

DACS: Efficient Resource Allocation and Power Control for D2D Communication Considering RAN Sharing

You-Chiun Wang and Wei-Ting Chen

Abstract—*Device-to-device (D2D) communication plays a key role in improving the performance and flexibility of mobile networks. Without the relay via a base station (BS), it allows two user equipments (UEs) to directly communicate with each other. D2D pairs can share the spectrum resources given to cellular UEs, but doing so may cause interference. As a result, it is critical to manage resources and transmitted power for UEs. In addition, the radio access network (RAN) sharing technique is designed to enable multiple service providers (SPs) to share network infrastructure (e.g., BSs) for reducing deployment and operational expenses. Consequently, this paper proposes a D2D resource allocation and power control with RAN sharing (DACS) framework in which some SPs each possess a portion of a BS's resources and their UEs require service. Each SP first allocates the owned resources to its UEs and decides power to meet SINR demands of UEs while reducing interference. To serve more UEs, SPs can borrow resources from one another. Afterward, DACS moderately enhances the power of senders to improve the quality of the signal received by UEs, thereby improving their throughput. Simulation results reveal that DACS can achieve a high service ratio, increase both throughput and energy efficiency, and maintain fairness among SPs.*

Index Terms—D2D communication, RAN sharing, resource and power management.

1 INTRODUCTION

The increasingly rapid growth in the number of *user equipments (UEs)*, such as mobile phones and IoT devices, has made spectrum resources scarce. One promising solution is *device-to-device (D2D) communication* [1], which accomplishes the direct communication between two UEs (referred to as a *D2D pair*) without asking a *base station (BS)* to relay their messages. This technique brings three benefits. First, the spectral efficiency is improved, since resources can be shared by D2D pairs and *cellular UEs (CUs)*, that is, UEs in contact with a BS or with other UEs via the BS). Second, UEs in a D2D pair (called *D2D UEs*, or *DUs* for short) are close to each other, so a D2D sender can reduce the transmitted power (below, we call it *power* for short) to save energy. Third, using D2D communication helps expand a BS's service coverage [2].

In D2D communication, both *resource allocation* and *power control* are critical issues [3]. For a BS, its spectrum resources are usually divided into *resource blocks (RBs)* to be units for allocation [4]. Then, the resource allocation problem asks how to assign the BS's RBs to CUs and D2D pairs for communication. On the other hand, when a CU and some D2D pairs share an RB, their senders will impose interference on these UEs. Hence, the power control issue asks how to find the power for senders to improve the throughput of UEs while mitigating interference.

Given the exorbitant expenses associated with telecommunications infrastructure, 3GPP suggests the *radio access network (RAN) sharing* technique [5] to assist *service providers (SPs)* in lowering network deployment and operating costs. The RAN sharing technique enables multiple SPs to share spectrum resources and cooperate on BSs. Using RAN sharing substantially extends services and coverage of a mobile network.

In addition, RAN sharing facilitates the deployment of 5G standalone and non-standalone networks [6].

In light of the advantages of RAN sharing, this paper considers D2D communication in a RAN sharing scenario. However, doing so raises challenges. In particular, suppose that the BS offers a set $\mathcal{R}$ of RBs. Two SPs, s_1 and s_2 , own subsets $\mathcal{R}_1$ and $\mathcal{R}_2$ of the BS's RBs, where $\mathcal{R}_1 \cup \mathcal{R}_2 = \mathcal{R}$, $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$, and $|\mathcal{R}_1| = |\mathcal{R}_2|$. Traditional solutions make each SP use only its RBs to serve the subscribed UEs. Consider that the BS's cell covers many of s_1 's UEs but only a few of s_2 's UEs. Even if these solutions help s_1 fully utilize each RB in $\mathcal{R}_1$, there will be some UEs not served (as $\mathcal{R}_1$'s RBs are insufficient). On the other hand, some RBs in $\mathcal{R}_2$ may be wasted due to just a few requests from its UEs. Since UEs of each SP in the cell may dynamically change, it is difficult to let SPs adjust the number of their RBs in $\mathcal{R}$ (e.g., based on the number of their UEs).

To tackle the above challenges, we propose a *D2D resource allocation and power control with RAN sharing (DACS)* framework. Each SP allots the owned RBs to its UEs and adjusts power for these UEs to meet their SINR demands while reducing interference. If there are UEs still not served, an SP can borrow resources from other SPs to serve them, which is carried out via an RB lending mechanism. Then, the power of some senders is increased to improve UE throughput. Compared to existing approaches, our DACS framework has two key innovations. First, DACS makes good use of RAN sharing by enabling SPs to borrow resources from each other, especially when their resources are not enough. This issue is rarely discussed and investigated in the literature. Second, in addition to raising efficiency, the RB lending mechanism in DACS takes account of maintaining fairness among SPs. More concretely, there are two metrics used to quantify SP fairness:

- 1) The ratio of UE throughput under each SP is close to the ratio of the BS's resources owned by the SP. Suppose that n SPs, $s_1, s_2, \dots, s_n$, own subsets

$\mathcal{R}_1, \mathcal{R}_2, \dots$, and $\mathcal{R}_n$ of a BS's RBs. If each SP has ever borrowed RBs from another SP, we expect that the ratio of UE throughput under $s_1, s_2, \dots$, and s_n is close to $|\mathcal{R}_1| : |\mathcal{R}_2| : \dots : |\mathcal{R}_n|$.

- 2) The total resource amount that each SP borrows from others can be close to the total resource amount that the SP lends to others. How to calculate the resource amount will be discussed later in Section 4.2.

Through simulations, we demonstrate that DACS can attain a high service ratio, improve UE throughput, raise energy efficiency, and maintain SP fairness.

This paper is outlined as follows: Section 2 discusses related work, and Section 3 describes the system model. We detail the DACS framework in Section 4, followed by experimental evaluation in Section 5. Then, Section 6 concludes this paper.

2 RELATED WORK

The resource allocation problem for D2D communication is NP-hard [7]. Some studies convert it to a matching issue. For each CU, the work [8] chooses a D2D pair to share its RB through the Hungarian algorithm. To overcome information ambiguity in the Hungarian algorithm, the study [9] uses an autoencoder that addresses the random weight selection issue when some links between CUs and D2D pairs have the same weight. Zhou et al. [10] select pairs of CUs and DUs for sharing RBs to improve energy efficiency. The study [11] employs the Gale-Shapley method to assign DUs to reuse those RBs allocated to CUs. However, each RB can be shared by at most one D2D pair, resulting in low RB utilization.

Game-theoretic approaches are also developed for D2D resource allocation. Both studies [12], [13] treat UEs as players and employ a Stackelberg game for RB allocation. Considering the uncertainty of channel state information, Liu et al. [14] impose the probability constraint to optimize the user utility in the Stackelberg game. Two coalition formation algorithms are proposed in [15] by using game theory to allocate RBs to D2D pairs with the objective of minimizing interference. In [16], D2D pairs engage in strategic games with nearby BSs for RB allocation to improve their utility. Akhyani et al. [17] combine artificial intelligence and coalition game theory to resolve the issues of resource allocation and security in D2D communication. As can be seen, the above approaches do not take account of RAN sharing.

How to allocate resources to UEs via graph coloring attracts attention [18]–[22]. These studies construct a graph to express the interference relationship between UEs. Then, adjacent vertices (i.e., UEs) cannot be painted with the same color; in other words, these UEs are not allowed to share RBs due to non-negligible interference with each other. Raghu et al. [23] divide DUs into two categories and propose an iterative-based method to manage their resources and power. Both studies [24], [25] apply the deep learning technique to D2D resource allocation. Alghazali et al. [26] solve the D2D resource allocation problem using genetic algorithms. However, these studies do not exploit the property of RAN sharing to improve system performance.

Regarding RAN sharing, the work [27] first allots RBs to the CUs belonging to each SP and then makes DUs reuse CUs' RBs with the least interference. Based on binary integer programming, the study [28] formulates an RB allocation problem that takes both the sum rate of UEs and the cost of

SPs into consideration. In [29], SPs may trade RBs according to their channel conditions. The study [30] considers two types of 5G slices: an eMBB (enhanced Mobile Broadband) slice that needs a high data rate and an mMTC (massive Machine Type Communication) slice that requires low latency. Then, RBs are assigned to UEs by referring to their types to meet QoS requirements. Nevertheless, these studies assume fixed power for the BS and D2D senders (i.e., no power control). The work [31] adjusts power for senders and allows SPs borrowing RBs from each other to increase throughput. However, some SPs may borrow more RBs from others, which leads to unfairness in UE throughput under different SPs.

Table 1 presents a comparison between the prior work about D2D resource allocation and our DACS framework. Apparently, there has not been much research done on the problem of applying D2D communication in a RAN sharing scenario. Only the work [31] and DACS jointly handle resource allocation and power control for this situation and take advantage of the property of RAN sharing by allowing SPs to borrow RBs from each other (i.e., the RB lending mechanism). In comparison to the work [31], our DACS framework is more capable of ensuring that SPs maintain fairness in the context of RB lending, which underscores the distinction between DACS and the existing work.

3 SYSTEM MODEL

Consider a BS offering a set $\mathcal{R}$ of downlink RBs to serve UEs every *transmission time interval* (TTI). In the RAN sharing scenario, there is a set $\mathcal{S}$ of SPs that cooperate on the BS. Specifically, each SP $s_k \in \mathcal{S}$ owns a subset $\mathcal{R}_k \subset \mathcal{R}$ of RBs with three conditions as follows:

- 1) All RBs in $\mathcal{R}$ are assigned to SPs in $\mathcal{S}$ (i.e., $\bigcup_{s_k \in \mathcal{S}} \mathcal{R}_k = \mathcal{R}$).
- 2) Each SP possesses non-zero RBs (i.e., $|\mathcal{R}_k| > 0, \forall s_k \in \mathcal{S}$).
- 3) No two SPs have overlapped RBs (i.e., $\mathcal{R}_k \cap \mathcal{R}_m = \emptyset, \forall s_k, s_m \in \mathcal{S}$ and $s_k \neq s_m$).

SPs may borrow RBs from each other. This can be carried out using a *capacity broker*, as proposed by 3GPP, which takes charge of managing requests and leases for resources between SPs [32].

