Joint Optimization of Signal Design and Resource Allocation in Wireless D2D Edge Computing

Junghoon Kim*, Taejoon Kim†, Morteza Hashemi†, Christopher G. Brinton*, and David J. Love*

*Electrical and Computer Engineering, Purdue University, IN, USA
†Electrical Engineering and Computer Science, University of Kansas, KS, USA
Email: {*kim3220, cgb, djlove}@purdue.edu, †{taejoonkim, mhashemi}@ku.edu

Abstract—In this paper, we study the distributed computational capabilities of device-to-device (D2D) networks. A key characteristic of D2D networks is that their topologies are reconfigurable to cope with network demands. For distributed computing, resource management is challenging due to limited network and communication resources, leading to inter-channel interference. To overcome this, recent research has addressed the problems of wireless scheduling, subchannel allocation, power allocation, and multiple-input multiple-output (MIMO) signal design, but has not considered them jointly. In this paper, unlike previous mobile edge computing (MEC) approaches, we propose a joint optimization of wireless MIMO signal design and network resource allocation to maximize energy efficiency. Given that the resulting problem is a non-convex mixed integer program (MIP) which is prohibitive to solve at scale, we decompose its solution into two parts: (i) a resource allocation subproblem, which optimizes the link selection and subchannel allocations, and (ii) MIMO signal design subproblem, which optimizes the transmit beamformer, transmit power, and receive combiner. Simulation results using wireless edge topologies show that our method yields substantial improvements in energy efficiency compared with cases of no offloading and partially optimized methods and that the efficiency scales well with the size of the network.

I. INTRODUCTION

The number of wireless devices is now over 8.6 billion, and with the advent of new 5G-and-beyond technologies it is expected to grow to 12.3 billion by 2022 [1]. Many of these devices will be data-processing-capable nodes that facilitate the rapidly growing data-intensive applications running at the network edge, e.g., social networking video streaming and distributed data analytics. While some devices are occupied processing computationally-intensive applications, e.g., facial recognition, location-based augmented/virtual reality (AR/VR), and online 3D gaming [2]–[4], it may be desirable for them to offload their data to devices with underutilized resources. Traditionally, cloud computing architectures have been adopted for such data intensive applications, e.g., Amazon Web Services and Microsoft Azure, but the exponential rise in data generation at the edge means that D2D is becoming centralizing architectures infeasible for providing latency-sensitive quality of service at scale [1].

As a current trend in wireless networks is reducing cell sizes [5], many 5G networks will be dense with short distances, forming a magnitude of smaller subnets [6]. Networks of small subnets combined with improved computational and storage capabilities of edge devices have enabled mobile edge computing (MEC), a recently popularized line of research. At a high level, MEC leverages radio access networks (RANs) to boost computing power in close proximity to end-users, thus enabling the users to offload their computations to an edge server (central processor) [7]–[11] as shown in Fig. 1(a). In an MEC architecture, the edge servers are assumed to be predetermined, as they have high-performance computing units which can process large amounts of computationally intensive tasks efficiently. This MEC concept was extended to a helper-edge server architecture to exploit the computation resources of idle devices [12], [13].

The current trend in distributed computing, though, is a migration to more decentralized architectures. This is due to the fact that all edge devices can take part in data offloading, given the advances in 5G communication technologies in conjunction with improved computational capabilities of individual devices. For this reason, device-to-device (D2D) network architectures that were previously studied in 4G LTE standards have the potential of providing distributed computing at scale. Unlike the MEC system in Fig. 1(a), distributed computing in the D2D network of Fig. 1(b) will have a more complicated topology given the additional coordination requirements.

Judicious management of network and communication resources is essential for efficient D2D network because wireless transmissions among edge nodes participating in data offloading incur inevitable inter-channel interference. In particular, joint management of network and communication resources with proper coordination mechanisms among the distributed nodes is critical. Network and communication resource man-
management methodologies must be developed to adapt to configurable network topologies in D2D networks.

A. Related work

Previous research for edge computing considering network and communication resources has been largely conducted in MEC system. However, the network and communication aspects have mainly been studied separately. In particular, wireless scheduling [14], [15], interference management through allocation of different subchannels [16], power allocation [12], [13], [17], and multiple-input multiple-output (MIMO) signal design [18]–[20] have each been independently addressed. In reality, there is a need to understand how MEC systems should be designed to jointly optimize link selection and resource management in D2D scenarios.

On the other hand, some research on D2D networks has addressed resource management such as subchannel allocation [21] and power allocation [22], or jointly both [23]–[25]. Also, MIMO signal design is considered in D2D communication underlaying cellular networks in order to mitigate inter channel interferences [26], [27], or to maximize overall utilities for cellular users communicating with base station and D2D [28]. However, for the above works, link selection is not considered. Although some recent works addressed the link selection [29] and also with power allocation [30] in D2D network, many other resources were not jointly addressed yet.

B. Motivations and Contributions

Motivated by this, we develop an optimization methodology for network and communication resource management in wireless D2D networks, considering the problems of link selection, subchannel allocation, power allocation, and MIMO signal design jointly. Our approach is driven by two design principles:

1. **Link selection for energy minimization.** The network topology should be defined so that energy efficiency is maximized. Given a reconfigurable network topology, link selection design between nodes in D2D network is more complicated than offloading decisions between nodes and a server in MEC systems.

