IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

SPECIAL SECTION ON SMART CACHING, COMMUNICATIONS, COMPUTING AND CYBERSECURITY FOR INFORMATION-CENTRIC INTERNET OF THINGS

Received May 31, 2019, accepted June 27, 2019, date of publication July 15, 2019, date of current version August 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2928891

D2D Computation Offloading Optimization for Precedence-Constrained Tasks in Information-Centric IoT

JINDOU XIE, YUNJIAN JIA, ZHENGCHUAN CHEN, ZHAOJUN NAN, AND LIANG LIANG
School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China

Corresponding author: Yunjian Jia (yunjian@cqu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFE0121100, in part by the Program for Innovation Team Building at Colleges and Universities in Chongqing, China, under Grant CXTDX201601006, and in part by the National Natural Science Foundation of China under Grant 61601067.

ABSTRACT This paper proposes a computation offloading scheme for precedence-constrained tasks in a base station-assisted device-to-device (D2D) scenario for the information-centric Internet of Things (IC-IoT). When specified precedence among subtasks cannot be described as simple sequential or parallel relations in a task, the selection of task execution helper for subtasks offloading becomes complex due to the constraints of latency and resources. We define this type of precedence and aim to minimize the time and financial cost of computation task offloading for the user by optimizing subtask-helper pairs. This problem is modeled as a dynamic generalized multi-resource-constrained assignment problem. The optimal offloading policy is offered by searching minimum weight matchings in a bipartite graph. Computer simulations indicate the effectiveness of the proposed approach compared with the random helper selection and priority-based offloading scheme.

INDEX TERMS Computation offloading, precedence-constrained task, D2D, information-centric IoT, weighted bipartite graph.

I. INTRODUCTION

Ubiquitous connections and heterogeneous devices show new requirements for massive wireless access and complex mobility support in various Internet of things (IoT) scenarios [1] e.g. smart home, intelligent transport system, and smart healthcare. The traditional IP-centric IoT architecture now faces limits of extensible capacity and frequent updates of route table. A new network paradigm, information centric networking (ICN) is proposed to address these challenges [2]. ICN decouples in terms of the content and location by content-based naming and name-based content discovery and delivery. Such feature makes ICN has potential to support various IoT applications that involve different perceptions and automations. Therefore, the integration of information-centric networking and Internet of things forming new paradigm information-centric Internet of things (IC-IoT) has been discussed. It can directly locate heterogeneous IoT services by the specified content, which enables in-network caching to reduce the network cost on the duplicate content transmission, and support a highly efficient and scalable content retrieval [3].

To meet emerging demand of intense computing capacity, the cloud function sinks to the edge of networks to form mobile edge computing (MEC) [4], where both edge nodes and cloud can help user to execute tasks. As the task execution is closer to user, edge nodes can provide low-latency and flexible computing augmentation services (e.g. large-scale sensing tasks [5]) for users. On the other hand, device-to-device (D2D) communication is recognized as a promising solution to reduce the heavy load of cellular network [6]. D2D allows mobile devices to communicate directly to achieve dependable content distribution [7] and information dissemination [8] without cellular relay. Mature process technology brings mobile devices much more powerful computing capabilities for data analysis, which indicates that smart mobile devices constitute to intelligent groups. In order to achieve the multiplexing gain of available resources on these intelligent devices, the idea of combination of D2D and MEC come out. A novel D2D-MEC technique integrates D2D communications and MEC to enhance the computation capacity of system [9] and support task collaborative execution [10] with assistance of base station (BS).

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenyu Zhou.

Applying D2D-MEC into IC-IoT, mobile devices can share not only the communication and computing resources among each other but also the popular task contents. In the integrated structure, mobile devices share resources and contents successfully relying on interest packet and data packet transmission. User device floods interest packet to request execution help and send data packet to provide task content for helper devices. In the other hand, helpers send back task execution results in a data packet to user. The cooperation of user and helpers couples high-speed task execution and short-range data transmission to cut task service time. Besides, information-centric task addressing could reduce the occupancy of resources for repeated content transmission and duplicate task execution to improve the resources utility efficiency. However, appropriate helpers selection in dynamic networks is a big challenge. Because of time-varying communication resource volume and computing capability offered by helpers, the transmission and task execution cost is sophisticated and unpredictable. Thus, the framework requires to update cost of possible task-helper pairs according to the live system information.

Generally, a task is composed of several divisible and logical dependent subtasks. To meet user specified needs, there are certain precedence relations among these subtasks, such as sequential precedence, parallel precedence and general precedence [11]. The precedence determines the execution order and processing time of subtasks. In practice, the precedence of a large delay-constrained task cannot be simply described as sequential or parallel precedence, such as mobile games. Considering data transfer delay and game rules, there are some hard precedence constraints on execution steps, e.g. the second step has not to start until a specified interval later since the first step finished or a specified interval later since the first step started. To the knowledge of our best, few works concentrate on computation task offloading problem with this type of precedence relations.

Combing these aspects, the strategy of task offloading is supposed to consider system dynamics and specified precedence relations among subtasks. In summary, the main contributions of this paper include:

- A new task graph model represents an application with given precedence relations. For the precedence-constrained task, the problem formulation aims to minimize the cost of task offloading jointly considering subtasks delay constraints, association states between user and helpers and available resources constraints.
- We propose an efficient task offloading scheme based on weighted bipartite graph matching to pair subtasks and helpers. By constructing appropriate weight of bipartite graph according to the time-varying system information, we search minimum cost (weight) subtask-helper pairs as the optimal offloading policy.
- Extensive simulations are conducted to compare the performance of proposal with random offloading and priory-based offloading. Finally, we investigate the effect of system parameters on performance.

The rest of the paper is organized as follows. Related works about task offloading are reviewed in Section II. Section III introduces the system model; Section IV formulates the problem for precedence-constrained task offloading via D2D in IC-IoT. A bipartite graph matching-based task offloading algorithm is proposed, and its computing complexity is analyzed in Section V. Then, Section VI evaluates the cost performance of the proposed offloading algorithms, and discusses the effect of model parameters. Finally, conclusions are provided in Section VII.

II. RELATED WORKS

In this section, a brief overview about some related works in regards to task offloading in edge networks is given. For user perspective, the proper use case of task offloading can cut energy cost and speed up the process of computing.

A crucial part regarding task offloading is weather to offload or not, which is called binary offloading [11]. It mainly focuses on the whole task scheduling. In order to minimize execution delay, offloading policy decides local computing or task offloading according to the length of application buffer queues, available processing powers, and characteristic of the channel between local device and edge server [12]–[14]. Another offloading strategy in binary offloading is minimizing the energy consumption satisfying the execution delay of the application. This optimization problem is formulated as a constrained Markov decision process in [15], which is solved by an online learning and a pre-calculated offline approach. Labidi *et al.* [16] optimize scheduling and computing offloading initiatives for each user to guarantee QoE, with low energy consumption and average queuing constraints. Moreover, some works discuss the task offloading decision aiming to minimize the weighted sum of energy consumption and execution delay. Chen *et al.* [17] consider a policy for multi-user multi-channel environment to tradeoff the energy consumption of users and the task service delay. Zhang *et al.* [18] propose an online dynamic tasks assignment to investigate the tradeoff between energy consumption and execution delay for an MEC system with energy harvest function.

