Energy Efficient Relay Selection and Resource Allocation in D2D-Enabled Mobile Edge Computing

Yang Li, *Member, IEEE*, Gaochao Xu, *Member, IEEE*, Kun Yang, *Senior Member, IEEE*, Jiaqi Ge, Peng Liu, and Zhenjun Jin

***Abstract*—In order to improve resource utilization and network capacity, we propose the Device-to-Device (D2D) enabled Mobile Edge Computing (MEC) system, where multiple Smart Devices (SDs) transmit the offloading data to the MEC server with the help of wireless access point (WAP) selected from multiple WAPs. The SD uses the chosen WAP as the communication relay between the MEC server and itself. Aimed to minimize the total energy consumption of the system and satisfy the SDs demand on delay, we jointly optimize relay selection and resource allocation in D2D-enabled MEC system. The problem is formulated as an integer-mixed non-convex optimization problem which is a NP-hard problem. We thus propose a two-phase optimization algorithm that jointly optimizes relay selection policy and resource allocation strategy. In first phase, the original problem is converted into a convex optimization problem by using convex optimization techniques, and the optimal relay selection policy can be achieved by solving the relay selection problem. After obtaining the relay selection policy, the original problem is transformed into a resource allocation problem solved by leveraging the Lagrange Method in the second phase. Furthermore, the proposed algorithm is a low-complexity algorithm which is associated with the root finding method. The optimal relay selection policy and resource allocation strategy can be found in polynomial time. The extensive simulation results are provided to indicate that the D2D-enabled MEC system achieves remarkable results in energy saving. Compared with other baseline methods, our proposed algorithm can not only achieve the optimal solution with less time cost, but also improve the energy efficiency and network capacity.**

***Index Terms*—Mobile edge computing, convex optimization, D2D communication, relay selection, resource allocation.**

Manuscript received May 14, 2020; revised October 1, 2020; accepted October 27, 2020. Date of publication November 6, 2020; date of current version January 22, 2021. This work was supported in part by the National Key R&D Plan of China under Grant 2017YFA0604500, in part by the Jilin Province Science and Technology Development Plan Project under Grants 20190201180JC and 20200401076GX, in part by the Jilin Province Education Department 13th Five-Year Science and Technology Project under Grant JJKH20191310KJ. The review of this article was coordinated by Dr. Kaigui Bian. *(Corresponding author: Zhenjun Jin.)*

Yang Li is with the School of Information Science and Technology, North China University of Technology, Beijing 100144, China, and also with the College of Computer Science and Technology, Jilin University, Changchun 130012, China (e-mail: liyang17@mails.jlu.edu.cn).

Gaochao Xu and Jiaqi Ge are with the College of Computer Science and Technology, Jilin University, Changchun 130012, China (e-mail: xugc@jlu.edu.cn; 752332557@qq.com).

Kun Yang is with the School of Information Science and Technology, Northwest University, Xi'an, China, and also with the School of Computer Science and Electronic Engineering (CSEE), University of Essex, Colchester, U.K. (e-mail: kunyang@essex.ac.uk).

Peng Liu is with the College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China (e-mail: pliu16@mails.jlu.edu.cn).

Zhenjun Jin is with the College of Computer and Engineering, Changchun University of Technology, Changchun 130012 (e-mail: 34712384@qq.com).

Digital Object Identifier 10.1109/TVT.2020.3036489

I. Introduction

With the development of wireless communication, the demand for spectrum resources is growing. With the landing of the 5th generation wireless systems (5G [1]), the access of large-scale smart devices (SDs) will inevitably lead to a shortage of spectrum resources [2]. With the application of artificial intelligence, healthcare, voice recognition and virtual reality in the field of mobile communication, the computation tasks put forward higher requirements on the computation capacity and data access of SD. However, limited by the finite spectrum resources [3], [4], transmitting the offloading task to the edge server directly will result in an increase in transmission energy of SDs under the influence of bandwidth, transmission distance and other factors, which means that computation offloading [5] cannot achieve high energy efficiency. Therefore, the finite energy, bandwidth, and computation and communication resource bandwidth of SD pose great challenge on accomplishing the computation-intensive delay-sensitive task [6].

Although Mobile Edge Computing (MEC) [7] provides powerful computation capacity, the large-scale of transmission data can not be completed due to the latency constraint and the low data transmission rate. Therefore, edge caching [8], [9] is proposed to avoid unnecessary repetitive transmission of data and tasks. However, the input data of mobile application is constantly updated, and the continuous data caching increases the communication overhead so that the issue of high energy consumption cannot be solved substantially. Improving communication capacity is the fundamental solution. Hence, Device-to-Device (D2D) communication technology [10], [11] is proposed to solve the problem perfectly. In order to enable the remote SDs to utilize the computation resource of the MEC server, an optimal relay should be opted for accomplishing the transmission of the computing data [12]. Therefore, as shown Fig. 1, this paper proposes a D2D-enabled MEC system to maximize resource utilization, improve energy efficiency and communication capacity, where multi-SD and multiple wireless access points (i.e., Laptop, Tablet, iPAD) are included in a certain area. Meanwhile, joint optimization of wireless access point selection and resource allocation is studied for energy saving.

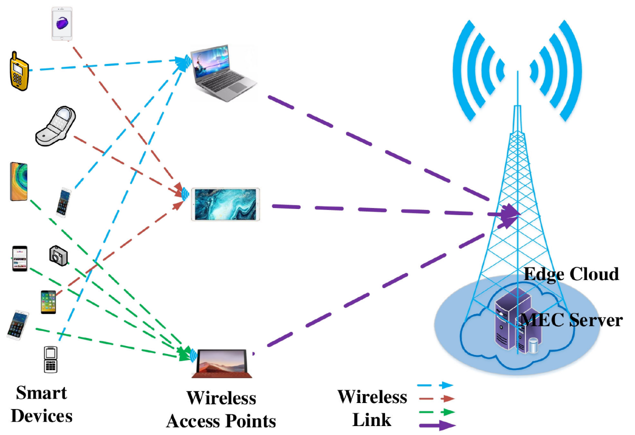

Fig. 1. A D2D-enabled MEC system with multiple smart devices offloading computation tasks to the MEC server through the WAP selected from the multiple Wireless Access Points.

A. Related Work

Because of the finite battery power and computation and communication resource on the SD, SD can offload the computation task to the MEC server and utilize the powerful computing ability of the MEC server to accomplish the task. By integrating the characteristics of computation and communication resource in the MEC system, jointly optimizing resource allocation strategy of MEC server and SDs can improve resource utilization and energy efficiency [13]–[22].

Optimization Resource Allocation: Wang *et al.* [13] integrated dynamic voltage scaling into single-user MEC system to minimize the energy consumption of SD and the latency of application execution by jointly optimizing the CPU cycles frequency, transmission power and offloading ratio. Liu *et al.* [14] achieved the trade-off between the computation model, local computing or edge computing fully, and the transmission power in a multi-channel multiuser MEC system. Liu *et al.* [15] made a trade-off between the reliability and latency in a MEC-enabled Augmented Reality (AR) system, and proposed an online offloading decision algorithm to optimize the selection of transmission rate and MEC server. Kuang *et al.* [16] proposed an energy efficient and low latency partial offloading mechanism by jointly optimizing the task offloading decision, offloading scheduling and power allocation in a single-user MEC system which consisted of a single mobile device with multiple independent computation tasks and parallel implementation of task offloading and execution. Rodrigues *et al.* [17] presented a heuristic algorithm by utilizing virtual machine migration and transmission power control for reconfiguring cloudlets to balance the workload between cloudlets and consequently maximize cost-effectiveness in a multi-user multi-cloudlet scenario. Rodrigues *et al.* [18] presented an approach for minimizing delay that considered both communication and computation elements by using virtual machine migration and transmission power control in a scenario with two cloudlet servers. Ren *et al.* [19] aimed to improve the quality of experience (QoE) for users by jointly optimizing communication and computation resource allocation in a TDMA-based multi-user MEC system. Tran *et al.* [20] studied the joint task offloading and resource allocation problem in a multi-cell MEC network and achieved the trade-off between task offloading and resource allocation. Ning *et al.* [21] aimed at minimizing the overall energy consumption in a three-layer system model with Edmonds-Karp and deep reinforcement learning methods. Ning *et al.* [22] comprehensively considered both of cloud computing and MEC from single-user computation offloading problem to the multiuser model.

Up to now, it is difficult to improve resource utilization and energy efficiency simply by jointly optimizing computation and communication resource. By analyzing the communication model of related works, the transmission energy consumption is affected by many factors, i.e., bandwidth, channel gain, signal-to-noise ratio and transmission power. Low bandwidth, long distance and large-scale data transmission will lead to more energy consumption. For further improvement, some researches [23]–[28] proposed edge caching.

Edge Caching: Wang *et al.* [23] formulated an optimization problem that jointly optimized the computation offloading decision, radio resource allocation, MEC computation resource allocation and content caching policy in wireless cellular network with MEC. Liu *et al.* [24] considered a multi-user cache-enhanced MEC system that jointly optimized data caching policy, computation and communication resource, and applied the block coordinate descent and convex optimization techniques to minimize energy consumption. Tang *et al.* [25] studied the resource allocation strategy between the MEC server and BSs, and proposed a Stackelberg game method to obtain the caching and computation resource allocation strategy. Zhang *et al.* [26] proposed a centralized MEC network which included one macro eNodeB and multi-SD, and improved the user QoS by jointly optimizing the computation offloading, content caching and resource allocation. [27], [28] pursued the studies with the purpose of increasing network throughput and the successful ratio of task offloading through combining edge caching.

Although data caching can decline the total energy consumption, the whole data caching consumes more energy than the energy cost of the partial data offloading. In order to decrease the energy consumption of the SD and enable remote SDs to utilize MEC server resource, D2D communication [29]–[35] and relay selection [12], [36]–[44] are integrated into the mobile edge computing to improve network capacity.

D2D Communication: Hou *et al.* [29] proposed a novel framework that SD could choose the optimal computation offloading mode between MEC-offloading and D2D-offloading in D2D-assisted cellular networks integrated with MEC system. Liu *et al.* [30] proposed an edge resource pooling framework with multi-SD and an edge server. The proposed framework utilized D2D task offloading to achieve computation resource pooling and sharing among SDs. Liu *et al.* [31] adopted the MEC-enabled network integrated with D2D communication to enhance traditional wireless networks capacity, and the proposed resource allocation method significantly improved the system throughput. Xing *et al.* [32] studied the TDMA-based D2D-enabled multi-helper MEC system where SD offloaded the computation task to its nearby SDs served as helpers for cooperative computation. Kuang *et al.* [33] proposed a joint

D2D User Equipments mode selection, base station selection, power allocation and channel allocation algorithm based on particle swarm optimization algorithm to maximize the system energy efficiency in multiple base stations heterogeneous cellular networks with D2D communication. Tang *et al.* [34] proposed a distributed anti coordination game based partially overlapping channels (POCs) assignment algorithm to solve the channel assignment problem in a combined UAV and D2D based network with the dynamic topology and high mobility of nodes. He *et al.* [35] integrated MEC system and D2D communication to further improve the computation capacity of the cellular networks, and aimed to maximize the supported SDs number and formulated the problem as a mixed-integer non-linear problem solved by using a two-phase optimization algorithm.

