(Received 15 October 2024, Accepted 19 December 2024, Scheduled 9 January 2025)

Minimization of Latency in D2D-Assisted MEC Collaborative Offloading Based on Intelligent Reflecting Surface

Jun Zhou, Chenwei Feng*, Yawei Sun, and Jiaxing Guo

School of Opto-Electronic and Communication Engineering, Xiamen University of Technology, Xiamen 361024, China

ABSTRACT: With the rapid development of various intelligent scenarios, the demand for low latency, efficient processing, and energy optimization is increasing. In smart communities, intelligent transportation, industrial environments, and other scenarios, a large amount of data is generated that needs to be processed in a short time. Traditional cloud computing models are difficult to meet the requirements for real-time and computing efficiency due to the long data transmission distance and high latency. Therefore, this paper introduces Intelligent Reflecting Surfaces (IRS) into the optimization model of Device-to-Device (D2D) communication and Mobile Edge Computing (MEC) collaborative offloading to enhance system performance and minimize total latency. This paper proposes a latency minimization problem for joint offloading mode selection, computing resource allocation, and IRS phase beamforming. The original problem is decoupled into three subproblems using the Block Coordinate Descent (BCD) algorithm. Through precise potential game theory, the Nash equilibrium (NE) is achieved, and multi-objective optimization is realized using the Lagrangian multiplier method and KKT conditions. Finally, a phase shift optimization problem is solved using the gradient descent algorithm. Simulation results show that the proposed algorithm outperforms other benchmark schemes in terms of performance.

1. INTRODUCTION

With the rapid advancement of technology, particularly the rapid iteration of big data, cloud computing, and artificial intelligence, the demand for information processing and data transmission is growing. Real-time processing and decision-making of massive data have become critical needs for current intelligent applications. From smart cities to the Industrial Internet of Things (IIoT), various scenarios are increasingly demanding low latency and efficient computing. However, traditional cloud computing architectures, due to long data transmission distances and high latency, are unable to meet the low latency and real-time requirements of these applications [1]. In response, mobile edge computing (MEC) has emerged, deploying edge servers locally to process data near its source, effectively reducing transmission latency. However, MEC systems still face challenges such as limited computing resources and bandwidth, especially in device-dense environments, where resource allocation and communication efficiency become critical issues that need to be addressed.

To address these challenges, Device-to-Device (D2D) communication technology has been introduced to support direct communication between devices, reducing reliance on edge servers. Through D2D communication, devices can transmit data directly, significantly reducing transmission delays and alleviating the burden on edge servers. Furthermore, reconfigurable intelligent surfaces (RISs), also known as intelligent reflective surfaces (IRS) or large intelligent surfaces (LIS), have received much attention for their potential to enhance the capacity and coverage of wireless networks by intelligently reconfiguring the wireless propagation environment [2]. In this paper, intelligent reflective surfaces (IRSs) are proposed as a promising new solution to achieve these goals. Specifically, an IRS is a planar array of a large number of reconfigurable passive elements (e.g., low-cost printed dipoles), where each element is capable of independently generating a certain phase shift to the incident signal (controlled by an additional intelligent controller), thus collaboratively changing the reflected signal [3]. These elements enhance the signal transmission by adjusting the phase, thus reducing the system delay. In existing research, significant progress has been made in MEC, D2D communication, and IRS technology, but their applications and optimizations are mainly focused on individual technical layers.

2. RELATED WORK

Firstly, MEC technology reduces computing latency and local device energy consumption by offloading computationally intensive tasks to MEC servers, such as base stations, access points, or roadside units, where tasks can be processed closer to the data source. Refs. [4–11] cover various aspects such as task offloading decisions, service caching, and workload scheduling. For example, the introduction of the Gibbs sampling algorithm helps minimize service delays while also reducing outsourced traffic. Additionally, other studies have explored efficient allocation of computing resources by optimizing network access selection and service placement to handle the challenge of concurrent tasks from multiple users. Further work considers the impact of heterogeneous edge servers and different user locations on offloading strategies, implementing dynamic offloading decisions through methods like Markov De-

[*] Corresponding author: Chenwei Feng (cwfeng@xmut.edu.cn).

doi:10.2528/PIERB24101501

Published by THE ELECTROMAGNETIC ACADEMY

cision Processes (MDPs), which greatly reduce task processing times and communication delays. Research [12] investigated an IRS-assisted NOMA-MEC system, which significantly reduced the system delay by optimising the power allocation and IRS phase matrix, and verified the key role of IRS for channel enhancement. Research [13] introduced IRS in a cell-free MEC system and reduced the system delay by jointly optimising IRS reflections, user power and computational resources, further demonstrating the potential of IRS for enhancing communication efficiency. In addition, [14] proposed an IRS optimisation algorithm based on deep reinforcement learning, which effectively improved the performance of edge computing systems by jointly optimising IRS phases and computing offloading decisions. Different from the above studies, this research introduced IRS technique in D2D collaborative MEC systems, which achieved lower latency and higher resource utilisation by jointly optimising task offloading, resource allocation, and IRS dynamic phase adjustment, providing a new solution for complex communication scenarios.

D2D-assisted MEC offloading, as another important technology, has gained widespread research attention. D2D communication allows devices to transmit data directly, bypassing the edge server, thereby reducing network load and latency. Various D2D communication frameworks proposed in [15–20] focus on optimizing energy consumption and task allocation strategies. For example, the D2D offloading framework in cognitive radio networks combines power control and deadline optimization in direct communication between users to minimize system energy consumption. Researchers have also proposed innovative mechanisms for D2D-assisted computation offloading, such as pricing-based matching algorithms to optimize resource sharing and task allocation between users, particularly for delay-sensitive applications. In these frameworks, D2D communication effectively improves resource utilization, especially on devices with underutilized computing resources.

The collaborative offloading between MEC and D2D further optimizes the performance of mobile edge computing systems. By jointly using MEC and D2D, researchers can flexibly schedule tasks across multiple devices, achieving efficient resource utilization. Refs. [21–26] not only optimize offloading decisions and resource allocation but also propose solutions that treat both MEC servers and devices with spare computing capacity as edge nodes, jointly handling computational tasks. This collaborative mechanism effectively reduces task processing delays and local device energy consumption, while further optimizing device selection, task partitioning, and caching strategies through reinforcement learning. Moreover, the introduction of a collaborative mechanism helps address communication bottlenecks under high load conditions, enhancing the overall processing capability of the edge network. In recent years, [27] proposed a D2D collaborative offloading framework based on game theory, which achieves real-time task offloading decision-making in dynamic MEC environments by introducing satisfaction metrics and no-regret dynamics mechanisms, improving the collective system benefits and quality of service. The CODE framework proposed in [28] focuses on dealing with the offloading problem of large-scale video traffic and significantly reduces network latency through dual schemes of maximising offloading and minimising latency. Research [29], on the other hand, uses attention mechanism and deep reinforcement learning to optimise resource allocation and collaboration decisions in D2D-MEC systems, and demonstrates good performance in large-scale user scenarios.

In recent years, the application of Intelligent Reflecting Surfaces (IRS) in Mobile Edge Computing (MEC) has made significant progress. By adjusting the phase of reflective units, IRS changes the propagation paths of wireless signals, improving the performance of communication links and providing new optimization methods for MEC task offloading. Ref. [30] proposes an IRS-assisted MEC system, where the optimization of IRS phase settings and computing resource allocation significantly reduces the total task offloading delay. Ref. [31] explores the application of IRS in multi-user MEC systems, significantly reducing communication and computational delays in multi-user scenarios through joint optimization of IRS phase and task offloading strategies. To reduce the complexity of IRS phase optimization, [32] proposes a low-complexity phase adjustment algorithm, effectively addressing computational bottlenecks in large-scale IRS applications. Ref. [33] introduces an IRS-assisted multi-group multicast MISO system model and employs a joint design method based on semi-definite relaxation (SDR) and alternating optimization to simultaneously optimize transmission beamforming and IRS phase adjustment. Through such joint optimization, the system's total transmission rate is maximized while ensuring the quality of service (QoS) for each user group. Ref. [34] proposes an efficient joint optimization framework combining IRS phase adjustment with active beamforming at the base station to maximize the transmission rate of task offloading systems. Specifically, [35, 36] delve into the potential of RIS-assisted edge-D2D collaborative computing in industrial applications. This study proposes an innovative collaborative offloading mechanism that dynamically adjusts RIS phases and offloading strategies between D2D devices, improving computational efficiency and communication reliability in industrial applications. Overall, IRS technology enhances the performance of MEC task offloading by optimizing signal propagation paths, providing more efficient solutions for edge computing. Inspired by this, this paper introduces IRS into D2D communication and MEC collaborative offloading optimization, aiming to minimize total system latency through joint optimization of offloading selection, resource allocation, and IRS phase control. The Block Coordinate Descent (BCD) algorithm is adopted to decompose the problem into three subproblems, which are solved using game theory, KKT conditions, and gradient descent methods to achieve collaborative optimization.

Although significant progress has been made in reducing computing latency and optimizing communication performance with MEC, D2D, and IRS technologies, most existing studies focus on optimizing two of these technologies, with little research exploring how to organically combine all three to harness their synergistic effects and further improve overall system performance. By integrating MEC, D2D communication, and IRS technologies, future intelligent cell systems will be better able to optimize resource utilization, significantly reduce data processing delays, and enhance the efficiency of communication systems. Thus, based on the above analysis, this study aims to fill the gap in existing research. While D2D-assisted MEC offloading helps reduce task delays, this collaborative offloading also incurs additional migration costs due to unstable mobility, and some areas experience severe signal blocking and multipath effects due to dense cell buildings and multi-story structures. Therefore, integrating IRS to improve efficiency is crucial. In this study, we investigate D2D-assisted MEC collaborative offloading based on IRS, aiming to minimize system latency and improve performance by integrating MEC, D2D communication, and IRS.