The drawback of above-mentioned works is that the offloading decision focus on an integrated task schedule and an optimal edge node selection. Obviously, it is not acceptable to the case when a task is relatively large and has to be processed by multiple devices in the mobile edge. Therefore, an application is considered to be divided into several smaller subtasks, and some of them can be offloaded while others only accept local computing. Partial offloading is carried out to discuss how much and what should be offloaded [11]. The main objective of [19] is to decide which parts should be offloaded to the edge nodes to minimize energy consumption with service delay constraints. Zhao *et al.* [20] address the partial offloading problem for multiple user scenario. With wider use of multi-antennas, the divided task parts can be transmitted to different edge nodes simultaneously and be executed in parallel. Authors in [21] propose to offload a

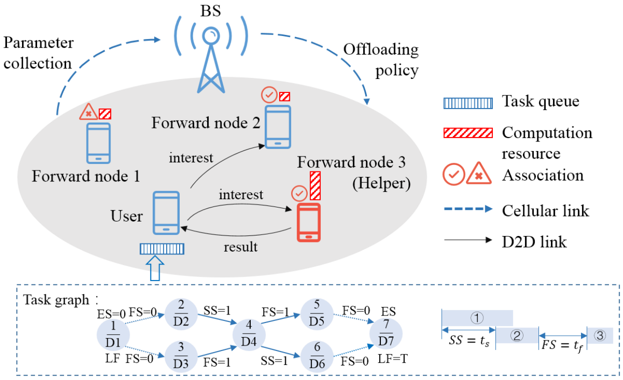


FIGURE 1. **Illustration of precedence-constrained task offloading via D2D in IC-IoT with one user and several helper candidates in the same base station coverage.**

task from a user to multiple edge nodes through single and multiple hops, to minimize the overall task response time.

To ensure the task can be finished on schedule, some works take subtasks precedence into account for task offloading policy design. Zhang *et al.* [22] jointly consider resources schedule and precedence among subtasks in task offloading policy that supports hybrid strategies of sequential and parallel task execution. The task with a sequential subtask arrangement is considered in [23] that jointly optimizes user-program partitioning and server-computation scheduling to minimize the average service time. For component dependency graph structures, authors in [24] suggest to process appropriate components in parallel in the mobile and cloud to shorten execution times. Furthermore, Kim *et al.* [25] take the cost of mobile user into consideration by constructing the objective function that includes financial cost (such as network price and computing price) and energy consumption.

However, these works have considered pure relations (i.e. sequential or parallel) among subtasks to construct simple task graph. In this paper, we focus on offloading scheme for a new task model with specified precedence relations aiming to minimize the cost of task scheduling (i.e. weighted sum of service time and financial cost).

III. SYSTEM MODEL

As illustrated in Fig. 1, task can be offloaded via D2D from user to helpers in an information-centric IoT network. Mobile user with precedence-constrained task tends to ask computing help from neighboring mobile devices with ample computing resources. These mobile devices are called forward nodes in this work (e.g. forward nodes 1-3 in Fig.1). Motivated by the fact that the BS generally has global network information and high computation power, a BS-assisted D2D structure is employed. The BS collects relative parameters, makes the task offloading policy and manage both cellular and D2D connections of mobile devices. The cellular links carry control information (e.g. task parameters, offloading policy). In the other hand, task contents and execution results are transmitted in data packet between mobile devices via D2D. So far, it is possible but expensive and inflexible to support D2D by modifying many entities and protocols [6]. Fortunately, the combination of D2D and novel network technologies, such as software-defined networking (SDN) and network function virtualization (NFV), facilitates the coexistence of D2D and cellular communication [26]. Moreover, we consider a slotted structure and each time interval has the equal length.

A. PRECEDENCE-CONSTRAINED TASK GRAPH

Task decomposition methods are based on divided-and-conquer technique, such as decomposition based on deep priori knowledge, and processing data relations [27]. In general, the subtask precedence management can be carried out by a task queue controller in the upper layer of user (e.g., application layer) according to specified need. This work concentrates on a precedence-given task offloading problem. Especially, unlike sequential, parallel and general precedence, subtasks with specified precedence relations construct a precedence-constrained task graph.

We mainly consider two specified precedence relations, as depicted in Fig.1. 1) Finish-to-Start(FS): current subtask i cannot start until its predecessor pre_i has been finished with an additional given interval denoted as $FS_{pre_i,i}$. As shown in Fig.1, subtask ③ cannot start until t_f later after completion of subtask ②, recorded as $F_{23} = t_f$. When $t_f = 0$, the precedence relation between ② and ③ becomes conventional sequential. 2) Start-to-Start(SS): the current subtask i is able to begin after the predecessor started for a fixed interval expressed $SS_{pre_i,i}$. In Fig.1, subtask ② can start till t_s later after beginning of ①, i.e. $SS_{12} = t_s$. $t_s = 0$ presents the traditional parallel precedence relation.

After decomposition, task graph $G(V, Z, B, E)$ is given, in which V is the set of subtasks, Z expresses the set of corresponding computing resources demands (e.g. the number of CPU cycles), B contains the amount of transmitted subtasks, and E is the specified precedence relations among subtasks.

B. TASK OFFLOADING PROCEDURE IN IC-IoT

In view of characteristic of IC-IoT, mobile devices can play three roles in the procedure of task offloading as depicted in Fig.1: user, forward node, and helper. The BS is not only a controller but also a powerful alternative forward node with high resource rental cost. The network allows one-hop D2D communication in interest or data packet.

When user suffers from computing resources limit and cannot execute the remaining task, it broadcasts interest packet to forward nodes (including BS) for task execution help request. Interest packet includes parameters of task and identity of user. The user may receive some responses of the interest from forward nodes. BS selects suitable forward nodes from them to offload subtasks.

We call the set of helper candidates as forward nodes $\mathcal{J} = \{1, 2, \cdots j\}$ that contains BS specially. The components of $\mathcal{J}$ are time-varying due to devices mobility and dynamic private task generation. As forward nodes receive interest packet, they would determine if computing resource they carries matches this interest (e.g. processor version, application compatibility). If they are compatible, and candidates are willing to share computing resources, they will answer the interest packet to user and controller BS. When the controller BS receives answers, it analyzes mobility information of forward nodes to get the association states between candidates and user. Further, BS also examines the computing capabilities of these nodes who reply to the interest. The results of analysis are used to select appropriate forward nodes as helper(s) (e.g. forward node 3 in Fig.1) to reduce the cost of trial error for user. According to the matching between selected forward nodes and subtasks, BS will help user establish D2D links with helpers for subtasks and results transmission.