2. **Joint resource allocation and MIMO signal design.** The link selection is accompanied with efficient management of available resources. We jointly consider network resources that involve topology configuration, through link selection and frequency assignment, and communication resources that involve transmission power and MIMO antennas. Especially, with a large number of links and limited subchannels, MIMO signal design, i.e., beamforming, is essential to mitigate inevitable inter-channel interferences for robust data transfer and optimization.

The contributions of this paper are as follows:

- We formulate a novel optimization problem for D2D edge computing networks that minimizes the total energy consumption required to process given data samples at the D2D edge nodes, based on a framework for joint wireless MIMO signal design, resource allocation (subchannel and power allocation), and link selection (Section II).

  - We decompose the integrated framework into two subproblems to make it computationally tractable: one for network resource allocation and the other for MIMO signal design. This facilitates the network resource optimization with integer variables and MIMO signal design optimization with non-integer variables (Section II).

  - In the optimization of network resource allocation, we jointly determine the link with subchannel allocation. For solving this problem, we modify our objective function and exploit an integer programming. We propose a greedy algorithm as another solution to reduce the computational complexity (Section III).

  - With respect to MIMO signal design optimization, we determine the transmit beamformer, transmit power and receive combiner by exploiting interference coordination techniques for MIMO systems. (Section III). The energy minimization problem is converted to the weighted minimum mean square error (WMMSE) minimization problem for tractability.

  - Extensive simulations are presented to evaluate the integrated framework and our proposed algorithms (Section IV).

II. SYSTEM MODEL AND OPTIMIZATION FORMULATION

In this section, we formulate the problem of jointly optimizing MIMO signal design and network resource allocation to minimize the total energy consumption in device-to-device (D2D) network. We first outline preliminaries of the underlying wireless D2D network model in Sec. II-A and then present the optimization in Sec. II-C.

A. Wireless Device-to-device (D2D) Network Model

Fig. 2 demonstrates a wireless D2D network model of $K$ nodes that can transmit using multiple antennas.

**Data transfer.** Consider a multi-point to multi-point wireless channel where a group of $K$ nodes can simultaneously transfer...
(offload) data to one another. With \( k, k' = 1, ..., K \), we define \( a_{k,k'} \in \{0,1\} \) as the binary variable describing whether node \( k \) transmits data to node \( k' \), and \( I_k \) (in bits) as the length of data present at node \( k \) for potential transfer, which is a task eligible for distributed processing. That is, \( a_{k,k'} = 1 \) implies that the node \( k \) sends the data \( I_k \) to the node \( k' \) while \( a_{k,k'} = 0 \) means the node \( k \) is holding it for local processing. Note that data transfer is not allowed from a node to itself, i.e., \( a_{k,k} = 0 \) for all \( k \).

We assume that each task should be processed as a whole, i.e., the task is divisible. In other words, If \( a_{k,j} = 1 \), then \( a_{k,j'} = 0 \), since the node \( k \) will not transmit its data to two or more nodes. We denote the set of link pairs attempting to transmit data as

\[ \mathcal{A} = \{(k,k') : a_{k,k'} = 1\}, \quad (1) \]

with \((k,k')_n = (k_n,k_n')\) denoting the \( n \)-th pair of transmit and receive node where \( n = 1, ..., L \), and \( L = |\mathcal{A}| \) being the total number of link pairs. For example, if we have the connected link from node 3 to node 1 and it is the first link pair \((n=1)\), then \((k_1,k_1') = (3,1)\).

**Frequency subchannels.** To facilitate multiple access transmission, the pairs in \( \mathcal{A} \) will leverage multiple frequency subchannels \( \mathcal{B} = \{b_1, ..., b_S\} \), where \( S \) is the total number of subchannels, each having the same bandwidth \( W \). In general, the subchannels will be limited, i.e., \( S \ll L \), meaning subchannels will be shared across multiple links. With \( m_i \in \{0,1\}^L \) as the binary vector of which active links are using the \( i \)-th subchannel, \( i = 1, ..., S \), the \( n \)-th element \( m_{n,i} \in \{0,1\} \) for \( n = 1, ..., L \) denotes whether the \( n \)-th link uses subchannel \( i \) \((m_{n,i} = 1)\) or not \((m_{n,i} = 0)\). For example, suppose we have three links and two subchannels. Then \( L = 3 \) and \( S = 2 \). One possible subchannel allocation is \( m_1 = (1,0,0)^T \) and \( m_2 = (0,1,1)^T \), meaning the first link is using the first subchannel \( b_1 \) and the second and third links are using the second subchannel \( b_2 \). We also denote \( M = [M_{m,n}] = \{0,1\}^{L \times L} \) as the binary subchannel occurrence matrix \( M = \sum_{i=1}^{S} m_i m_i^T \); if \( M_{m,n} = \sum m_{m,i} m_{n,i} = 1 \), then links \( n \) and \( m \) share the same subchannel, and they interfere with each other, and if \( M_{m,n} = 0 \) they do not.