The contributions of this paper are summarized as follows:

- Proposed collaborative optimization model: This paper combines Intelligent Reflecting Surface (IRS), Device-to-Device (D2D) communication, and Mobile Edge Computing (MEC), proposing a new collaborative optimization model. This model systematically addresses the total latency minimization problem in cell scenarios by jointly optimizing offloading mode selection, computing resource allocation, and IRS phase beamforming, effectively improving system communication efficiency and resource utilization.
- Application of a precise potential game model: For the offloading mode selection problem, a precise potential game model is introduced, forming a stable offloading strategy among user devices through game theory. This model balances competition among user devices, achieving global latency minimization and ensuring that the system converges at the Nash equilibrium (NE), providing a solid theoretical foundation for collaborative optimization.
- Multi-objective resource collaborative optimization: To address the complex resource allocation problem, this paper proposes a multi-objective resource collaborative optimization algorithm. By employing the Lagrangian multiplier method and KKT conditions, the joint optimization of offloading ratio and resource allocation is achieved under constraints, further reducing task processing latency. This method effectively allocates limited computing resources within the system, ensuring the stability and efficiency of multi-user task processing.
- IRS-GD phase optimization algorithm: To further optimize system performance, a gradient descent (GD)-based IRS phase optimization algorithm is designed. By dynamically adjusting the IRS phase, data transmission rates are significantly improved, reducing system latency. This algorithm provides an effective solution for the practical deployment and optimization of IRS, improving system performance.

The rest of this paper is organized as follows. In Section 2, the system model is established, and the latency minimization problem is formulated and decomposed. Section 3 designs an algorithm to solve the latency minimization problem. Section 4 presents simulation experiments and discusses the results. Finally, Section 5 concludes the paper.

3. SYSTEM MODELLING AND PROBLEM FORMULATION

3.1. Communications Model

As shown in Fig. 1, this paper considers a D2D-assisted MEC cooperative offloading system based on the assistance of multiple IRSs in a cellular scenario. The system consists of a cellular access network and a D2D communication network, a base station (BS) with edge servers located at the centre of the cellular network, user devices (UDs) and service devices (SeDs). These User Devices have limited computational resources and latency-sensitive computationally intensive tasks to be performed, and Service Devices are composed of idle users with relatively high computational power, but still limited with respect to the edge servers to handle multiple computationally intensive tasks at the same time. Define $\boldsymbol{\mathcal{M}} = \{1, 2, \dots, \boldsymbol{M}\}$ and $\boldsymbol{\mathcal{K}} = \{1, 2, \dots, \boldsymbol{K}\}$ as the sets of UDs and SeDs of the access system, respectively. Define the sets of BS0 and SeDs as $\boldsymbol{\mathcal{S}} = \{0, \boldsymbol{\mathcal{K}}\}$. The UDs compete for communication and computational resources for task processing, and the SeDs comprise multiple computationally-assisted nodes (e.g., smart terminals and small-volume devices with smaller computational capabilities). Multiple IRSs and SeDs that assist BSs and D2Ds are deployed around the cell to provide quality for smart application services in the cell. The UDs can communicate with the edge servers through the base station by using wireless cellular access technology; they can also communicate with the SeDs by using D2D communication technology, but the UDs are allowed to be within the maximum distance of the D2D communication only to establish with any of the devices in the SeDs and D2D link. The applications considered in this study are data partitioning oriented. The computational tasks on the UDs can be arbitrarily divided into three parts, and the computations are simultaneously executed locally, on the edge server and on the SeDs in parallel.

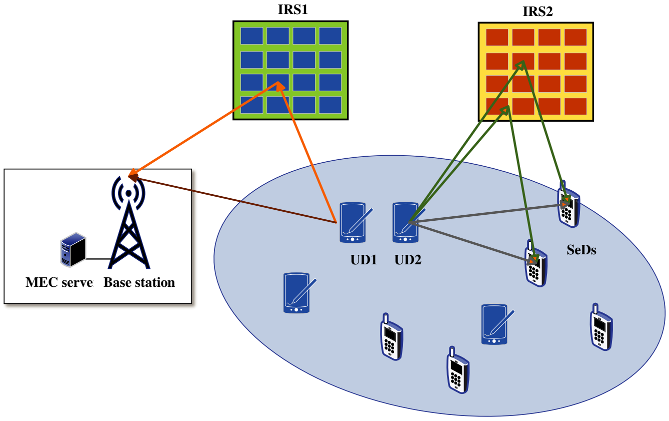

FIGURE 1. Model diagram of D2D-assisted MEC co-unloading system with IRS.

Multiple IRSs are deployed to improve the communication efficiency of UDs with BSs and SeDs in cellular and D2D networks, respectively. Controllers of the multiple IRSs are deployed in the system to control the phase shifts of the reflection elements in the BS-based and D2D-based IRSs. It is assumed that there exist L IRSs, and each IRS is equipped with

N passive reflection elements. Therefore, the channel coefficients from UD_m to BS, from UD_m to the IRS of the l-th auxiliary BS, and from the IRS of the l-th auxiliary BS to BS are $h_{B,m} \in \mathbb{C}^{1\times 1}$, $\mathbf{h}_{r,m}^{\mathrm{B}^{(l)}} \in \mathbb{C}^{N\times 1}$, $\mathbf{G}_m^{(l)} \in \mathbb{C}^{1\times N}$, respectively. Similarly, the channel coefficients from UD_m to SeD_k, UD_m to the IRS of the l-th auxiliary D2D, and the IRS of the l-th auxiliary D2D to SeD_k are $h_{k,m} \in \mathbb{C}^{1\times 1}$, $\mathbf{h}_{r,m}^{\mathrm{D}^{(l)}} \in \mathbb{C}^{N\times 1}$, $\mathbf{G}_d^{(l)} \in \mathbb{C}^{1\times N}$, respectively. In this paper, we assume that the above channel coefficients are perfectly known. For IRSs, simply set the amplitude reflection coefficients of all reflection units to 1 and denote the phase shift coefficient vector of the l-th IRS by $\theta^{(l)} = [\theta_1^{(l)}, \theta_2^{(l)}, \dots, \theta_N^{(l)}]^T$, where, for all $n \in \{1, 2, \dots, N\}$, $\theta_n^{(l)} \in [0, 2\pi)$. Then, we obtain the matrix of reflection coefficients of the l-th IRS, $\mathbf{\Theta}^{(l)} = \mathrm{diag}\left\{e^{j\theta_1^{(l)}}, e^{j\theta_2^{(l)}}, \dots, e^{j\theta_N^{(l)}}\right\}$, where j represents the imaginary unit. Let $\mathbf{\Phi} = \{\mathbf{\Theta}^{(1)}, \mathbf{\Theta}^{(2)}, \dots, \mathbf{\Theta}^{(L)}\}$ denote the set of phase variables for all L IRSs.

In communication systems, IRS enhances the channel gain between user devices (UDs) and BSs or SeDs by dynamically adjusting the phases of reflection units to improve the data transmission rate and reliability. In order to optimise the system performance, we need to reasonably allocate IRS resources to maximise the communication performance of each UD. Therefore, we choose to satisfy the principle of maximising channel gain to select IRSs and allocate them to each UD to enhance the overall communication performance of the system. The binary variable $A_{m,l}$ denotes whether the l-th IRS is assigned to UD_m, specifically, $A_{m,l} = 1$ to denote that the l-th IRS is assigned to UD_m and $A_{m,l} = 0$ to denote that the l-th IRS is not assigned to UD_m. Additionally, each UD_m can only choose to have one and only one IRS.

For a cellular link, let $H_{B,m} = h_{B,m} + \mathbf{G}_m^{(l)}\mathbf{\Theta}^{(l)}\mathbf{h}_{r,m}^{\mathbf{B}(l)}$ denote the total channel gain between the user device UD_m and the base station (BS). For each UD_m, iterate through all IRSs, $l \in \{1, 2, \dots, L\}$, select the one with the largest channel gain IRS_{l^*}, $l^* = \arg\max_l H_{k,m}$, assign the best IRS to UD_m, then $A_{m,l^*} = 1$. Similarly, for a D2D link, let $H_{k,m} = h_{B,m} + \mathbf{G}_m^{(l)}\mathbf{\Theta}^{(l)}\mathbf{h}_{r,m}^{\mathbf{B}(l)}$ denote the total channel gain between the user devices UD_m and serving devices SeDs, For each UD_m, iterate through all IRSs, $l \in \{1, 2, \dots, L\}$, select the one with the largest channel gain IRS_{l^*}, $l^* = \arg\max_l H_{k,m}$, assign the best IRS to UD_m, then $A_{m,l^*} = 1$. For computational tasks generated by UDs, UDs are able to offload the tasks to edge computing servers and SeDs for processing. Define $x_{ms} \in \{0, 1\}$ as the collaborative communication indicator, where $m \in \mathcal{M}, s \in \mathcal{S}$, then $\mathbf{x} = \{x_{ms} | m \in \mathcal{M}, s \in \mathcal{S}\}$ is the set of offloading patterns for all UDs.

3.1.1. Cellular Links

The indicator variable $x_{m0} = 1$ indicates that UD_m's task is offloaded to the edge server for processing over the cellular link. In cellular networks, to avoid interference between different users, partial offloading is performed using an orthogonal frequency division multiple access (OFDMA) scheme. Each user device (UD) is assigned a separate subchannel of the cellular link when offloading a task, ensuring that multiple UDs can perform task offloading at the same time without interfering with each other. In OFDMA, each subchannel is orthogonal in the frequency domain, which means that despite the overlap between the spectra, due to its orthogonality, UDs transmitting data simultaneously do not interfere with each other, thus improving the overall efficiency and stability of the system. In MEC offloading, each UD is assigned a cellular link subchannel for transmitting the offloaded task to the target MEC server. Edge servers can process multiple tasks in parallel, while each SeD can only serve one task at a time. Then the set of cellular-UDs is defined as $\mathcal{U}_c = \{m | x_{m0} = 1, \forall m \in \mathcal{M}\}$. We define p_m to be the transmission power between UD_m and the edge server, and furthermore, we assume that the wireless bandwidth and noise power remain constant while each computational task is being transmitted, denoted as B_m and δ_m^2, respectively. Thus, the communication rate between UD_m and BS (we ignore the delay from the BS to the edge server) can be expressed as:

$$R_{B,m}^{\mathrm{mec}} = B_m \log_2\left(1 + \frac{p_m \left|h_{B,m} + \sum_{l=1}^{L} A_{m,l}\mathbf{G}_m^{(l)}\mathbf{\Theta}^{(l)}\mathbf{h}_{r,m}^{\mathbf{B}(l)}\right|^2}{\delta_m^2}\right) \tag{1}$$

where $m \in \mathcal{U}_c$.