Each UE cannot be a CU and a DU simultaneously. For DUs, we adopt in-band and underlaid D2D communication. In other words, cellular and D2D communication share the same frequency band, and D2D pairs can reuse RBs assigned to CUs. Each D2D pair has a sender and a receiver, where the receiver is used to represent the pair. Each DU belongs to exactly one D2D pair. Let us denote by $\mathcal{C}_k$ and $\mathcal{D}_k$ the sets of CUs and D2D receivers subscribed to SP s_k , where $\mathcal{C}_k \cap \mathcal{D}_k = \emptyset$. Moreover, $\mathcal{C}$ and $\mathcal{D}$ signify all CUs and D2D receivers; that is, $\mathcal{C} = \bigcup_{s_k \in \mathcal{S}} \mathcal{C}_k$ and $\mathcal{D} = \bigcup_{s_k \in \mathcal{S}} \mathcal{D}_k$.

RB allocation has to adhere to five rules. First, the use of the same RB by two CUs is prohibited due to significant interference. Second, D2D pairs may reuse a CU's RBs or share RBs not yet given to any CU. Third, each SP s_k must first allocate RBs in $\mathcal{R}_k$ to its UEs. Fourth, only after s_k allots RBs by the third rule can s_k lend other SPs the available RBs in $\mathcal{R}_k$ or borrow RBs from others. Finally, if a D2D pair's members belong to different SPs, the receiver's SP takes charge of RB allocation.

In addition, we make the following assumptions:

TABLE 1: Comparison between the prior work about D2D resource allocation and our DACS framework.

work	methodology	power control	RAN sharing	SP fairness
work [8]	matching			
work [9]	matching			
work [10]	matching	✓		
work [11]	matching	✓		
work [12]	game theory	✓		
work [13]	game theory	✓		
work [14]	game theory	✓		
work [15]	game theory	✓		
work [16]	game theory			
work [17]	game theory			
work [18]	graph coloring	✓		
work [19]	graph coloring	✓		
work [20]	graph coloring	✓		
work [21]	graph coloring	✓		
work [22]	graph coloring			
work [23]	iterative-based method	✓		
work [24]	deep learning	✓		
work [25]	deep learning	✓		
work [26]	genetic algorithm	✓		
work [27]	heuristic algorithm		✓	
work [28]	binary integer programming		✓	
work [29]	resource trading		✓	
work [30]	distributed algorithm		✓	
work [31]	RB lending mechanism	✓	✓	partial
DACS	RB lending mechanism	✓	✓	✓

 TABLE 2: Interference from u_j 's sender $\varepsilon(j)$ to u_i on RB r_x .

$\varepsilon(j)$	u_j	u_i	interference amount $\psi_{\varepsilon(j),i}^x$
BS	CU	CU	0 (since u_i and u_j cannot share r_x)
BS	CU	DU	$g_{BS,i} \times p_{BS,j}^x$
DU	DU	CU/DU	$g_{\varepsilon(j),i} \times p_{\varepsilon(j),j}^x$

- 1) Our work focuses primarily on the power of the BS and D2D senders. In other words, we consider downlink communication for CUs and omit their uplink communication.
- 2) Regarding RB sharing between SPs, all RBs (including adjacent ones) are treated as sharable units.
- 3) UEs may be stationary (e.g., fixed IoT devices) or move at relatively low speeds (e.g., pedestrians carrying UEs while walking or vehicles with UEs moving in a downtown area).

Each RB's capacity is decided by its receiver's SINR. When UE u_i receives data from its sender $\varepsilon(i)$ using RB r_x , the strength of $\varepsilon(i)$'s signal acquired by u_i is $g_{\varepsilon(i),i} \times p_{\varepsilon(i),i}^x$. Here, $g_{\varepsilon(i),i}$ and $p_{\varepsilon(i),i}^x$ denote $\varepsilon(i)$'s channel gain (hereinafter referred to as "gain") and power on r_x to transmit data to u_i . Given the environmental (or thermal) noise φ , we can estimate UE u_i 's SINR on RB r_x by

$$\sigma_i^x = \frac{z_i^x (g_{BS,i} \times p_{BS,i}^x)}{\sum_{u_j \in \mathcal{D}} z_j^x \psi_{\varepsilon(j),i}^x + \varphi} \quad \text{if } u_i \in \mathcal{C} \text{ (i.e., a CU),} \quad (1)$$

$$\sigma_i^x = \frac{z_i^x (g_{\varepsilon(i),i} \times p_{\varepsilon(i),i}^x)}{\sum_{u_j \in \mathcal{C}} z_j^x \psi_{BS,i}^x + \sum_{u_j \in \mathcal{D} \setminus \{u_i\}} z_j^x \psi_{\varepsilon(j'),i}^x + \varphi} \quad \text{otherwise (i.e., a D2D receiver),} \quad (2)$$

where $z_i^x = 1$ if u_i uses r_x , or $z_i^x = 0$ otherwise. When u_i is a CU, its interference sources will be D2D senders also using u_i 's RBs. In Eq. (1), the term $\psi_{\varepsilon(j),i}^x$ gives the amount of interference from a D2D sender $\varepsilon(j)$ (i.e., u_j 's sender, where $u_j \in \mathcal{D}$) to u_i on RB r_x , as calculated by $g_{\varepsilon(j),i} \times p_{\varepsilon(j),j}^x$. On the other hand, if u_i is a D2D receiver, its interference sources include not only the BS but also those D2D senders that share u_i 's RBs. As a result, in Eq. (2), the term $\psi_{BS,i}^x$ is the amount of interference

from the BS to u_i on RB r_x . It can be derived as $g_{BS,i} \times p_{BS,j}^x$, where $u_j \in \mathcal{C}$. Similarly, the term $\psi_{\varepsilon(j'),i}^x$ indicates the amount of interference from another D2D sender $\varepsilon(j')$ (i.e., the sender of $u_{j'}$, where $u_{j'} \in \mathcal{D}$ and $u_{j'} \neq u_i$) to u_i , as computed by $g_{\varepsilon(j'),i} \times p_{\varepsilon(j'),j'}^x$. We summarize the amount of interference from different interference sources in Table 2.

Let v_i^x be the indicator to reveal whether the channel quality of UE u_i on RB r_x fulfills its minimal required SINR $\sigma_i^{\min}$; in particular, $v_i^x = 1$ if $\sigma_i^x \geq \sigma_i^{\min}$, or $v_i^x = 0$ otherwise. Then, our problem can be expressed by

$$\text{maximize } \sum_{r_x \in \mathcal{R}} \sum_{s_k \in \mathcal{S}} \sum_{u_i \in \mathcal{C}_k \cup \mathcal{D}_k} z_i^x v_i^x, \quad (3)$$

subject to the following constraints:

$$\sum_{u_i \in \mathcal{C}} z_i^x \leq 1, \quad \forall r_x \in \mathcal{R}, \quad (4)$$

$$\sum_{u_i \in \mathcal{C} \cup \mathcal{D}} z_i^x \leq \delta \quad \forall r_x \in \mathcal{R}, \quad (5)$$

$$z_i^x \in \{0, 1\}, v_i^x \in \{0, 1\} \quad \forall r_x \in \mathcal{R}, \forall u_i \in \mathcal{C} \cup \mathcal{D}, \quad (6)$$

$$p_i^{\min} \leq p_{\varepsilon(i),i}^x \leq p_i^{\max} \quad \forall r_x \in \mathcal{R}, \forall u_i \in \mathcal{C} \cup \mathcal{D}. \quad (7)$$

In Eq. (3), the objective is to maximize the number of CUs and D2D pairs served by all SPs. Specifically, a UE u_i is viewed as *served* if it can obtain RBs that meet SINR demand $\sigma_i^{\min}$. Here, there are two control parameters: z_i^x (i.e., resource allocation) and v_i^x (i.e., power control, as the power decides u_i 's SINR). For constraints, Eq. (4) indicates that two CUs cannot use the same RB. Eq. (5) imposes the maximum number of UEs in $\mathcal{C} \cup \mathcal{D}$ allowed to share an RB, where $\delta \in \mathbb{Z}^+$ and $\delta > 1$. In Eq. (6), z_i^x and v_i^x are indicators whose values are 0 or 1. Eq. (7) gives both lower and upper bounds (i.e., $p_i^{\min}$ and $p_i^{\max}$) of the power of u_i 's sender. We summarize notations in Table 3.

4 THE PROPOSED DACS FRAMEWORK

Fig. 1 illustrates DACS's architecture, which is composed of three core algorithms. In the *basic resource allocation (BRA)* algorithm, each SP allocates the owned RBs to its UEs. When there are some UEs not served yet, the *inter-SP resource sharing (IRS)* algorithm enables resource lending among SPs to

TABLE 3: Summary of notations.

notation	definition
$\mathcal{S}$	set of SPs cooperating on the BS
$\mathcal{R}_k$	set of RBs that SP s_k possesses ($\mathcal{R}$: union of all $\mathcal{R}_k$)
$\mathcal{C}_k, \mathcal{D}_k$	sets of CUs and D2D receivers subscribed to s_k ($\mathcal{C}$: union of all $\mathcal{C}_k$, $\mathcal{D}$: union of all $\mathcal{D}_k$)
$\mathcal{C}'_k, \mathcal{D}'_k$	sets of s_k 's CUs and D2D receivers not served yet
$\mathcal{Q}_x$	set of UEs that share RB r_x , where $ \mathcal{Q}_x \leq \delta$
Θ	set of DUs in $\mathcal{Q}_x$ that cannot allocate power
$g_{\varepsilon(i),i}$	$\varepsilon(i)$'s gain to u_i , where $\varepsilon(i)$ denotes u_i 's sender
$p_{\varepsilon(i),i}^x$	$\varepsilon(i)$'s power to u_i on r_x , where $p_{\varepsilon(i),i}^x \in [p_i^{\min}, p_i^{\max}]$
σ_i^x	u_i 's SINR on r_x ($\sigma_i^{\min}$: minimal required SINR)
$\zeta_{i,j}$	indicator to check if two UEs u_i and u_j are neighbors
$\Omega_{k,m}^x$	resource amount of r_x that SP s_k lends to SP s_m
$f_{\text{SRT}}^{\text{DEC}}, f_{\text{SRT}}^{\text{INC}}$	decreasingly and increasingly sorting functions
Set size:	$ \mathcal{S} = \eta_S, \mathcal{D} = \eta_D, \mathcal{C} \cup \mathcal{D} = \eta_U, \mathcal{R} = \eta_R, \mathcal{R}_k \leq \eta_K$

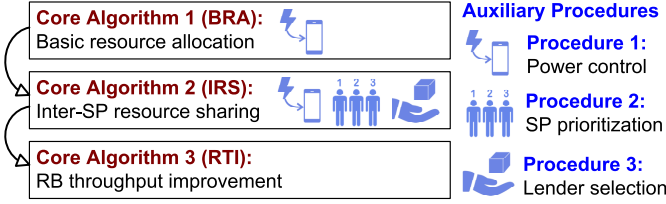


Fig. 1: The architecture of the DACS framework.

serve these UEs. Then, the *RB throughput improvement (RTI)* algorithm increases the number of data bits carried by each RB. Moreover, there are three auxiliary procedures. The power control procedure adjusts the power of senders on the same RB to meet SINR demands of their receivers while mitigating interference. This procedure is used by the BRA and IRS algorithms. Both SP prioritization and lender selection procedures are adopted in the IRS algorithm to implement the RB lending mechanism, which takes account of efficiency and SP fairness.