$\mathcal{H} = \{1, 2, \cdots h\}$ is the set of helpers. Helpers receive different subtasks and try their best to help task execution. Each helper executes subtasks in a best-effort and first-come-first-server manner. In other words, helpers execute subtasks with all current available resources and the execution of a given subtask is assumed to be non-preemptive. After helpers finishing subtask, the results are directly sent back via D2D and the occupied resources will be released.

C. TASK PROCESSING

Task processing including two main parts, subtask content transmission and subtask execution.

- **Subtask content transmission**
  In the BS-assisted D2D system, BS can manage the peer discovery, mode selection, power control and link quality feedback to coordinate the coexistence of D2D and cellular links. At the beginning of data transmission, BS will assign a D2D transmission power $P_j(t)$ for helper $j \in \mathcal{N}$. Thus the interference in the coverage between cellular and D2D will be controlled by BS [28]. We can get the D2D transmission rate r_j from user to helper j as $r_j(t) = W_j(t)\log_2\left(1 + \frac{H_j(t)P_j(t)}{\sigma^2+I_j}\right)$ based on Shannon theory, where $W_j(t)$ is the bandwidth allocated for D2D link between user and helper j, σ^2 is the noise power (e.g. background noise), and intra-interference is denoted as I_j. In time slotted structure, considering devices slow movement, limited sizes of subtasks, and heavy sensing overhead, D2D link transmission rate is assumed to update periodically and remain steady during subtask transmission period. Besides, the transmission cost of result and control information are negligible due to the small data size, so the backward link and cellular link quality are out of consideration. In addition, a user can remain several D2D links for data transmission at the same time with novel antenna technology (e.g. D2D MIMO [29]).
- **Subtask execution**
  Modern mobile device processors have many scheduling schemes to control their processing capacity such as Performance governor (i.e., locking the device CPU at maximum frequency) [10]. $l_j(t)$ denotes as the current load (i.e., proportion of occupied processing capacity because the helper j may run personal tasks locally). Then, the available processing capacity in a time interval t for user is denoted by $f_j(t) = \left(1 - l_j(t)\right) F_j$, where $l_j(t)$ is uniform distributed $l_i \in [0, L_i]$, and F_j is the maximum computing capability of helper j.
  With assistance of BS, user can offload subtask i to a nearby mobile helper j via D2D link. In this case, the time and resource rental cost for subtask i transmitting from user to helper j through D2D are given by $T_{ij}^t = b_i(t)/r_j(t), b_i \in B$, and $E_{ij}^t = p_i^b(t)W_j(t)$, respectively. $p_i^b(t)$ is the price of per bandwidth for transmission. In addition, the time and resources rental expenses for executing the offloaded subtask i in helper j are given by $T_{ij}^e = z_i/f_j(t), z_i \in Z$, and $E_{ij}^e = p_j^e(t)f_j(t)$. The $p_j^e(t)$ is the fees for unit computing capabilities from helper j. Therefore, the total service time for subtask i offloading from user to helper j is $T_{ij} = T_{ij}^t + T_{ij}^e$, and the corresponding total financial cost is $E_{ij} = E_{ij}^t + E_{ij}^e$.

IV. PROBLEM FORMULATION

Based on system model, this section describes the problem formulation for mobile task offloading in IC-IoT, considering necessary constraints.

A. DELAY CONSTRAINT

T_0 is defined as the user sojourn time within the coverage. The average velocity of user is v_0 that has inverse relations with T_0 due to the limited coverage area.

T presents the time constraint of a service. According to a given task graph $G = (V, Z, B, E)$, the time constraint of each subtask is estimated by computing the earliest start time and latest finish time recursively, denoted as ES_i and LF_i respectively. The estimated service time of each subtask D_i is computed, assuming that the processors run at the average frequency of the first time interval, data is transmitted with average rate and all subtasks run at their worst cases (e.g. maximal size). For each subtask, there are $EF_i = ES_i + D_i$, and $LF_i = LS_i + D_i$ where the EF_i, and LS_i are the earliest finish time and latest start time respectively. In the graph, a subtask without any predecessor is called an entry subtask (e.g. node 1 in Fig.1) and a subtask without any successor is called an exit subtask (e.g. node 7 in Fig.1). For the entry subtask $ES_{entry} = 0$, and the exit subtask $LF_{exit} = min(T, T_0)$.

By forward tracing along the graph, the earliest start time of subtask i can be calculated by

$$ES_i = max(EF_{pre_i} + FS_{pre_i,i}, ES_{pre_i} + SS_{pre_i,i}). \tag{1}$$

Through back stepping, the latest finish time can be computed starting from LF_{exit} by

$$LF_i = min(LS_{suc_i} - FS_{i,suc_i}, LS_i - SS_{i,suc_i} + D_i). \tag{2}$$

Actual service time of subtask i is $D_i^a(j) = T_{ij}$. Actual start time AS_i and the actual finish time AF_i of subtask i follow the rule $AF_i = AE_i + D_i^a$. Then, the time of each subtask should satisfy:

$$0 \le ES_i \le AS_i \le AF_i \le LF_i, (i \in V). \tag{3}$$

This constraint ensures that each subtask could be finished before the specified time constraint. After stopping the task scheduling, the schedule length will be the actual finish time of the exit task.

B. ASSOCIATION CONSTRAINT

One of the challenges of task offloading is the dynamic topology due to devices mobility, which influences the components of the forward node set $\mathcal{J}$.

There are some human mobility models being used for wireless networks, such as random walk model (RWM) and fluid flow model [30]. Because task computing interest is related to the usersąŕ activity tendency and habits tightly, we apply individual mobility model (IMM) [31] to evaluate the user mobility performance. Devices average velocity and the area of the community influence the probabilities of devices arrival, pause, and departure. The meeting periods of the meeting between helper candidates and user can be calculated with IMM. The period of forward node j is denoted as $< s_j, e_j >$, where s_j is the time of j arriving in the range of user's D2D link. e_j is the moment of node j stepping out the D2D range, which is not larger than T_0. For time interval t, the met forward nodes written as $\mathcal{M}(t) = \{m_1^t, m_2^t, \cdots, m_M^t\}$, where $M(t)$ is the number of components in $\mathcal{M}(t)$.

Then, we use a binary variable a_j to indicate whether a forward node j could link to user with D2D based on the meeting time. Then, we have the following association constraint:

$$\sum_{j \in \mathcal{J}} a_j(t) = M(t), \tag{4}$$

$$a_j \in \{0, 1\}. \tag{5}$$

The constraint of (4) represents that the number of forward nodes who are able to link to user with D2D is equal to the number of met forward nodes. $a_j = 1$ indicates forward node $j \in \mathcal{J}$ can communicate with user via D2D in time interval t.