Relay Selection: The user-relay assisted wireless networks [36] were promising candidates to fulfill the demanding coverage and capacity requirements of 5G based on orthogonal frequency division multiplexing access (OFDMA). In order to support long-distance users and improve energy efficiency of the wireless networks, relay-assisted D2D communication in 5G networks [37] was applied. Meanwhile, a single cell OFDMA-based user-relay aided downlink network [12] was considered. Similarly, a novel C-RAN architecture [38] was proposed to minimize the energy consumption. Li *et al.* [39] proposed an optimal relay selection policy to improve the system performance and conserve energy consumption with D2D relay communication technology. Omran *et al.* [40] developed a joint relay selection and load balancing scheme to determine which SD could act as a relay. Li *et al.* [41] aimed to minimize the total energy consumption in a MEC system with multiple smart wearable devices and a smart mobile device by jointly optimizing the two-tier computation and communication resource. An OFDMA MEC system with multi-SD and multi-relay, and a MEC server was proposed to minimize the total energy const by selecting the optimal computation model from executing locally, offloading to the relay device and accomplishing on the relay device, offloading to the MEC server directly or via a relay device [42]. A wireless relay-enabled task offloading framework [43] was proposed to improve the quality of the patient monitoring services, and the intra-body task can be executed by the selected relay, by other relays, or by the MEC server. Li *et al.* [44] aimed to minimize the system cost by providing the optimal computation offloading strategy, transmission power allocation, bandwidth assignment in a multiple wireless access points network with MEC. Huang *et al.* [45] jointly optimized the individual SD's task offloading decisions, transmission time allocation between wireless powered transfer and task offloading, and time allocation among multiple SDs according to the time-varying wireless channels in a wireless powered MEC network with one access point and multiple SDs. An online task offloading decisions and wireless resource allocations algorithm based on deep reinforcement learning was proposed to maximize the weighted sum computation rate with binary computation offloading in [45]. Since the massive amounts of devices will result in too much data and parameters to be processed in the MEC scenarios with many and varied users, servers and applications, Rodrigues *et al.* [46] demonstrated the best solution by utilizing Machine Learning (ML) algorithms, which enabled the computer to draw conclusions and made predictions based on existing data without human supervision, leading to quick near-optimal solutions even in problems with high dimensionality.

However, in these studies, the computation tasks are executed by the optimal computation model selected from local computing, offloading to the relaying and completing on the relay device, and offloading to the MEC server directly or through a relay. And these researches do not take the measure of partial computation offloading to achieve a more reasonable resource allocation in the MEC system. Therefore, the study of joint optimization wireless access point selection policy, partial computation offloading and resource allocation in a multi-user multi-relay MEC server has the important significance and value.

B. Contribution

In this paper, we study the D2D-enabled MEC system that wireless access points help the SD upload the offloading data to the MEC server. There are multiple WAPs and each SD chooses one as a repeater. Compared to the data transmission delay, the processing time of data packets is extremely small which can be ignored. Each SD contains a computation-intensive delay-sensitive task where input data can be partitioned into bitwise. The offloading data is transmitted to MEC server through WAP. The computation feedback to be downloaded to the SD is much shorter than the data offloaded to the MEC server. We safely neglect the time spent on task computing, downloading and processing data packets by the WAPs and the MEC server. Each time block is only occupied by task offloading from SD to MEC server with the help of WAP [12], [22], [38], [42]–[44]. We aim to minimize the total energy consumption of the MEC system by deciding the relay selection policy, the offloading ratio for each SD, the resource allocation in WAP and MEC server while meeting the SD's requirements of task delay. We formulate the energy consumption minimization problem as an integer-mixed non-convex optimization problem, and we propose a lower time complexity algorithm by using convex optimization techniques to solve the NP-hard problem. The main contributions of our work are summarized as follows:

The D2D-enabled MEC system allows the remote SDs to utilize the computation resource of the MEC server with the help of the optimal WAP. As such, it provides more abundant computation resource for the remote SD, and allows the SDs to perform more complex applications and services.

Unlike many existing relay selection and computation offloading strategy, we jointly optimize relay selection, computation offloading and computation and communication resource to minimize the total energy consumption. The remote SDs go for the optimal relay from the WAP set, and the optimal resource allocation strategy can achieve the optimal computation offloading ration and transmission power.

The formulated problem is an integer-mixed non-convex optimization problem which is not easy to solve. Therefore, we propose a joint optimization relay selection and resource allocation algorithm which splits the solution process into two phases.

In the first phase, we transform the original non-convex problem into a convex problem by using convex optimization techniques and obtain the optimal relay selection policy by using Lagrange Method to solve the convex problem. After achieving the optimal relay selection policy, the original problem is converted into a resource allocation problem. The optimal resource allocation strategy can be obtained by solving the convex problem in the second phase.

Simulation results show that not only the D2D-enabled MEC system decrease energy cost with increasing the number of WAPs, but also the relay selection policy and resource allocation strategy can make the minimum energy consumption. Meanwhile, our proposed algorithm can obtain the same energy consumption as the brute method in a shorter execution time.

The remaining part of this paper is organized as follows. System model and problem formulation are introduced in Section II. In Section III, convex optimization techniques are used to deal with the original problem, the optimal relay selection and resource allocation are given by using the Lagrange Method, and joint optimization algorithm of relay selection and resource allocation, i.e., JOSR is proposed. Section IV gives the simulation results and discussion. Finally, we summarize our paper in Section V.

II. System Model and Problem Formulation

As shown in Fig. 1, we study the D2D-enabled MEC system consisting of N smart devices and multiple cooperative wireless access points as relays in a given rectangular area, and one remote base station (BS) integrated with an edge server. The set of smart devices and multiple cooperative WAPs in the system are denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$ and $\mathcal{M} = \{1, 2, \ldots, M\}$, respectively. The offloading tasks are transmitted to MEC server with the help of WAP selected from $\mathcal{M}$. All the SDs are independent, randomly distributed in the rectangular area, and equipped with one antenna. The multiple WAPs which receive the offloading input data from SD and transmit the received data to the BS act as relays with multiple antennas. Each SD monopolizes one antenna of the wireless access point, and WAP supports simplex mode that only one operation can be performed in a time slice, i.e., receiving or sending. Therefore, the time-division-multiple-access (TDMA) protocol is adopted to accomplish the offloading operation, which the time block is divided into two time slots, i.e., SD offloads computation task to WAP and WAP transmits the task to edge server. In order to allow multiple accesses, all SDs within the network access multiple WAPs follow the OFDMA principle and ignore the interference caused by other relays [12], [42], [43]. Meanwhile, multiple WAPs transmit the offloading computation bits from each SD to BS using the OFDMA principle in parallel. Each SD has a computation-intensive delay-sensitive task which can be processed locally or partially offloaded to the edge server for execution. We use $\mathcal{A}_i \triangleq (I_i, C_i, T_i), \forall i \in \mathcal{N}$ denote the task, where I_i denotes the size of computation input data, C_i denotes the total number of CPU cycles to accomplish the task, T_i denotes the time constraint for task $\mathcal{A}_i$. We assume that the time constraints of all the tasks are identical and equal to the time block, i.e., T. We also assume that task $\mathcal{A}_i$ is divided into bitwise for partial local computing and partial computation offloading. Moreover, each SD chooses a relay node from $\mathcal{M}$ to offload data for long-distance data transmission. Meanwhile, multiple WAPs are heterogeneous and have different bandwidths, the Additive White Gaussian Noises and so on. When SDs select different WAPs, these hyperparameters will result in the differences in the tasks offloading ratios, which will further lead to the different energy consumption of the proposed MEC system. In this paper, we aim to minimize the energy consumption of the D2D-enabled MEC system subject to the constraints of communication, computation resource and latency. In this context, we give the detail description of the system model.

A. Relay Selection Model

Due to the constraint of limited computation capacity, the SD should offload partial computation task to the nearby edge server for accomplishing the task. The better wireless access point chooses, the less the energy consumption of the system consumes. Therefore, in order to minimize the energy consumption, SD needs to decide one of the WAPs acting as a relay for transmitting the offloading input data. To describe whether the SD $i \in \mathcal{N}$ transmits the offloading data with the help of the WAP $j \in \mathcal{M}$ or not, we introduce a relay selection decision as a decision matrix shown as follows

$$\boldsymbol{a} \in (a_{i,j})_{N*M} \tag{1}$$

where $a_{i,j}$ is a binary variable. Specifically, $a_{i,j} = 1$ if the offloading data of the SD i is transmitted by the WAP j. Otherwise, $a_{i,j} = 0$. We can get the following constraints which define the relay selection decision. (2) and (3) guarantee the uniqueness of the relay node.

$$a_{i,j} \in \{0, 1\}, i \in \mathcal{N}, j \in \mathcal{M} \tag{2}$$

$$\sum_{j=1}^{M} a_{i,j} = 1, i \in \mathcal{N} \tag{3}$$

We assume that SDs adopt a partial computation offloading model. That is to say, tasks can be partitioned into two parts in bitwise, which execute on the SD and MEC server in parallel. Note that $\lambda_{i,j}$ denotes the offloading ratio which SD $i \in \mathcal{N}$ transmits data through WAP $j \in \mathcal{M}$.

B. Computation Model

As for a target time block, we assume that the local computation task for each SD must be completed within the corresponding time block. Let F_i^{loc} denote the local computation capability of i-th SD assigned to task $\mathcal{A}_i$ in one time block. τ_i^{loc} denotes the required time to finish the partial local computation task, and λ_i^{loc} defines the ratio of local computing. Therefore, we can get the expression of F_i^{loc} as follows:

$$F_i^{loc} = \frac{\lambda_i^{loc} C_i}{\tau_i^{loc}}, i \in \mathcal{N} \tag{4}$$

And the energy consumption for one computing cycle can be given by $\kappa_i F_i^{loc^2}$ where κ_i is the effective switched capacitance

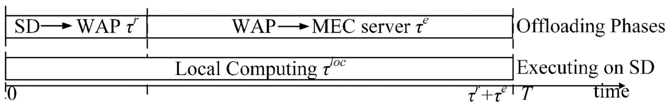

Fig. 2. Computation Model and Communication Model.

that relies on the chip architecture. Thus, the total energy consumption of local computing are as follows:

$$E_i^{loc} = \kappa_i F_i^{loc^2} \lambda_i^{loc} C_i, i \in \mathcal{N} \tag{5}$$

Based on (5), the energy consumption of local execution has a monotonous increasing trend with the increase of F_i^{loc}. F_i^{loc} is too small to accomplish task within T. Therefore, we set the maximum available time block as the deadline, that is $\tau_i^{loc} = T, i \in \mathcal{N}$, where the optimal value of F_i^{loc} can be achieved. Let $\lambda_{i,j}$ denote the offloading ratio from SD i to WAP j. If $a_{i,j} = 0$, $\lambda_{i,j} = 0$. Meanwhile, according to the former mentioned, the local ratio can be described as follows:

$$\lambda_i^{loc} = 1 - \sum_{j=1}^{M} a_{i,j}\lambda_{i,j}, i \in \mathcal{N} \tag{6}$$

Substituting (6) into them, (4), (5) are converted to (7), (8), separately.

$$F_i^{loc} = \frac{(1 - \sum_{j=1}^{M} a_{i,j}\lambda_{i,j})C_i}{\tau_i^{loc}}, i \in \mathcal{N} \tag{7}$$

$$E_i^{loc} = \frac{\kappa_i (1 - \sum_{j=1}^{M} a_{i,j}\lambda_{i,j})^3 C_i^3}{T^2}, i \in \mathcal{N} \tag{8}$$

Besides, let $F_i^{\max}$ denote the maximum CPU frequency of SD i. The local computation capacity constraint is given by:

$$0 \le \frac{(1 - \sum_{j=1}^{M} a_{i,j}\lambda_{i,j})C_i}{T} \le F_i^{\max}, i \in \mathcal{N} \tag{9}$$

C. Communication Model

As shown in Fig. 2, the communication model is divided into two phases, and correspondingly, the time block is partitioned into two time slots. In the first phase, the SD transmits the offloading input data to the chosen WAP in the first time slot. Because the WAP supports the simplex mode, the WAP will not send the offloading data to the remote MEC server until receives all the offloading input bits. The process that the WAP transmits the received data to the MEC server is completed in the second phase, i.e., the second time slot. Meanwhile, according to the wireless access point selection model, the offloading ratio $\lambda_{i,j}$ can be represented as $a_{i,j}\lambda_{i,j}$. Assume that the channel gain, the transmission power and other information are constant in one time block T, but change in different time block. We give the detail describe in the following subsection.