Next, we detail each core algorithm in Section 4.1 and each auxiliary procedure in Section 4.2. Then, Section 4.3 gives the design rationale of our DACS framework, followed by the discussion of some issues in Section 4.4.

4.1 Core Algorithms

Given the UEs of each SP $s_k \in \mathcal{S}$, the BRA algorithm carries out initial RB and power allocation for them. BRA's pseudocode is presented in Algorithm 1. Recall the RB allocation rules discussed in Section 3: multiple CUs cannot share the same RB, whereas D2D pairs are allowed to reuse a CU's RB or share an RB with other D2D pairs. To facilitate RB allocation in the BRA algorithm, we check if a D2D receiver $u_i \in \mathcal{D}$ has a neighboring relationship with another UE $u_j \in \mathcal{C} \cup \mathcal{D}$, where $u_j \neq u_i$. In other words, the sender of u_i , $\varepsilon(i)$, will impose non-negligible interference on u_j ¹. The corresponding code is given in lines 1–4. Specifically, line 3 checks if the gain $g_{\varepsilon(i),j}$ from $\varepsilon(i)$ to u_j exceeds threshold g_{th} . How to specify the value of g_{th} will be discussed later in Remark 1. In line 4, we use an indicator $\zeta_{i,j}$ to reveal whether u_i and u_j are neighbors. The default value of $\zeta_{i,j}$ is zero. If $g_{\varepsilon(i),j} > g_{\text{th}}$, we set $\zeta_{i,j} = 1$. In this case, as u_i is a neighbor of u_j , u_j must be a neighbor of u_i (i.e., symmetry). Hence, we also set $\zeta_{j,i} = 1$ in line 4.

The subsequent for-loop allocates RBs to s_k 's CUs (i.e., lines 7–10) and DUs (i.e., lines 11–14) and then decides power for them (i.e., lines 15–20). Specifically, BRA serves CUs with

1. In this situation, it is inappropriate to let u_i and u_j share the same RB.

Algorithm 1: Basic Resource Allocation (BRA)

```

1 foreach  $u_i \in \mathcal{D}$  do
2   foreach  $u_j \in (\mathcal{C} \cup \mathcal{D}) \setminus \{u_i\}$  do
3     if  $g_{\varepsilon(i),j} > g_{\text{th}}$  then
4        $\zeta_{i,j} \leftarrow 1$  and  $\zeta_{j,i} \leftarrow 1$ ;
5 foreach  $s_k \in \mathcal{S}$  do
6    $\mathcal{C}'_k \leftarrow f_{\text{SRT}}^{\text{DEC}}(\mathcal{C}_k, g_{\text{BS},i}), \mathcal{D}'_k \leftarrow f_{\text{SRT}}^{\text{DEC}}(\mathcal{D}_k, g_{\text{BS},i});$ 
7   foreach  $u_i \in \mathcal{C}'_k$  do
8     foreach  $r_x \in \mathcal{R}_k$  do
9       if  $\mathcal{Q}_x \cap \mathcal{C} = \emptyset$  and  $\zeta_{i,j} = 0, \forall u_j \in \mathcal{Q}_x$  then
10        Move  $u_i$  from  $\mathcal{C}'_k$  to  $\mathcal{Q}_x$  and break;
11  foreach  $u_i \in \mathcal{D}'_k$  do
12    foreach  $r_x \in \mathcal{R}_k$  do
13      if  $|\mathcal{Q}_x| < \delta$  and  $\zeta_{i,j} = 0, \forall u_j \in \mathcal{Q}_x$  then
14        Move  $u_i$  from  $\mathcal{D}'_k$  to  $\mathcal{Q}_x$  and break;
15  foreach  $r_x \in \mathcal{R}_k$  do
16     $\Theta \leftarrow \text{power\_control}(\mathcal{Q}_x);$ 
17    Move DUs in  $\Theta$  from  $\mathcal{Q}_x$  to  $\mathcal{D}'_k$ ;
18    foreach  $u_i \in \mathcal{Q}_x \cap \mathcal{D}$  do
19      if  $\sigma_i^x < \sigma_i^{\min}$  then
20        Move  $u_i$  from  $\mathcal{Q}_x$  to  $\mathcal{D}'_k$ ;

```

larger gains first, as their SINR demands can be met using lower power. Hence, line 6 sorts CUs in $\mathcal{C}_k$ by their gains from the BS decreasingly, as denoted by $f_{\text{SRT}}^{\text{DEC}}(\mathcal{C}_k, g_{\text{BS},i})$, and stores the result in a set $\mathcal{C}'_k$. Let $\mathcal{Q}_x$ be the set of UEs using RB r_x . For each CU u_i in $\mathcal{C}'_k$, we choose an RB r_x from $\mathcal{R}_k$ (i.e., s_k 's RBs) for u_i if two conditions both hold: 1) r_x is not assigned to any CU (i.e., $\mathcal{Q}_x \cap \mathcal{C} = \emptyset$; the first RB allocation rule) and 2) none of u_i 's neighbors share r_x (i.e., $\zeta_{i,j} = 0, \forall u_j \in \mathcal{Q}_x$; to avoid interference). If r_x can be found, we remove u_i from $\mathcal{C}'_k$ and add it to $\mathcal{Q}_x$.

Then, s_k assigns RBs to its DUs. The BRA algorithm prioritizes DUs that experience greater gains from the BS. That is because these DUs are more susceptible to the BS's interference. As a result, line 6 sorts DUs in $\mathcal{D}_k$ by gain $g_{\text{BS},i}$ in a decreasing order, and the result is stored in a set $\mathcal{D}'_k$. In lines 11–14, we pick an RB $r_x \in \mathcal{R}_k$ for each DU u_i in $\mathcal{D}'_k$ if two conditions both hold: 1) r_x can accommodate more UEs (i.e., $|\mathcal{Q}_x| < \delta$) and 2) there is no u_i 's neighbor also using r_x . After finding RB r_x , we remove u_i from $\mathcal{D}'_k$ and add it to $\mathcal{Q}_x$.

The remaining code computes power for UEs that use each RB $r_x \in \mathcal{R}_k$. This can be done by the power control procedure in line 16, which takes $\mathcal{Q}_x$ as its input parameter. As mentioned later in Section 4.2, the power control procedure returns a subset of DUs (denoted by Θ) from $\mathcal{Q}_x$ that cannot allocate power. Consequently, line 17 removes these DUs from $\mathcal{Q}_x$ and adds them back to $\mathcal{D}'_k$. Afterward, the for-loop in lines 18–20 checks if the SINR σ_i^x of each DU u_i in $\mathcal{Q}_x$ can meet its minimum requirement $\sigma_i^{\min}$. If not (i.e., $\sigma_i^x < \sigma_i^{\min}$), u_i will be excluded from $\mathcal{Q}_x$ and added to $\mathcal{D}'_k$. Theorem 1 discusses the time complexity of the BRA algorithm.

Remark 1 (gain threshold). In the BRA algorithm, we use a gain threshold g_{th} to check if the sender of a DU u_i , $\varepsilon(i)$, will cause non-negligible interference on another UE u_j . Generally

speaking, when the distance between $\varepsilon(i)$ and u_j is below a threshold d_{th} (that is, $\varepsilon(i)$ is close enough to u_j), the interference from $\varepsilon(i)$ to u_j becomes non-negligible. One feasible way is to set d_{th} equal to the maximum allowable distance between the sender and the receiver in a D2D pair. In addition, the gain from $\varepsilon(i)$ to u_j can be derived as follows [33]:

$$g_{\varepsilon(i),j} = 10^{-\Phi_{\varepsilon(i),j}/10}, \quad (8)$$

where $\Phi_{\varepsilon(i),j}$ denotes the amount of path loss from $\varepsilon(i)$ to u_j . It can be calculated by

$$\Phi_{\varepsilon(i),j} = 148 + 40 \log_{10} L(\varepsilon(i), u_j), \quad (9)$$

where $L(\varepsilon(i), u_j)$ is the distance from $\varepsilon(i)$ to u_j (in kilometers). In our simulation, d_{th} is set to 30 m. Based on Eqs. (8) and (9), we can obtain that $g_{th} \approx 1.9567 \times 10^{-9}$.

Theorem 1. Algorithm 1 has a time complexity of $O(\eta_D(\eta_U - 1) + \eta_R(T_{PC} + \delta))$, where $\eta_D = |\mathcal{D}|$, $\eta_U = |\mathcal{C} \cup \mathcal{D}|$, $\eta_R = |\mathcal{R}|$, and T_{PC} is the amount of time taken by the power control procedure.