**Communication demand.** The time it takes for the data transfer from node \( k \) to \( k' \) can be quantified as \( I_k/R_{k,k'} \), where \( R_{k,k'} \) is the maximum achievable data rate over the wireless channel. If a node \( k \) has transmit power \( P_k \), then we can calculate the total energy consumed for data transfer over the set of link pairs \( \mathcal{A} \) as in (2). Note that communication only occurs where the link pairs are connected \( a_{k,k'} = 1 \), i.e., \((k_n,k_n')\) pairs for \( n = 1, ..., L \). This means that we can write the communication energy \( E_M \) as

\[ E_M = \sum_{(k,k') \in \mathcal{A}} P_k \cdot \frac{I_k}{R_{k,k'}} = \sum_{n=1}^{L} P_{k_n} \cdot \frac{I_{k_n}}{R_{k_n,k_n'}}. \quad (2) \]

A lower \( E_M \) means the device has a lower energy requirement. Assuming each node \( k \) has \( N_{TX}^k \) transmit antennas and \( N_{RX}^k \) receive antennas, we consider the case of multiple-input multiple-output (MIMO) links between nodes. The channel matrix from transmitter \( k_n \) to receiver \( k_n' \) is then denoted \( H_{k_n,k_n'} \in \mathcal{C}^{N_{RX}^k \times N_{TX}^k} \). Transmit beamforming at \( k_n \) is accomplished by constructing the beampformer \( \sqrt{P_{k_n}} e_{k_n} \in \mathcal{C}^{N_{TX}^k} \) where \(|e_{k_n}| = 1\). The receiver \( k_n' \) behaves similarly, constructing the combiner \( z_{k_n'} \in \mathcal{C}^{N_{RX}^k} \) with \(|z_{k_n'}| = 1\). Using these assumptions, the maximum data rate on link \( n \) \([31]\) is

\[ R_{k_n,k_n'} = W \log_2 \left( 1 + \frac{P_{k_n} |z_{k_n'}^H H_{k_n,k_n'} f_{k_n'}|^2}{\sum_{m \neq n} M_{m,n} P_{k_m} |z_{k_m}^H H_{k_m,k_n'} f_{k_m}|^2 + \sigma^2} \right) \quad (3) \]

where \( W \) is the bandwidth of frequency subchannel, a superscript \( H \) denotes the conjugate transpose, and the noise power \( \sigma^2 = 1 \). Note that the data rate in (3) considers interference from other links \( m \neq n \) sharing the same subchannel allocation.

**Processing demand.** There are two possibilities for processing the data \( I_k \): the local processing and offloaded processing. The local processing means that the node \( k \) processes its own data \( I_k \), i.e., \( a_{k,k} = 0 \) for all \( k' \). However, the offloaded processing means that the node \( k \) transfers the data \( I_k \) to another node \( k' \) which then processes it, i.e., \( a_{k,k'} = 1 \) for some \( k' \). We denote the processing power of node \( k \) as \( F_k \), which is practically a function of CPU usage and energy efficiency of device \([32]\). Using this, the energy requirement in each of these cases is

\[ E_k = F_k \cdot \frac{I_k}{C_k}, \quad E_{k,k'} = P_k \cdot \frac{I_k}{R_{k,k'}} + F_{k'} \cdot \frac{I_{k'}}{C_{k'}}, \quad (4) \]

where \( C_k \) is the computation speed of node \( k \) and \( E_{k,k'} \) includes the energy of communication from the node \( k \) to \( k' \). With this, we can calculate the total energy consumed for data processing in an entire network over \( K \) nodes as

\[ E_P = \sum_{k=1}^{K} \left( \sum_{k' \neq k} a_{k,k'} \cdot \left( P_k \cdot \frac{I_k}{R_{k,k'}} + F_{k'} \cdot \frac{I_{k'}}{C_{k'}} \right) \right) + \left( 1 - \sum_{k' \neq k} a_{k,k'} \right) \cdot F_k \cdot \frac{I_k}{C_k}. \quad (5) \]

**B. Example for Wireless D2D Network Model**

Ideally, the total energy consumption with data offloading in wireless D2D network should be minimized. To illustrate this, consider a three node example where at most one link can be connected. For \( k = 1, 2, 3 \), each node is assumed to have the same length of data \( I_k = 10 \) Mbits, computing power \( F_k = 1 \) Watt, transmission power \( P_k = 1 \) Watt. The transmission data rate is assumed as \( R_{k,k'} = 2 \) Mbits/sec for \( k \neq k' \). However, we consider different computing speeds, \( C_1 = 10, C_2 = 2, \) and \( C_3 = 1 \) Mbits/sec. Using these definitions, the total energy consumption for local computation should be \( E_{P,local} = F_1 (I_1/C_1) + F_2 (I_2/C_2) + F_3 (I_3/C_3) = 16 \) Joules.