C. RESOURCES CONSTRAINT

The BS can maintain multiple D2D links for user and each subtask can be assigned to a helper. In this sense, in one time interval, multiple subtasks can be offloaded to several helpers coupling in pairs. If some helpers have additional spare resources after they received a subtask, they have chance to support more subtasks in following time interval. Thus, the subtask-helper matching can be only considered in current time interval.

In the procedure of task scheduling, subtasks can be classified into three types based on the state of service, *Can*, *Doing*, *Done*. When the time t goes to ES_i, subtask i is ready for transmission according to the precedence constraints and put into $Can(t)$ set who contains $O(t)$ elements. While there is an appropriate helper j for subtask i, subtask will be included by $Doing_j(t)$ and deleted by $Can(t)$. That means subtask i is offloaded to helper j, denoted as $u_{i,j}(t) = 1$, or there is $u_{i,j}(t) = 0$. Then, for subtask i, the maximal remaining time for task computing is $\tau_f = (e_j - t - d_t^m)$, where $d_t^m = \frac{b_i}{\max\limits_j r_j}$ is the minimal data transmission delay for subtask i in time interval t. Similarly, the maximal remaining time for task transmission is $\tau_b = (e_j - t - d_f^m)$, and $d_f^m = \frac{z_i}{\max\limits_j f_j}$. As the subtask i is completed, it will be removed from $Doing_j(t)$ to *Done*. Thus, the sum of sharing resources satisfies the constraint:

$$\sum_i u_{i,j}(t) = \sum_j u_{i,j}(t) \le min(O(t), M(t)). \tag{6}$$

$$u_{i,j}(t) z_i \le f_j(t) \tau_f, \tag{7}$$

$$u_{i,j}(t) b_i \le r_j(t) \tau_b, \tag{8}$$

$$u_{i,j} \in \{0, 1\}, \quad i \in V, j \in \mathcal{J} \tag{9}$$

The constraint of (6) represents that the assignment is pairing subtasks and helpers, and the numbers of pairs cannot exceed the minimum value of subtasks and helpers. The constraint (7) means the sum of computing demand of subtask offloaded to helper j in time interval t is no more than the computing capabilities provided by helper j. The constraint (8) clarifies the communication resources limits. It ensures that in time interval t, the transmission amount of subtask i cannot exceed the traffic bound between user and helper j. $u_{ij} \in \boldsymbol{u}$ denotes the offloading decision. The constraint (9) illustrates the control variable (offloading decision) is a binary variable.

D. PROBLEM OBJECTIVE AND FORMULATION

In this paper, in terms of the given task delay constraint, association constraint and resources constraint, we focus on an efficient task offloading decision to minimize the cost of task process.

Specifically, the integrated objective variable the cost for the offloading process is defined as

$$C_{ij} = T_{ij} + \varepsilon E_{ij}, i \in V, j \in \mathcal{J}. \tag{10}$$

Formally, the task offloading problem can be formulated as follows:

$$\begin{aligned}
\min_{u_{i,j}(t)} \; & \sum_{i \in V} \sum_{j \in \mathcal{J}} a_j u_{i,j} C_{ij} \\
\text{s.t.} \; & \sum_{j \in \mathcal{J}} a_j(t) = M(t), \\
& 0 \le ES_i \le AS_i \le AF_i \le LF_i, \\
& \sum_i u_{i,j}(t) = \sum_j u_{i,j}(t) \le min(O(t), M(t)), \\
& u_{i,j}(t) z_i < f_j(t) \tau_f, \\
& u_{i,j}(t) b_i < r_j(t) \tau_b, \\
& a_j \in \{0, 1\}, u_{i,j} \in \{0, 1\}, \\
& 1 \le t \le min(T_0, T), \quad \forall i \in V, \; \forall j \in \mathcal{J}.
\end{aligned} \tag{11}$$

C_{ij} is the weighted integration of service time and resource rental fees of subtask i offloaded to helper j. Thus, the total cost of task processing depends on the task offloading policy $\boldsymbol{u}$. That is, a selected helper with more ample resources will finish task quickly but cost much in rental fees for computing capability and bandwidth. Otherwise, the second resources-rich helper may lead a longer-term low payment. Therefore, the expression $\sum\limits_{i \in V} \sum\limits_{j \in \mathcal{J}} a_j u_{i,j} C_{ij}$ stands for the cost of the whole task processing. In addition, the value of tradeoff parameter ε depends on the specific application scenarios. For instance, in vehicular network the information about traffic situation is delay-strict so ε must be small for timely transmission to ensure security. ε in smart home is better to be larger for energy conservation.

V. PRECEDENCE-CONSTRAINED TASK OFFLOADING

To realize the offloading of task with precedence constraints in IC-IoT, this problem is modeled as a dynamic generalized assignment problem (DGAP) with multi-resources constraints in problem (11) which is a proved NP-hard problem [32].

Despite many efforts and achievements in solutions of this problem, most of them are feasible in specific given scenarios. Some methods reduce the DGAP model to a number of classical deterministic assignment problems stated at discrete time points but it is not feasible under multiple resources constraints. Exact dynamic programming methods (e.g., branch-and-bound) has higher time complexity, which are not scalable and acceptable.

A. TASK OFFLOADING ALGORITHM BASED ON BIPARTITE MATCHING

In general, key challenges of solving the problem are devices mobility and dynamic private task generation that causes the available resources changing with time. Dynamic environment urges us to sense the system information such as association states between user and helpers, and available devices resources in each time interval. For reducing the complexity of sense, the association states $\boldsymbol{a}$ among mobile devices are predicted at first according to the mobility information analysis with IMM.

After association states sensing, how to find optimal subtask-help pairs for lowest cost in every time interval is the important component of proposed task offloading algorithm.

1) FEASIBILITY DISCUSSION OF MATCHING THEORY

The centralized matching algorithms (i.e. there arises a trusted third party who collects information, runs the matching algorithm, and announces the matching results) are already used in wireless communication system resources allocation. For example, Gale-Shapley matching is used to realize energy-efficient task assignment and route planning in [33]. Gu *et al.* [34] find a stable matching between admissible D2D pairs and channel reuse partners to maximize the system throughput with matching algorithm.

In the BS-assisted D2D system, as mobile devices step into the BS coverage, they will register to the network through BS. Such registrations contain mobility and identity information such as velocity or type of offered/required services [35]. Naturally, BS can collect global information of its coverage to make D2D pairing process for more energy efficient and less time consuming task offloading. Yet, the centralized control brings more overheads. For overhead reduction, some works have proposed alternative approaches on schedule methods (e.g. building resources pool [35]) and control information transmission (e.g. sparse vector coding for short packet transmission [36]).