1) From SD to Wireless Access Point: In this phase, the offloading input data is uploaded to the relay wireless access point. For the relay WAP, we define $\tau_{i,j}^r$ as the assigned time to this phase, $p_{i,j}^r$ is the transmission power and $R_{i,j}^r$ is the corresponding achievable rate for SD i. As a result, the achievable rate in this phase can be defined as follows.

$$R_{i,j}^r = \frac{a_{i,j}\lambda_{i,j} I_i}{\tau_{i,j}^r}, i \in \mathcal{N} \tag{10}$$

Besides, according to the Shannon-Hartley Equation, the achievable rate can be denoted as:

$$R_{i,j}^r = B_j^r \log_2(1 + \frac{p_{i,j}^r h_{i,j}^r}{N_j^r}), i \in \mathcal{N} \tag{11}$$

where B_j^r is the bandwidth of the wireless subchannel provided by the j WAP, $h_{i,j}^r$ is the subchannel power gain from SD i to WAP j, N_j^r is the Additive White Gaussian Noise(AWGN) between SD i and WAP j. Hence, combining (10) with (11), the transmission power can be denoted as the following expression

$$p_{i,j}^r = \frac{N_j^r}{h_{i,j}^r}(2^{\frac{a_{i,j}\lambda_{i,j}I_i}{B_j^r \tau_{i,j}^r}} - 1), i \in \mathcal{N} \tag{12}$$

Specially, if $a_{i,j} = 0$, the corresponding offloading ratio, transmission power, the time slot are all equal to 0, i.e., $p_{i,j}^r = 0, \tau_{i,j}^r = 0, \lambda_{i,j} = 0$. Learning from this subsection, the energy consumption for sending the offloading input data from SD i to WAPs can be given by:

$$E_i^r = \sum_{j=1}^{M} p_{i,j}^r \tau_{i,j}^r, i \in \mathcal{N} \tag{13}$$

2) From Wireless Access Point to MEC Server: After the WAP received the offloading input bits, the WAP starts to upload the data to the remote base station integrated with a MEC server. Let B^e denote the bandwidth of the subchannel provided by MEC server. N^e is the AWGN between WAPs and MEC server. And then, for WAP j, we have already known that the offloading bits received through the occupied antenna are $a_{i,j}\lambda_{i,j}I_i$ during the first phase. $\tau_{i,j}^e$ is the second time slot which is used to transmit the offloading data from i-th WAP to MEC server. Thus, we can get the achievable rate $R_{i,j}^e$ which can be denoted as follows:

$$R_{i,j}^e = \frac{a_{i,j}\lambda_{i,j} I_i}{\tau_{i,j}^e}, i \in \mathcal{N} \tag{14}$$

Meanwhile, based on the Shannon Theorem, $R_{i,j}^e$ can also be presented as the following expression.

$$R_{i,j}^e = B^e \log_2(1 + \frac{p_{i,j}^e h_j^e}{N^e}), i \in \mathcal{N} \tag{15}$$

where $p_{i,j}^e$ denotes the transmission power of the occupied antenna, and h_j^e is the channel power gain from j-th WAP to BS. By integrating (14) and (15) together, the expression of $p_{i,j}^e$ can be inferred.

$$p_{i,j}^e = \frac{N^e}{h_j^e}(2^{\frac{a_{i,j}\lambda_{i,j}I_i}{B^e \tau_{i,j}^e}} - 1), i \in \mathcal{N} \tag{16}$$

Specially, if $a_{i,j} = 0$, $p_{i,j}^e$ and $\tau_{i,j}^e$ are 0. Therefore, the energy consumption of the second phase for SD i can be presented as follows.

$$E_i^e = \sum_{j=1}^{M} p_{i,j}^e \tau_{i,j}^e, i \in \mathcal{N} \tag{17}$$

Besides, we suppose that the computation capacity of the edge server is finite. Let $F_e^{\max}$ indicate the edge computation capacity measured by CPU cycles per time slice. Hence, the total number of CPU cycles accomplished by the edge server is limited by the following constraint.

$$\sum_{i=1}^{N}\sum_{j=1}^{M} a_{i,j}\lambda_{i,j}C_i \leq F_e^{\max} \tag{18}$$

The constraint (18) demonstrates that we neglect the edge computing latency. Then, we consider the delay caused by the offloading data transmission for the j WAP in $\mathcal{M}$. We give the delay constraints as follows:

$$0 \leq \tau_{i,j}^r + \tau_{i,j}^e \leq T, i \in \mathcal{N}, j \in \mathcal{M} \tag{19}$$

Ultimately, the offloading ratios should also satisfy the following constraint for each SD.

$$0 \leq \sum_{j=1}^{M} a_{i,j}\lambda_{i,j} \leq 1, i \in \mathcal{N} \tag{20}$$

Although we are not sure which WAP acts as the relay for sending the offloading bits from SD i, the the offloading ratios satisfy constraint (20) after we get the optimal relay selection policy, i.e., $a_{i,j} = 0$ or $a_{i,j} = 1$.

D. Problem Formulation

Under the previous discussion, our objective is to minimize the total energy consumption of the D2D-enabled MEC system by jointly optimizing the communication, computation resource and the relay selection policy. We consider the computation, communication resource and delay constraints, and formulate the problem as follows:

$$\begin{aligned} P_1 : \min_{\boldsymbol{a},\boldsymbol{\lambda},\boldsymbol{\tau}^r,\boldsymbol{\tau}^e} \quad & \sum_{i=1}^{N} \left(E_i^{loc} + E_i^r + E_i^e\right) \\ \text{s.t. } & (2), (3), (9), (18), (19), (20), \end{aligned} \tag{21}$$

where $\boldsymbol{a}, \boldsymbol{\lambda}, \boldsymbol{\tau}^r, \boldsymbol{\tau}^e$ denote the solution of P_1. (2) and (3) represent the relay selection constraint. (9) denotes the finite CPU frequency of SD. (18) indicates that the total offloading CPU cycles should less than or equal to the maximum computation capacity of the edge server. (19) ensures the offloading operation be completed during a time block. (20) shows the offloading ratio constraint under different relay selection policy. Now, the problem is formulated as a mixed integer nonlinear optimization problem. Obviously, problem P_1 is a non-convex problem because of discrete variables and second-order terms of the form $x \cdot y$. Simultaneously, because problem P_1 contains two types of variables, namely discrete variable (relay selection policy) and continuous variables (the offloading ratio and the two time slots), P_1 is an integer-mixed non-convex problem, which is NP-hard problem [24], [33], [47]. Therefore, fixing the problem P_1 is a challenge.

III. Problem Solution

In this section, we propose an optimal algorithm, jointly optimizing relay selection and resource allocation, named as JOSR to solve P_1.

A. Reformulation and Linearization P_1

$a_{i,j}$ is a 0-1 integer variable, and in order to solve the original problem P_1, an intuitive method is to enumerate all M^N possible relay selection policies and find out the optimal relay selection policy and resource allocation strategy with the minimum energy consumption. However, the time complexity of the exhaustive method is $\mathcal{O}(M^N)$, which is acceptable while M and N are smaller, but cannot work out as M and N increasing. This straightforward method can be used as a benchmark to evaluate the performance of actually proposed low complexity algorithm. In order to facilitate subsequent optimization, we first relax the discrete variable $a_{i,j}$ by $0 \leq a_{i,j} \leq 1$. That is to say, constraints (2) and (3) are converted into the following relaxed constraint (22).

$$0 \leq \sum_{j=1}^{M} a_{i,j} \leq 1, i \in \mathcal{N}, j \in \mathcal{M} \tag{22}$$

Learning from [48], we can adopt Reformulation-Linearization Technique (RLT) to eliminate all quadratic terms. RLT can linearize the objective function and constraints (9), (18), (20), in Problem P_1. For the second order term $a_{i,j}\lambda_{i,j}$,let the intermediate variable $l_{i,j} = a_{i,j}\lambda_{i,j}$ where $a_{i,j}$ and $\lambda_{i,j}$ are bounded as $0 \leq a_{i,j} \leq 1$ and $0 \leq \lambda_{i,j} \leq 1$, respectively. And then, the RLT bound-factor product constraints for $l_{i,j}$ can be described as follows:

$$\begin{cases} \{[a_{i,j} - 0] \cdot [\lambda_{i,j} - 0]\}_{LS} \\ \{[a_{i,j} - 0] \cdot [1 - \lambda_{i,j}]\}_{LS} \\ \{[1 - a_{i,j}] \cdot [\lambda_{i,j} - 0]\}_{LS} \\ \{[1 - a_{i,j}] \cdot [1 - \lambda_{i,j}]\}_{LS} \end{cases}, i \in \mathcal{N}, j \in \mathcal{M} \tag{23}$$

$\{\}_{LS}$ represents a linearization operation under $l_{i,j} = a_{i,j}\lambda_{i,j}$. Substituting into (23), we obtain

$$\begin{cases} l_{i,j} \geq 0 \\ l_{i,j} - a_{i,j} \leq 0 \\ l_{i,j} - \lambda_{i,j} \leq 0 \\ a_{i,j} + \lambda_{i,j} - l_{i,j} - 1 \leq 0 \end{cases}, i \in \mathcal{N}, j \in \mathcal{M} \tag{24}$$

After we substitute $l_{i,j}$ into the objective function and constraints (9), (18), (20) in P_1, P_1 is converted to the problem P_2 as:

$$\begin{aligned} P_2 : \min_{\boldsymbol{a},\boldsymbol{l},\boldsymbol{\lambda},\boldsymbol{\tau}^r,\boldsymbol{\tau}^e} \quad & \sum_{i=1}^{N} h\left(l_{i,j}, \tau_{i,j}^r, \tau_{i,j}^e\right) \\ \text{s.t. } (25a) : 0 \leq & \frac{(1-\sum_{j=1}^{M} l_{i,j})I_i D_i}{T} \leq F_i^{\max}, i \in \mathcal{N} \\ (25b) : & \sum_{j=1}^{M} l_{i,j} \leq 1, i \in \mathcal{N} \\ (25c) : & \sum_{i=1}^{N}\sum_{j=1}^{M} l_{i,j}C_i \leq F_e^{\max} \\ & (19), (22), (24), \end{aligned} \tag{25}$$

where

$$h\left(l_{i,j},\tau_{i,j}^{r},\tau_{i,j}^{e}\right)=\frac{\kappa_i(1-\sum_{j=1}^{M}l_{i,j})^3C_i^3}{T^2}+\sum_{j=1}^{M}\frac{N_j^r}{h_{i,j}^r}(2^{\frac{l_{i,j}I_i}{B_j^r\tau_{i,j}^r}}-1)\tau_{i,j}^r+\sum_{j=1}^{M}\frac{N_e}{h_j^e}(2^{\frac{l_{i,j}I_i}{B^e\tau_{i,j}^e}}-1)\tau_{i,j}^e \tag{26}$$

As $f(y)\triangleq\frac{h_x}{N_x}(2^y-1)$ is a monotonically increasing and convex function with respect to $y>0$, its perspective function $f(\frac{l_x}{\tau_x})\tau_x$ is a convex function when $\tau_x\leq 0$ and $l_x\leq 0$. Note that we define $\tau_x=0$ and $f(\frac{l_x}{\tau_x})\tau_x=0$ when $l_x=0$ holds. Then, the first item of the objective function is convex, and we verify the conclusion by using Python tool package for convex optimization CVXPY [49]. Since the objective function is the sum of affine and convex function, and constraints are all convex, problem P_2 is convex [50].

Therefore, we obtain an equivalence problem P_2 by relaxing and RLT operations. Now, by analyzing the objective function, the objective can be seen as the total energy consumption cost by accomplishing the local computation cycles $l_i^{loc}C_i$, completing the transmission of offloading data $l_{i,j}I_i$ through j-th WAP, respectively. The computation input bits of $\mathcal{A}_i$ are partitioned into $M+1$ parts in bitwise, i.e., $l_i^{loc}, l_{i,j}, j\in\mathcal{M}$, and $l_i^{loc}=1-\sum_{j=1}^{M}l_{i,j}$. Besides, the objective of problem P_2 is to minimize the total energy consumption of the D2D-enabled MEC system spent on local computing and computation offloading of all SDs. However, the difference between P_2 and P_1 is that some SDs may perform computation offloading through M relays. Moreover, more constraints (24) are used to result the optimal solution.