3.1.2. D2D Links

UD_m can form a D2D link with any of the sets SeD_k within the maximum distance d^max. Indicator variable $x_{mk} = 1$ indicates that UD_m's tasks are offloaded to SeD_k for remote processing over a D2D link. Therefore, the D2D link is denoted by x_mk. For a given time range, the feasible D2D links of UDs remain the same. Each UD can establish a D2D link with only one SeD within the current time slot, and the connection relationship between all UDs and SeDs is fixed within this time slot. In addition, a UD can offload its computation to at most one SeD, and each SeD can serve at most one offloading device, thus forming non-overlapping D2D pairs in the network. Then, the set of D2D-UDs is defined as $\mathcal{U}_d = \{m | x_{mk} = 1, \forall m \in \mathcal{M}, \forall k \in \mathcal{K}\}$. We define p_d to be the transmission power between UD_m and SeD_k, given a fixed bandwidth B_d and noise power δ_d^2 in D2D offloading, and obtain the communication rate between UD_m and SeD_k, denoted as:

$$R_{k,m}^{\mathrm{d2d}} = B_{d2d} \log_2\left(1 + \frac{p_d |h_{k,m} + \sum_{l=1}^{L} A_{m,l}\mathbf{G}_d^{(l)}\Theta^{(l)}\mathbf{h}_{r,m}^{\mathrm{D}(l)}|^2}{\delta_{\mathrm{d2d}}^2}\right) \tag{2}$$

where $m \in \mathcal{U}_d$.

3.2. Computational Model

For each UD, there is a latency-sensitive application task that needs to process a large amount of input data. We consider applications oriented towards data partitioning, for which the input data is known in advance and can be arbitrarily partitioned for parallel processing due to per-bit independence. Typical examples are virus scanning, file/graphics compression, recognition, and vision applications [37]. Divide the system time into time slots. The system state is constant within time slots but changes between time slots. Each time slot BS allocates computational resources. A computational task on UD_m can be described as $\mathcal{I}_m = \{Q_m, C_m, \tau_m, f_m\}$, where Q_m is the size of the task data (in bits), C_m the computational resources required to compute one bit of the task (measured in CPU cycles per second), τ_m the task deadline, i.e., the maximum tolerable delay of the task execution (in seconds), and let f_m denote the local computational power. Let f_{m0} and f_{mk} denote the computational resources allocated to UD_m for performing offloading tasks at BS0 and SeD_k per second, respectively, and the computational resource allocation profile is defined as $\mathbf{f} = \{f_{ms} | m \in \mathcal{M}, s \in \mathcal{S}\}$.

As mentioned above, consider the partial offloading policy, where each user has only one task to be offloaded at a given time in a short time computational offloading problem, and the application data partitioning of the UD, where a part of the task is processed locally, and the rest of the task is offloaded to be executed remotely. Define the variable $\alpha_m \in [0, 1]$ to denote the proportion of partial task offloading for UD_m. Then, $(1 - \alpha_m)Q_m$ bit is processed locally, and $\alpha_m Q_m$ bit is offloaded to be processed on the remote device. $x_{m0} = 1, \forall m \in \mathcal{M}$ to denote the task is offloaded to the edge server for processing via a cellular link. Tasks are offloaded to the edge server for processing via cellular links, and $x_{mk} = 1, \forall m \in \mathcal{M}, \forall k \in \mathcal{K}$ indicates that tasks are offloaded to the SeDs for remote processing via D2D links.

3.2.1. Local Computing

The UDs will process a small portion of their tasks locally. Each UD has a fixed CPU frequency, and the time consumption for local computation depends on the CPU clock frequency f_m and the number of CPU cycles required per bit C_m. Then, the local computation delay $D_m^{\rm L}$ for UD_m is

$$D_m^{\rm L} = \frac{(1 - \alpha_m) Q_m C_m}{f_m} \tag{3}$$

3.2.2. Edge Computing

The total latency of offloading to the edge server for remote processing consists of three components, namely the time $D_m^{{\rm mec},t}$ for uploading the computational task, the time $D_m^{{\rm mec},c}$ for executing the task on the MEC server, and the time for downloading the computational results by the UD. Since the size of the downloaded results is usually much smaller than the size of the transmitted data, it is ignored. Therefore, the latency to complete the edge computation can be calculated as:

$$D_m^{\rm mec} = D_m^{{\rm mec},t} + D_m^{{\rm mec},c} = \frac{\alpha_m Q_m}{R_{B,m}^{\rm mec}} + \frac{\alpha_m Q_m C_m}{f_{m0}}, \quad m \in \mathcal{U}_c. \tag{4}$$

It is assumed that the computational capacity at the edge servers is limited, so the feasible computational resource allocation must satisfy $\sum_{m=1}^{\mathcal{M}} x_{m0} f_{m0} \leq \mathbf{F}_0$, where $\mathbf{F}_0$ is the total computational capacity of the edge servers (in CPU cycles per second). On the other hand, the computational resources of SeDs are fully allocated to offloading tasks because a SeD serves only one UD.

3.2.3. D2D-SeDs Computing

Similar to edge computing, the latency of UD_m to complete D2D-SeDs computation can be obtained from the D2D transmission latency $D_m^{{\rm d2d},t}$ and remote execution latency $D_m^{{\rm d2d},c}$, denoted as:

$$\begin{aligned} D_m^{\rm d2d} &= D_m^{{\rm d2d},t} + D_m^{{\rm d2d},c} \\ &= \sum_{k=1}^{\mathcal{K}} x_{mk} \left(\frac{\alpha_m Q_m}{R_{k,m}^{\rm d2d}} + \frac{\alpha_m Q_m C_m}{f_{mk}} \right), \quad m \in \mathcal{U}_d. \end{aligned} \tag{5}$$

3.3. Problem Formulation

In this work, the objective of this study is to minimise the sum of task computation delays for all UDs, which offload part of the computation to the edge servers or SeDs of the BS by establishing cellular or D2D links. In the case of partial offloading, then two processes are involved, i.e., local computation and computation offloading (offloading plus remote execution). Since local computation can be performed simultaneously with the computation offloading process [38], the total task computation delay of UD_m is determined by the longer process, which can be expressed as:

$$D_m = \max\{D_m^{\rm L}, D_m^{\rm mec}, D_m^{\rm d2d}\} \tag{6}$$

Based on the system model of this study, considering the maximum delay constraints of the application and limited computing resources, the problem of minimizing the total system delay is formulated as a joint optimization problem for offloading decisions, computing resource allocation, and IRS phase optimization, as follows:

$$\mathcal{P}0: \quad \min_{\mathbf{x}, \boldsymbol{\alpha}, \mathbf{f}, \boldsymbol{\Theta}} \sum_{m=1}^{M} D_m \tag{7a}$$

$$\text{s.t.} \quad D_m \leq \tau_m, \quad \forall m \in \mathcal{M}, \tag{7b}$$

$$\sum_{k=1}^{K} x_{m,k} \leq 1, \quad \forall m \in \mathcal{M}, \tag{7c}$$

$$x_{ms} \in \{0, 1\}, \quad \forall m \in \mathcal{M}, \quad \forall s \in \mathcal{S}, \tag{7d}$$

$$0 < \alpha_m < 1, \quad \forall m \in \mathcal{M}, \tag{7e}$$

$$\sum_{m=1}^{\mathcal{M}} x_{m0} f_{m0} \leq \mathbf{F}_0, \tag{7f}$$

$$f_{m0} \geq 0, \quad m = 1, 2, \ldots, M, \tag{7g}$$

$$0 \leq \theta_n^{(l)} < 2\pi, \quad \forall n \in \{1, 2, \ldots, N\} \tag{7h}$$

$$\sum_{l=1}^{L} A_{m,l} = 1, \quad \forall m \in \mathcal{M}. \tag{7i}$$

As shown above, the objective function in (7a) represents the minimization of the total execution delay of all UD tasks. The constraint in (7b) implies that the execution delay of task UD_m must not exceed the maximum tolerable delay. The user association constraint in (7c) and (7d) ensures that UD_m can only select one SeD for task offloading from multiple SeDs, where user association is a binary variable. (7e) defines the range of the offloading ratio, ensuring that the offloading ratio for the part of UD_m's task processed remotely by the edge server or offloaded through a D2D link to the SeD is positive and does not exceed 1. (7f) and (7g) represent the constraints on the computing resources allocated to each device. (7h) indicates the phase beamforming constraint of the l-th IRS, while (7i) specifies that each UD_m is assigned exactly one IRS.

3.4. Problem Decomposition

According to the constraints in (7b)–(7h), it can be observed that the proposed total delay minimization problem is a Mixed Integer Non-linear Programming (MINLP) problem, which is NP-hard. Therefore, this section first decomposes the optimization problem into three subproblems: offloading mode selection, joint offloading ratio and computing resource allocation, and IRS phase optimization.

3.4.1. Offloading Decision Problems

This subproblem addresses the decision-making problem of UD_m's offloading mode in the cell network under delay constraints. By selecting the offloading mode of UD_m's task to either edge computing or D2D communication, the total system delay is minimized. When switching modes, the allocated computing resources and IRS-assisted phase are fixed to determine the offloading mode for UD_m across the entire system. The problem is then transformed into:

$$\mathcal{P}1: \quad \min_{\mathbf{x}} \sum_{m=1}^{M} D_m$$
$$\text{s.t.} \quad (7b),\ (7c),\ (7d). \tag{8}$$

This problem is a combinatorial optimization problem, where the task of each user device must be determined as being offloaded via either the cellular link or the D2D link. Heuristic algorithms, integer programming, or dynamic programming can be employed to solve this problem. For example, greedy algorithms based on network topology and channel state can be used for decision-making, or integer programming models can be used for precise solutions.

3.4.2. Joint Offloading Ratios and Computational Resource Allocation Problems

This subproblem solves the system's total delay minimization problem by determining the offloading ratio for each user device and the computing resource allocation scheme for the edge server, given the delay constraints, offloading decisions, and the correlation constraints between offloading decisions and resource allocation. In this subproblem, the phase and offloading mode are fixed, simplifying the problem into:

$$\mathcal{P}2: \quad \min_{\boldsymbol{\alpha},\mathbf{f}} \sum_{m=1}^{M} D_m$$
$$\text{s.t.} \quad (7b),\ (7e),\ (7f),\ (7g). \tag{9}$$

These two problems can be combined into a multi-objective optimization problem, which simultaneously considers offloading ratios and computing resource allocation. Multi-objective optimization algorithms, such as Multi-Objective Particle Swarm Optimization (MOPSO) or Multi-Objective Genetic Algorithm (MOGA), can be used to solve this. Alternatively, the two problems can be solved independently, and iterative optimization can be used to gradually optimize the offloading decisions and resource allocation, for example, using an Alternating Minimization algorithm or a step-by-step optimization approach.