Algorithm 1 Bipartite Graph Matching With Hungarian Algorithm

```
input: weight matrix $C^0_{O(t)\times M(t)}$
output: matching $u_{ij}$, sum of cost $C$
Construct cost matrix $C^0_{n\times n} \leftarrow C^0_{O(t)\times M(t)}$
for $i = 1 \rightarrow n$ do
    $C_1 \leftarrow c_{ij} - \min\limits_{j} c_{ij}$
end for
for $j = 1 \rightarrow n$ do
    $C_2 \leftarrow c^1_{ij} - \min\limits_{i} c^1_{ij}$
end for
$lines \leftarrow 0$
while $lines < n$ do
    Cover all zeros in $C_2$ with least $lines$.
    $k \leftarrow$ the smallest element that is not covered
    $c^3_{ij} \leftarrow$ elements that are not covered.
    $C_3 \leftarrow c^3_{ij} - k$
    $c^4_{ij} \leftarrow$ elements that are covered twice.
    $C_4 \leftarrow c^4_{ij} + k$
end while
$W \leftarrow$ The elements in $C^0$ corresponds to the 0 elements
    in $C_4$ with the same index.
$C \leftarrow \sum W$
$u_{i,j} \leftarrow$ index of 0 elements in $C_4$.
```

To take full advantage of network control, we draw on the matching solution for this problem. In each time interval, the subtasks offloading problem is regarded as a weighted bipartite matching, where the one side represents the set of all subtasks that need to be offloaded, the other side is the set of forward/fog nodes who are willing and able to share resources for task execution. Moreover, the objective function values (cost) of subtask-helper pairs are used to label the edges of bipartite graph.

2) SUBTASK-HELPER MATCHING BY HUNGARIAN ALGORITHM

Hungarian algorithm is accepted to find optimal assignment in graph matching. The main idea of it is replacing the previous matching policy with a new augmenting path. The optimal assignment come out until there is no new augmenting path. Specifically, the minimum weight binary graph matching could follow algorithm 1.

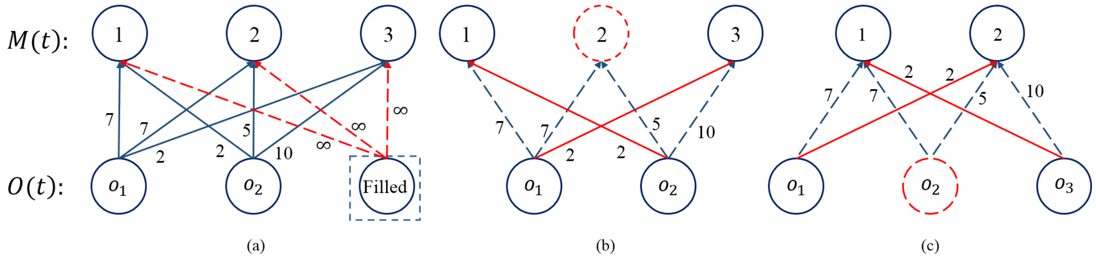


FIGURE 2. Illustration of IC-IoT with different number of subtasks and helpers in time interval t.

To satisfy the input requirement (i.e. a square matrix), the cost matrix is reconstructed. If there are $O(t)$ tasks in $Can(t)$ ($Can(t) = \mathcal{O}(t)$), $M(t)$ helpers candidates could provide help and $M(t) \neq O(t)$, then a weight matrix $C(t)$ with the $max(M(t), O(t)) \times max(M(t), O(t))$ dimensions is constructed by adding extra elements. The cost of filled subtask(helper)- helpers(subtasks) pairs is ∞ to express there is no matching. Besides, if the forward nodes cannot support the execution of subtask i under delay constraint (i.e. $min(e_j, LF_i)$) with their resources, the weight is also set to ∞ to meet the delay and resources constraints.

Take an example, there is a bipartite graph $BG(t) = \{\mathcal{O}(t), \mathcal{M}(t), C(t)\}$. As illustrated in Fig.2, in time t, there are two subtasks that need to be executed in $Can(t)$ set, denoted as $\mathcal{O}(t) = \{o_1(t), o_2(t)\}$, by three helper candidates $\mathcal{M}(t) = \{m_1(t) = 1, m_2(t) = 2, m_3(t) = 3\}$. The cost $C(t)^0_{2\times 3} = \{c_{i,j}\}$ indicates the cost of subtask o_i offloaded to helper $m_j(t)$ is c_{ij}. Each helper can serve exactly one subtask in one time interval. The objective is to minimize the total cost required to perform all subtasks. To satisfy the input constraint of Hungarian algorithm, one filled element is added in $\mathcal{O}(t)$ and the weight is set to ∞ shown in Fig.2(a). In this example, it turns out to be optimal to assign helper 1 to subtask o_2, helper 3 to subtask o_1, and helper 2 is not to be assigned in time t as presented in Fig.2(b). The total cost required is $2+2=4$. All other assignments lead a larger cost. If in time interval t the number of helpers is less than that of subtasks which need to be offloaded as Fig.2(c), the extra subtasks o_2 would be computed locally or wait for next chance in following time intervals when the time does not exceed LS_i ($i = o_2$). In other words, after matching in time t subtask o_2 is still in *Can* and other two subtasks is in *Doing*.

More details about proposed task offloading algorithm is clarified in algorithm 2 that is accomplished by BS. It includes delay constraints estimation and the minimum weight matching between subtasks and helpers in each time interval by Hungarian algorithm.

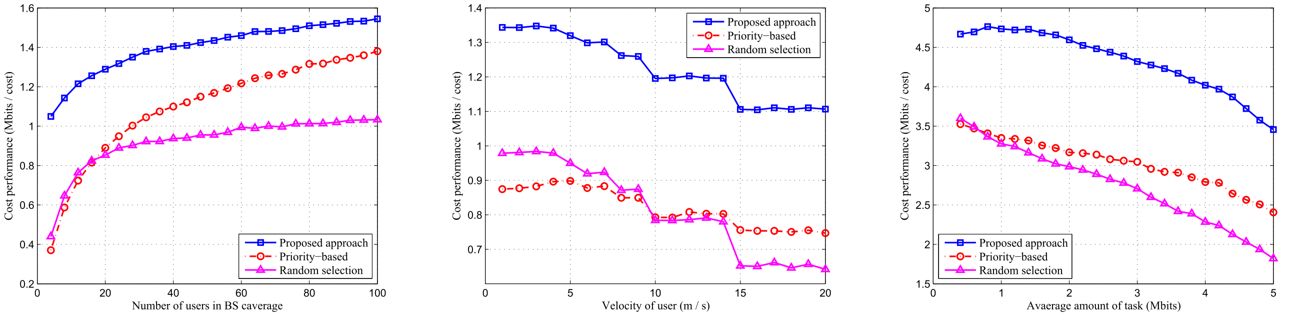
Algorithm 2 proposed Task Offloading Algorithm Based on Weighted Bipartite Matching

```
input: $G\{V, Z, B, E\}, T, v_0$
output: $C, \boldsymbol{u}, b_f$
$T_0 \leftarrow v_0$
$ES \leftarrow ES_{entry} = 0, eq.(1)$
$LF \leftarrow LF_{exist} = min(T, T_0), eq.(2)$
if $ES_i < 0, or\ LF_i > T$ then
    $ES_i \leftarrow 0$
    $LF_i \leftarrow T$
end if
Get $a(t), M(t), < s_j, e_j >$ from IMM.
$t = 1$
while $t < min(T, T_0)$ or $\forall i \in Done$ do
    if $t > ES_i$ and $t > AF_{suc_i}$ then
        Put $i$ into $Can(t)$, $Can(t) \leftarrow i$
    end if
    Compute cost $C_{O(t)\times M(t)}$ for each pair based on eq. (10).
    $C(t), u_{i,j}(t) \leftarrow Hungarian(C_{O(t)\times M(t)})$.
    Update sets: $Can, Doing, Done$.
    $t = t + 1$
end while
$C = \sum C(t)$.
$\boldsymbol{u} = \bigcup_t u_{i,j}(t)$.
$b_f = \sum_{i \in Done} b_i(t)$.
```