Proof: The code in lines 1–4 takes $O(\eta_D(\eta_U - 1))$ time. The for-loop in lines 5–20 allots RBs and power for UEs of each SP, so its computation time is similar to that of handling all UEs in $\mathcal{C} \cup \mathcal{D}$. In line 6, sorting $\mathcal{C}_k$ and $\mathcal{D}_k$ sets (for all SPs) spends time of no more than $O(|\mathcal{C}| \log_2 |\mathcal{C}| + |\mathcal{D}| \log_2 |\mathcal{D}|)$. The code in lines 7–10 consumes $O(|\mathcal{C}| \eta_R)$ time, and that in lines 11–14 requires $O(|\mathcal{D}| \eta_R)$ time. Since $|\mathcal{Q}_x| \leq \delta$, the code in lines 15–20 takes $O(\eta_R(T_{PC} + \delta))$ time. Because $|\mathcal{C}| + |\mathcal{D}| = \eta_U$, $\eta_R < \eta_U$, and $\eta_C \leq \eta_D$, we obtain that $\eta_D(\eta_U - 1) > \max\{|\mathcal{C}| \log_2 |\mathcal{C}|, |\mathcal{D}| \log_2 |\mathcal{D}|, |\mathcal{C}| \eta_R, |\mathcal{D}| \eta_R\}$. Thus, the time complexity can be simplified to $O(\eta_D(\eta_U - 1) + \eta_R(T_{PC} + \delta))$. $\square$

Once an SP s_k does not have enough RBs to serve all of its UEs (i.e., $\mathcal{C}'_k \cup \mathcal{D}'_k \neq \emptyset$), the IRS algorithm helps s_k borrow RBs from other SPs to serve these UEs (i.e., the RB lending mechanism). Algorithm 2 presents IRS's pseudocode. Line 1 invokes the SP prioritization procedure to obtain a sorted set $\mathcal{S}_B$. It determines the order in which each SP can borrow RB from others. As discussed later in Section 4.2, the SP prioritization procedure has two different modes. Hence, we maintain a global variable *mode*, whose value is set to zero in the beginning, and pass this variable as the input parameter of the SP prioritization procedure. Afterward, the subsequent for-loop iteratively picks an SP s_k from $\mathcal{S}_B$. In line 3, we call the lender selection procedure to acquire a sorted set $\mathcal{S}_L$ that decides the order in which SPs lend RBs to s_k . Obviously, $\mathcal{S}_L$ does not include s_k .

Like BRA, line 4 sorts CUs in $\mathcal{C}'_k$ and DUs in $\mathcal{D}'_k$ based on their gains from the BS decreasingly. Algorithm 2 first handles CUs in $\mathcal{C}'_k$ (in lines 5–13), followed by DUs in $\mathcal{D}'_k$ (in lines 14–22). For each CU u_i in $\mathcal{C}'_k$, we iteratively pick an SP s_m from $\mathcal{S}_L$ and find an RB r_x from $\mathcal{R}_m$ (i.e., the set of s_m 's RBs). Then, r_x is a candidate RB used to serve u_i if three conditions all hold: 1) r_x is not assigned to any CU yet (i.e., $\mathcal{Q}_x \cap \mathcal{C} = \emptyset$), 2) r_x still has room for serving u_i (i.e., $|\mathcal{Q}_x| < \delta$), and 3) there is no u_i 's neighbor also using r_x (i.e., $\zeta_{i,j} = 0, \forall u_j \in \mathcal{Q}_x$). If so, line 9 calls the power control procedure to recalculate power for each UE in $\mathcal{Q}_x \cup \{u_i\}$. However, if the power control procedure cannot allocate power for some UEs in $\mathcal{Q}_x \cup \{u_i\}$ (i.e., $\Theta \neq \emptyset$) or some UEs in $\mathcal{Q}_x \cup \{u_i\}$ cannot meet their SINR demands (i.e., $\sigma_j^x < \sigma_j^{\min}, \exists u_j \in \mathcal{Q}_x \cup \{u_i\}$), it is inappropriate to let u_i use r_x . In this case, we restore the original power for all UEs in $\mathcal{Q}_x \cup \{u_i\}$, as shown in line 11. Otherwise, u_i can safely use

Algorithm 2: Inter-SP Resource Sharing (IRS)

```

1  $\mathcal{S}_B \leftarrow \text{SP\_prioritization}(\text{mode});$ 
2 foreach  $s_k \in \mathcal{S}_B$  do
3    $\mathcal{S}_L \leftarrow \text{lender\_selection}(s_k);$ 
4    $\mathcal{C}'_k \leftarrow f_{\text{SRT}}^{\text{DEC}}(\mathcal{C}'_k, g_{\text{BS},i}), \mathcal{D}'_k \leftarrow f_{\text{SRT}}^{\text{DEC}}(\mathcal{D}'_k, g_{\text{BS},i});$ 
5   foreach  $u_i \in \mathcal{C}'_k$  do
6     foreach  $s_m \in \mathcal{S}_L$  do
7       foreach  $r_x \in \mathcal{R}_m$  do
8         if  $\mathcal{Q}_x \cap \mathcal{C} = \emptyset$  and  $|\mathcal{Q}_x| < \delta$  and  $\zeta_{i,j} = 0, \forall u_j \in \mathcal{Q}_x$  then
9            $\Theta \leftarrow \text{power\_control}(\mathcal{Q}_x \cup \{u_i\});$ 
10          if  $\Theta \neq \emptyset$  or  $\sigma_j^x < \sigma_j^{\min}, \exists u_j \in \mathcal{Q}_x \cup \{u_i\}$  then
11            Restore power for UEs in  $\mathcal{Q}_x \cup \{u_i\}$ ;
12          else
13            Move  $u_i$  from  $\mathcal{C}'_k$  to  $\mathcal{Q}_x$  and go to line 5;
14        foreach  $u_i \in \mathcal{D}'_k$  do
15          foreach  $s_m \in \mathcal{S}_L$  do
16            foreach  $r_x \in \mathcal{R}_m$  do
17              if  $|\mathcal{Q}_x| < \delta$  and  $\zeta_{i,j} = 0, \forall u_j \in \mathcal{Q}_x$  then
18                 $\Theta \leftarrow \text{power\_control}(\mathcal{Q}_x \cup \{u_i\});$ 
19                if  $\Theta \neq \emptyset$  or  $\sigma_j^x < \sigma_j^{\min}, \exists u_j \in \mathcal{Q}_x \cup \{u_i\}$  then
20                  Restore power for UEs in  $\mathcal{Q}_x \cup \{u_i\}$ ;
21                else
22                  Move  $u_i$  from  $\mathcal{D}'_k$  to  $\mathcal{Q}_x$  and go to line 14;
23  $\text{mode} \leftarrow (\text{mode} + 1) \% 2;$ 

```

r_x . Hence, we move u_i from $\mathcal{C}'_k$ to $\mathcal{Q}_x$ and go back to line 5 to pick the next UE in $\mathcal{C}'_k$ to serve.

The way to handle DUs of $\mathcal{D}'_k$ by lines 14–22 is similar to that to handle CUs, except that we check only two conditions, $|\mathcal{Q}_x| < \delta$ and $\zeta_{i,j} = 0, \forall u_j \in \mathcal{Q}_x$, in line 17. Then, line 23 changes the value of variable *mode*. As can be seen, this value will alternate between zero and one every time when the IRS algorithm is used to carry out the RB lending mechanism.

In Remark 2, we discuss how to adapt the IRS algorithm to the situation when orthogonality constraints on some RBs are applied. Then, Theorem 2 analyzes IRS's time complexity.

Remark 2 (RB orthogonality constraints). As mentioned in Section 3, we assume that all RBs are treated as sharable units when performing RB sharing between SPs. In practice, some adjacent RBs may have orthogonality constraints, making them non-sharable. To adapt the IRS algorithm to this situation, in both lines 7 and 16 of Algorithm 2, we can replace the set $\mathcal{R}_m$ (i.e., the set of all RBs of SP s_m) by another set $\mathcal{R}_{m,k}^{\text{SHA}}$. Here, $\mathcal{R}_{m,k}^{\text{SHA}}$ is a subset of *sharable* RBs in $\mathcal{R}_m$ that SP s_m can lend to SP s_k .

Theorem 2. Let T_{SP} and T_{LS} be the amount of time required by the SP prioritization and lender selection procedures. Given η_S SPs in $\mathcal{S}$, the time complexity of Algorithm 2 is $T_{\text{SP}} + \eta_S T_{\text{LS}} + O(\eta_R \eta_U (T_{\text{PC}} +$

Algorithm 3: RB Throughput Improvement (RTI)

```

1 foreach  $r_x \in \mathcal{R}$  do
2   foreach  $u_i \in \mathcal{Q}_x$  do
3      $\tilde{g}_i^{\text{sum}} \leftarrow \sum_{u_j \in \mathcal{Q}_x \setminus \{u_i\}} g_{\varepsilon(i),j}$ ;
4      $\mathcal{Q}_x \leftarrow f_{\text{SRT}}^{\text{INC}}(\mathcal{Q}_x, \tilde{g}_i^{\text{sum}})$ ;
5     foreach  $u_i \in \mathcal{Q}_x$  do
6       if  $f_{\text{CQI}}(p_i^{\text{max}}) = f_{\text{CQI}}(p_{\varepsilon(i),i}^x)$  then
7         continue;
8        $\alpha \leftarrow 0, \alpha_{\text{upd}} \leftarrow 0, \beta \leftarrow (p_i^{\text{max}} - p_{\varepsilon(i),i}^x) / \alpha_{\text{max}}$ ;
9       while  $\alpha < \alpha_{\text{max}}$  do
10         $\alpha \leftarrow \alpha + 1$ ;
11        if  $f_{\text{CQI}}(p_{\varepsilon(i),i}^x + \alpha\beta) = f_{\text{CQI}}(p_{\varepsilon(i),i}^x + (\alpha - 1)\beta)$ 
12          then
13            continue;
14         $\Gamma_{\text{pre}} \leftarrow \sum_{u_j \in \mathcal{Q}_x} \gamma_j^x$ ;
15         $p_{\varepsilon(i),i}^x \leftarrow p_{\varepsilon(i),i}^x + \alpha\beta, \Gamma_{\text{alt}} \leftarrow \sum_{u_j \in \mathcal{Q}_x} \gamma_j^x$ ;
16        if  $\Gamma_{\text{alt}} \leq \Gamma_{\text{pre}}$  or  $\sigma_j^x < \sigma_j^{\text{min}}, \exists u_j \in \mathcal{Q}_x$  then
17          restore  $p_{\varepsilon(i),i}^x$  and break;
18        else
19          restore  $p_{\varepsilon(i),i}^x, \alpha_{\text{upd}} \leftarrow \alpha$ ;
20       $p_{\varepsilon(i),i}^x \leftarrow p_{\varepsilon(i),i}^x + \alpha_{\text{upd}} \times \beta$ ;

```

δ)).