3.4.3. IRS Phase-Shift Optimisation Problem

By fixing UD_m's offloading mode and allocated computing resources, the IRS phase is adjusted to minimize system delay. At this point, the optimization problem is transformed into:

$$\mathcal{P}3: \quad \min_{\mathbf{\Theta}} \sum_{m=1}^{M} D_m$$
$$\text{s.t.} \quad (7b),\ (7h). \tag{10}$$

The IRS phase problem is typically a continuous optimization problem, requiring the adjustment of IRS phase settings to maximize system performance. Numerical optimization methods such as Gradient Descent, Conjugate Gradient, or Quasi-Newton methods can be used to solve this. Additionally, convex optimization theory can be considered, using convex optimization algorithms such as the interior-point method or projected gradient method to solve the problem.

4. ALGORITHM DESIGN AND IMPLEMENTATION

In this section, the solutions to the above three subproblems will be discussed in detail. The offloading mode selection problem will be solved in Subsection 3.1; the joint offloading ratio and computational resource allocation problem will be solved in Subsection 3.2; the IRS phase optimisation problem will be solved in Subsection 3.3; and in Subsection 3.4, an algorithm for delay minimisation in the alternating iteration method will be proposed until the algorithm converges, and a suboptimal solution is obtained.

4.1. Precise Potential Game Optimisation Algorithms

From the perspective of game theory, this subsection utilises the accurate potential game to solve the task offloading strategy problem, i.e., the choice of UD_m offloading mode. The objective of the whole optimisation problem is to minimise the total delay of the whole system, and based on this condition, an accurate potential game game-theoretic model is introduced. In this paper, UD_m can choose to offload to the edge servers via the cellular network, or to the SeDs via D2D communication. For all UDs in the system, the offloading modes are assumed to be X_{-ms} when the offloading modes of all other user devices are determined, except for UD_m. Then, UD_m will choose the optimal offloading mode that minimises its own delay, so the offloading decision subproblem is denoted as:

$$\min_{\mathbf{x}} D_m(x_{ms}, X_{-ms}), \quad \forall m \in \mathcal{M}, \quad \forall s \in \mathcal{S} \tag{11}$$

Since there is a competitive relationship between end-users in the system, the offloading mode selection game can be defined as $G = \{M, x_{ms}, D_m | m \in \mathcal{M}, s \in \mathcal{S}\}$, where **M** denotes the set of participants in the game, i.e., all end-devices in the system; x_{ms} denotes the strategy space of UD_m, i.e., the offloading modes, and D_m denotes the delay function of the participants.

Definition 1: For offload mode policy x^*_{ms}, if all UDSs in the system satisfy

$$D_m(x^*_{ms}, X^*_{-ms}) \leq D_m(x_{ms}, X^*_{-ms}), \quad \forall m \in \mathcal{M}, \quad \forall s \in \mathcal{S} \tag{12}$$

then the offloading mode strategy x^*_{ms} is said to be the NE of the game G. The NE is a state that makes the system stable, and when the Nash equilibrium is reached, and none of the other players change their strategies, none of the players in the game can further increase their utility by unilaterally changing their own strategies, i.e., all the players in the game have reached the equilibrium state.

Definition 2: If there exists a potential function $P(X)$ for the game process, when the unloading pattern of UD_m changes unilaterally from x_{ms} to x'_{ms} and $x_{ms}, x'_{ms} \in \mathbf{x}$, there is the following relation.

$$\begin{aligned} & D_m(x_{ms}, X_{-ms}) - D_m(x'_{ms}, X_{-ms}) \\ = & P(x_{ms}, X_{-ms}) - P(x'_{ms}, X_{-ms}), \ \forall m \in \mathcal{M}, \ \forall s \in \mathcal{S} \end{aligned} \tag{13}$$

Then the game is an exact potential game. Every precise potential game with a finite set of strategies has NE [39] and has Finite Improvement Property (FIP), i.e., any step of the update process for a better response must be finite and lead to NE.

Corollary 1: The game G is an exact potential game with the potential function shown in Equation (14), and the game G always converges to NE with FIP.

$$\begin{aligned} P(X) = & x_{mk} \sum_{m=1}^{M} D_m^{d2d} \\ & + (1 - x_{mk}) \left(x_{m0} D_m^{mec} + \sum_{m'=1, m' \neq m}^{M} D_m^{d2d} \right), \\ & \forall m \in M \end{aligned} \tag{14}$$

Proof: When UD_m updates the unloading mode from x_{ms} to x'_{ms}, the potential function $P(X)$ should satisfy Equation (13), and when UD_m's unloading mode changes from D2D unloading mode to MEC unloading mode, $x_{ms} = x_{mk} = 1$, $x'_{ms} = x_{m0} = 1$. Then there is

$$\begin{aligned} & P(x_{mk}, X_{-ms}) - P(x_{m0}, X_{-ms}) \\ = & \sum_{m=1}^{M} D_m^{d2d} - D_m^{mec} - \sum_{m'=1, m' \neq m}^{M} D_m^{d2d} - D_m^{mec} \\ = & D_m(x_{mk}, X_{-ms}) - D_m(x_{m0}, X_{-ms}) \end{aligned} \tag{15}$$

The derivation reveals that the potential function $P(X)$ always satisfies Equation (13) for a change in UD_m's offloading pattern. Thus, the game G satisfies the conditions for an accurate potential game, and there always exists NE. After obtaining the optimal unloading mode selection strategy via FIP, none of UD_m has an incentive to deviate unilaterally.

Algorithm 1 leverages a precise potential game model to optimize the task offloading strategy for user devices (e.g., choosing edge computing or D2D communication). By updating the offloading modes iteratively based on changes in the potential function, it minimizes the total system delay and ensures convergence to a Nash Equilibrium.

Algorithm 1 Task Offloading Strategy Optimization Based on Precise Potential Game

```
Input: $M, K, L, N, Q_m, C_m, \alpha_{mec}, \alpha_{d2d}, B_{\mathrm{d2d}}, B_{\mathrm{mec}}, p_d, p_m, G_d, G_m, h_d, h_m, f_{m0}, f_{mk}, \delta_{\mathrm{d2d}}, \delta_{\mathrm{mec}}$, iter1, $\epsilon$
Initialization Offloading modes $x_{ms}$ for all $UD_m, m \in \mathcal{M}, s \in \mathcal{S}$, and delay $D_0$
Compute the initial potential function value $P(X)$
for $t = 0$ to iter1 do
    for each user device $UD_m \in \mathcal{M}$ do
        for each offloading mode $s \in \mathcal{S}$ do
            Compute the delay $D_m(x_{ms}, X_{-ms})$ of $UD_m$ under mode $s$
        end for
        Find the offloading mode that minimizes delay $s^* = \arg\min_s D_m(x_{ms}, X_{-ms})$
        Update the offloading mode of $UD_m$, $x_{ms} \leftarrow s^*$
    end for
    Compute the new potential function value $P(X)$
    Calculate the change in the potential function $\Delta P = |P_{\mathrm{new}}(X) - P_{\mathrm{old}}(X)|$
    if $\Delta P < \epsilon$ then
        Break the loop
    end if
    Update the potential function value $P_{\mathrm{old}}(X) \leftarrow P_{\mathrm{new}}(X)$
end for
Output: Optimal offloading mode $x_{ms}$
```

4.2. Resource Co-optimisation Algorithm

After determining the offloading pattern of UD_m within the system, it is necessary to identify a multi-objective optimisation problem consisting of the offloading ratio and computational resource allocation to the MEC server. The computational resource allocation problem is usually a minimisation problem containing at least two inequality constraints; therefore, in this study, the Lagrange multiplier method with KKT conditions is used to solve the problem. The original problem is transformed into a dyadic problem to be solved by integrating the constraints into the optimisation problem, and the subproblems are shown in Equation (9).

Firstly, the task offloading ratio needs to be determined, and it is an important factor for partial offloading, which affects the latency of local and remote execution. When UD_m chooses the D2D offloading mode, the optimisation problem is expressed as follows:

$$\min_{\boldsymbol{\alpha}} \max \left\{ \frac{\alpha_m Q_m}{R^{\text{d2d}}_{k,m}} + \frac{\alpha_m Q_m C_m}{f_{mk}}, \frac{(1-\alpha_m)Q_m C_m}{f_m} \right\}$$
$$\text{s.t.} \quad (7b),\ (7e),\ (7f),\ (7g). \tag{16}$$

According to Equation (16), in the D2D offloading mode, the latency consists of two parts: local execution and remote execution. Since these two parts are simultaneous, the total delay in the D2D offloading mode should be the larger of the two, and it is obvious that the delay of UD_m is minimised when the two parallel processes spend the same amount of time, under the constraint of the total delay Equation (7b)

$$\frac{(1-\alpha_m)Q_m C_m}{f_m} \le \tau_m. \tag{17}$$

From the constraints in Equation (17), we can obtain the minimum value of α_m

$$\alpha_m^{\min} = 1 - \frac{\tau_m f_m}{Q_m C_m}. \tag{18}$$

Define variable α_{d2d} as the offloading ratio that achieves the minimum value of the total delay in the D2D offloading mode, and when the two parts of the delay are equal, we can obtain Equation (19)

$$\alpha_m = \frac{1}{1 + \frac{f_m}{C_m R^{\text{d2d}}_{k,m}} + \frac{f_m}{f_{mk}}} = \alpha_{d2d}. \tag{19}$$

It is shown below that the delay D_m is minimised in the D2D offloading mode when $\alpha_m = \alpha_{d2d}$.

Derivation of the objective function when the latency is large for local and remote execution, respectively, leads to Equations (20) and (21)

$$\frac{\partial D_m}{\partial \alpha_m} = -\frac{Q_m C_m}{f_m} < 0,\ \forall m \in M,\ \alpha_m^{\min} \le \alpha_m \le \alpha_{d2d}, \tag{20}$$

$$\frac{\partial D_m}{\partial \alpha_m} = \frac{Q_m}{R^{\text{d2d}}_{k,m}} + \frac{Q_m C_m}{f_{mk}} > 0, \forall m \in M, \alpha_{d2d} \le \alpha_m \le 1. \tag{21}$$

It can be seen that the objective function is monotonically decreasing on the interval $\alpha_m^{\min} \le \alpha_m \le \alpha_{d2d}$ and monotonically increasing on $\alpha_{d2d} \le \alpha_m \le 1$. Therefore, the time delay D_m is minimised at $\alpha_m = \alpha_{d2d}$.