B. COMPUTATIONAL COMPLEXITY ANALYSIS

As to the complexity of the proposed task offloading algorithm, the number of vertexes is N ($N = O(t)+M(t)$) and the maximum number of edges is $N^2/2$ in the bipartite graph. The complexity of Hungarian algorithm is classic $O(N^3)$ implementation [10]. Therefore, the proposed task offloading algorithm has a $O(N^3)$ complexity in each time interval. The iterations in Hungarian algorithm depend on the number of vertexes requiring matchings. In practice, the numbers of mobile subtasks $O(t)$ and the number of met mobile helpers $M(t)$ in general are limited in a time interval. In addition, the bipartite graph normally will be not fully connected, and the resources constraints reduce the research space.

In the proposed algorithm, the matching between subtasks and helpers occurs in each interval so the total complexity of task offloading is $O(T_0N^3)$. The delay constraint T_0 is another impact factor. When user moves quickly,



(a) Number of helpers with $b \in 2-5$ Mbits, $v_0 = 5$

(b) User velocity with $j = 25$, $b \in 2-5$Mbits

(c) Amount of subtasks with $j = 25$, $v_0 = 5$

FIGURE 3. Cost performance with delay constraint $T = 20$, tradeoff parameter $\varepsilon = 1$, maximum current load percentage of helper $L = 0.7$.

the residence time T_0 is limited decreasing the scheduling times. Otherwise, when user move slowly, it in one time interval gets more helpers which speed up task execution and reach the terminating condition iteration quickly.

Therefore, we consider that the computation complexity of the proposed task offloading algorithm should be acceptable in practice. Besides, the BS can also leverage multiple processors to adopt some parallel implementations for graph matching.

VI. PERFORMANCE ANALYSIS

In this section, a simulation experiment is provided concerning the precedence-constrained task offloading via D2D in IC-IoT. Firstly, the performance of proposed task offloading scheme is evaluated by comparing with priory-based offloading and random helper selection. Then, the effects of tradeoff parameter ε and maximum private load percentage of helper L on the performance of task offloading are discussed.

A. EXPERIMENT SETUP

The BS coverage area is assumed as $100m \times 100m$, and the D2D range is $20m$. The user average velocity in the coverage is 5m/s. As for the propagation gain, the path loss of D2D links is -3 [10]. Besides, channel bandwidth W=$20MHz$, noise power $\sigma^2 = 2 \times 10^{-8}W$, intra-interference $I_n \propto d_0^{-4}$ (d_0 is the average distance between helpers and neighboring BS.), transmission power for user is $P_{\text{tx}} = 200mW$. The data size of subtask is $b_i \in [0.5, 2]Mb$, and the computing demands of subtasks are uniformly distributed $z_i \in [0.5, 2]$Gigacycles. The upper bound of helpers' CPU frequency are uniformly distributed $F_j \in [3, 5]$GHz. Moreover, the task graph is assumed as Fig.1, which includes seven subtasks connected with precedence relations. The price of unit bandwidth and computing resource are set to unit cost for simple simulation.

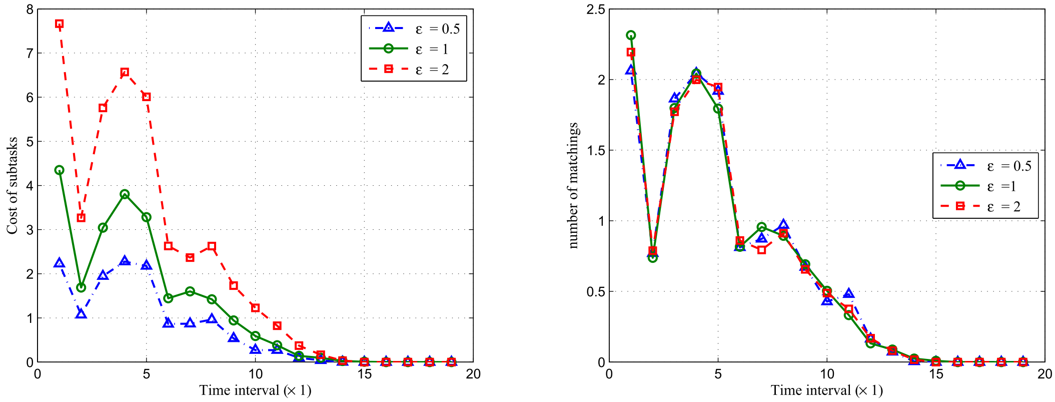
B. PERFORMANCE COMPARISON

The proposed task offloading approach in this paper is compared with other two different offloading strategies: priority-based offloading scheme and random selection scheme. We formulate 2000 times for average value of cost performance.

As for priority-based offloading scheme, every subtask has a priority that is predefined. Then, subtasks are scheduled in the descending order of priorities. Moreover, the BS sorts all feasible helpers for each ready subtask, and chooses the lightest pair for subtask among the remaining helpers. The priority is defined as the order of getting into *Can*. The earlier subtask is put into *Can*, the higher priority it will get. In random selection, BS chooses the subtask-helper pairs randomly for task execution among all the feasible pairs. We evaluate offloading performance according to cost for per unit amount of finished task, called cost performance (i.e. b_f/C (Mb/cost), b_f is the amount of finished task.). A larger value of cost performance means a lower cost for per-bit task service and a more efficient offloading policy.