Proof: Line 1 obviously spends T_{SP} time. The for-loop in lines 2–22 repeats η_S times. Line 3 takes T_{LS} time to run the lender selection procedure. In line 4, sorting $\mathcal{C}'_k$ and $\mathcal{D}'_k$ requires $O(|\mathcal{C}'_k| \log_2 |\mathcal{C}'_k|)$ and $O(|\mathcal{D}'_k| \log_2 |\mathcal{D}'_k|)$ time. Since s_k is excluded from $\mathcal{S}_L$, we have $|\mathcal{S}_L| = \eta_S - 1$. Line 9 takes T_{PC} time to execute the power control procedure, and line 10 consumes $O(\delta)$ time (as $|\mathcal{Q}_x \cup \{u_i\}| \leq \delta$). Hence, the triple for-loop in lines 5–13 spends time of $|\mathcal{C}'_k|(\eta_S - 1)|\mathcal{R}_m|O(T_{\text{PC}} + \delta)$. Similarly, the triple for-loop in lines 14–22 takes time of $|\mathcal{D}'_k|(\eta_S - 1)|\mathcal{R}_m|O(T_{\text{PC}} + \delta)$. Thus, the time complexity is $T_{\text{SP}} + \eta_S(T_{\text{LS}} + O(|\mathcal{C}'_k| \log_2 |\mathcal{C}'_k|) + O(|\mathcal{D}'_k| \log_2 |\mathcal{D}'_k|) + |\mathcal{C}'_k|(\eta_S - 1)|\mathcal{R}_m|O(T_{\text{PC}} + \delta) + |\mathcal{D}'_k|(\eta_S - 1)|\mathcal{R}_m|O(T_{\text{PC}} + \delta))$. Since $(\eta_S - 1)|\mathcal{C}'_k| < |\mathcal{C}|$, $(\eta_S - 1)|\mathcal{D}'_k| < |\mathcal{D}|$, $\eta_S|\mathcal{R}_m| \leq \eta_R$, and $|\mathcal{C}| + |\mathcal{D}| = \eta_U$, we can simplify the time complexity to $T_{\text{SP}} + \eta_S T_{\text{LS}} + O(\eta_R \eta_U (T_{\text{PC}} + \delta))$. $\square$

For each RB $r_x \in \mathcal{R}$, the RTI algorithm (whose pseudocode is shown in Algorithm 3) increases sender power for UEs in $\mathcal{Q}_x$ to improve their throughput on r_x . Since raising power would increase interference to other UEs, we prioritize increasing power for UEs with less impact on others. Lines 2–4 give the code, where $\tilde{g}_i^{\text{sum}}$ is the sum of gains on all UEs in $\mathcal{Q}_x$ (excluding u_i) from u_i 's sender $\varepsilon(i)$. A smaller $\tilde{g}_i^{\text{sum}}$ value implies that $\varepsilon(i)$ has less impact. Hence, line 4 employs this value to sort UEs in $\mathcal{Q}_x$ increasingly, as denoted by $f_{\text{SRT}}^{\text{INC}}(\mathcal{Q}_x, \tilde{g}_i^{\text{sum}})$.

The for-loop in lines 5–19 iteratively picks a UE u_i from $\mathcal{Q}_x$ and seeks to raise the power of u_i 's sender. In effect, the number of data bits that RB r_x can carry for UE u_i depends on the *modulation and coding scheme* (MCS), which is decided by u_i 's SINR. Taking LTE-A as an example, Table 4 lists MCSs along with their minimum required SINRs [34]. Each MCS is associated with a *channel quality indicator* (CQI). A larger CQI value indicates that the MCS is more complex, allowing r_x to carry more data bits (in other words, u_i 's throughput can

improve), but it needs a higher SINR.

Let $f_{\text{CQI}}(p_{\varepsilon(i),i}^x)$ denote the best CQI that r_x can use for u_i as the power of $\varepsilon(i)$ (i.e., u_i 's sender) is $p_{\varepsilon(i),i}^x$. If $f_{\text{CQI}}(p_i^{\text{max}})$ is equal to $f_{\text{CQI}}(p_{\varepsilon(i),i}^x)$, no matter how we raise $\varepsilon(i)$'s power, the CQI cannot increase. Since adjusting $\varepsilon(i)$'s power is meaningless, we skip u_i , as shown in lines 6–7. Otherwise, we gradually increase $\varepsilon(i)$'s power and check if UE throughput on r_x can improve. More concretely, $\varepsilon(i)$ is allowed to raise power by at most $p_i^{\text{max}} - p_{\varepsilon(i),i}^x$ which we divide into α_{max} equal parts (i.e., β) and $\alpha_{\text{max}} > 1$. In addition, we use α_{upd} to record the largest α index that can effectively increase $\varepsilon(i)$'s power, as presented in line 8.

In the while-loop (i.e., lines 9–18), we increase the α index by one each time (until reaching the maximum value α_{max}). Each α index corresponds to the case of raising $\varepsilon(i)$'s power by an amount of $\alpha\beta$. The code in lines 11–12 is used to skip those α indices such that r_x 's CQI does not change. In the case that r_x 's CQI increases, we execute the code in lines 13–18. Let Γ_{pre} and Γ_{alt} be r_x 's throughput before and after increasing $\varepsilon(i)$'s power by an amount of $\alpha\beta$, where γ_j^x is the number of data bits that a UE u_j acquires from r_x . Line 15 gives two conditions to leave the while-loop: 1) raising $\varepsilon(i)$'s power has no benefit (i.e., $\Gamma_{\text{alt}} \leq \Gamma_{\text{pre}}$) and 2) doing so makes some UEs in $\mathcal{Q}_x$ fail to meet their SINR demands (i.e., $\sigma_j^x < \sigma_j^{\text{min}}, \exists u_j \in \mathcal{Q}_x$). If the check on line 15 can pass (that is, neither of the above two conditions holds), we store the current α index to α_{upd} in line 18. Since $p_{\varepsilon(i),i}^x$ is temporarily increased for computing Γ_{alt} , we need to restore $p_{\varepsilon(i),i}^x$ in lines 16 and 18. After exiting the while-loop, line 19 officially increases $\varepsilon(i)$'s power by an amount of $\alpha_{\text{upd}} \times \beta$. Notice that if no suitable α index can be found by the while-loop, we have $\alpha_{\text{upd}} = 0$. In this case, line 19 will keep $\varepsilon(i)$'s original power.

Let us give an example in Table 5 to demonstrate how the RTI algorithm works, where $\alpha_{\text{max}} = 5$. Since RB r_x 's CQI is 9 for α indices 0, 1, and 2, the code in lines 10–12 increases the α index to 3. Then, we run the code in lines 13–18. As the check on line 15 can pass, line 18 sets $\alpha_{\text{upd}} = 3$. After increasing the α index to 4 by line 10, since r_x 's CQI changes from 10 to 11, the code in lines 13–18 is carried out. However, the check on line 15 cannot pass, so we jump out of the while-loop. Finally, r_x 's power is increased by an amount of $\alpha_{\text{upd}} \times \beta = 3 \times (p_i^{\text{max}} - p_{\varepsilon(i),i}^x) / 5$. Theorem 3 discusses the time complexity of the RTI algorithm.

Theorem 3. The time complexity of Algorithm 3 is $O(\alpha_{\text{max}} \eta_R \delta^2)$.

Proof: The outermost for-loop runs η_R times. Since $|\mathcal{Q}_x| \leq \delta$, the code in lines 2–3 takes $O(\delta^2)$ time, and line 4 uses $O(\delta \log_2 \delta)$ time to sort $\mathcal{Q}_x$. The for-loop in lines 5–19 has δ iterations, and the while-loop runs at most α_{max} times. Since lines 13, 14, and 15 take $O(\delta)$ time, the code in lines 5–19 needs $O(\delta(\alpha_{\text{max}} \delta))$ time. Hence, the time complexity is $\eta_R O(\delta^2 + \delta \log_2 \delta + \delta(\alpha_{\text{max}} \delta)) = O(\alpha_{\text{max}} \eta_R \delta^2)$. $\square$

4.2 Auxiliary Procedures

Given a set $\mathcal{Q}_x$ of UEs that share RB r_x , the power control procedure finds the amount of power for the sender $\varepsilon(i)$ of each UE u_i in $\mathcal{Q}_x$ such that u_i 's SINR demand (i.e., σ_i^{min}) is met while $\varepsilon(i)$'s interference can be reduced. This procedure returns a subset Θ of DUs in $\mathcal{Q}_x$ whose senders cannot be allocated power, which is initialized to $\emptyset$ in line 1. Then, the for-loop in lines 2–5 estimates the initial power (denoted by p_i) for u_i 's sender. More concretely, if u_i is a CU, we optimistically

TABLE 4: MCS supported by each CQI and its minimum required SINR.

CQI	MCS (code rate)	SINR	CQI	MCS (code rate)	SINR
1	QPSK (78/1024)	-6.936 dB	9	16QAM (616/1024)	8.573 dB
2	QPSK (120/1024)	-5.147 dB	10	64QAM (466/1024)	10.366 dB
3	QPSK (193/1024)	-3.180 dB	11	64QAM (567/1024)	12.289 dB
4	QPSK (308/1024)	-1.253 dB	12	64QAM (666/1024)	14.173 dB
5	QPSK (449/1024)	0.761 dB	13	64QAM (772/1024)	15.888 dB
6	QPSK (602/1024)	2.699 dB	14	64QAM (873/1024)	17.814 dB
7	16QAM (378/1024)	4.694 dB	15	64QAM (948/1024)	19.829 dB
8	16QAM (490/1024)	6.525 dB			

TABLE 5: An example to show how the RTI algorithm works.