The objective is to minimize total energy to find best link pair among \( 6 \) possible links: \((1,2), (1,3), (2,1), (2,3), (3,1), (3,2)\).
and (3, 2). It is noteworthy that the link must be properly selected to minimize the total energy. Suppose we have link (3, 1). Then the total energy including transmission energy should be \( E_{P,31} = F_1(I_1/C_1) + F_2(I_2/C_2) + (P_3(I_3/R_{3,1}) + F_1(I_3/C_1)) = 17 \) Joules, which is larger than the local computation energy \( E_{P,local} \). This implies that if transmission channel is available and energy-efficient, we can exploit the communication to minimize total energy for data processing. If we consider link (2, 1) instead of link (3, 1), the total energy would be \( E_{P,21} = F_1(I_1/C_1) + (P_2(I_2/R_{2,1}) + F_1(I_2/C_1)) + F_3(I_3/C_3) = 17 \) Joules, which is larger than local computation energy \( E_{P,local} \). Further, we are aware that link (3, 1) is the best link pair for minimizing total energy consumption with the relationship \( E_{P,31} < E_{P,local} < E_{P,21} \) for any \((k, k') \neq (3, 1)\), where \( E_{P,kk'} \) is the total energy consumption with the link pair \((k, k')\). Therefore, the link selection optimization is essential for minimizing total energy consumption. Although this example assumed only one link and identical data rates of the nodes, we are aware that minimizing the inter-channel interferences in many links environment are critical to data rate maximization for selected links, which will lead to total energy minimization.

### C. Optimization and Decomposition

We are interested in the combination of MIMO signal design parameters – i.e., transmit powers \( \{P_k\} \), transmit beamformers \( \{f_k\} \), and receive combiners \( \{z_k'\} \) – and network resource allocation parameters – i.e., transmission link pairs \( \{a_{k,k'}\} \) and subchannel allocations \( \{m_{n,i}\} \) – that will lead to processing the network data \( I_1, \ldots, I_K \) with minimum energy consumption. We can formulate the following optimization problem:

\[
\text{minimize} \quad E_P
\]

subject to

\[
R_{k,k'} \text{ defined in (3) } \forall a_{k,k'} = 1,
\]

\[
||f_k||_2 = 1 \quad \forall k, \quad ||z_k'||_2 = 1 \quad \forall k',
\]

\[
\sum_{k=1}^{K} P_k \leq P,
\]

\[
0 \leq \sum_{j=1}^{K} a_{k,j} + \sum_{j=1}^{K} a_{j,k} \leq 1 \quad \forall k,
\]

\[
a_{k,k'} \in \{0, 1\} \quad \forall k, k', \quad a_{k,k} = 0 \quad \forall k,
\]

\[
\sum_{(k,k') \notin \mathcal{X}} a_{k,k'} = 0,
\]

\[
\sum_{i=1}^{S} m_{n,i} = 1 \quad \forall n, \quad m_{n,i} \in \{0, 1\} \quad \forall n, i \quad \text{variables} \ \{f_k\}, \{z_k'\}, \{P_k\}, \{a_{k,k'}\}, \{m_{n,i}\}
\]

Constraints (7-9) relate to the MIMO signal design variables, as described in Sec. II-A. In (7), since \( R_{k,k'} \) is only defined when \( a_{k,k'} = 1 \), Eqn. (7) can be considered as \( R_{k,k'} \), for \( n = 1, \ldots, L \) as in (2). In (9), we impose an overall transmission power budget \( P \) that the network cannot exceed. Constraints (10-13), by contrast, restrict the link pair and subchannel allocations, as follows:

**Transmit and receive streams (10&11).** Because each task is indivisible, we consider that each node can have at most one transmit stream, i.e., \( \sum_j a_{k,j} \leq 1 \). Also, we assume that each receive node can have at most one receive stream to reduce the complexity of designing the combiner of receive node, i.e., \( \sum_j a_{j,k} \leq 1 \). Further, we restrict that each node cannot be in transmit and receive mode simultaneously, which leads to Constraint (10). Constraint (11) means data transfer is taken as a binary variable and not allowed for same node.

**Link pair candidates (12).** Only the link pairs that would lead to a lower energy requirement should be considered for the network through optimization. However, we can define the link pair candidates based on comparison of the computation energy, i.e., \( F_k \cdot \frac{I_k}{C_k} \) and \( F_{k'} \cdot \frac{I_{k'}}{C_{k'}} \) for \( k \neq k' \), denoting the computation energy for local processing and offloaded processing, respectively. In other words, if \( F_k \cdot \frac{I_k}{C_k} > F_{k'} \cdot \frac{I_{k'}}{C_{k'}} \), the pair \((k, k')\) can be the candidate for link pairs. Therefore, we define the link pair candidate set as \( \mathcal{X} = \{(k, k') : F_k \cdot \frac{I_k}{C_k} > F_{k'} \cdot \frac{I_{k'}}{C_{k'}}\} \). Note that the optimized link pairs \( \mathcal{X} \) should be among the candidates, i.e., \( \mathcal{X} \subset \mathcal{A} \).

**Subchannel allocation (13).** Each link pair \((k_n, k'_n) \in \mathcal{A} \) must be allocated to a subchannel. With the allocation variables \( m_{n,i} \) restricted to binary values, the sum over subchannels \( i \) for each node \( n \) must be 1.