The cost performance of different schemes for different numbers of users in BS coverage are depicted in Fig.3(a). As can be seen, cost performances of three schemes become stable after a slight increase with the increase of helpers number in the BS coverage. In addition, the performance of proposed scheme represents notable improvement in all conditions while the figure of priority-based offloading policy takes over that of random selection at the helpers' number in 14. The reason is that a larger number of helpers in this coverage would bring more neighbors for user in each time interval. Thus, it leads user have more opportunities to select suitable helpers to help task execution on the average, which leads all the three scheme increase in general. Moreover, the performance of random selection is better than that of priority-based method when the helpers number is less than 14 because priority-based method selected the optimal helper for the subtask with high priority ignoring the sum cost of all available subtasks in the time interval. With the increase of helpers number, user selects the helpers according to task priority that leads first-arrival first-service, which reduces the risk of following subtasks violating the delay constraints.

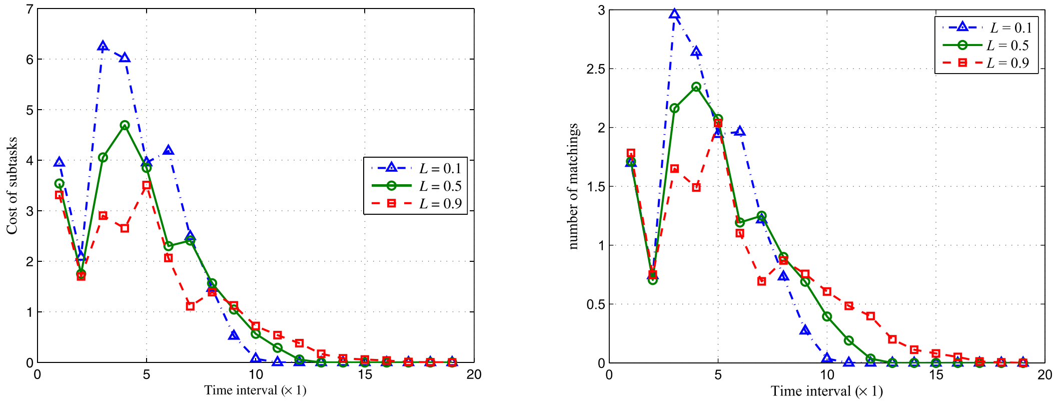
Fig. 3(b) shows the cost performances of three schemes with different user velocities. With the increase of user velocity, the figure of them all decrease in terraces but greater changes has occurred in the figure of random selection.



(a) Cost of subtask

(b) Number of matchings

FIGURE 4. **Effect of tradeoff parameter ε.**



(a) Cost of subtask

(b) Number of matchings

FIGURE 5. **Effect of helpers' upper bound of current load percentage L.**

Proposed approach shows the best performance while the random one presents better than the priority-based method until the user velocity is 10m/s. As user speed increases, the situation of random and priority-based become opposite. The reason of the result is that the faster user moves, the shorter residence time in BS coverage is. That results in less scheduling times for task and less task is completed by helpers with the BS assistance. Hence, the high-speed user movement affects the overall performance.

The cost performances of different schemes with different subtask sizes are discussed in Fig. 3(c). Cost performance of all three schemes are in decrease with average subtask amount increasing. The proposed scheme keeps the best result while the figure of priority-based method remain larger than that of random selection. The reason is that size of subtask becomes larger and delay constraint T_0 does not change. More helpers cannot support task execution with spare resources under the delay constraint. The number of idle helpers decreases, and thus there are less feasible subtask-helper pairs. That will raise the opportunities for helpers who carries ample resources with high rental charges, thus it brings high cost.

C. EFFECT OF PARAMETERS

This part discusses the effects of the tradeoff parameter ε and upper bound of helpers current load L on the performance. We formulate the scheduling process of one precedence-constrained task and record cost of subtasks and number of matchings in each time interval.

1) TRADEOFF PARAMETER ε

The effects of tradeoff parameter on subtasks cost and matching numbers in each time interval are discussed in Fig. 4. In different tradeoff parameters simulations, the value of ε is respectively set to 0.5, 1, and 2 with the current load upper bound $L = 0.7$. The cost of subtask in one time interval increases as ε increases as shown in Fig. 4(a). For each time interval, the numbers of matchings are closed with different ε in Fig. 4(b). Larger ε means BS prefers the helpers whose fees of bandwidth and computing capabilities are lower among the candidates even though it will bring long execution period. That is suitable for delay sensitive applications. Because different selection criterions decide different helper-selection results among the same candidates, that does not affect the number of matchings. Besides, ε changes from 0.5 to 2 doubling partial cost that leads the linear growth of subtasks cost in one time interval.

2) CURRENT LOAD UPPER BOUND OF HELPER L

Fig. 5 shows the effect of the upper bound of helper current load L on processing cost and matchings number in each time interval. To analyze the effects, the load percentage upper bound L is respectively set to 0.1, 0.5, 0.9 with the tradeoff parameter $\varepsilon = 1$. The result indicates that the cost of subtasks and the number of matchings will get heavier tail and lower peak value with larger L. The larger L means there are more private applications for helpers who remain less computing resources for task offloading. Thus, when L is large, it is difficult to find an appropriate helper who may provide ample resources to execute subtask satisfying the delay constraint. That leads less matchings in each time interval, and naturally less cost of subtasks. To finish the task, subtasks has to be scheduled in more time intervals, which leads the heavier tail.

VII. CONCLUSION

In this paper, an efficient D2D computation task offloading scheme is proposed for precedence-constrained task in IC-IoT. It aims to minimize the weighted sum of task processing delay and resources rental fees jointly considering the constraints of task delay, association states and available resources. Specifically, a precedence-constrained task graph is constructed to calculate the subtask delay constraints and mobility information is modeled to estimate the association states between user and helper candidates. Then, because the available resources is time-varying, Hungarian algorithm is employed to find lightest subtask-helper pairs by updating the cost matrix in each time interval. Finally, the simulation results indicate the effectiveness of proposed approach on cost performance comparing with priority-based offloading scheme and random selection.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[2] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of green information-centric networking: Research issues and challenges," *IEEE Commun. Surv. Tuts.*, vol. 17, no. 3, pp. 1455–1472, 3rd Quart., 2015.

[3] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the Internet of Things: Challenges and opportunities," *IEEE Netw.*, vol. 30, no. 2, pp. 92–100, Mar. 2016.

[4] Y. Mao, J. Zhang, Z. Chen, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[5] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," *IEEE Netw.*, vol. 32, no. 4, pp. 54–60, Jul./Aug. 2018.

[6] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 42–49, Dec. 2009.

[7] Z. Zhou, H. Yu, C. Xu, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Dependable content distribution in D2D-based cooperative vehicular networks: A big data-integrated coalition game approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 953–964, Mar. 2018.

[8] Z. Zhou, C. Gao, C. Xu, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Social big-data-based content dissemination in Internet of vehicles," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 768–777, Feb. 2018.

[9] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Trans. Commun.*, vol. 18, no. 3, pp. 1750–1763, Feb. 2019.

[10] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficientand incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[11] 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, 4th Quart., 2017.

[12] S. Bi and Y. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.

[13] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "A cooperative scheduling scheme of local cloud and Internet cloud for delay-aware mobile cloud computing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–6.