α index	0	1	2	3	4	5
r_x 's CQI	9	9	9	10	11	12
check on line 15	-	-	-	pass	not pass	-

Procedure `power_control`(Q_x):

```

1   $\Theta \leftarrow \emptyset$ ;
2  foreach  $u_i \in Q_x$  do
3    Estimate initial power  $p_i$  for  $u_i$ 's sender;
4    if  $u_i \in \mathcal{D}$  and  $p_i > p_i^{\max}$  then
5       $\Theta \leftarrow \Theta \cup \{u_i\}$ ;
6   $Q'_x \leftarrow Q_x \setminus \Theta$ ;
7  foreach  $u_i \in Q'_x$  do
8     $\tilde{g}_i^{\text{sum}} \leftarrow \sum_{u_j \in Q_x \setminus \{u_i\}} g_{\varepsilon(i),j}$ ;
9   $Q'_x \leftarrow f_{\text{SRT}}^{\text{INC}}(Q'_x, \tilde{g}_i^{\text{sum}})$ ;
10 foreach  $u_i \in Q'_x$  do
11    $\Lambda_i^{\text{LB}} \leftarrow \max\{p_i, p_i^{\min}\}$ ,  $\Lambda_i^{\text{UB}} \leftarrow p_i^{\max}$ ;
12   do
13      $P_L \leftarrow \Lambda_i^{\text{LB}} + (\Lambda_i^{\text{UB}} - \Lambda_i^{\text{LB}}) \times 1/4$ ;
14      $P_M \leftarrow \Lambda_i^{\text{LB}} + (\Lambda_i^{\text{UB}} - \Lambda_i^{\text{LB}}) \times 1/2$ ;
15      $P_H \leftarrow \Lambda_i^{\text{LB}} + (\Lambda_i^{\text{UB}} - \Lambda_i^{\text{LB}}) \times 3/4$ ;
16     if  $\sigma_i^x \geq \sigma_i^{\min}$  and decrement in  $\sigma_i^x < 50\%$  when
        using  $P_L$  then
17        $p_{\varepsilon(i),i}^x \leftarrow P_L$ ,  $\Lambda_i^{\text{UB}} \leftarrow \Lambda_i^{\text{UB}} \times 3/4$ ;
18     else if increment in  $\sigma_i^x > 50\%$  when using  $P_H$ 
        then
19        $p_{\varepsilon(i),i}^x \leftarrow P_H$ ,  $\Lambda_i^{\text{LB}} \leftarrow \Lambda_i^{\text{LB}} \times 5/4$ ;
20     else
21        $p_{\varepsilon(i),i}^x \leftarrow P_M$ ;
22        $\Lambda_i^{\text{UB}} \leftarrow \Lambda_i^{\text{UB}} \times 3/4$ ,  $\Lambda_i^{\text{LB}} \leftarrow \Lambda_i^{\text{LB}} \times 5/4$ ;
23   while  $P_H - P_L \geq \rho$ ;
24 return  $\Theta$ ;

```

assume that it is interfered with by merely thermal noise φ . Hence, p_i can be derived as follows:

$$(g_{\text{BS},i} \times p_i) / \varphi \geq \sigma_i^{\min} \Rightarrow p_i = (\sigma_i^{\min} \times \varphi) / g_{\text{BS},i}. \quad (10)$$

When u_i is a DU, we additionally take account of interference from the BS (using the minimum power to a CU u_j):

$$\frac{g_{\varepsilon(i),i} \times p_i}{g_{\text{BS},i} \times p_j^{\min} + \varphi} \geq \sigma_i^{\min} \Rightarrow p_i = \frac{\sigma_i^{\min} (g_{\text{BS},i} \times p_j^{\min} + \varphi)}{g_{\varepsilon(i),i}}. \quad (11)$$

However, if $p_i > p_i^{\max}$, it means that $\varepsilon(i)$'s power cannot be allocated, so we add u_i to Θ . The code is given in lines 4–5. Let $Q'_x \subseteq Q_x$ be the set of UEs whose sender power is allocatable (i.e., line 6). Since raising power may increase interference with other UEs, we prioritize increasing power for UEs with less effect. To do so, the for-loop in lines 7–8 computes the sum of

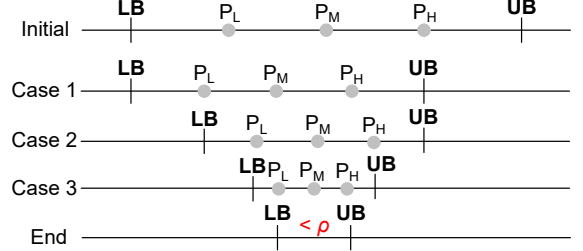


Fig. 2: Ternary search in power control (LB/UB: lower/upper bound).

gains $\tilde{g}_i^{\text{sum}}$ on all UEs in Q_x (excluding u_i) by u_i 's sender $\varepsilon(i)$, and line 9 then sorts Q'_x increasingly by the $\tilde{g}_i^{\text{sum}}$ value.

The subsequent for-loop allocates power for the sender of each UE u_i in Q'_x . It is based on a ternary search, as shown in Fig. 2. Line 11 sets a power lower bound Λ_i^{LB} to the maximum of p_i (i.e., the initial power) and $p_i^{\min}$ (i.e., the minimal power) and a power upper bound Λ_i^{UB} to $p_i^{\max}$. Moreover, we set three power values, P_L , P_M , and P_H , located at $1/4$, $1/2$, and $3/4$ of the range from Λ_i^{LB} to Λ_i^{UB} , in lines 13–15. Then, three cases are adopted to adjust power. In case 1 (i.e., lines 16–17), u_i can meet its SINR demand by using a lower power, and the decrement in σ_i^x is less than a half. Thus, it is safe to set $p_{\varepsilon(i),i}^x = P_L$, and we lower the power upper bound Λ_i^{UB} . In case 2 (i.e., lines 18–19), using a high power can significantly improve u_i 's SINR, so we set $p_{\varepsilon(i),i}^x$ to P_H and raise the power lower bound Λ_i^{LB} . In case 3 (i.e., lines 20–22; the remaining case), we set $p_{\varepsilon(i),i}^x$ to P_M (i.e., the medium power) and shrink the range from Λ_i^{LB} to Λ_i^{UB} . The above iterations are repeated until the difference between P_H and P_L is below a threshold ρ . Then, line 24 returns Θ .

The convergence of the ternary search is the subject of Remark 3. Remark 4 discusses why we choose to use the ternary search in the power control procedure. Then, Theorem 4 analyzes the time complexity of the power control procedure.

Remark 3 (convergence of ternary search). The ternary search is done by the while-loop in lines 12–23 of the power control procedure, where the termination condition is $P_H - P_L < \rho$. Whenever an iteration of the while-loop is carried out, one of the three cases must be performed. Specifically, case 1 lowers Λ_i^{UB} in line 17, thereby decreasing P_H . Case 2 raises Λ_i^{LB} in line 19, which increases P_L . Case 3 decreases P_H and increases P_L by line 22. In other words, these three cases keep reducing the range from P_L to P_H . Eventually, the difference between P_L and P_H will fall below threshold ρ , which breaks the while-loop. This verifies that the ternary search must converge. Actually, the convergence speed is fast due to its low time complexity, as discussed later in Theorem 4.

Remark 4 (choice of ternary search). The ternary search helps

Procedure SP_prioritization(*mode*):

```

1  foreach  $s_k \in \mathcal{S}$  do
2       $\Omega_k^{\text{sum}} \leftarrow 0$ ;
3      foreach  $r_x \in \mathcal{R}_k$  do
4          foreach  $s_m \in \mathcal{S} \setminus \{s_k\}$  do
5               $\Omega_k^{\text{sum}} \leftarrow \Omega_k^{\text{sum}} + \Omega_{k,m}^x$ ;
6  foreach  $s_k \in \mathcal{S}$  do
7       $\Psi_k^{\text{sum}} \leftarrow 0$ ;
8      foreach  $s_m \in \mathcal{S} \setminus \{s_k\}$  do
9          foreach  $r_x \in \mathcal{R}_m$  do
10              $\Psi_k^{\text{sum}} \leftarrow \Psi_k^{\text{sum}} + \Omega_{m,k}^x$ ;
11  if  $mode = 0$  then
12      return  $f_{\text{SRT}}^{\text{DEC}}(\mathcal{S}, \Omega_k^{\text{sum}})$ ;
13  return  $f_{\text{SRT}}^{\text{INC}}(\mathcal{S}, \Psi_m^{\text{sum}})$ ;

```

quickly find a suitable power for the sender of each UE $u_i \in \mathcal{Q}_x$ such that u_i 's SINR demand can be met while the interference is reduced. There are some other methods, such as gradient-based or convex optimization methods, that may find out more accurate sender power for u_i . However, these methods usually incur high computation costs. In the power control procedure, we choose to use the ternary search instead of these methods due to two reasons. First, the power control procedure is frequently called by both BRA and IRS algorithms. Using gradient-based or convex optimization methods will greatly increase the time complexity of our DACS framework. Second, after allocating RBs to CUs and D2D pairs, the RTI algorithm will adjust their sender power (not using the power control procedure) to improve throughput. As a result, finding an accurate power value for each UE in the power control procedure becomes unnecessary.

Theorem 4. The power control procedure takes $O(\delta^2)$ time.

Proof: Since $|\mathcal{Q}_x| \leq \delta$, the for-loop in lines 2–5 takes $O(\delta)$ time, and the for-loop in lines 7–8 spends $O(\delta^2)$ time. Sorting $\mathcal{Q}'_x$ in line 9 takes no more than $O(\delta \log_2 \delta)$ time. The for-loop in 10–23 repeats at most δ times. The while-loop in lines 12–23 needs $O(\log_3((p_i^{\text{max}} - p_i^{\text{min}})/\varsigma))$ time to carry out a ternary search, where ς is the minimum adjustable power range. Because $\log_3((p_i^{\text{max}} - p_i^{\text{min}})/\varsigma) < \delta$, the time complexity is $O(\delta + \delta^2 + \delta \log_2 \delta + \delta(\log_3((p_i^{\text{max}} - p_i^{\text{min}})/\varsigma))) = O(\delta^2)$. $\square$

The SP prioritization procedure finds the order of SPs that can borrow RBs from others. Given an RB r_x in $\mathcal{R}_k$ (i.e., the set of SP s_k 's RBs), we define the *resource amount* of r_x that s_k lends to another SP s_m as follows:

$$\Omega_{k,m}^x = (|\mathcal{Q}_x \cap (\mathcal{C}_m \cup \mathcal{D}_m)|) / |\mathcal{Q}_x|, \quad (12)$$

where the numerator is the number of s_m 's UEs using r_x and the denominator gives the number of total UEs that share r_x . For example, suppose that there are 15 UEs sharing r_x , six of which belong to s_m . Then, the resource amount of r_x that s_k lends to s_m is $6/15 = 0.4$.