Assuming all nodes have \( N \) transmit and receive antennas, i.e., \( N = N_{Tx} = N_{Rx} \) for all \( k \), the optimization (6-13) is a mixed integer program (MIP) with \((2N + 1)K\) non-integer and \( K(K-1)(1+S) \) integer variables. Also, since the term \( P_k \cdot I_k / R_{k,k'} \) as a function of \( P_k \) has the form \( x / \log(x) \), which is non-convex, the problem is a non-convex MIP. Existing solvers for non-convex MIPs do not scale well with the number of variables [33], and even with \( K = 10 \) nodes equipped with \( N = 5 \) antennas and \( S = 10 \) subchannels, our problem already has more than 1000 variables. One of the challenges we address in Sec. III is developing an effective algorithm to solve this problem. Shown in Fig. 3, as a first step, we decompose it into two interrelated subproblems, one for the network resource allocation variables (integer) and the other for the MIMO signal design variables (non-integer) as follows:

1. **Network resource optimization:** The variables \( \{a_{k,k'}\} \) and \( \{m_{n,i}\} \) appear in each term of the objective (6). With \( \{f_k\}, \{z_k'\}, \) and \( \{P_k\} \) fixed, then, we have the following

\[
\begin{align*}
\text{minimize} & \quad E_P \\
\text{subject to} & \quad R_{k,k'} \text{ defined in (3) } \forall a_{k,k'} = 1, \\
& \quad ||f_k||_2 = 1 \quad \forall k, \\
& \quad ||z_k'||_2 = 1 \quad \forall k', \\
& \quad \sum_{k=1}^{K} P_k \leq P, \\
& \quad 0 \leq \sum_{j=1}^{K} a_{k,j} + \sum_{j=1}^{K} a_{j,k} \leq 1 \quad \forall k, \\
& \quad a_{k,k'} \in \{0, 1\} \quad \forall k, k', \\
& \quad \sum_{(k,k') \notin \mathcal{X}} a_{k,k'} = 0, \\
& \quad \sum_{i=1}^{S} m_{n,i} = 1 \quad \forall n, \\
& \quad m_{n,i} \in \{0, 1\} \quad \forall n, i \\
& \quad \text{variables} \ \{f_k\}, \{z_k'\}, \{P_k\}, \{a_{k,k'}\}, \{m_{n,i}\}
\end{align*}
\]

Fig. 3: Overall optimization is decomposed into two interrelated subproblems: 1) network resource optimization and 2) MIMO signal design optimization.
Given the dependence of $R_{k,k'}$ on these variables, this problem is a non-linear integer program, but is no longer an MIP.

2) MIMO signal design optimization: The variables $\{f_k\}$, $\{z_k\}$, and $\{P_k\}$ only appear in the term of (6) captured by the communication demand $E_M$ in (2). With $\{a_{k,k'}\}$ and $\{m_{n,i}\}$ fixed, it reduces to the following optimization:

$$
\text{minimize } E_M
\text{ subject to } \text{Constraints (7) – (9)}
\text{variables } \{f_k\}, \{z_k\}, \{P_k\}
$$

Given the dependence of $R_{k,k'}$ on these non-integer variables, the problem is a non-convex optimization, but is also no longer an MIP. In fact, the dependence of $R_{k,k'}$ on both sets of variables renders the optimization (6-13) non-separable, and hence the subproblems must be solved alternately [34]. Although the decomposition for the overall optimization problem does not enable to obtain optimal solution, the obtained (suboptimal) solution from the decomposition can reduce the energy consumption significantly by efficient data offloading compared to that of local processing. This is approximately a 17% reduction in the Fig. 4 example. We will develop this decomposition approach in Section III.

III. DATA TRANSFER AND PROCESSING ALGORITHMS

A. Network resource optimization

We propose two different algorithms to solve for $\{a_{k,k'}\}$ and $\{m_{n,i}\}$ with $\{f_k\}$, $\{z_k\}$ and $\{P_k\}$ fixed.

1) Non-linear integer program: The variables $\{a_{k,k'}\}$ and $\{m_{n,i}\}$ cannot be separable because the subchannel allocation $m_{n,i}$ is determined only for the connected link $a_{k,k'} = 1$. This means that, in order to find the best combinations of $\{a_{k,k'}\}$ and $\{m_{n,i}\}$, the subchannel should be determined jointly with link pairs. Therefore, for joint optimization for $\{a_{k,k'}\}$ and $\{m_{n,i}\}$, three dimensional (3D) variable, $a_{k,k',i}$ is introduced, meaning whether $k$ transmits data to $k'$ through the subchannel $i$. The variable $\{a_{k,k',i}\}$ is linked with $\{a_{k,k'}\}$ and $\{m_{n,i}\}$ by

$$
a_{k,k'} = \sum_{i=1}^{S} a_{k,k',i}, \quad m_{n,i} = a_{k_n,k_n',i}, \quad (14)
$$

where the meaning of two relationships is that the link pair $(k,k')$ is mapped to only one subchannel and that $n$-th link pair $(k_n,k_n')$ is using subchannel $i$.