When UD_m selects the MEC offloading mode, according to Equation (6), the magnitude of the total system delay is determined by the larger value of the local computation delay and the MEC task offloading delay, and it is necessary to take into account the offloading ratio and the computational resource allocation of the MEC server at the same time; therefore, the optimisation problem is expressed as:

$$\min_{\boldsymbol{\alpha},\mathbf{f}} \sum_{m=1}^{M} \max \left\{ \frac{\alpha_m Q_m}{R^{\text{mec}}_{B,m}} + \frac{\alpha_m Q_m C_m}{f_{m0}}, \frac{(1-\alpha_m)Q_m C_m}{f_m} \right\}$$
$$\text{s.t.} \quad (7d),\ (7f),\ (7g). \tag{22}$$

Similarly, from the minimum delay in the D2D offloading mode, the variable α_{mec} is defined as the offloading ratio in which the total delay in the MEC offloading mode obtains the minimum value, and the delay D_m in the MEC offloading mode is minimum when $\alpha_m = \alpha_{mec}$.

$$\alpha_m = \frac{1}{1 + \frac{f_m}{C_m R^{\text{mec}}_{k,m}} + \frac{f_m}{f_{m0}}} = \alpha_{mec}. \tag{23}$$

Substituting Equation (23) into the optimisation problem Equation (22), i.e.,

$$\min_{\mathbf{f}} \sum_{m=1}^{M} \left(1 - \frac{1}{1 + \frac{f_m}{C_m R^{\text{mec}}_{k,m}} + \frac{f_m}{f_{m0}}} \right) \frac{Q_m C_m}{f_m} \tag{24a}$$

$$\text{s.t.} \quad \sum_{m=1}^{\mathcal{M}} x_{m0} f_{m0} \le \mathbf{F}_0, \tag{24b}$$

$$f_{m0} \ge 0, \quad m = 1, 2, \ldots, \mathcal{M}, \tag{24c}$$

Notice that constraints (24b) and (24c) are convex such that the objective function of (24a) is denoted as D_m, which is a multivariate function. The function is convex if its Hessian matrix is positive definite. The Hessian matrix can be expressed as:

$$H = \begin{bmatrix} \frac{\partial^2 D_m}{\partial f_{10}^2} & \cdots & \frac{\partial^2 D_m}{\partial f_{10} \partial f_{m0}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 D_m}{\partial f_{m0} \partial f_{10}} & \cdots & \frac{\partial^2 D_m}{\partial f_{m0}^2} \end{bmatrix} \tag{25}$$

By obtaining the second order derivatives of D_m with respect to f_{m0}, we have

$$\frac{\partial^2 D_m}{\partial f_{m0}^2} = \frac{2AB}{(Af_{m0} + B)^3} \cdot \frac{Q_m C_m}{f_m} \ge 0, \quad \forall m \in \mathcal{M} \tag{26}$$

where $A = 1 + \frac{f_m}{C_m R^{\text{mec}}_{k,m}}$, $B = f_m$. Similarly, the second-order mixed partial derivatives of D_m are

$$\frac{\partial^2 D_m}{\partial f_{m0} \partial f_{m'0}} = 0, \quad \forall m, \quad m' \in \mathcal{M}, \quad m \ne m' \tag{27}$$

By Equations (26) and (27), we can determine that the Hessian matrix H is positive definite, and hence (24a) is a convex optimisation problem that can be solved by the KKT condition. Its Lagrangian function is expressed as

$$\begin{aligned} &L(f_{10}, \dots, f_{mo}, \lambda, \mu_1, \dots, \mu_M) \\ &= \sum_{m=1}^{M} \left(1 - \frac{1}{1 + \frac{f_m}{C_m R_{k,m}^{\text{mec}}} + \frac{f_m}{f_{m0}}} \right) \frac{Q_m C_m}{f_m} \\ &\quad + \lambda \left(\sum_{m=1}^{\mathcal{M}} x_{m0} f_{m0} - \mathbf{F}_0 \right) \\ &\quad + \sum_{m=1}^{\mathcal{M}} \mu_m (\mathbf{F}_0 - f_{m0}) \end{aligned} \tag{28}$$

$$\frac{\partial L}{\partial f_{m0}} = -\frac{B}{(A f_{m0} + B)^2} \cdot \frac{Q_m C_m}{f_m} + \lambda - \mu_m = 0, \quad \forall m \in \mathcal{M} \tag{29}$$

$$\sum_{m=1}^{\mathcal{M}} x_{m0} f_{m0} - \mathbf{F}_0 = 0, \quad \forall m \in \mathcal{M} \tag{30}$$

$$\mu_m (f_{m0} - \mathbf{F}_0) = 0, \quad \forall m \in \mathcal{M} \tag{31}$$

where $\lambda, \mu_1, \dots, \mu_M$ is a non-negative Lagrange multiplier.

Thus, the Lagrange multipliers can be updated by gradient descent as

$$\lambda(t+1) \;=\; \left[\lambda(t) + \delta(t) \left(\sum_{m=1}^{\mathcal{M}} x_{m0} f_{m0} - \mathbf{F}_0 \right) \right]^+ \tag{32}$$

$$\mu_m(t+1) \;=\; [\mu_m(t) + \delta(t)(\mathbf{F}_0 - f_{m0})]^+ \tag{33}$$

where t is the iteration index, and $\delta(t)$ is the positive step size at iteration t.

Algorithm 2 uses the Lagrangian multiplier method and Karush-Kuhn-Tucker (KKT) conditions to solve the multi-objective optimization problem of offloading ratios and edge server resource allocation. It iteratively updates resource allocation variables and Lagrange multipliers to optimize resource distribution under constraints, further reducing system latency.

4.3. IRS-GD Phase Shift Optimisation Algorithm

As described in Section 3.2, the optimal solution to Problem P2 leads to $D_m = D_m^L = D_m^{d2d}$ in the D2D offloading mode and $D_m = D_m^L = D_m^{mec}$ in the MEC offloading mode. Therefore, with fixed offloading modes, the problem formulation after replacing D_m with D_m^{d2d}, D_m^{mec}, respectively, and deleting the constant term is as follows:

$$\begin{aligned} &\min_{\mathbf{\Theta}} \sum_{m=1}^{M} \frac{\alpha_m Q_m}{x_{mk} R_{k,m}^{\text{d2d}} + x_{mo} R_{B,m}^{\text{mec}}} \\ &\quad\text{s.t.} \quad (7h) \end{aligned} \tag{34}$$

Algorithm 2 Resource Collaborative Optimization Algorithm

```
1: Input: $M, Q_m, C_m, x_{ms}, \theta_n^{(l)}, R_{k,m}^{\text{d2d}}, R_{B,m}^{\text{mec}}, \tau_m, \mathbf{F}_0$, iter2, $\epsilon$
2: Initialization Lagrange multipliers: $\lambda, \mu_1, \dots, \mu_M$
3: for $t = 1$ to iter2 do
4:     Update Lagrange multipliers:
5:         $\lambda(t+1) = [\lambda(t) + \delta(t)(\sum_{m=1}^{\mathcal{M}} x_{m0} f_{m0} - \mathbf{F}_0)]^+$
6:     for $m = 1$ to $M$ do
7:         $\mu_m(t+1) = [\mu_m(t) + \delta(t)(\mathbf{F}_0 - f_{m0})]^+$
8:     end for
9:     Solve subproblem (24):
10:        Use KKT conditions and gradient descent to update $f_{m0}$ until convergence
11:            Compute the gradient of the objective function
12:            Use gradient descent to update $f_{m0}$ until KKT conditions are met
13:    Solve subproblem (16):
14:        Calculate $\alpha_m$ based on $\alpha_{mec}$ and $\alpha_{d2d}$
15:        Compute the corresponding delay $D_m$ and select the $\alpha_m$ with the minimum delay
16:    Check if convergence conditions are met:
17:        If the changes in Lagrange multipliers and resource allocation variables are less than $\epsilon$, break the loop
18: end for
```

4.3.1. D2D Offloading Mode

When $x_{mk} = 1, \forall m \in \mathcal{M}, \forall k \in \mathcal{K}$, the D2D offloading mode is selected, then the minimisation objective function is:

$$\mathcal{L}_{d2d} = \min_{\mathbf{\Theta}} \sum_{m=1}^{M} \frac{\alpha_m Q_m}{R_{k,m}^{\text{d2d}}} \tag{35}$$

For each phase shift θ_n, the gradient of the objective function $\mathcal{L}_{d2d}$ with respect to θ_n is computed:

$$\frac{\partial \mathcal{L}_{d2d}}{\partial \theta_n} = -\sum_{m=1}^{M} \frac{\alpha_m Q_m}{(R_{k,m}^{\text{d2d}})^2} \cdot \frac{\partial R_{k,m}^{\text{d2d}}}{\partial \theta_n} \tag{36}$$

In order to calculate $\frac{\partial R_{k,m}^{\text{d2d}}}{\partial \theta_n}$, first find the derivative of the interior term so that $A = \mathbf{G}_m^{(l)} \mathbf{\Theta}^{(l)} \mathbf{h}_{k,m}^{\text{D}(l)} + h_{k,m}$, then

$$\frac{\partial |A|^2}{\partial \theta_n} = \frac{\partial (A^* A)}{\partial \theta_n} = 2\Re \left(A^* \frac{\partial A}{\partial \theta_n} \right)$$

Since $\mathbf{\Theta} = \operatorname{diag}\left(e^{j\theta_1}, e^{j\theta_2}, \dots, e^{j\theta_N}\right)$, then $\frac{\partial A}{\partial \theta_n} = j e^{j\theta_n} [\mathbf{G}_d^{(l)} \mathbf{h}_{k,m}^{\text{D}(l)}]_n$, therefore:

$$\frac{\partial |A|^2}{\partial \theta_n} = 2\Re \left(A^* \cdot j e^{j\theta_n} [\mathbf{G}_d^{(l)} \mathbf{h}_{k,m}^{\text{D}(l)}]_n \right)$$