In order to solve problem P_2, we derive its optimal solution by leveraging the Lagrange method and give the detailed mathematical derivation processes. We introduce the following Lagrange multipliers $\alpha_i\geq 0$, $\beta_i\geq 0$, $\gamma_{i,j}\geq 0$, $\mu\geq 0$, $\delta_{i,j}\geq 0$, $\omega_{i,j}\geq 0$, $\nu_{i,j}\geq 0$ and $\eta_{i,j}\geq 0$ associated with the corresponding constraints. And then the partial Lagrange function of problem P_2 is presented as follows.

$$\begin{aligned}L=&\sum_{i=1}^{N}\left(\frac{\kappa_i(1-\sum\limits_{j=1}^{M}l_{i,j})^3C_i^3}{T^2}+\sum_{j=1}^{M}\frac{N_j^r}{h_{i,j}^r}(2^{\frac{l_{i,j}I_i}{B_j^r\tau_{i,j}^r}}-1)\tau_{i,j}^r\right)\\&+\sum_{i=1}^{N}\left(\sum_{j=1}^{M}\frac{N_e}{h_j^e}(2^{\frac{l_{i,j}I_i}{B^e\tau_{i,j}^e}}-1)\tau_{i,j}^e+\beta_i(\sum_{j=1}^{M}l_{i,j}-1)\right)\\&+\sum_{i=1}^{N}\left(\alpha_i(\frac{(1-\sum\limits_{j=1}^{M}l_{i,j})C_i}{T}-F_i^{\max})+\delta_i(\sum_{j=1}^{M}a_{i,j}-1)\right)\\&+\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\gamma_{i,j}(\tau_{i,j}^r+\tau_{i,j}^e-T)+\omega_{i,j}(l_{i,j}-\lambda_{i,j})\right)\\&+\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\nu_{i,j}(l_{i,j}-a_{i,j})+\eta_{i,j}(a_{i,j}+\lambda_{i,j}-l_{i,j}-1)\right)\\&+\mu\left(\sum_{i=1}^{N}\sum_{j=1}^{M}l_{i,j}C_i-F_e^{\max}\right)\end{aligned} \tag{27}$$

Let $\boldsymbol{a}^*$, $\boldsymbol{\lambda}^*$, $\boldsymbol{l}^*$, $\boldsymbol{\tau}^{r*}$ and $\boldsymbol{\tau}^{e*}$ denote the optimal solution for P_2 and $\boldsymbol{\alpha}^*$, $\boldsymbol{\beta}^*$, $\boldsymbol{\gamma}^*$, μ^*, $\boldsymbol{\delta}^*$, $\boldsymbol{\omega}^*$, $\boldsymbol{\nu}^*$ and $\boldsymbol{\eta}^*$ denote the optimal Lagrange multipliers. According to the Karush-Kuhn-Tucker (KKT) dual complementarity conditions $\beta_i(\sum_{j=1}^{M}l_{i,j}-1)=0$, we obtain $\beta_i=0$ because of the total offloading ratio constraint $\sum_{j=1}^{M}l_{i,j}<1$. Meanwhile, learning from the stationary equations $\frac{\partial L}{\partial\lambda_{i,j}^*}=0$, we can get $\boldsymbol{\omega}^*=\boldsymbol{\eta}^*$. Based on the stationary equation $\frac{\partial L}{\partial\tau_{i,j}^{r*}}=0$, we can get the following lemma.

Lemma 1: For any given $\boldsymbol{\gamma}\geq 0$, the optimal solutions $(l_{i,j}^*,\tau_{i,j}^{r*})$ to P_2 can be shown as follows: If $\gamma_{i,j}=0$, we obtain $l_{i,j}^*=0$, $\tau_{i,j}^{r*}=0$ and $a_{i,j}^*=0$; If $\gamma_{i,j}>0$, we obtain

$$r_{i,j}^r=1+W_0\left[\left(\frac{\gamma_{i,j}^*h_{i,j}^r}{N_j^r}-1\right)\Big/e\right] \tag{28}$$

where $r_{i,j}^r=\frac{I_i\ln 2l_{i,j}^*}{B_j^r\tau_{i,j}^{r*}}$ denotes the ratio between the offloading data and its corresponding time slot, W_0 is the principal branch of the Lambert W function defined as the solution for $W_0(x)e^{W_0(x)}=x$.

Proof: Its process is similar to Appendix A of [51]. ■

Similarly, according to the stationary equation $\frac{\partial L}{\partial\tau_{i,j}^{e*}}=0$ and Lemma 1, the optimal ratio between $l_{i,j}^*$ and $\tau_{i,j}^{e*}$ can be obtained with a given $\gamma_{i,j}>0$. The results are described as the following equation.

$$r_{i,j}^e=1+W_0[(\frac{\gamma_{i,j}^*h_j^e}{N^e}-1)/e] \tag{29}$$

Learning from (26), the objective function is monotonic decreasing function with respect to $\boldsymbol{\tau^r},\boldsymbol{\tau^e}\geq 0$. The lager $\boldsymbol{\tau^r},\boldsymbol{\tau^e}$ are, the less the objective value is. And then, we can get Lemma 2 through analyzing the KKT dual complementarity conditions $\boldsymbol{\gamma}(\boldsymbol{\tau^r}+\boldsymbol{\tau^e}-T)=0$.

Lemma 2: For any given $\boldsymbol{\gamma}\geq 0$, the optimal solution $l_{i,j}^*$ to P_2 can be expressed as: If $\gamma_{i,j}=0$, then $l_{i,j}^*=0$, $\tau_{i,j}^{r*}=0$, $a_{i,j}^*=0$; If $\gamma_{i,j}>0$, then we have (30), shown at the bottom of the next page.

Proof: See [41]. ■

Through analyzing (30), $l_{i,j}^*$ is monotonic increasing concave functions of $\gamma_{i,j}>0$. Meanwhile, when $\gamma_{i,j}\to+\infty$, we have $l_{i,j}^*\to+\infty$.

B. Optimization of Relay Selection

Aimed to get the optimal relay selection policy, we assume that the computation capacity of MEC server and SD is infinite, i.e., $\sum_{i=1}^{N}\sum_{j=1}^{M}l_{i,j}C_i\ll F_e^{\max}$, $\frac{(1-\sum_{j=1}^{M}l_{i,j})I_iD_i}{T}<F_i^{\max}$. And $\mu=0$ and $\alpha_i=0$ are obtained by their KKT dual complementarity conditions. According to the value range of $0\leq a_{i,j}\leq 1$, we study the optimal relay selection policy while $0<a_{i,j}<1$. Therefore, $l_{i,j}<a_{i,j}$. Moreover, we obtain $\nu_{i,j}=0$ through the KKT dual complementarity condition $\sum_{i=1}^{N}\sum_{j=1}^{M}\nu_{i,j}(l_{i,j}-a_{i,j})=0$. Meanwhile, plugging (28), $\beta_i=0$ and $\omega_{i,j}=\eta_{i,j}$ into $\frac{\partial L}{\partial l_{i,j}^*}=0$, we can get the local computation ratio l_i^{loc} described as (31), shown at the bottom of the next page.

Based on the two stationary equations $\frac{\partial L}{\partial l^*_{i,j}} = 0$, $\frac{\partial L}{\partial l^*_{i,k}} = 0, k \in \mathcal{M}, k \neq j$, $\omega_{i,j} = \eta_{i,j}, \nu_{i,j} = 0$, we achieve the following result from the difference between the both stationary equations.

$$\begin{aligned}&\frac{N_j^r I_i \ln 2}{h_{i,j}^r B_j^r} e^{r_{i,j}^r} + \frac{N^e I_i \ln 2}{h_j^e B^e} e^{r_{i,j}^e} = \\ &\frac{N_k^r I_i \ln 2}{h_{i,k}^r B_k^r} e^{r_{i,k}^r} + \frac{N^e I_i \ln 2}{h_k^e B^e} e^{r_{i,k}^e}, k \in \mathcal{M}, k \neq j\end{aligned} \tag{32}$$

We assume that the computation capacity of MEC server and SD is sufficient, as mentioned above, then $\mu = 0$ and $\alpha_i = 0$ are satisfied. (31) can be simplified to $l_i^{loc(0)}$ where $\mu^* = 0$ and $\alpha_i^* = 0$. We can obtain the intermediate variables $l_{i,j}, l_i^{loc}$ through solving (33a) combining with the well-studied algorithms, i.e., Hybr, Broyden1, Linearmixing, which were implemented or inherited in SciPy, an open-source software for mathematics, science, and engineering, in the Complex Domain. The solutions are defined as $\gamma_{i,j}^{(0)}, j \in \mathcal{M}$.

$$\begin{cases}(32)\\ \sum_{j=1}^M l^*_{i,j} + l_i^{loc(0)} = 1\end{cases} \text{(33a)} \quad \begin{cases}(32)\\ l_i^{loc(0)} = l_i^{loc(\max)}\\ \sum_{j=1}^M l^*_{i,j} + l_i^{loc(0)} = 1\end{cases} \text{(33b)} \tag{33}$$

Because $\gamma_{i,j}^*$ is the Lagrange multiplier, $\gamma_{i,j} \geq 0$. To simply the problem solving process, we introduce three positive constants $b_{\min}$, b_x, $b_{\max}$ and a monotonically increasing function. The monotonically increasing function is denoted as (34).

$$\psi_{i,j}(\gamma_{i,j}) = \frac{N_j^r I_i \ln 2}{h_{i,j}^r B_j^r} e^{r_{i,j}^r} + \frac{N^e I_i \ln 2}{h_j^e B^e} e^{r_{i,j}^e} \tag{34}$$

The three constants $b_{\min}, b_x, b_{\max}$ are denoted as follows:

$$\begin{cases} b_{\min} = \min\left(\psi_{i,j}(0)\right), j \in \mathcal{M}\\ b_{\max} = \max\left(\psi_{i,j}(0)\right), j \in \mathcal{M}\\ b_x = \psi_{i,j}(\gamma_{i,j}^{(0)})\end{cases} \tag{35}$$

In order to achieve the roots of (33a), we integrate the binary section algorithm and the root finding algorithm in Scipy. Firstly, we adjust the value of $b_{\max}$ with 0.1 as the step length until $\sum_{j=1}^M l^*_{i,j} + l_i^{loc(0)} > 1$. Secondly, with a given value b_x between $b_{\min}$ and $b_{\max}$, we use the binary section algorithm solve (32), and obtain the corresponding the value of $\gamma_{i,j}, j \in \mathcal{M}$. Thirdly, we achieve the optimal values $\gamma_{ij}^{(0)}$ which satisfy $\sum_{j=1}^M l^*_{i,j} + l_i^{loc(0)} = 1$ through continuously adjusting the value of b_x. We can get the values of $l_i^{loc}, l_{i,j}$ by substituting $\gamma_{ij}^{(0)}$ into (31), (30), denoted as $l_i^{loc(0)}, l_{i,j}^{(0)}$. According to the constraint of the finite CPU frequency of SD, we can get the constraint $l_i^{loc} \leq l_i^{loc(\max)} = \frac{F_i^{\max} T}{C_i}$. If $l_i^{loc(0)} > l_i^{loc(\max)}$, the intermediate variable l_i^{loc} equals $l_i^{loc(\max)}$ and $\alpha_i^* > 0$. The expression of l_i^{loc} can be overwritten as $l_i^{loc(1)}$ where $\mu = 0$. $\gamma_{ij}^{(1)}$, $\alpha_i^{(1)}$ are obtained by solving (33b). Finally, we obtain the intermediate variables $l_{i,j}, l_i^{loc}$. Let $l_i^{\max}$ denote $\max(l_{i,j}, j \in \mathcal{M})$. Meanwhile, the offloading ratios must satisfy (25c). If (25c) is true, we can obtain the optimal relay selection policy a^* based on the following constraint.

