conference on Hot Topics in Cloud Computing (HotCloud)*, 2010.