Next:

$$\frac{\partial R_{k,m}^{\text{d2d}}}{\partial \theta_n} = \frac{B_{\text{d2d}}}{\ln 2} \cdot \frac{p_d \cdot 2\Re \left(A^* \cdot j e^{j\theta_n} [\mathbf{G}_d^{(l)} \mathbf{h}_{k,m}^{\text{D}(l)}]_n \right)}{\left(\delta_{\text{d2d}}^2 + p_d |A|^2\right) \left(1 + \frac{p_d |A|^2}{\delta_{\text{d2d}}^2}\right)}$$

4.3.2. MEC Unloading Mode

When $x_{m0} = 1, \forall m \in \mathcal{M}, \forall k \in \mathcal{K}$, the MEC offloading mode is selected, then the minimisation objective function is:

$$\mathcal{L}_{mec} = \min_{\boldsymbol{\Theta}} \sum_{m=1}^{M} \frac{\alpha_m Q_m}{R_{B,m}^{\text{mec}}} \tag{37}$$

Let $B = \mathbf{G}_m^{(l)} \boldsymbol{\Theta}^{(l)} \mathbf{h}_{r,m}^{\mathbf{B}(l)} + h_{B,m}$ and compute the gradient of the objective function $\mathcal{L}_{mec}$ with respect to θ_n:

$$\frac{\partial \mathcal{L}_{mec}}{\partial \theta_n} = -\sum_{m=1}^{M} \frac{\alpha_m Q_m}{(R_{B,m}^{\text{mec}})^2} \cdot \frac{\partial R_{B,m}^{\text{mec}}}{\partial \theta_n} \tag{38}$$

$$\frac{\partial R_{B,m}^{\text{mec}}}{\partial \theta_n} = \frac{B_{\text{mec}}}{\ln 2} \cdot \frac{p_m \cdot 2\Re\left(B^* \cdot j e^{j\theta_n} [\mathbf{G}_m^{(l)} \mathbf{h}_{r,m}^{\text{B}(l)}]_n\right)}{(\delta_{\text{mec}}^2 + p_m |B|^2)\left(1 + \frac{p_m |B|^2}{\delta_{\text{mec}}^2}\right)}$$

According to the gradient descent method, update each θ_n:

$$\theta_n^{(t+1)} = \theta_n^{(t)} - \mu \cdot \left(\frac{\partial \mathcal{L}_{d2d}}{\partial \theta_n} + \frac{\partial \mathcal{L}_{mec}}{\partial \theta_n}\right) \tag{39}$$

This algorithm employs gradient descent to optimize the phase shifts of IRS reflective elements, enhancing wireless channel performance. It computes the gradients of the objective functions for D2D and MEC modes, updates the phase vector iteratively, and improves data transmission rates, significantly reducing system latency.

5. SIMULATIVE RESULTS AND ANALYSIS

This section presents the results of the IRS-assisted MEC-D2D collaborative offloading to reduce latency in cellular scenarios, including the properties of the proposed algorithm and the latency performance in simulated environmental scenarios.

In the simulation, the base station is set to $(0, 0)$, and the user devices and service devices are uniformly and randomly distributed in a circular area with a radius of 300 m, while the maximum distance of D2D association is set to $d_{\text{max}} = 50$ m. Other key parameters are listed in Table 1. In the following, simulation tests on the total system delay are performed for the variables of user devices (UDs), service devices (SeDs), the number of IRS configurations L, reflective elements N, and task size, respectively.

TABLE 1. Main parameters of the simulation [42].

Description	Parameter and Value
Location model	$R = 300$ m
	$d_{\text{max}} = 50$ m
Communication model	$B_{\text{mec}} = 10$ MHz, $B_{\text{d2d}} = 5$ MHz
	$\sigma_M = 10^{-6}, \sigma_D = 7 \times 10^{-7}$
	$Q_m = [250, 300]$ Kb
Computing model	$C_m = [700, 800]$ cycle/s
	$F_0 = 50 \times 10^9$, $f_m = 0.5 \times 10^9$ cycle/s
Convergence criterion	$\epsilon = 0.001$

To demonstrate the superiority of the performance of the proposed algorithmic system, the performance of the proposed scheme is evaluated by comparing it with the following four benchmark schemes:

(1) Greedy Edge: each UD performs task offloading via edge servers, no D2D offloading involved [40].

(2) Stochastic phase-shift: Algorithm 1 and Algorithm 2 are used to optimise the offloading mode, edge computing resource allocation and offloading ratio of user devices. The step of designing the IRS phase shift is also skipped, and the IRS phase shift is set randomly, obeying a uniform distribution in the range of [0, 2π).

(3) No IRS: Consider setting the reflection channel of IRS to 0. The offloading mode selection, edge computing resource allocation and offloading ratio of user devices are designed according to Algorithm 1 and Algorithm 2.

(4) Rate maximisation: the offloading decision is made in order to achieve the maximum total rate, so the offloading mode with the maximum rate is selected for each user device, and the total rate is calculated to include the sum of the transmission rates provided by all the tasks delivered by the cellular and D2D users in the network [41]. Edge computing resource allocation, offloading ratio, and optimal phase shift are implemented through Algorithm 2 and Algorithm 3.

Algorithm 3 IRS-GD Phase Optimization Algorithm

```
Input: $M, K, \alpha, Q_m, B_d, B_m, p_d, p_m, \delta_d, \delta_m, \mathbf{G_d}, \mathbf{G_m}, h_d, h_m$, iter3, $\mu, \epsilon$
Initialization Phase vector $\theta^{(0)}$, set the maximum number of iterations iter3 and learning rate $\mu$
for $t = 0$ to (iter3 $- 1$) do
    Compute the transmission rates $R_{k,m}^{\text{d2d}}$ and $R_{B,m}^{\text{mec}}$ for all users
    Compute the gradients of the objective functions $\mathcal{L}_{d2d}$ and $\mathcal{L}_{mec}$ for D2D and MEC modes based on equations (36) and (38)
    Update the phase vector according to Equation (39)
    if $\|\theta^{(t+1)} - \theta^{(t)}\| < \epsilon$ then
        Stop the iteration
    end if
end for
Output: Optimal phase vector $\theta^*$
```

This study mentions IRS-assisted communication, D2D and MEC collaborative computational offloading to reduce the total delay; therefore, it is important to verify the performance gain due to the number of IRS reflective elements as well as the SeDs of the service devices. Indeed, the computational and communication loads (in terms of computational offload requests) may vary considerably with increasing user density. In this case, the scalability and robustness of the proposed scheme need to be verified. Therefore, in Fig. 2, the performance of the scheme is analysed by varying the number of UDs while the number

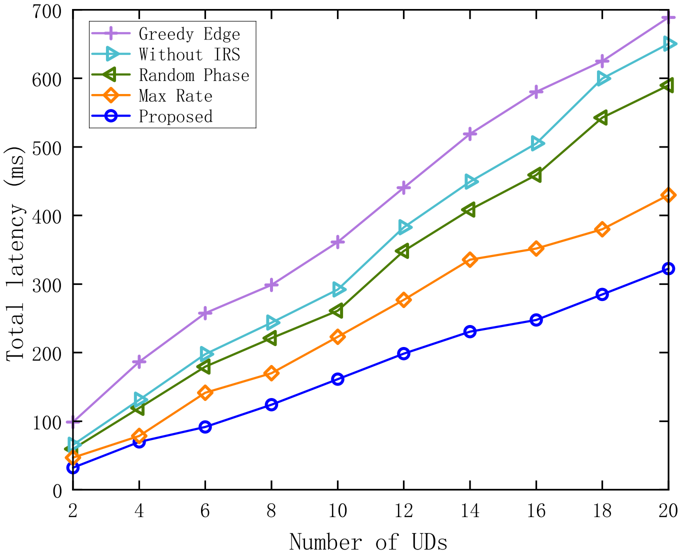

FIGURE 2. Total system delay as a function of the number of user devices (UDs).

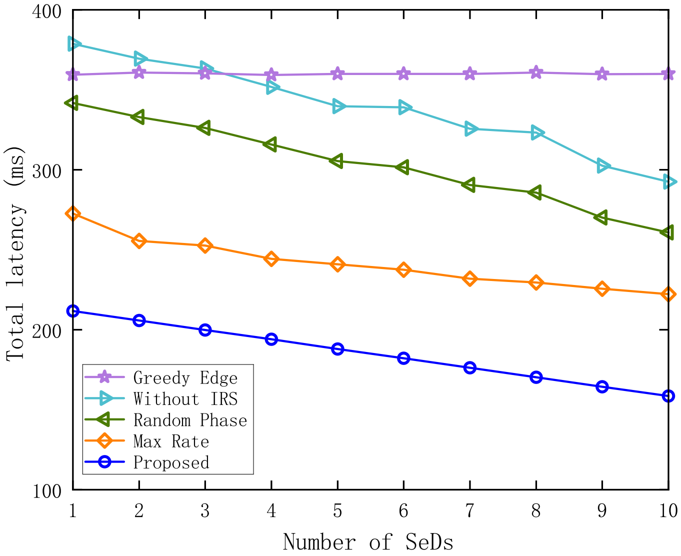

FIGURE 3. Total system delay as a function of the number of service devices (SeDs).

of SeDs, K, is set to 10, the number of IRS configurations, L, set to 10, and the number of reflective elements set to $N = 40$. Specifically, Fig. 2 plots the total latency of all the schemes and shows an increasing trend with the increase in the number of UDs. It is clear that as the number of users increases, the MEC computational resources allocated to the user devices decrease. Initially, the maximised rate algorithm and the algorithm in this paper achieve almost the same latency with a small number of UDs, which is due to the availability of sufficient resources on the edge servers. However, the increase in UDs leads to resource contention at the edge servers, and thus the collaborative offloading scheme achieves better performance. Comparing all the schemes, the algorithm in this paper has the smallest latency growth rate. In particular, it is about 58% lower than that without IRS, which itself confirms the effectiveness of the task offloading strategy of this algorithm to reduce the execution latency by efficiently utilising the MEC and D2D computational resources.