Algorithm 1: Optimal Relay Selection Policy Algorithm for Solving P_2 with $\mu = 0$.

for $i = 1; i \leq N; i = i + 1$ **do**
  $b_{min} = min\left(\psi_j(0)\right)$;$b_{max} = max\left(\psi_j(0)\right)$;
  Adjust b_{max} until $\sum_{j=1}^{M} l_{i,j} + l_i^{loc} > 1$;
  while $\sum_{j=1}^{M} l_{i,j} + l_i^{loc} \neq 1$ **do**
    $\gamma_i = bisect(\psi_{i,j}(\gamma_{i,j}) - b_x, b_{min}, b_{max}), j \in \mathcal{M}$;
    compute l_i^{loc} by using $l_i^{loc(0)}$ with (31);
    if $(l_i^{loc} > l_i^{loc(max)})$ **then**
      $l_i^{loc} = l_i^{loc(max)}$;
      compute α_i by using $l_i^{loc(1)}$ and γ_i ;
    compute $l_{i,j}$ by using (30) and γ_i;
    Adjust b_x with $\sum_{j=1}^{M} l_{i,j} + l_i^{loc} - 1$;
obtain $l_i^{loc}, l_{i,j}$;calculate $a_{i,j}$ based on (36);
return $\boldsymbol{a}$;

$$\boldsymbol{a} = \left[a_{i,j} | a_{i,j} = \begin{cases}0, l_{ij} \neq l_i^{\max}, j \in \mathcal{M}\\ 1, l_{ij} = l_i^{\max}\end{cases}\right] \tag{36}$$

According to (36), the optimal relay for SD i is the wireless access point with the maximum value of $l_{i,j}, j \in \mathcal{M}$. The processes can be shown as Algorithm 1.

When the computation capacity of MEC server cannot satisfy the total need of all SDs, we can get the following conclusions through analyzing the dual complementarity condition of μ.

$$\sum_{i=1}^{N} \sum_{j=1}^{M} l_{i,j} C_i = F_e^{\max} \tag{37}$$

We assume that the CPU frequency of SD is infinite $\frac{(1 - \sum_{j=1}^M l_{i,j}) I_i C_i}{T} < F_i^{\max}$ and $\alpha_i = 0$. l_i^{loc} can be rewritten as

$$l_{i,j}^* = \frac{T B_j^r B^e (1 + W_0[(\frac{\gamma_{i,j}^* h_{i,j}^r}{N_j^r} - 1)/e])(1 + W_0[(\frac{\gamma_{i,j}^* h_j^e}{N^e} - 1)/e])}{I_i \ln 2 (B_j^r (1 + W_0[(\frac{\gamma_{i,j}^* h_{i,j}^r}{N_j^r} - 1)/e]) + B^e (1 + W_0[(\frac{\gamma_{i,j}^* h_j^e}{N^e} - 1)/e]))} \tag{30}$$

$$l_i^{loc*} = \sqrt{\frac{T^2}{3\kappa_i C_i^3}\left(\frac{N_j^r I_i \ln 2}{h_{i,j}^r B_j^r} e^{1 + W_0[(\frac{\gamma_{i,j}^* h_{i,j}^r}{N_j^r} - 1)/e]} + \frac{N^e I_i \ln 2}{h_j^e B^e} e^{1 + W_0[(\frac{\gamma_{i,j}^* h_j^e}{N^e} - 1)/e]} - \alpha_i^* \frac{C_i}{T} + \mu^* C_i\right)} \tag{31}$$

$l_i^{loc(2)}$ where $\alpha_i^* = 0$.

$$\begin{cases}(32)\\ \sum_{j=1}^{M} l_{i,j}^* + l_i^{loc(2)} = 1\end{cases} \text{(38a)} \begin{cases}(32)\\ l_i^{loc*} = l_i^{loc(\max)}\\ \sum_{j=1}^{M} l_{i,j}^* + l_i^{loc*} = 1\end{cases} \text{(38b)} \tag{38}$$

With a given μ_x, the Lagrange Multipliers $\gamma_{i,j}$ can be obtained by solving (38a), defined as $\gamma_{i,j}^{(2)}$. If $l_i^{loc(2)} > l_i^{loc(\max)}$, $l_i^{loc(2)}$ equals $l_i^{loc(\max)}$. The Lagrange Multipliers $\gamma_{i,j}^{(3)}$, $\alpha_i^{(3)}$ are the root of (38b). And then, adopting the root finding methods from the SciPy library, the Lagrange Multiplier μ_x is continuously adjusted until its value satisfies (37) where $l_{i,j}, l_i^{loc}$ can be computed with a given μ_x. According to (36), the optimal relay selection policy $\boldsymbol{a}^*$ can be obtained. The processes are shown as Algorithm 2.

Hence, the proposed method for calculating $\boldsymbol{a}^*$ can be described as Algorithm 1 and Algorithm 2 through summarizing the solving process. Meanwhile, by the analysis of Algorithm 1 and Algorithm 2, the maximum computation complexity of the two algorithms is associated with (33), (37) and (38). We use the iterative algorithms(Newton and Gradient Methods) to solve (33), (37) and (38). Therefore, we set the maximum iterations of (33a) and (33b) as $\mathcal{C}_1^1, \mathcal{C}_2^1$ respectively. Then, the maximum computational complexity of the solution process for Algorithm 1 is $\mathcal{O}(N(\mathcal{C}_1^1 + \mathcal{C}_2^1))$. Hence, we denote the maximum iterations of (38a) and (38b) as $\mathcal{C}_3^1$, $\mathcal{C}_4^1$, separately. Meanwhile, in order to obtain the optimal μ, we define the maximum iterations of the method of finding root used as $\mathcal{C}_0^1$. Then, the maximum computation complexity of Algorithm 2 is $\mathcal{O}(N\mathcal{C}_0^1(\mathcal{C}_3^1 + \mathcal{C}_4^1))$. In summary, the maximum computation complexity of the relay selection algorithm is $\mathcal{O}(N\mathcal{C}_0^1(\mathcal{C}_3^1 + \mathcal{C}_4^1))$.

Algorithm 2: Optimal Relay Selection Policy Algorithm for Solving P_2 with $\mu > 0$.

$\mu_x = 1e-9$; $res = F_e^{max}$;
repeat
  for $i = 1; i \le N; i = i+1$ **do**
    $b_{min} = min(\psi_j(0))$; $b_{max} = max(\psi_j(0))$;
    Adjust b_{max} until $\sum_{j=1}^{M} l_{i,j} + l_i^{loc} > 1$;
    while $\sum_{j=1}^{M} l_{i,j} + l_i^{loc} \neq 1$ **do**
      $\gamma_i = bisect(\psi_{i,j}(\gamma_{i,j}) - b_x, b_{min}, b_{max}), j \in \mathcal{M}$;
      compute l_i^{loc} by using $l_i^{loc(2)}$ with(31);
      if $(l_i^{loc} > l_i^{loc(max)})$ **then**
        $l_i^{loc} = l_i^{loc(max)}$;
        compute α_i by using (31) and γ_i;
      compute $l_{i,j}$ by using (30) and γ_i;
      Adjust b_x with $\sum_{j=1}^{M} l_{i,j} + l_i^{loc} - 1$;
  Adjust μ_x with $res = \sum_{i=1}^{N}\sum_{j=1}^{M} l_{i,j}C_i - F_e^{max}$;
until $(res == 0)$;
obtain $l_i^{loc}, l_{i,j}$;calculate $a_{i,j}$ based on (36);
return $\boldsymbol{a}$;

C. Optimization of Resource Allocation

We introduce the following parameters for reformulating P_1 to obtain the optimal resource allocation strategy. According to the decision matrix $\boldsymbol{a}$, $\mathcal{N}_j, j \in \mathcal{M}$ is the set of SDs that selects j-th WAP. $\mathcal{N}_j \cap \mathcal{N}_k = \varnothing, j,k \in \mathcal{M}, j \neq k$ and $\mathcal{N}_1 \cup \mathcal{N}_2 \cup \ldots \cup \mathcal{N}_M = \mathcal{N}$. $\lambda_i, i \in \mathcal{N}_j$ represents the offloading ratio through j-th WAP. $\tau_i^r, i \in \mathcal{N}_j$ defines the time that i-th SD spends on transmitting the offloading bits to j-th WAP, and $\tau_i^e, i \in \mathcal{N}_j$ denotes the time used to upload the offloading data from j-th WAP to MEC server. $h_j(\lambda_i, \tau_i^r, \tau_i^e)$ defines the total energy consumption of SD i. After we calculate the optimal relay selection policy $\boldsymbol{a}^*$, P_1 can be reformulated as P_3.

$$\begin{aligned} P_3: &\min_{\boldsymbol{\lambda}, \boldsymbol{\tau}^r, \boldsymbol{\tau}^e} \sum_{j=1}^{M}\sum_{i\in\mathcal{N}_j} h_j(\lambda_i, \tau_i^r, \tau_i^e)\\ \text{s.t. } &(39a): 0 \le \tfrac{(1-\lambda_i)C_i}{T} \le F_i^{\max}, i \in \mathcal{N}_j\\ &(39b): \sum_{j=1}^{M}\sum_{i\in\mathcal{N}_j} \lambda_i C_i \le F_e^{\max}\\ &(39c): \tau_i^r + \tau_i^e \le T, i \in \mathcal{N}_j\\ &(39d): 0 \le \lambda_i \le 1, i \in \mathcal{N}_j \end{aligned} \tag{39}$$

where

$$\begin{aligned} h_j(\lambda_i, \tau_i^r, \tau_i^e) = &\tfrac{\kappa_i(1-\lambda_i)^3 C_i^3}{T^2} + \tfrac{N_j^r}{h_{i,j}^r}(2^{\frac{\lambda_i I_i}{B_j^r \tau_i^r}} - 1)\tau_i^r\\ &+ \tfrac{N^e}{h_j^e}(2^{\frac{\lambda_i I_i}{B^e \tau_i^e}} - 1)\tau_i^e \end{aligned} \tag{40}$$

(39a) denotes the constraint of the finite CPU frequency for each SD. (39b) means that the total offloading CPU cycles cannot more than the maximum computation capacity of the MEC server. (39c) defines the delay constraint. We know that the objective function of P_3 is a convex function and the whole constraints are linear function. P_3 is a convex optimization problem which is similar to P_2. Therefore, P_3 can be solved by using the Lagrange Method. And then the partial Lagrange function of problem P_3 can be expressed as follows.

$$\begin{aligned} \Gamma = &\sum_{j=1}^{M}\sum_{i\in\mathcal{N}_j} h_j(\lambda_i, \tau_i^r, \tau_i^e) + \alpha_i(\frac{(1-\lambda_i)C_i}{T} - F_i^{\max})\\ &+ \sum_{j=1}^{M}\sum_{i\in\mathcal{N}_j} \beta_i(\tau_i^r + \tau_i^e - T) + \gamma_i(\lambda_i - 1) + \omega\lambda_i C_i\\ &- \omega F_e^{\max} \end{aligned} \tag{41}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \omega$ define the optimal Lagrange multipliers. According to Lemma 1, we can obtain the relationships of $(\lambda_i^*, \tau_i^{r*})$ and $(\lambda_i^*, \tau_i^{e*})$, which are associated with β_i^*. Similarly, based on Lemma 2, we can get the expressions of $(\lambda_i^*$, which are similar to (30). Meanwhile, according to the stationary equations $\frac{\partial\Gamma}{\partial\lambda_i^*} = 0$,

Algorithm 3: Optimal Resource Allocation Algorithm for Solving Case I.

for $j = 1; j \leq M; j = j + 1$ **do**
  for $i \in \mathcal{N}_j$ **do**
    $\beta_i \leftarrow solve(42a)$;
    compute $\lambda_i, \lambda_i^{loc}, \alpha_i = 0, \omega = 0$;
    if $(\lambda_i^{loc*} > \lambda_i^{loc(max)})$ **then**
      $\lambda_i^{loc} = \lambda_i^{loc(max)}$;
      $\alpha_i, \beta_i \leftarrow solve(42b)$;
      compute λ_i;
return $\boldsymbol{\lambda}$, $\boldsymbol{\tau}^r$, $\boldsymbol{\tau}^e$;

we can get the local computation ratio for the SD in the set $\mathcal{N}_j, j \in \mathcal{M}$. $\gamma = 0$ because $\lambda_i^* < 1$. We use λ_i^{loc*} denote the local ratio. For Problem P_3, with the number of SDs increasing, the computation capacity of MEC server is not able to meet the requirement of the offloading tasks. Thus, two cases should be considered:

Case I: The computation capacity of MEC server is enough to accommodate all SDs. We obtain $\sum_{j=1}^{M} \sum_{i \in \mathcal{N}_j} \lambda_i C_i \ll F_e^{\max}$, $\omega = 0$.