There are two modes in the SP prioritization procedure (which mode is used will be decided by the input parameter *mode*):

- ◊ **[Mode 0]** Lines 1–5 and 11–12: When s_k has lent more resources to others (i.e., with a larger Ω_k^{sum} value), it

Procedure lender_selection(s_k):

```

1  foreach  $s_m \in \mathcal{S} \setminus \{s_k\}$  do
2       $\Psi_m \leftarrow 0$ ;
3      foreach  $r_x \in \mathcal{R}_k$  do
4           $\Psi_m \leftarrow \Psi_m + \Omega_{k,m}^x$ ;
5  return  $f_{\text{SRT}}^{\text{DEC}}(\mathcal{S} \setminus \{s_k\}, \Psi_m)$ ;

```

has a higher priority to borrow RBs. This mode is to encourage SPs to lend their RBs.

- ◊ **[Mode 1]** Lines 6–10 and 13: If s_k has borrowed fewer resources from other SPs (i.e., with a smaller Ψ_m^{sum} value), it can have a higher priority to borrow RBs. This mode is for fairness consideration.

Remark 5 discusses what triggers the mode switch, and Theorem 5 analyzes the time complexity of the SP prioritization procedure.

Remark 5 (mode switch). The three core algorithms (i.e., BRA, IRS, and RTI) of the DACS framework will be executed once in sequence during each scheduling period (usually a TTI, whose length is 1 ms). However, if every SP in $\mathcal{S}$ has enough RBs to serve all of its UEs in a scheduling period, there is no need to perform the IRS algorithm in that period to reduce the computation cost. Evidently, only the IRS algorithm can invoke the SP prioritization procedure. Based on line 1 in Algorithm 2, IRS will pass a global variable *mode* as the input parameter to the SP prioritization procedure, which decides the mode used. Moreover, when the IRS algorithm terminates, it changes the mode in line 23 (where the value of the variable *mode* alternates between zero and one). This triggers a mode switch for the SP prioritization procedure the next time it is invoked again. As a result, the switching is based on the time interval between two consecutive executions of the IRS algorithm. In general, there are usually SPs that do not have enough resources for serving their UEs, so the IRS algorithm will be executed periodically. This makes the mode switch in the SP prioritization procedure occur every TTI.

Theorem 5. The SP prioritization procedure takes $O(\eta_R(\eta_S - 1))$ time.

Proof: The for-loop in lines 1–5 has η_S iterations. It spends $O(|\mathcal{R}_k|(\eta_S - 1))$ time to execute the double for-loop in lines 3–5. Because $\eta_S |\mathcal{R}_k| = \eta_R$ (i.e., the total RBs in $\mathcal{R}$), the code in lines 1–5 takes $O(\eta_R(\eta_S - 1))$ time. Similarly, the triple for-loop in lines 6–10 requires $O(\eta_R(\eta_S - 1))$ time. Then, sorting $\mathcal{S}$ in lines 12 and 13 needs $O(\eta_S \log_2 \eta_S)$ time. Hence, the time complexity is $2O(\eta_R(\eta_S - 1)) + O(\eta_S \log_2 \eta_S) = O(\eta_R(\eta_S - 1))$. $\square$

Given an SP s_k , the lender selection procedure determines the order of the other SPs in $\mathcal{S}$ that lend RBs to s_k . In general, when an SP s_m has borrowed more resources from s_k , s_m should lend back its RBs to s_k first. This approach enables us to more effectively ensure that SPs are treated fairly. In the lender selection procedure, for every SP s_m in $\mathcal{S}$ (except s_k), the for-loop in lines 1–4 computes the total resource amount (denoted by Ψ_m) that s_m has borrowed from s_k . Then, line 5 sorts all SPs in $\mathcal{S} \setminus \{s_k\}$ decreasingly by their Ψ_m values and returns the sorted result. Theorem 6 gives an analysis of the time complexity for the lender selection procedure.

Theorem 6. Let η_K be the maximum number of RBs that an SP can own in $\mathcal{R}$, where $\eta_K < \eta_R$. Then, the lender selection procedure requires $O(\eta_K(\eta_S - 1))$ time.

Proof: The outer for-loop in lines 1–4 runs $(\eta_S - 1)$ times. The inner for-loop in lines 3–4 takes at most $O(\eta_K)$ time (as $|\mathcal{R}_k| \leq \eta_K$). Then, it takes $O((\eta_S - 1) \log_2(\eta_S - 1))$ time to do sorting in line 5. As $\eta_K > \log_2(\eta_S - 1)$, the time complexity is $O(\eta_K(\eta_S - 1))$. $\square$

4.3 Design Rationale

DACS has three core algorithms, including BRA (Algorithm 1), IRS (Algorithm 2), and RTI (Algorithm 3). With the BRA algorithm, each SP s_k serves its UEs (i.e., $\mathcal{C}_k$ and $\mathcal{D}_k$) using its own RBs (i.e., $\mathcal{R}_k$). When s_k cannot serve all of its UEs, s_k may borrow RBs from other SPs via the IRS algorithm. In view that CUs usually perform conventional cellular communication while DUs may often be IoT devices [35], both BRA and IRS first allot RBs to CUs. Then, a DU can reuse an RB if there are no neighbors using that RB (to avoid interference) and the number of UEs sharing the RB is below δ (i.e., the constraint in Eq. (5)). Finally, for each RB $r_x \in \mathcal{R}$, the RTI algorithm aims to improve signal quality of UEs using r_x by increasing the power of their senders, thereby raising r_x 's throughput accordingly.

DACS also has three auxiliary procedures. Both BRA and IRS algorithms employ the power control procedure to find a suitable power for the sender of each UE u_i that shares an RB r_x . This power should meet u_i 's SINR demand (i.e., $\sigma_i^{\min}$), while the interference from u_i 's sender to other UEs sharing r_x can be reduced. To quickly compute this power, we design a ternary search. Then, the SP prioritization procedure as well as the lender selection procedure carry out the RB lending mechanism. The SP prioritization procedure provides two modes: 1) giving an SP that has lent more resources a high priority and 2) giving an SP that has borrowed fewer resources a high priority. This way, we could achieve a good balance between efficiency and SP fairness. For each SP s_k , the lender selection procedure asks an SP that has borrowed more resources from s_k to first “repay” RBs to s_k . Doing so can further improve the fairness of the RB lending mechanism. Theorem 7 analyzes the time complexity of our DACS framework.

Theorem 7. The time complexity of the DACS framework is $O(\eta_D(\eta_U - 1) + \eta_R\eta_U\delta^2)$.

Proof: By combining Theorems 1–6, we derive DACS's time complexity as $O(\eta_D(\eta_U - 1) + \eta_R\delta^2) + O(\eta_R(\eta_S - 1) + \eta_S\eta_K(\eta_S - 1) + \eta_R\eta_U\delta^2) + O(\alpha_{\max}\eta_R\delta^2)$. Since $\eta_U > \max\{\alpha_{\max}, \eta_S\}$ and $\eta_K < \eta_R$, we can simplify the time complexity to $O(\eta_D(\eta_U - 1) + \eta_R\eta_U\delta^2)$. $\square$

4.4 Issue Discussion

In this section, we discuss two issues in our DACS framework: 1) how to handle dynamic UE mobility and 2) how to adapt DACS in large-scale deployments (e.g., dense urban areas).

For the first issue, we assume that the moving speeds of UEs are low, as indicated in Section 3. For instance, pedestrians who carry UEs walk, or cars with UEs move downtown. Since the DACS framework is run in each scheduling period (typically one TTI, which lasts for 1 ms), the relative positions of UEs will not likely change during the period. As such, the channel quality calculation (in terms of interference and SINR) is essentially accurate. Nevertheless, the high mobility

TABLE 6: Simulation parameters.

BS parameters:	
cell range	750 m (channel bandwidth: 180 kHz)
maximum power	46 dBm (≈ 40 W)
RBs offered	90 RBs/TTI
number of SPs	3 (SP s_1, s_2, s_3 own 20, 30, 40 RBs)
UE parameters:	
D2D distance	maximum: 30 m
number of UEs	CUs: 60, D2D pairs: 30, 60, 90, 120, 150, 180
maximum power	23 dBm (≈ 0.2 W)
throughput demand	1 Mbps (i.e., $\sigma_i^{\min} \approx 17$ dB)
Channel parameters:	
propagation loss	urban macrocell model [37]
path loss	BS to UE: $128.1 + 37.6 \log_{10} L(\text{BS}, u_i)$ UE to UE: $148 + 40 \log_{10} L(\varepsilon(i), u_i)$
shadowing	log-normal distribution [38]
thermal noise	-174 dBm/Hz

of UEs may result in frequent network dynamics or even severe interference. This has an impact on the efficiency of resource utilization, especially for D2D communications [36]. In future work, we will investigate how to adapt our DACS framework to an environment with high UE mobility.

Regarding the second issue, the DACS framework is done by each BS *individually* to handle RB allocation and power control in its cell. Coordinating BSs in a centralized manner is unnecessary. As a result, we can state that BSs implement the DACS framework in a distributed fashion from the standpoint of the entire system. On the other hand, the core algorithms in DACS are run centrally from the viewpoint of a single BS. If a cell covers many CUs and D2D pairs, the centralized nature might cause a bottleneck (at that BS). To address this issue, we can provide a distributed alternative for some core algorithms. For the BRA algorithm, the code in lines 6–20 is used for each SP to allot the owned RBs to its UEs. The program of each SP can run this code in parallel. Moreover, for each RB $r_x \in \mathcal{R}$, the RTI algorithm adjusts the power for UEs that share r_x . To have a distributed alternative, the program of each SP s_k can be configured to perform power adjustment for UEs on merely s_k 's RBs (i.e., $\mathcal{R}_k$). By doing this, the bottleneck effect is lessened.

5 EXPERIMENTAL EVALUATION

Our simulation is built using MATLAB, and Table 6 presents its parameters. Let us consider a BS that provides 90 RBs per TTI, of which three SPs, s_1, s_2 , and s_3 , own 20, 30, and 40 RBs. Each SP takes charge of serving 20 CUs and 10–60 D2D pairs. In particular, we test the impact on SP fairness by making each SP have a different number of RBs but serve the same number of CUs and D2D pairs. The minimum power for the BS and a D2D sender is -40 dBm. The distance between two DUs in a D2D pair does not exceed 30 m. Regarding path loss, $L(\cdot, \cdot)$ is the distance between a sender and a receiver, as measured in kilometers. Shadowing fading is modeled through a log-normal distribution, whose mean and standard deviation are set to 0 dB and 8 dB, respectively.