The number of user devices, IRS configurations, and reflective elements are set to $M = 10$, $L = 10$, and $N = 40$, respectively. The total delay performance curves for the five schemes are shown in Fig. 3. From Fig. 3, it can be seen that the total delay of the system decreases with the increase of the SeDs of the service devices, and the performance of the algorithm in this paper outperforms that of other similar schemes. When the number of SeDs is 10, the total delay of this paper's algorithm is 28.8% lower than the Max Rate scheme, 39.2% lower than the Random Phase scheme, and 45.8% lower than without RIS scheme, and note that the Greedy Edge algorithm does not have SeDs for auxiliary task offloading, thus the change in the number of SeDs has no effect on this algorithm. As the number of SeDs users increases, some UDs perform task offloading via SeDs to alleviate the resource constraints of the edge servers, and this collaborative task offloading strategy effectively utilises the D2D-MEC computational resources to reduce the total system latency.

The numbers of user devices and service devices are set to $M = 10$ and $K = 10$, respectively, and the number of reflective elements is set to $N = 40$. Specifically, Fig. 4 compares the total system delay with the number of IRSs. It can be seen that the total system delay decreases with the increase in the number of IRS units L in all scenarios except the without IRS deployment scenario. This indicates that increasing the number of IRSs improves the system performance, mainly by enhancing the channel quality and signal transmission efficiency. As can be seen in Fig. 4, with the number of IRSs set to 10, the total system delay of this paper's algorithm is 30.46% lower than that of the Max Rate scheme, 38.76% lower than that of the Random Phase scheme, 49.49% lower than that of the Without RIS scheme, and 57.38% lower than that of the Greedy Edge scheme; however, with further increase in the L, the delay reduction gradually slows down, showing diminishing benefits of performance enhancement, and when the number of IRS units is increased to a certain level, the further reduction of delay is no longer as pronounced as it was initially. Although increasing the number of IRS units can optimise the system performance within a certain range, its improvement effect on the total system delay gradually saturates when L exceeds a certain threshold.

The number of user devices, service devices, and IRS configurations are set to $M = 10$, $K = 10$, and $L = 10$, respectively. Fig. 5 evaluates the effect of the number of reflective elements of the IRS on the total system delay. Firstly, the algorithm of this paper outperforms the other research schemes in terms of performance as shown in Fig. 2. Comparing the random phase-shift and the case with no IRS yields the lowest total system delay as shown in Fig. 5 and once again validates the effectiveness of D2D-MEC co-unloading with IRS assistance. As the number of IRS reflective elements increases, the effective signal-to-noise ratio (SNR) of the user-to-BS and user-to-D2D communication paths increases, which reduces the upload latency of the user device to the edge server and inter-user communica-

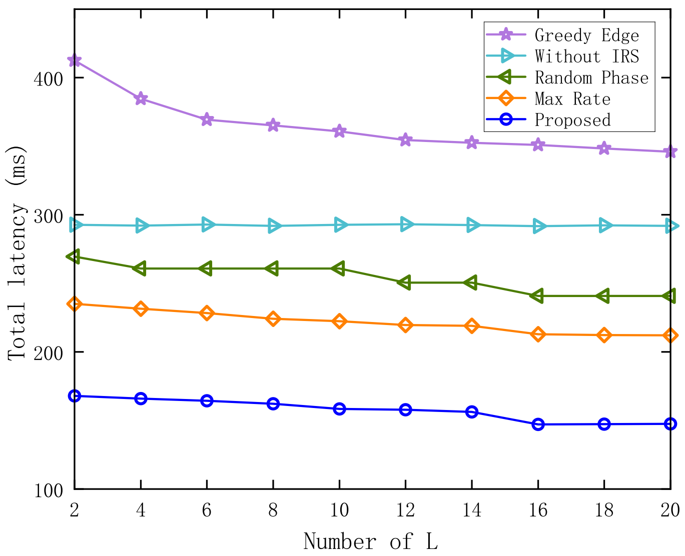

FIGURE 4. Impact of the number of IRS configurations (L) on total system delay.

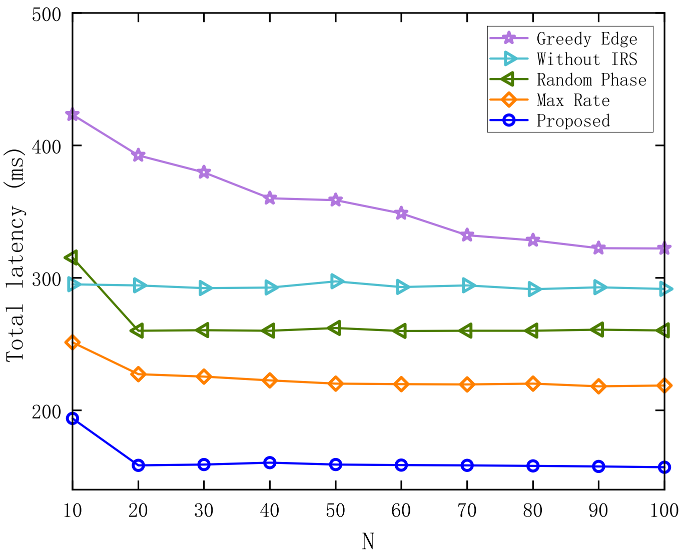

FIGURE 5. Total system delay as a function of the number of IRS reflective elements (N).

tion. However, the improvement in system performance (i.e., delay reduction) due to an increase in the number of reflective elements is compromised by the fact that once a sufficient number of IRS reflective elements have been deployed, the wireless channel is enhanced, with limited incremental gains from the additional elements. As the number of IRS reflective elements N increases, the total delay of the algorithms in this paper, Max Rate and Random Phase algorithms decreases, but after $N = 20$, since the wireless channel is already sufficiently enhanced by the available IRS reflective elements, and the performance gain reaches saturation, the total delay performance improvement of the system is not significant when $N > 20$. In particular, the performance gain is very significant when the offloading resources are limited, e.g., the total delay of the Greedy Edge algorithm is reduced by 23.8% from 423 ms to 322 ms when N is increased from 10 to 100. Note that the performance fluctuations given by the without IRS algorithm are due to the dynamic nature of the wireless network, and the change in N has no effect on the algorithm.

The numbers of user devices and service devices are set to $M = 10$ and $K = 10$, respectively, and the number of reflective elements is set to $N = 40$. The effect of task size on the total delay performance is shown in Fig. 6. Firstly, the proposed algorithm significantly outperforms other schemes. Secondly, it can be seen from Fig. 6 that the total system delay of all the studied schemes increases with the increase of task size. By comparing the performance trends of Random Phase and Without IRS, it can be concluded that if IRS assists in reducing the task transmission delay, compared with Greedy Edge, the algorithm of this paper optimises the offloading decision, MEC resource allocation and offloading ratio to significantly reduce the total system delay, which also verifies the effectiveness of the algorithmic scheme of this paper. Compared to other studied schemes, the Greedy Edge scheme has worse performance because the total uplift rate is the highest as the task size increases. The reason for this is that the total computational resources of the edge server are tight as the task size increases, and therefore, the total latency under this algorithm increases faster than the schemes of other studies.

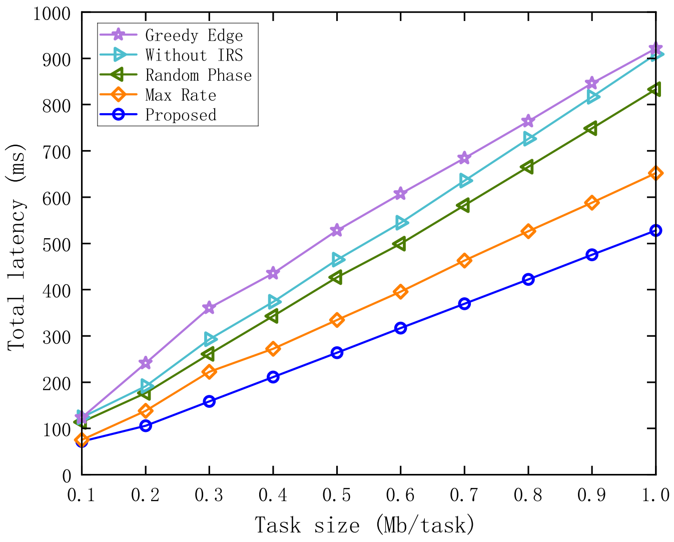

FIGURE 6. Impact of task size on total system delay for different schemes.

Our simulation model assumes ideal conditions, such as perfect phase adjustments for the IRS and accurate channel state information (CSI), which may not fully reflect real-world deployment scenarios. For instance, hardware imperfections in IRS or errors in CSI estimation could degrade performance. Additionally, the model does not account for the computational complexity or energy consumption of large-scale IRS deployments, which could become significant in practical implementations. Future studies could incorporate these factors and explore the robustness of the proposed algorithms under dynamic and imperfect conditions, such as user mobility and environmental changes.

6. CONCLUSIONS

In this study, the significant advantages of the proposed IRS-assisted MEC-D2D cooperative offloading algorithm in reducing system delay in cellular scenarios are verified through simulation results. The experimental results show that the algorithm is able to minimise the total system delay in various test scenarios, especially when the number of user devices increases, demonstrating good scalability and robustness. The algorithm is able to significantly reduce the task execution delay by rationally utilising the computational resources for mobile edge computing (MEC) and device-to-device (D2D) communication. In particular, the introduction of the Intelligent Reflective Surface (IRS) greatly improves the overall performance of the system. By increasing the number of reflective elements, IRS effectively improves the signal-to-noise ratio of the channel, which further reduces the data upload delay. Simulation results show that IRS plays a key role in enhancing the communication efficiency of D2D-MEC cooperative offloading. Compared with other benchmark schemes, the algorithm proposed in this paper performs well under all test conditions, especially in scenarios where the number of serving devices increases, and effectively reduces the total system delay through better task offloading strategies and resource allocation. In addition, the algorithm is still able to maintain low latency when dealing with large-scale tasks, which fully demonstrates its adaptability and effectiveness in complex network environments.

Overall, the IRS-assisted MEC-D2D cooperative offloading algorithm proposed in this paper performs well in reducing system delay, improving communication efficiency and resource utilisation, and demonstrates a wide range of application potentials and research values.

The simulation models in this study assume ideal conditions, such as perfect IRS phase alignment and accurate CSI, which may not reflect real-world scenarios. Factors like IRS hardware defects, CSI estimation errors, computational complexity, and energy consumption in large-scale deployments are not considered. Future research could address these limitations by exploring algorithm robustness under dynamic and imperfect conditions, including user mobility and environmental changes.