Case II: The computation capacity of MEC server is not enough to accommodate all SDs. We get $\sum_{j=1}^{M} \sum_{i \in \mathcal{N}_j} \lambda_i C_i = F_e^{\max}, \omega \geq 0$.

Thus, we will demonstrate how to deal with Case I and Case II above, which are shown in Algorithm 3 and Algorithm 4, respectively.

1) Case I: If the computation capacity of MEC server is large enough, MEC server will accept all the offloading tasks from SDs. Thus, the optimal resource allocation strategy can be obtained by dealing with the following equations.

$$\begin{cases} \lambda_i^* + \lambda_i^{loc*} = 1 \\ \alpha_i^* = 0 \\ \omega^* = 0 \end{cases} \text{(42a)} \quad \begin{cases} \lambda_i^* + \lambda_i^{loc*} = 1 \\ \lambda_i^{loc*} = \lambda_i^{loc(\max)} \\ \omega^* = 0 \end{cases} \text{(42b)} \tag{42}$$

We can get the Lagrange Multipliers β_i^* or (α_i^*, β_i^*) by solving (42a), (42b), respectively. If $\lambda_i^{loc*} > \lambda_i^{loc(\max)} = \frac{F_i^{\max} T}{C_i}$, we compute the optimal solution by solving (42b).

Those equation set can be solved by using the well-studied methods (Newton, Conjugate Direction and Gradient Methods) in SciPy library. Hence, we can summarize the process to solve Case I in Algorithm 3.

2) Case II: We assume that the CPU frequency of MD is enough for local computing. With a given ω, we can get the corresponding Lagrange Multipliers α_i^*, β_i^*. We get a ω^* which satisfies the following equality equation after several iterations.

$$\sum_{j=1}^{M} \sum_{i \in \mathcal{N}_j} \lambda_i C_i = F_e^{\max} \tag{43}$$

Algorithm 4: Optimal Resource Allocation Algorithm for Solving Case II.

$\omega_k = 1e-9; res = F_e^{max}$;
repeat
  for $j = 1; j \leq M; j = j + 1$ **do**
    for $i \in \mathcal{N}_j$ **do**
      $\beta_i \leftarrow solve(44a)$;
      compute $\lambda_i, \lambda_i^{loc}, \alpha_i = 0, \omega = \omega_k$;
      if $(\lambda_i^{loc*} > \lambda_i^{loc(max)})$ **then**
        $\lambda_i^{loc} = \lambda_i^{loc(max)}$;
        $\alpha_i, \beta_i \leftarrow solve(44b)$;
        compute λ_i;
  $res = \sum_{j=1}^{M} \sum_{i \in \mathcal{N}_j} \lambda_i C_i - F_e^{max}$;
  Adjust ω_k with res by using the method in SciPy;
until $(res == 0)$;
return $\boldsymbol{\lambda}$, $\boldsymbol{\tau}^r$, $\boldsymbol{\tau}^e$;

Meanwhile, we assume that the Lagrange Multiplier ω^* is defined as ω_k for k-th iteration.

$$\begin{cases} \lambda_i^* + \lambda_i^{loc*} = 1 \\ \alpha_i^* = 0 \\ \omega^* = \omega_k \end{cases} \text{(44a)} \quad \begin{cases} \lambda_i^* + \lambda_i^{loc*} = 1 \\ \lambda_i^{loc*} = \lambda_i^{loc(\max)} \\ \omega^* = \omega_k \end{cases} \text{(44b)} \tag{44}$$

We can get the Lagrange Multipliers by solving (44a), (44b). And (44b) is used to calculate the solution as $\lambda_i^{loc} = \lambda_i^{loc(\max)}$. Repeatedly, the solution can be obtained through solving (44a), (44b).

As for ω_k, we use the existing methods in SciPy library adjust the value of ω_k for each iteration. Therefore, the process of solving Case II can be summarized as Algorithm 4.

By analyzing Algorithm 3, (42) can be found the solutions in polynomial time by the way of well-studied root finding methods. And then we assume that the maximum numbers of iteration for solving (42a), (42b) are $\mathcal{C}_1^2, \mathcal{C}_2^2$. Hence, the maximum computation complexity of Algorithm 3 is $\mathcal{O}(N(\mathcal{C}_1^2 + \mathcal{C}_2^2))$. By the same reason, we define the maximum numbers of iteration as $\mathcal{C}_3^2, \mathcal{C}_4^2$ while finding the root of (44a), (44b). Meanwhile, we denote the maximum iterations of the used root finding method for adjusting the value of ω_k as $\mathcal{C}_0^2$. Therefore, the maximum computation complexity of Algorithm 4 is $\mathcal{O}(N\mathcal{C}_0^2(\mathcal{C}_3^2 + \mathcal{C}_4^2))$. Finally, the maximum overall computation complexity of the resource allocation algorithm can be represented as $\mathcal{O}(N\mathcal{C}_0^2(\mathcal{C}_3^2 + \mathcal{C}_4^2))$.

D. Algorithm Design

Fig. 3 shows the solving process and transformation relationship from problem P_1 to P_3. We give the pseudo program of our proposed algorithm JOSR. Learned from the discussion of B and C, the maximum computational complexity of the relay selection policy algorithm and the resource allocation strategy algorithm are $\mathcal{O}(N\mathcal{C}_0^1(\mathcal{C}_3^1 + \mathcal{C}_4^1))$ and $\mathcal{O}(N\mathcal{C}_0^2(\mathcal{C}_3^2 + \mathcal{C}_4^2))$, respectively. Thus, the maximum computation complexity of our proposed algorithm JOSR is $\mathcal{O}(N \max(\mathcal{C}_0^1(\mathcal{C}_3^1 + \mathcal{C}_4^1), \mathcal{C}_0^2(\mathcal{C}_3^2 + \mathcal{C}_4^2))) =$

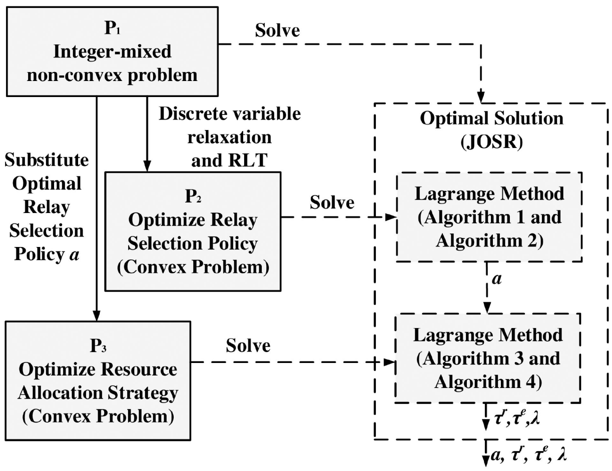

Fig. 3. Process of problem transformation and solving.

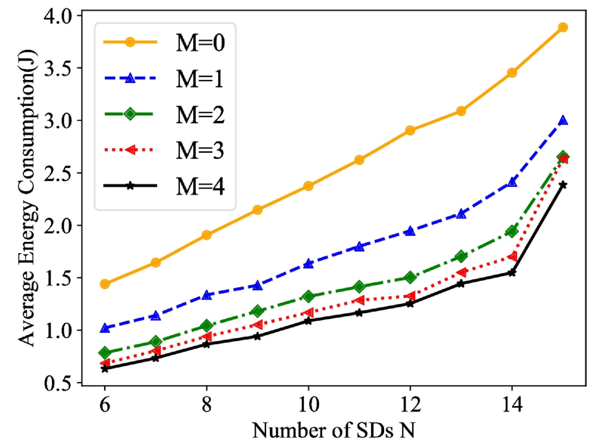

Fig. 4. Average energy consumption versus number of SDs N under different number of WAPs M.

Algorithm 5: Joint Optimization of Relay Selection and Resource Allocation Algorithm (JOSR) for Solving P_1.

compute $\boldsymbol{l}$, $\boldsymbol{l}^{loc}$, $\boldsymbol{a}$, by using Algorithm 1;
if $(\sum_{i=1}^{N}\sum_{j=1}^{M} l_{i,j}C_i - F_e^{max}) > 0$ **then**
  compute $\boldsymbol{l}$, $\boldsymbol{l}^{loc}$, $\boldsymbol{a}$ by using Algorithm 2;
obtain $\mathcal{N}_j, j \in \mathcal{M}$ with $\boldsymbol{a}^*$;
compute $\boldsymbol{\lambda}$, $\boldsymbol{\lambda}^{loc}$, $\boldsymbol{\tau}^r$, $\boldsymbol{\tau}^e$ by using Algorithm 3;
if $(\sum_{j=1}^{M}\sum_{i\in\mathcal{N}_j} \lambda_i C_i - F_e^{max} > 0)$ **then**
  compute $\boldsymbol{\lambda}$, $\boldsymbol{\lambda}^{loc}$, $\boldsymbol{\tau}^r$, $\boldsymbol{\tau}^e$ by using Algorithm 4;
return $\boldsymbol{a}, \boldsymbol{\lambda}$, $\boldsymbol{\tau}^r$, $\boldsymbol{\tau}^e$;

$\mathcal{O}(N\mathcal{C}_0^1(\mathcal{C}_3^1 + \mathcal{C}_4^1))$ The maximum computation complexity of Algorithm 5, i.e., JOSR is not only affected by the iteration times to solve the nonlinear equations, but also by the number N of SDs.

IV. Simulation Results

In this section, the simulation environment for the proposed D2D-enabled MEC system is designed. The effectiveness of the proposed algorithm JOSR is evaluated by simulation. Unless otherwise stated, all hyperparameters are listed as follows. We assume that the MEC system has three wireless access points and each mobile device has a computation-intensive task in a given rectangular area. Then the SD and the WAP are located with a distance followed the uniform distribution with $d \sim U(1, 20)$ m. It is assumed that the subchannel number is more than SDs number. While SD transmits the offloading data to WAP, we set AWGN and the subchannel bandwidth followed the uniform distribution with $N^r \sim U(1, 5) \cdot 10^{-7}W$ and $B^r \sim U(1, 5)$ Mbps, respectively. Meanwhile, the distance between the WAP and the AP follows the uniform distribution with $D \sim U(100, 150)$ m. We set $N^e = 10^{-7}W$ and $B^e = 20$ Mbps [26], respectively. And the channel power gain in the MEC system is modeled as $h = 10^{-3}d^{-\varsigma}\phi$[31], where ϕ defines short-term fading which is assumed to follow an approximately Gaussian distributed as $\phi \sim CN(0, 1)$. ς represents the path-loss exponent and here set $\varsigma = 3$. For computation model, we set the maximum computation capacity of MEC server $F_e^{\max} = 12000 Megacycles$, the CPU frequency for each SD follows the uniform distribution $F_i^{\max} \sim U(0.4, 1)$ GHz. The input data size of the computation task follows the uniform distribution $I_i \sim U(100, 500) KB$, and the total CPU cycles for each task follow Gaussian distribution $C_i \sim CN(1000, 250) Megacycles$ [20]. The effective switched capacitance for each SD follows the uniform distribution $\kappa_i \sim U(1, 10) \cdot 10^{-27}$. Please note that the following simulations are performed 100 times and averaged to eliminate contingency. The simulation results are obtained by performing the programs coded with Python on a regular PC with $2 * 2.88$ GHz CPU, 4G memory. Besides, setting the time-block $T = 1.0$ s.