[14] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1451–1455.

[15] M. Kamoun, W. Labidi, and M. Sarkiss, "Joint resource allocation and offloading strategies in cloud enabled cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 5529–5534.

[16] W. Labidi, M. Sarkiss, and M. Kamoun, "Energy-optimal resource scheduling and computation offloading in small cell networks," in *Proc. 22nd Int. Conf. Telecommun. (ICT)*, Apr. 2015, pp. 313–318.

[17] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[18] G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang, "Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4642–4655, Oct. 2018.

[19] M. Deng, H. Tian, and B. Fan, "Fine-granularity based application offloading policy in cloud-enhanced small cell networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, May 2016, pp. 638–643.

[20] Y. Zhao, S. Zhou, T. Zhao, and Z. Niu, "Energy-efficient task offloading for multiuser mobile cloud computing," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Nov. 2015, pp. 1–5.

[21] S. Zhang, J. Wu, and S. Lu, "Distributed workload dissemination for makespan minimization in disruption tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 7, pp. 1661–1673, Jul. 2016.

[22] W. Zhang, Z. Zhang, S. Zeadally, and H.-C. Chao, "Efficient task scheduling with stochastic delay cost in mobile edge computing," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 4–7, Jan. 2019.

[23] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 319–333, Feb. 2019.

[24] S. E. Mahmoodi, R. N. Uma, and K. P. Subbalakshmi, "Optimal jointscheduling and cloud offloading for mobile applications," *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 301–313, Jun. 2019.

[25] Y. Kim, J. Kwak, and S. Chong, "Dual-side optimization for cost-delay tradeoff in mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1765–1781, Feb. 2018.

[26] Y. Cai, F. R. Yu, C. Liang, B. Sun, and Q. Yan, "Software-defined device-to-device (D2D) communications in virtual wireless networks with imperfect network state information (NSI)," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7349–7360, Sep. 2016.

[27] B.-L. Lu and M. Ito, "Task decomposition and module combination based on class relations: A modular neural network for pattern classification," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1244–1256, Sep. 1999.

[28] C.-H. Yu, K. Doppler, C. B. Ribeiro, and O. Tirkkonen, "Resource sharing optimization for device-to-device communication underlaying cellular networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2752–2763, Aug. 2011.

[29] D. Feng, L. Lu, Y. Yuan-Wu, G. Li, S. Li, and G. Feng, "Device-to-device communications in cellular networks," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 49–55, Apr. 2014.

[30] X. Wang, X. Lei, P. Fan, R. Q. Hu, and S.-J. Horng, "Cost analysis of movement-based location management in PCS networks: An embedded Markov chain approach," *IEEE Trans. Veh. Technol.*, vol. 63, no. 4, pp. 1886–1902, May 2014.

[31] X. Ge, J. Ye, Y. Yang, and Q. Li, "User mobility evaluation for 5G small cell networks based on individual mobility model," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 528–541, Mar. 2016.

[32] O. E. Kundakcioglu and S. Alizamir, *Generalized Assignment Problem*. 2nd ed. Springer: Boston, MA, USA, 2008.

[33] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5526–5538, Nov. 2018.

[34] Y. Gu, Y. Zhang, M. Pan, and Z. Han, "Matching and cheating in device to device communications underlying cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2156–2166, Oct. 2015.

[35] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklós, and Z. Turányi, "Design aspects of network assisted device-to-device communications," *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 170–177, Mar. 2012.

[36] H. Ji, S. Park, and B. Shim, "Sparse Vector Coding for Ultra Reliable and Low Latency Communications," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6693–6706, Oct. 2018.

JINDOU XIE received the B.E. degree in communication engineering from Chongqing University, Chongqing, China, in 2018, where she is currently pursuing the Ph.D. degree in information and communication engineering.

Her research interests include intelligent mobile networks, mobile edge computing, and the Internet of Things (IoT).

YUNJIAN JIA received the B.S. degree in engineering from Nankai University, Tianjin, China, and the M.E. and Ph.D. degrees in engineering from Osaka University, Suita, Japan, in 1999, 2003, and 2006, respectively.

From 2006 to 2012, he was a Researcher with the Central Research Laboratory, Hitachi, Ltd., Japan, where he was involved in research and development on wireless networks, and contributed to LTE/LTE-Advanced standardization in 3GPP. He is currently a Professor with the College of Communication Engineering, Chongqing University, Chongqing, China. He is the author of more than 70 published papers. He holds 30 granted patents. His research interests include future radio access technologies, intelligent mobile networks, and the Internet of Things (IoT).

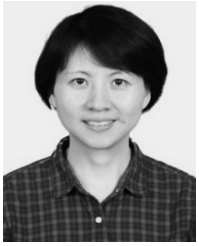
Dr. Jia has received several prizes from the industry and academia, including the IEEE Vehicular Technology Society Young Researcher Encouragement Award, the IEICE Paper Award, the Yoko-suka Research Park R&D Committee YRP Award, and the Top 50 Young Inventors of Hitachi. He received the Research Fellowship Award of the International Communication Foundation, in 2004, and the Telecommunications Advancement Foundation Japan, in 2005.

ZHENGCHUAN CHEN received the B.E. degree from Nankai University, China, in 2010, and the Ph.D. degree from Tsinghua University, China, in 2015.

He visited The Chinese University of Hong Kong, in 2012, and the University of Florida, USA, in 2013. From 2015 to 2017, he was a Postdoctoral Research Fellow with the Singapore University of Technology and Design. He is currently an Assistant Professor with the College of Communication Engineering, Chongqing University, China. His current research interests include the Internet of Things, age of information, and network information theory. He has served in several IEEE conferences, including the IEEE GLOBECOM, as a Technical Program Committee Member. He was a co-recipient of the Best Paper Award at the International Workshop on High Mobility Wireless Communications, in 2013. He was selected as an Exemplary Reviewer of the IEEE TRANSACTIONS ON COMMUNICATIONS, in 2015.

ZHAOJUN NAN received the B.Eng. degree in automation from Jamusi University, Jamusi, China, in 2009, and the M.Eng. degree in navigation guidance and control from Harbin Engineering University, Harbin, China, in 2012. He is currently pursuing the Ph.D. degree in information and communication engineering with Chongqing University, Chongqing, China.

His research interests include wireless communication and optimization, edge caching, and fog computing.



LIANG LIANG received the B.Eng. and M.Eng. degrees from the Southwest University of Science and Technology (SWUST), China, in 2003 and 2006, respectively, and the Ph.D. degree in communication and information system from the University of Electronic Science and Technology of China (UESTC), in 2012. From 2011 to 2012, she was an International Visitor with the Institute for Infocomm Research (I2R), Singapore. She is currently an Associate Professor with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, China. Her research interests include wireless communication and optimization, wireless network virtualization, mobile edge computing, and the Internet of Things (IoT).