Then, by substituting $a_{k,k',i}$ for $a_{k,k'}$ and $m_{n,i}$, the formulation (5) can be rewritten as

$$
E_P = \sum_{k=1}^{K} \left( \sum_{k' \neq k}^{K} \sum_{i=1}^{S} a_{k,k',i} \cdot \left( P_k \cdot \frac{I_k}{R_{k,k'}} + F_{k'} \cdot \frac{I_{k'}}{C_{k'}} \right) + \left( 1 - \sum_{k' \neq k}^{K} \sum_{i=1}^{S} a_{k,k',i} \right) \cdot F_k \cdot \frac{I_k}{C_k} \right). \quad (15)
$$

where

$$
R_{k,k'} = \log_2 \left( 1 + \frac{P_k |z_k|^2 H_{k,k'} f_k |^2}{\sum_{l \neq k,k'}^{K} \sum_{l'=1}^{L_{l,k}} a_{l,l',k} P_l |z_l|^2 H_{l,k',l'} f_l |^2 + 1} \right). \quad (16)
$$

The constraints (10 – 12) change to

$$
0 \leq \sum_{j=1}^{K} \sum_{i=1}^{S} a_{j,k,i} + \sum_{j=1}^{K} \sum_{i=1}^{S} a_{j,k,i} \leq 1 \quad \forall k, \quad (17)
$$

$$
a_{k,k'} \in \{0,1\} \quad \forall k, k', i, \quad \sum_{i=1}^{S} a_{k,k,i} = 0 \quad \forall k, \quad (18)
$$

where by introducing $\{a_{k,k'}\}$, the constraint (13) is satisfied since we get $\sum_{i=1}^{S} m_{n,i} = \sum_{i=1}^{S} a_{k_n,k_n',i} = a_{k_n,k_n'} = 1$ from the two relationships in (14). Note that $n$-th link pair $(k_n,k_n')$ comes from the connected link $A$, which leads to $a_{k_n,k_n'} = 1$.

Then, we obtain a non-linear integer program problem:

$$
\text{minimize } E_P \text{ in (15)}
\text{ subject to } \text{Constraints (17 – 19)}
\text{variables } \{a_{k,k'}\}
$$

We implemented this algorithm using the Solving Constraint Integer Programs (SCIP) solver [35] in the Matlab environment. Since it is $NP$-hard problem, a large computational time is required when the number of variables is large. For example, for running with $K = 20$ and $S = 2$, we have total $K \times K \times S = 800$ variables and a computation time of approximately an hour on a 3.5 GHz CPU.

2) Greedy algorithm: For all link pairs $(k,k')$ for $k,k' = 1,...,K$, we find the best pair to maximize the saved energy by data offloading, $D_{k,k'}$ as

$$
\maximize_{(k,k')} D_{k,k'} = E_k - E_{k,k'}, \quad (20)
$$

where $E_k$ and $E_{k,k'}$ are given in (4). The $(k,k')$ link will be connected when $D_{k,k'} > 0$. The link pairs are sequentially determined until all possible links are connected, which leads to $a_{k,k'} = 1$ when links are connected and $a_{k,k'} = 0$, otherwise.

Given the possible link candidates $(k,k') \in A$, we need to allocate the subchannels to the link pairs to minimize the interference. At the beginning, when new subchannels are available, we allocate the empty subchannel which is interference-free, to the selected link pair until all subchannels are used. However, after allocating all interference-free subchannels to the link pairs, the subchannel allocation to the remaining link pairs will incur inter-channel interference. In this case, we should compare all combinations of link pairs and subchannels and choose the best combination. We performed this process until all possible pairs are determined. During this process, whenever we determine the best pair, we update the candidate set $\mathcal{E}$ with the constraint (10) i.e., removing the candidates from the determined links so far. A formal description of the algorithm for joint link pair and subchannel is provided in Algorithm 1.
\[ \{k, k'\} = \{ (i, k, k') \mid e_{i, k, k'} = 1 \} \] from (10), \( \sum_{j=1}^{K} e_{k, j} = 0 \) for \( k = k_p, k_p' \). Then, we fix \( e_{i, k, k'} \) and \( m_{n,i} \) and solve (28). Substituting (29) into (28), we get
\[ w_n = \frac{1}{e_n \log(e_n)} \]
where \( e_n \) is given by (29) and \( \log(e_n) \) is the natural logarithm of \( e_n \).

We fix \( \{z_{k_n}'\} \) and the auxiliary weight \( \{w_n\} \), and optimize the beamformer \( \{g_{k_n}\} \) by solving (28). Substituting \( \{e_n\} \) found from (29) in (28) makes the beamformers

\[ E_{up}^M = P \sum_{n=1}^{L} I_{k_n} \]  \tag{21} \]
where \( P \) is the total transmission power budget in (9). The problem in (21) is simplified to a problem of minimizing the time delay, which has a similar form as maximizing harmonic-mean rate \([36]\). Combining the power \( P_{k_n} \) and \( f_{k_n} \) into \( g_{k_n} \), i.e., \( g_{k_n} = \sqrt{P_{k_n} f_{k_n}} \), leads to
\[ T = \sum_{n=1}^{L} \frac{I_{k_n}}{R_{k_n, k_n'}} \]  \tag{22} \]
subject to \( \sum_{n=1}^{L} \|g_{k_n}||^2 \leq P \) and \( \|z_{k_n}'\||^2 = 1 \) \tag{23} \]
variables \( \{g_{k_n}\}, \{z_{k_n}'\} \) \tag{24} \]
where \( R_{k_n, k_n'} \) is given in (3) and the constraint (23) comes from (8) and (9). This problem is non-convex and NP-hard. Therefore, suboptimal solution can be found by alternatively solving for the variables in (24).
Algorithm 2 Algorithm for transmit beamformer containing power and receive combiner optimization