A. Effect of Energy Saving

In this simulation, we provide numerical results to verify the performance of the proposed D2D-enabled MEC system. We compare the effect of energy saving with the different number of wireless access points.

In Fig. 4, we show the average energy consumption with N increasing from 6 to 15 under different number of WAPs M. One can see that with the increase of the number of SDs, the average energy cost increases correspondingly, and more and more fast with increasing, as the computation capacity of MEC server is occupied and more computation tasks are executing in SDs. Also, with the increase of the number of WAPs M, the average energy decrease, as SDs can have better relay selection policy to offload the computation data under more WAPs in the MEC system.

In Fig. 5, we show the average energy consumption with T increasing from 1.0 s to 1.6 s under different number of WAPs M. $N = 10$ is set in this figure. one can see that with the increase of the time block T, the energy consumption decrease correspondingly, but slowly. Similarly, in Fig. 5, with the increase of

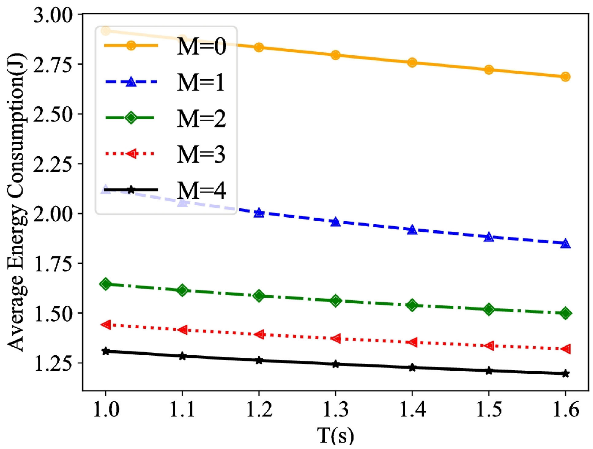

Fig. 5. Average energy consumption versus time block T under different number of WAPs M.

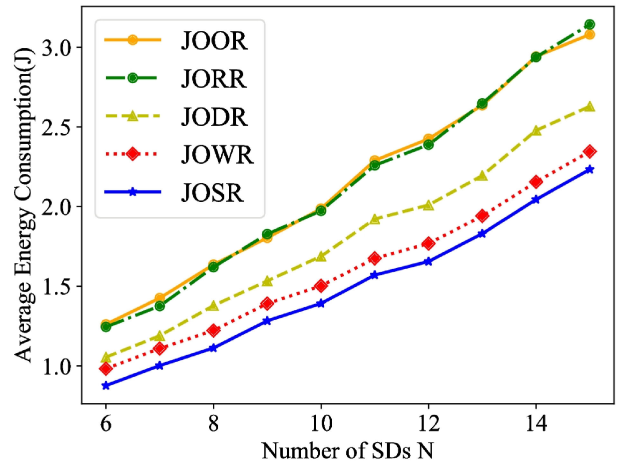

Fig. 6. Average energy consumption versus number of SDs N with different relay selection policy.

the number of WAPs, the average energy cost decrease. The proposed MEC system can effectively reduce energy consumption through analysis of Figs. 4 and 5.

B. Effect of Proposed Relay Selection Policy and Resource Allocation Strategy

In this simulation, we evaluate the performance of the JOSR algorithm from the relay selection policy and resource allocation strategy.

We firstly verify the effects of proposed relay selection policy to solve the mixed-integer non-convex optimization problem. According to the researches on relay selection [36], [44], the effectiveness of the proposed JOSR algorithm is compared with the following methods. The comparative methods include selecting the WAP based on the distance from SD to WAP, randomly-selected WAP, choosing WAP in order and going for the relay WAP according to the information for the wireless link, i.e., $\psi_{ij}(0)$. These comparison algorithms are named JODR, JORR, JOOR, JOWR. Specially, we set the number of WAPs $M = 3$.

In Fig. 6, we compare the effectiveness of the proposed relay selection policy with the other four benchmarks under the number of SDs increasing from 6 to 15 with $T = 1.0$ s. One can see that the average energy consumption increases with N increasing. Compared with the other four benchmarks, our proposed algorithm has the least energy consumption. Specially, compared with the JOWR algorithm, our proposed JOSR algorithm can reduce energy cost by 10%–20%. Therefore, the JOSR algorithm can achieve the best relay selection policy. Meanwhile, the value of $\psi_{i,j}(0)$ can be used as a judgment attribute to obtain an approximate optimal relay selection policy.

In Fig. 7, it demonstrates the curves of average energy consumption with T increasing from 1.0 s to 1.6 s under $N = 10$. One can see that the average energy cost decrease with T increasing, and our proposed algorithm has the minimum average energy consumption. Through analyzing Figs. 6 and 7, we can get a conclusion that the optimal relay selection policy

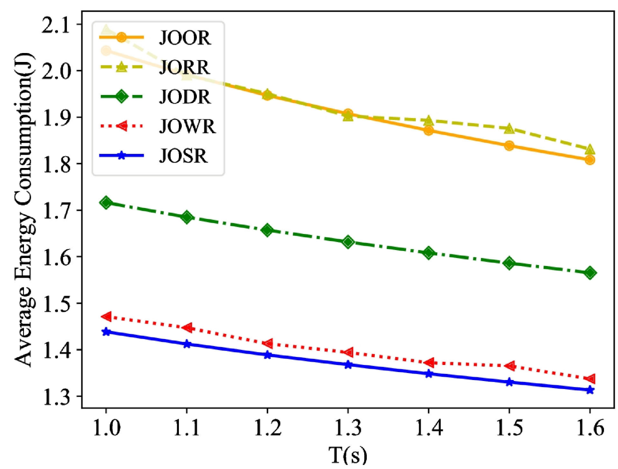

Fig. 7. Average energy consumption versus time block T with different relay selection policy.

is not only associated with its bandwidth, distance, channel gain, AWGN and the effective combination of these factors, but also has something to do with other joint factors. In the second simulation, we compare the proposed resource allocation strategy with the fixed offloading ratio $\boldsymbol{\lambda}$ under the optimal relay selection policy, which has been used in some works such as [36], [38]. $T = 1.0$ s, $N = 10$ is set in both Figs. 8 and 9, respectively. One can see that the joint optimization of relay selection and resource allocation algorithm achieves the best performance, followed by the second best solution when setting $\boldsymbol{\lambda} = 0.8$ in both Figs. 8 and 9. The performance of $\boldsymbol{\lambda} = 0.4$ can be shown as the worst solution among the test ones in both figures. As a consequence, the simulation results demonstrate that the proposed JOSR algorithm outperforms the fixed offloading ratio.

Ultimately, through the above analysis of the two simulation results, the optimal solution obtained by our proposed JOSR algorithm achieves less energy consumption than the relay selection policy and resource allocation strategy corresponding to approximate algorithms.

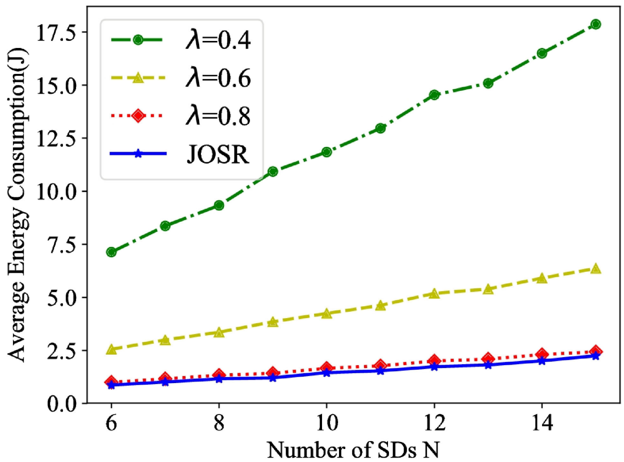

Fig. 8. Average energy consumption versus number of SDs N with different resource allocation strategy.

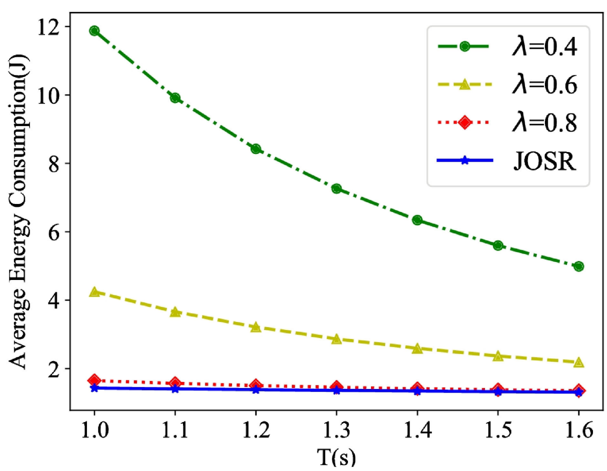

Fig. 9. Average energy consumption versus number of SDs N with different resource allocation strategy.

C. Effect of Proposed Algorithm

In this simulation, we verify the effect of the proposed algorithm JOSR. We compare the energy consumption and time cost of the proposed JOSR algorithm with the brute method (BM) and the Genetic Algorithm (GA), respectively. To avoid the local extremum in GA, we set the percent of crossover and mutation as 0.6, 0.01, separately. And we use the energy consumption of each relay selection policy as the fitness.

Fig. 10 indicates the energy cost and execution time with N increasing from 6 to 15 under T = 1.0 s. From the histogram in Fig. 10, we can see that the proposed method can get the optimal solution the same as BM. However, GA cannot obtain the optimal strategy because of trapping into local extremum. The curves in Fig. 10 show that our proposed algorithm has the lowest increasing rate in execution time. Fig. 11 shows the energy consumption and execution time with T increasing from 1.0 s to 1.6 s under $N = 10$. From the histogram in Fig. 11, the energy consumption obtained from our proposed algorithm is

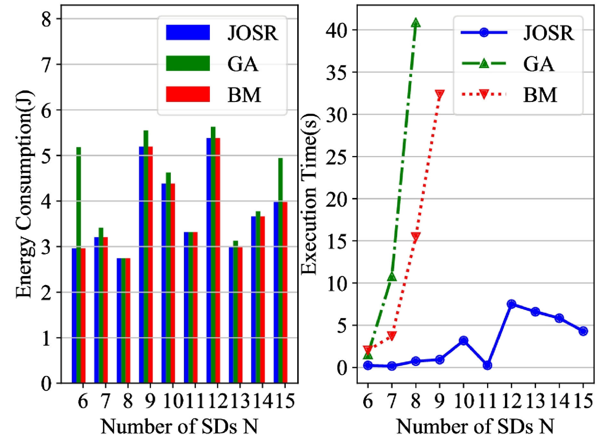

Fig. 10. Energy consumption and execution time versus number of SDs N.

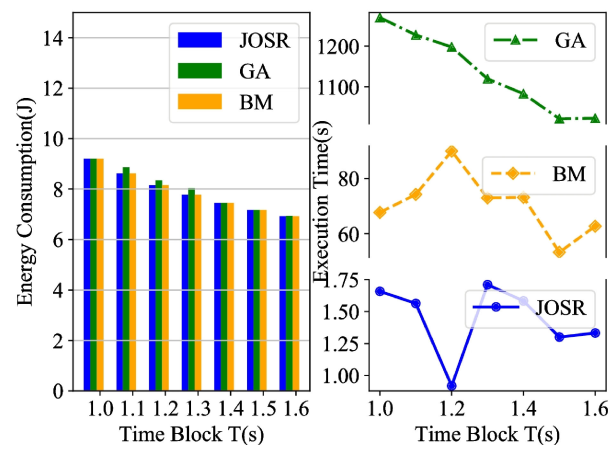

Fig. 11. Energy consumption and execution time versus time block T.

equal to the optimal value of BM, which verify that our proposed algorithm is feasible and effective. Meanwhile, from the curves in Fig. 11, the time cost is much less than other two algorithms. This is because adopted optimization techniques reduce the computational complexity. Therefore, our proposed JOSR algorithm is more efficient than other baselines, and usually achieves the ideal solution within an acceptable time cost. The simulation results of Fig. 11 verify the conclusion of Fig. 10 again. Hence, our proposed algorithm reduces the execution time and produces the optimal solution.