ACKNOWLEDGEMENT

This work was supported by Natural Science Foundation of Fujian Province (Grant No. 2022J011276, and No. 2023I0044), High-level Talent Project of Xiamen University of Technology (Grant No. YKJ22030R), and Postgraduate Science and Technology Innovation Project of Xiamen University of Technology (Grant No. YKJCX2022122).

REFERENCES

[1] Zhang, K., X. Gui, D. Ren, J. Li, J. Wu, and D. Ren, "Survey on computation offloading and content caching in mobile edge networks," *Journal of Software*, Vol. 30, No. 8, 2491–2516, 2019.

[2] Liu, Y., X. Liu, X. Mu, T. Hou, J. Xu, M. D. Renzo, and N. Al-Dhahir, "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Communications Surveys & Tutorials*, Vol. 23, No. 3, 1546–1577, 2021.

[3] Wu, Q. and R. Zhang, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," *IEEE Communications Magazine*, Vol. 58, No. 1, 106–112, 2020.

[4] Ma, X., A. Zhou, S. Zhang, and S. Wang, "Cooperative service caching and workload scheduling in mobile edge computing," in *IEEE INFOCOM 2020 — IEEE Conference on Computer Communications*, 2076–2085, Toronto, ON, Canada, 2020.

[5] Gao, B., Z. Zhou, F. Liu, F. Xu, and B. Li, "An online framework for joint network selection and service placement in mobile edge computing," *IEEE Transactions on Mobile Computing*, Vol. 21, No. 11, 3836–3851, 2021.

[6] Yang, G., L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via markov decision process in mobile-edge computing," *IEEE Internet of Things Journal*, Vol. 8, No. 4, 2483–2493, 2020.

[7] Chen, C.-L., C. G. Brinton, and V. Aggarwal, "Latency minimization for mobile edge computing networks," *IEEE Transactions on Mobile Computing*, Vol. 22, No. 4, 2233–2247, 2021.

[8] Liu, T., Y. Zhang, Y. Zhu, W. Tong, and Y. Yang, "Online computation offloading and resource scheduling in mobile-edge computing," *IEEE Internet of Things Journal*, Vol. 8, No. 8, 6649–6664, 2021.

[9] Xu, X., Q. Wu, L. Qi, W. Dou, S.-B. Tsai, and M. Z. A. Bhuiyan, "Trust-aware service offloading for video surveillance in edge computing enabled internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 3, 1787–1796, 2020.

[10] Lai, P., Q. He, G. Cui, F. Chen, M. Abdelrazek, J. Grundy, J. Hosking, and Y. Yang, "Quality of experience-aware user allocation in edge computing systems: A potential game," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, 223–233, Singapore, 2020.

[11] Zhou, Z., Z. Wang, H. Yu, H. Liao, S. Mumtaz, L. Oliveira, and V. Frascolla, "Learning-based URLLC-aware task offloading for internet of health things," *IEEE Journal on Selected Areas in Communications*, Vol. 39, No. 2, 396–410, 2020.

[12] Li, G., M. Zeng, D. Mishra, L. Hao, Z. Ma, and O. A. Dobre, "Latency minimization for IRS-aided NOMA MEC systems with WPT-enabled IoT devices," *IEEE Internet of Things Journal*, Vol. 10, No. 14, 12 156–12 168, 2023.

[13] Li, N., W. Hao, F. Zhou, S. Yang, and N. Al-Dhahir, "Min-max latency optimization for IRS-aided cell-free mobile edge computing systems," *IEEE Internet of Things Journal*, Vol. 11, No. 5, 8757–8770, 2024.

[14] Liu, K., F. Lin, Y. Zhao, and J. Zhang, "Deep reinforcement learning optimization algorithm designed for IRS-assisted edge computing," in *2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT)*, 835–840, Qingdao, China, 2023.

[15] Cheng, Y., C. Liang, Q. Chen, and F. R. Yu, "Energy-efficient D2D-assisted computation offloading in NOMA-enabled cognitive networks," *IEEE Transactions on Vehicular Technology*, Vol. 70, No. 12, 13 441–13 446, 2021.

[16] Budhiraja, I., S. Tyagi, S. Tanwar, N. Kumar, and J. J. P. C. Rodrigues, "DIYA: Tactile internet driven delay assessment NOMA-based scheme for D2D communication," *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 12, 6354–6366, 2019.

[17] Zhou, H., T. Wu, H. Zhang, and J. Wu, "Incentive-driven deep reinforcement learning for content caching and D2D offloading," *IEEE Journal on Selected Areas in Communications*, Vol. 39, No. 8, 2445–2460, 2021.

[18] Hamdi, M., A. B. Hamed, D. Yuan, and M. Zaied, "Energy-efficient joint task assignment and power control in energy-harvesting D2D offloading communications," *IEEE Internet of Things Journal*, Vol. 9, No. 8, 6018–6031, 2021.

[19] Saleem, U., Y. Liu, S. Jangsher, Y. Li, and T. Jiang, "Mobility-aware joint task scheduling and resource allocation for cooperative mobile edge computing," *IEEE Transactions on Wireless Communications*, Vol. 20, No. 1, 360–374, 2020.

[20] Peng, J., H. Qiu, J. Cai, W. Xu, and J. Wang, "D2D-assisted multi-user cooperative partial offloading, transmission scheduling and computation allocating for MEC," *IEEE Transactions on Wireless Communications*, Vol. 20, No. 8, 4858–4873, 2021.

[21] Wang, Y., K. Wang, H. Huang, T. Miyazaki, and S. Guo, "Traffic and computation co-offloading with reinforcement learning in fog computing for industrial applications," *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 2, 976–986, 2018.

[22] Tang, J., H. Tang, X. Zhang, K. Cumanan, G. Chen, K.-K. Wong, and J. A. Chambers, "Energy minimization in D2D-assisted cache-enabled Internet of Things: A deep reinforcement learning approach," *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 8, 5412–5423, 2019.

[23] Sun, M., X. Xu, Y. Huang, Q. Wu, X. Tao, and P. Zhang, "Resource management for computation offloading in D2D-aided wireless powered mobile-edge computing networks," *IEEE Internet of Things Journal*, Vol. 8, No. 10, 8005–8020, 2020.

[24] He, Y., J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Transactions on Wireless Communications*, Vol. 18, No. 3, 1750–1763, 2019.

[25] Tan, L., Z. Kuang, L. Zhao, and A. Liu, "Energy-efficient joint task offloading and resource allocation in OFDMA-based collaborative edge computing," *IEEE Transactions on Wireless Communications*, Vol. 21, No. 3, 1960–1972, 2021.

[26] Li, G. and J. Cai, "An online incentive mechanism for collaborative task offloading in mobile edge computing," *IEEE Transactions on Wireless Communications*, Vol. 19, No. 1, 624–636, 2019.

[27] Qian, C., G. Zhao, and H. Luo, "Game theory based D2D collaborative offloading for workflow applications in mobile edge computing," in *2022 IEEE International Conference on Web Services (ICWS)*, 276–285, Barcelona, Spain, 2022.

[28] Khan, M. A., E. Baccour, A. Erbad, R. Hamila, and M. Hamdi, "CODE: Computation offloading in D2D-edge system for video streaming," *IEEE Systems Journal*, Vol. 17, No. 3, 4014–4025, 2023.

[29] Li, K., X. Wang, Q. He, M. Yang, M. Huang, and S. Dustdar, "Task computation offloading for multi-access edge computing via attention communication deep reinforcement learning," *IEEE Transactions on Services Computing*, Vol. 16, No. 4, 2985–2999, 2023.

[30] Wu, Q. and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Transactions on Wireless Communications*, Vol. 18, No. 11, 5394–5409, 2019.

[31] Wu, Q. and R. Zhang, "Beamforming optimization for intelligent reflecting surface with discrete phase shifts," in *ICASSP 2019 — 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7830–7833, Brighton, UK, 2019.

[32] Guo, H., Y.-C. Liang, J. Chen, and E. G. Larsson, "Weighted sum-rate maximization for intelligent reflecting surface enhanced wireless networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 1–6, Waikoloa, HI, USA, 2019.

[33] Ye, J., S. Guo, and M.-S. Alouini, "Joint reflecting and precoding designs for SER minimization in reconfigurable intelligent surfaces assisted MIMO systems," *IEEE Transactions on Wireless Communications*, Vol. 19, No. 8, 5561–5574, 2020.

[34] Pan, C., H. Ren, K. Wang, W. Xu, M. Elkashlan, A. Nallanathan, and L. Hanzo, "Intelligent reflecting surface for multicell MIMO communications," *ArXiv Preprint ArXiv:1907.10864*, 2019.

[35] Zhou, G., C. Pan, H. Ren, K. Wang, and A. Nallanathan, "Intelligent reflecting surface aided multigroup multicast MISO communication systems," *IEEE Transactions on Signal Processing*, Vol. 68, 3236–3251, 2020.

[36] Yang, Y., B. Zheng, S. Zhang, and R. Zhang, "Intelligent reflecting surface meets OFDM: Protocol design and rate maximization," *IEEE Transactions on Communications*, Vol. 68, No. 7, 4522–4535, 2020.

[37] Muñoz, O., A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Transactions on Vehicular Technology*, Vol. 64, No. 10, 4738–4755, 2014.

[38] Zhang, H., X. Liu, Y. Xu, D. Li, C. Yuen, and Q. Xue, "Partial offloading and resource allocation for MEC-assisted vehicular networks," *IEEE Transactions on Vehicular Technology*, Vol. 73, No. 1, 1276–1288, 2024.

[39] Morris, S., D. Oyama, and S. Takahashi, "Implementation via information design in binary-action supermodular games," *Econometrica*, Vol. 92, No. 3, 775–813, 2024.

[40] Dai, X., Z. Xiao, H. Jiang, M. Alazab, J. C. S. Lui, S. Dustdar, and J. Liu, "Task co-offloading for D2D-assisted mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 1, 480–490, 2022.

[41] Ioannou, I., V. Vassiliou, C. Christophorou, and A. Pitsillides, "Distributed artificial intelligence solution for D2D communication in 5G networks," *IEEE Systems Journal*, Vol. 14, No. 3, 4232–4241, 2020.

[42] Bai, T., C. Pan, Y. Deng, M. Elkashlan, A. Nallanathan, and L. Hanzo, "Latency minimization for intelligent reflecting surface aided mobile edge computing," *IEEE Journal on Selected Areas in Communications*, Vol. 38, No. 11, 2666–2682, 2020.