1: \textbf{Initialize}
2: \(u_n = 2, w_n = 1\)
3: \textbf{repeat}
4: \(z_{k,n}^\text{mmse} = J_n^{-1}(g)H_{k,n,k'}g_{k,n} \forall n \) where \(J_n(g)\) in (27)
5: \(w_n' \leftarrow w_n \forall n\)
6: \(w_n = 1/e_n((\log(e_n))^2) \forall n \) where \(e_n\) in (29)
7: \textbf{Find} \(k_n = \arg \max_k \sum_{n=1}^L I_{k,n}w_n'\)
8: \(\text{while } \sum_{n=1}^L \log(w_n' - 1\sum_{n=1}^L \log(w_n') \ge \varepsilon \text{ do } \)}
9: \textbf{return} \(k_n = \arg \max_k \sum_{n=1}^L I_{k,n}w_n't\), \(P_{k,n} = \|g_{k,n}'\|_2^2\), and \(z_{k,n}' = z_{k,n}^\text{mmse}/\|z_{k,n}^\text{mmse}\|_2 \forall n\)

\(g_{k,n}\) for all \(k, n\) decoupled across the links, resulting in the following optimization problem:

\[
\begin{align*}
\text{minimize} \quad & \sum_{n=1}^L I_{k,n}w_n'|1 - z_{k,n}'^H H_{k,n,k'}g_{k,n}|^2 \\
+ & \sum_{n=1}^L \sum_{m,n} I_{k,n}w_n'M_{n,m}z_{k,m}^H H_{k,n,k'}g_{k,n}^H H_{k,m,k'} g_{k,m} z_{k,m} \\
\text{subject to} \quad & \sum_{n=1}^L \|g_{k,n}'\|_2^2 \le P.
\end{align*}
\]

This problem is a standard quadratic convex optimization problem, which can be solved by using Karush-Kuhn-Tucker (KKT) conditions [39]. Since it is a standard procedure, we omit the details here. An algorithmic description for the MIMO signal design is outlined in Algorithm 2.

IV. ALGORITHM EVALUATION AND DISCUSSION

In this section, we perform simulations to validate the proposed framework for MIMO signal design and resource allocation through our proposed algorithm.

A. Simulation setup

For resource allocation, we can apply one of our two proposed schemes, the non-linear integer programming (NLIP) in III-A1 or greedy algorithm in III-A2. For MIMO signal design, the WMMSE algorithm in III-B is used. Therefore, we evaluate two combinations of algorithm, NLIP–WMMSE and Greedy–WMMSE, which will be denoted by NLIP and Greedy in Figs. 5-7. The decomposed optimization depends on initial values of \(\{f_k\}, \{z_k\}, \{P_k\}, \{a_{k,k'}\}, \) and \(\{m_{n,i}\}\). For our proposed algorithm, NLIP–WMMSE and Greedy–WMMSE, we run each of them 10 times with different initializations and keep the mean of the results and the minimizer, called the best in Fig. 5-7. The minimizer is selected as the one minimizing the total energy most. For initializations, the link pair \(\{a_{k,k'}\}\) is randomly chosen from link candidate set \(\mathcal{X}\), the subchannel \(\{m_{n,i}\}\) is allocated to connected links \(a_{k,k'} = 1\) randomly, \(P_k = P/L_{\text{max}}\), and \(\{f_k\}, \{z_k\}\) are generated to be uniformly distributed on the complex unit sphere [40], where \(L_{\text{max}} = \lceil K/2 \rceil\) is the maximum number of link pairs with total \(K\) nodes.

For all our simulations, we generated random Gaussian channel \(H_{k,k'}\) for \(k, k' = 1, \ldots, K\), and used the bandwidth \(W = 1\) MHz for each subchannel, and \(P = 5\) Watt with noise power \(\sigma^2 = 1\). As for computational speed \(C_k\), the general unit Hz (or cycles/s) can be converted to bits/sec according to the application it processes. For example, in the work of the audio signal detection [41], 500 cycles are required to process 1 bit, i.e., 500 cycles/bit. In other words, 500 MHz CPU can be converted to 1 Mbits/sec. As an example, for computation power \(F_k\) the mobile devices with 500 MHz CPU require about 0.6 Watt for CPU processing [42]. Based on trying to emulate heterogeneous devices with different energy efficiency, we generate the length of data \(I_k \sim U(1,20)\) Mbits, computational speed \(C_k \sim U(0.1, 2)\) Mbits/sec, and computing power \(F_k \sim U(0.5, 1)\) Watt for \(k = 1, \ldots, K\), where \(U(a,b)\) is continuous uniform distribution ranging from \(a\) to \(b\). For all our simulations, all nodes have \(N\) transmit and receive antennas, i.e., \(N = N^T_{k} = N^R_{k}\) for all \(k\). We select the number of antennas based on properness of interference alignment [43], \(2N \ge L_{\text{max}} + 1\).

B. Varying number of links

Fig. 4 depicts the relationship among the total energy \(E_P\), communication energy \(E_M\), and computation energy \(E_F\), along the number of links \(L\) where \(E_P\) can be calculated as \(E_P = E_F - E_M\). Interestingly, the total energy \(E_P\) decreases below the local computation energy as the available communication resources (links) are more heavily exploited, while the energy for communication increases and that for computation decreases. In other words, since more nodes participate in offloading, we can further reduce energy consumption up to approximately 17 %, reducing energy from 20 Joules to 16.6 Joules. This simulation was performed by using Greedy-WMMSE, when \(K = 10\) nodes, \(N = 6\) antennas for both transmit and receive and \(S = 3\) subchannels.