Through analyzing the three sets of simulation results, our presented algorithm is suitable for solving the integer-mixed non-convex optimization problem in the D2D-enabled MEC system.

V. Conclusion

In this paper, we propose a joint optimization relay selection and resource allocation problem in the D2D-enabled MEC system. The system model is presented, and the energy consumption minimization problem is proposed. The optimization problem is formulated as an integer-mixed non-convex optimization problem, which is a NP-hard problem. The convex optimization techniques, i.e., the discrete variable relaxation, RLT, are used to convert Problem P_1 to problem P_2, and the optimal relay selection policy can be obtained by solving problem P_2. After obtaining relay selection policy, the original problem P_1 is transformed into problem P_3 which can be solved by using the Lagrange Method. Hence, a lower complexity algorithm, i.e., JOSR is proposed. Simulation results verify the effectiveness of the D2D-enabled MEC system in energy saving and the feasibility and efficiency of the proposed algorithm. In the future work, we will focus on selecting the best computation model [42], [43] from executing locally, fully offloading to MEC server without or with the help of relay, and fully offloading to nearby SD.

REFERENCES

[1] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014

[2] S.-H. Kim and S.-J. Han, "Contour routing for peer-to-peer dtn delivery in cellular networks," in *Proc. COMSNETS*, Jan. 2012, pp. 1–9.

[3] E. Dutkiewicz, "Radio spectrum research for emerging iot and 5G networks: applications to smart buildings and smart cities," in *Proc. Symp. Inf. Commun. Technol.*, 2016, pp. 2–2.

[4] E. Dutkiewicz, Y. He, B. A. Jayawickrama, and H. V. Abeywickrama, Radio environment maps generation and spectrum sensing testbed for spectrum sharing in 5G networks," in *Proc. APWC*, Sep. 2017, pp. 33–36.

[5] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: A survey," *IEEE Access*, vol. 7, pp. 131543–131558, 2019.

[6] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[7] M. Patel *et al.*, "Mobile-edge computing introductory technical white paper," *White Paper, Mobile-Edge Computing (MEC) Industry Initiative*, pp. 1089–7801, 2014.

[8] A. Sengupta, R. Tandon, and O. Simeone, "Cloud RAN, and edge caching, Fundamental performance trade-offs," in *Proc. IEEE SPAWC*, 2016, pp. 1–5.

[9] F. Gabry, V. Bioglio, and I. Land, "On energy-efficient edge caching in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3288–3298, Dec. 2016.

[10] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, and M. A. Javed, "A survey of device-to-device communications: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2133–2168, 2018.

[11] F. S. Shaikh and R. Wismller, "Routing in multi-hop cellular device-to-device (D2D) networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2622–2657, 2018.

[12] I. Basturk, Y. Chen, and M. Alouini, "Energy-efficient communication for user-relay aided cellular networks with OFDMA," *Phys. Commun.*, vol. 33, pp. 153–164. 2019.

[13] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[14] J. Liu, P. Li, J. Liu, and J. Lai, "Joint offloading and transmission power control for mobile edge computing," *IEEE Access*, vol. 7, pp. 81640–81651, 2019.

[15] J. Liu, and Q. Zhang, "Edge computing enabled mobile augmented reality with imperfect channel knowledge," in *Pro. Eur. Wireless*, 2019, pp. 1–6.

[16] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial offloading scheduling and power allocation for mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6774–6785, Aug. 2019.

[17] T. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, "Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration," *IEEE Trans. Comput.*, vol. 67, no. 9, pp. 1287–1300, Sep. 2018.

[18] T. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2017.

[19] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[20] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[21] Z. Ning *et al.*, "Deep reinforcement learning for intelligent internet of vehicles: An energy-efficient computational offloading scheme," *IEEE Trans. Cogn. Commum. Netw.*, vol. 5, no. 4, pp. 1060–1072, Dec. 2019.

[22] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, Jun. 2019.

[23] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[24] P. Liu, G. Xu, K. Yang, K. Wang, and X. Meng, "Jointly optimized energy-minimal resource allocation in cache-enhanced mobile edge computing systems," *IEEE Access*, vol. 7, pp. 3336–3347, 2018.

[25] Q. Tang, R. Xie, T. Huang, and Y. Liu, "Jointly caching and computation resource allocation for mobile edge networks," *IET Netw.*, vol. 8, no. 5, pp. 329–338, 2019.

[26] J. Zhang *et al.*, "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4283–4294, Jun. 2019.

[27] J. Ren, T. Hou, H. Wang, H. Ren, and X. Zhang, "Increasing network throughput based on dynamic caching policy at wireless access points," *Wireless Netw.*, vol. 26, no. 8, pp. 1–9, 2019.

[28] Y. Liao *et al.*, "Caching-aided task offloading scheme for wireless body area networks with mec," in *Proc. IEEE AHS*, 2019, pp. 49–54.

[29] J. Hou, X. Wang, D. Wang, Y. Lan, and Z. Liu, "Computation offloading strategy in d2d-assisted cellular networks with mobile edge computing," in *Proc. IEEE ICCC Workshops*, Aug. 2019, pp. 251–256.

[30] J. Liu, K. Luo, Z. Zhou, and X. Chen, "A D2D offloading approach to efficient mobile edge resource pooling," in *Proc. IEEE WiOpt*, 2018, pp. 1–6.

[31] Y. Liu, Y. Wang, R. Sun, S. Meng, and R. Su, "Energy efficient downlink resource allocation for D2D-assisted cellular networks with mobile edge caching," *IEEE Access*, vol. 7, pp. 2053–2067, 2018.

[32] H. Xing, L. Liu, J. Xu, and A. Nallanathan, "Joint task assignment and resource allocation for D2D-enabled mobile-edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4193–4207, Jun. 2019.

[33] Z. Kuang, G. Li, L. Zhang, H. Zhou, C. Li, and A. Liu, "Energy efficient mode selection, base station selection and resource allocation algorithm in D2D heterogeneous networks," *Peer-to-Peer Netw. Appl.*, vol. 13, no. 5, pp. 1814–1829, Jun. 2020.

[34] F. Tang, Z. Fadlullah, N. Kato, F. Ono, and R. Nallanathan, "AC-POCA: Anticoordination game based partially overlapping channels assignment in combined UAV and D2D-based networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1672–1683, Feb. 2018.

[35] Y. He, J. Ren, G. Yu, and Y. Cai, "Joint computation offloading and resource allocation in d2d enabled mec networks," in *Proc. IEEE ICC*, Oct. 2019, pp. 1–6.

[36] I. Basturk, and B. Qzbek, "Radio resource management for user-relay assisted OFDMA-based wireless networks," *Int. J. Electron. Commun.*, vol. 70, no. 5, pp. 643–651, 2016.

[37] M. Rahman, Y. Lee, and I. Koo, "Energy-efficient power allocation and relay selection schemes for relay-assisted D2D communications in 5G wireless networks," *Sensors*, vol. 18, no. 9, pp. 2865, 2018.

[38] K. Wang, K. Yang, and C. S. Magurawalage, "Joint energy minimization and resource allocation in C-RAN with mobile cloud," *IEEE Trans. Cloud Comput.*, vol. 6, no. 3, pp. 760–770, Sep. 2018.

[39] C. Li, F. Jiang, X. Wang, and B. Shen, "Optimal relay selection based on social threshold for d2d communications underlay cellular networks," in *Proc. IEEE WCSP*, 2016, pp. 1–6.

[40] A. Omran, L. Sboui, B. Rong, H. Rutagemwa, and M. Kadoch, "Joint relay selection and load balancing using D2D communications for 5G hetnet mec," in *Proc. IEEE ICC Workshops*, 2019, pp. 1–5.

[41] Y. Li, G. Xu, J. Ge, X. Fu, and P. Liu, "Communication and computation cooperation in wireless network for mobile edge computing," *IEEE Access*, vol. 7, pp. 106260–106274, 2019.

[42] M. Yao, L. Chen, T. Liu, and J. Wu, "Energy efficient cooperative edge computing with multi-source multi-relay devices," in *Proc. IEEE HPCC/SmartCity/DSS*, 2019, pp. 865–870.

[43] Y. Liao, Q. Yu, Y. Han, and M. Leeson, "Relay-enabled task offloading management for wireless body area networks," *Appl. Sci.*, vol. 8, no. 8, pp. 1409, 2018.

[44] Q. Li, J. Zhao, and Y. Gong, "Computation offloading and resource allocation for mobile edge computing with multiple access points," *IET Commun.*, vol. 13, no. 17, pp. 2668–2677, 2019.

[45] L. Huang, S. Bi, and Y. J. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, Nov. 2020.

[46] T. Rodrigues, K. Suto, H. Nishiyama, J. Liu, and N. Kato, "Machine learning meets computation and communication control in evolving edge and cloud: Challenges and future perspective," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 38–67, 2020.

[47] K. Zhu and E. Hossain, "Virtualization of 5G cellular networks as a hierarchical combinatorial auction," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2640–2654, Oct. 2016.

[48] Y. Niu, C. Gao, Y. Li, D. Jin, L. Su, and D. Wu, "Boosting spatial reuse via multiple-path multihop scheduling for directional mmwave wpans," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6614–6627, Aug. 2016.

[49] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2909–2913, 2016.

[50] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[51] X. Hu, K.-K. Wong, and K. Yang, "Wireless powered cooperation-assisted mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2375–2388, Apr. 2018.

Yang Li (Member, IEEE) received the B.S., M.S., and the Ph.D. degrees from the College of Computer Science and Technology, Jilin University, China, in 2014, 2017, and 2020, respectively. He is currently a Lecturer with the School of Information Science and Technology, North China University of Technology, Beijing, China. His main research interests include mobile cloud computing, mobile edge computing, D2D communication and distributed computing.

Gaochao Xu (Member, IEEE) received the B.S., M.S., and the Ph.D. degrees from the College of Computer Science and Technology, Jilin University, China, in 1988, 1991, and 1995, respectively. He is currently a Professor and a Ph.D. Supervisor with the College of Computer Science and Technology, Jilin University. His main research interests include distributed system, grid computing, cloud computing, the Internet of Things, information security, software testing, and software reliability assessment. As a person-in-charge or a principal participant, he has finished more than 10 national, provincial, and ministerial level research projects of China.

Kun Yang (Senior Member, IEEE) received the Ph.D. degree from the Department of Electronic & Electrical Engineering, University College London (UCL), U.K. He is currently a Chair Professor with the School of Computer Science & Electronic Engineering, University of Essex, leading the Network Convergence Laboratory (NCL), U.K. He is also an Affiliated Professor at Northwest University, China. Before joining the University of Essex at 2003, he was with UCL on several European Union (EU) research projects for several years. His main research interests include wireless networks and communications, IoT networking, data and energy integrated networks and mobile computing. He manages research projects funded by various sources such as UK EPSRC, EU FP7/H2020 and industries. He has published 300+ papers and filed 20 patents. He serves on the Editorial Boards of both IEEE and non-IEEE journals. He is an IEEE ComSoc Distinguished Lecturer (2020-2021).

Jiaqi Ge received the B.S. and M.S. degrees from the College of Computer Science and Technology, Jilin University, China, in 2018, where she is currently working toward the Ph.D. degree. Her main research interests include mobile edge computing, blockchain, D2D communication and distributed computing in the Internet of Things.

Peng Liu received the M.S. degree from the College of Computer Science and Technology, Jilin University, China, in 2016, and the Ph.D. degree, in 2019. Currently, he is a Lecturer with the College of Information and Computer Engineering, Northeast Forestry University, Harbin, China. His main research interests include mobile cloud computing, mobile edge computing, edge caching, and the Internet of Things.

Zhenjun Jin received the B.S. and Ph.D. degrees from the College of Computer Science and Technology, Jilin University, China, in 2002 and 2017, and the M.S degree from the Changchun University of Science and Technology, China, in 2009. He is currently a Lecturer with the School of Computer and Engineering, Changchun University of Technology. His research interests include cloud computing, mobile edge computing, and the Internet of Things.