C. Algorithm convergence analysis

Fig. 5 illustrates the convergence behavior of the proposed algorithm along the iteration number under different number
of antennas with $K = 6$ nodes $S = 3$ subchannels, where from the top to bottom, the total energy $E_P$, communication energy $E_M$, and computation energy $E_F$. The total energy decreases as the iteration proceeds and converges after 2-3 iteration. After first iteration the total energy $E_P$ is greatly decreased because of the sizable reduction in communication. This implies that both the MIMO signal design and network resource optimization are critical for minimizing the total energy. Further, the total energy becomes less than that of local computation without offloading, which is denoted by bold line in Fig. 5. Also, as the number of antennas $N$ is increasing, from $N = 5$ to $N = 15$, the interferences from transmit nodes are suppressed further due to the proposed MIMO signal design algorithm for transmit power, beamformer and receive combiner, which leads to less communication energy $E_M$ and total energy $E_P$.

**D. Varying number of subchannels**

We consider $K = 10$ nodes and $N = 7$ antennas for both transmit and receive. To address joint optimization for MIMO signal design, resource allocation and link selection, we introduce a random matching method, Random–WMMSE which is composed of the random network resource allocation–link selection and subchannel allocation, and MIMO signal design optimization–transmit beamformer, transmit power, and receive combiner with WMMSE approach, i.e., Random–WMMSE is a partially optimized solution. To be more specific, for Random–WMMSE, 1) $L_{\text{max}} = \lceil K/2 \rceil$ link pairs are randomly selected among $K$ nodes and subchannels are allocated to the selected link pairs randomly, 2) the direction of the links (transmit and receive nodes) is determined by comparing the computation speed $C_k$, and 3) the WMMSE
approach in Algorithm 2 is exploited for MIMO signal design. Random–WMMSE method is denoted by Random in Figs. 6-7.

Fig. 6 compares the energy of the two proposed methods, NLIP–WMMSE and Greedy–WMMSE, with Random–WMMSE across different numbers of subchannels, $S$. The proposed methods, NLIP–WMMSE and Greedy–WMMSE, give better performance as compared with Random–WMMSE due to the efficient link allocation to limited number of subchannels. The total energy $E_P$ decreases across the number of subchannels. This is a direct consequence from the reduction of $E_M$ and $E_F$. First, with respect to communication energy $E_M$, more subchannels enable the receive nodes to have less interferences, which increases the data rate and hence decreases communication energy $E_M$. As the data rates increase, more links are established, which make the computation energy $E_F$ further decreased. In other words, increasing the ratio $S/K$ enables more reduction of total energy consumption. The obtained total energy can fluctuate due to different initializations when the number of subchannels is greater than 3 shown in Fig. 6. This will be mitigated with much more number of initializations.

E. Varying number of nodes

Fig. 7 plots the energy $E_P$, $E_M$, and $E_F$ versus number of nodes, $K$. We consider $N = 10$, $S = 2$, and $P = 10$. As the nodes increase in number, it is obvious that the total energy $E_P$ and computation energy $E_F$ are increasing. However, under constant power constraints $P = 10$, the communication energy $E_M$ may not be increasing across the number of nodes because the nodes shares the total power $P$ among nodes. Interestingly, the Greedy–WMMSE and the NLIP–WMMSE gives almost the same result even when $K = 20$. However, the total energy gap between the proposed methods and Random–WMMSE becomes remarkable as the number of nodes is increasing. This stresses that efficient, joint MIMO signal design and network resource allocation is critical to maximize the energy efficiency in the environment surrounded by many devices with different computation speed $C_k$, power $F_k$ and length of data $I_k$.

V. Conclusion and Future Work

In this paper, we proposed a novel optimization methodology for joint MIMO signal design and resource allocation to maximize the energy efficiency of data processing in wireless device-to-device (D2D) edge networks. Given that the problem is a non-convex mixed integer program, we decomposed it into two subproblems for solvability at scale: (1) network resource allocation, i.e., link selection with subchannel allocation, and (2) MIMO signal design, i.e., transmit beamformer, transmit power and receive combiner, and then optimized each problem alternately. Our evaluation showed that substantial improvements in resource utilization can be achieved through this joint optimization when there are more available link pairs and subchannels for data transfer and interference mitigation.

Fig. 7: (a) Total energy $E_P$ (b) Energy for communication $E_M$ (c) Energy for computation $E_F$ comparison of three different schemes along the number of nodes, $K$. As the number of nodes increases, the Greedy-WMMSE and NLIP-WMMSE both considering joint optimization shows better performance in minimizing the energy, compared to the partially optimized solution, the Random–WMMSE.

Additionally, we showed that the solution scales well with the size (number of nodes) comprising the network.

For future work, time consumption for processing the data can also be included together with energy consumption. By extension, each receive node can be relaxed to have multiple receive streams. We are required to design proper receive combiner and to solve the computing resource allocation problem. Further work is needed to understand the algorithm behavior, convergence and sub-optimality.

Acknowledgment

We thank the reviewers for their valuable comments. D.J. Love was supported in part by the National Science Foundation (NSF) under grants CNS1642982 and CCF1816013.