The following methods are selected for comparison with our DACS framework:

- 1) *Joint mode selection and resource allocation (JMSRA)* [22]: JMSRA builds a D2D inter-user interference graph and adopts the matching theory to find the solution. As JMSRA does not consider RAN sharing, each SP can only use the owned RBs to serve its CUs and D2D pairs.

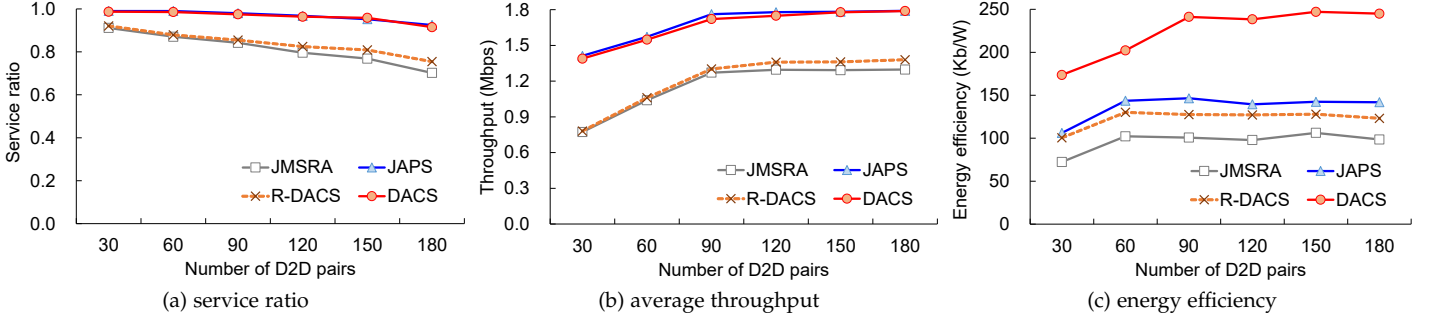


Fig. 3: Comparison of performance in the JMSRA, JAPS, R-DACS, and DACS methods.

- 2) *Joint resource allocation and power control with RAN sharing (JAPS)* [31]: JAPS adjusts the power of senders and allows SPs to borrow RBs from each other for increasing UE throughput.
- 3) *Reduced version of DACS (R-DACS)*: It is a combination of only BRA and RTI algorithms. We use R-DACS to evaluate the effect of the RB lending mechanism by the IRS algorithm on DACS.

Moreover, we set δ to 30. In other words, for each RB in $\mathcal{R}$, at most 30 UEs in $\mathcal{C} \cup \mathcal{D}$ can share it.

5.1 Comparison on Performance

First, we look at the service ratio, which is defined as follows:

$$\frac{\sum_{s_k \in \mathcal{S}} |\mathcal{C}_k \setminus \mathcal{C}'_k| + |\mathcal{D}_k \setminus \mathcal{D}'_k|}{\sum_{s_k \in \mathcal{S}} |\mathcal{C}_k| + |\mathcal{D}_k|}. \quad (13)$$

Specifically, the numerator denotes the number of CUs and D2D pairs successfully served by each SP (i.e., their SINR demands are satisfied), and the denominator is the total number of CUs and D2D pairs in the cell. Fig. 3(a) presents the experimental result. As can be seen, the service ratio will decrease as the number of D2D pairs grows, because more DUs compete for the fixed number of RBs. This phenomenon is particularly apparent in JMSRA and R-DACS.

JMSRA applies a matching solution to the D2D inter-user interference graph. R-DACS adopts the BRA algorithm for RB allocation and increases UE throughput using the RTI algorithm. Both methods serve a similar number of CUs and D2D pairs. In fact, R-DACS performs slightly better than JMSRA in terms of the service ratio, especially when there are more D2D pairs. However, JMSRA and R-DACS do not consider RAN sharing. When an SP does not have enough RBs (e.g., SP s_1), the SP has to leave some D2D pairs unserved. As a result, the service ratios of JMSRA and R-DACS drop to 0.7 and 0.75 when there are 180 D2D pairs. On the other hand, if an SP has insufficient RBs to serve all of its UEs, JAPS and DACS enable the SP to borrow RBs from others. Their service ratios are close to each other and keep above 0.91. This result demonstrates that both JAPS and DACS can exploit RAN sharing to increase the service ratio significantly.

Fig. 3(b) compares the average throughput of UEs. When the number of D2D pairs increases, the utilization of RBs may improve (as there are more D2D pairs sharing each RB). Therefore, the average throughput of UEs can increase as more D2D pairs are added to the network. However, the increase is less obvious when there are more than 90 D2D pairs, because network resources are approaching saturation.

TABLE 7: Ratio of UE throughput under each SP.

D2D pairs	JAPS	DACS
	$s_1 : s_2 : s_3$	$s_1 : s_2 : s_3$
60	10.51% : 31.53% : 57.96% (JFI = 0.6373)	29.53% : 31.07% : 39.41% (JFI = 0.8647)
120	15.05% : 24.00% : 60.95% (JFI = 0.6086)	27.83% : 30.32% : 41.86% (JFI = 0.8343)
180	13.95% : 24.08% : 61.97% (JFI = 0.5987)	29.35% : 31.73% : 38.91% (JFI = 0.8695)

Generally speaking, if a method has a higher service ratio, it can improve UE throughput accordingly. Hence, the trend in Fig. 3(b) is similar to that in Fig. 3(a), where JMSRA performs similarly with R-DACS while JAPS performs similarly with DACS. Thanks to RAN sharing, JAPS and DACS can substantially improve throughput compared to other methods.

Then, Fig. 3(c) measures the amount of energy efficiency, as calculated by

$$\frac{\sum_{u_i \in \mathcal{C} \cup \mathcal{D}} \sum_{r_x \in \mathcal{R}} \gamma_i^x z_i^x}{\sum_{u_i \in \mathcal{C} \cup \mathcal{D}} p_{\varepsilon(i), i}^x}. \quad (14)$$

Here, we assume that each UE in $\mathcal{C} \cup \mathcal{D}$ always has data to receive from the BS or its D2D sender (i.e., a full buffer model [39]). In Eq. (14), the numerator is the maximum number of data bits that each UE can obtain (i.e., γ_i^x) from its assigned RBs. Moreover, the denominator is the amount of power that the BS and all D2D senders spend on transmitting data to their receivers.

JMSRA considers fixed power (i.e., no power control). Though it has similar throughput with R-DACS, JMSRA results in lower energy efficiency than R-DACS. This phenomenon reflects the importance of power control. On the other hand, JAPS has similar throughput to DACS, but its energy efficiency is significantly lower than that of DACS. According to Eq. (14), this result proves that DACS performs much better than JAPS in power control (as senders can use less power to achieve similar throughput).

5.2 Comparison on Fairness

According to Table 1, only JAPS and DACS support RB lending mechanisms that enable SPs to borrow RBs from one another to improve the service ratio and throughput. In this section, we judge whether their RB lending mechanisms can maintain fairness among SPs.

Fig. 4 shows the amount of UE throughput under each SP, where there are 60, 120, and 180 D2D pairs. Among 90 RBs provided by the BS per TTI, SPs s_1 , s_2 , and s_3 own 20, 30, and 40 RBs, occupying around 22.22%, 33.33%, and 44.45% of the

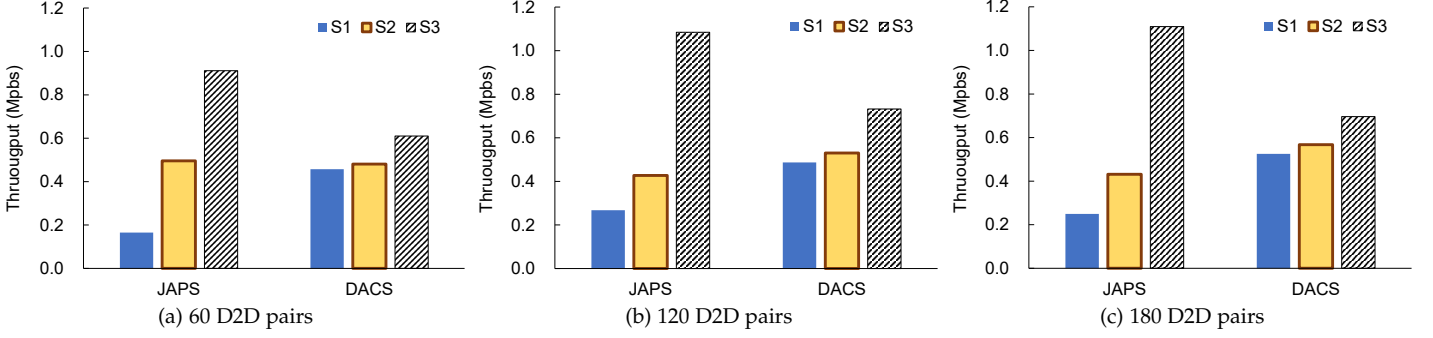


Fig. 4: Comparison of UE throughput under each SP in the JAPS and DACS methods.

BS's resources. As can be seen from Fig. 4, the gap between s_1 's throughput and s_3 's throughput is much larger in JAPS than that in DACS, particularly when there are more D2D pairs.

To conduct quantitative comparison on fairness more effectively, Table 7 displays the ratio of UE throughput under each SP. As mentioned in Section 1, one metric used to measure SP fairness is whether the ratio of UE throughput under each SP is close to that of the BS's resources owned by the SP. Thus, we compute the *Jain's fairness index (JFI)* for UE throughput under each SP by

$$\frac{(\sum_{s_k \in \mathcal{S}} \tilde{x}_k)^2}{|\mathcal{S}| \sum_{s_k \in \mathcal{S}} \tilde{x}_k^2} \text{ and } \tilde{x}_k = x_k \times \frac{|\mathcal{R}_k|}{|\mathcal{R}|}, \forall s_k \in \mathcal{S}, \quad (15)$$

where x_k is the amount of UE throughput under SP s_k , and $\tilde{x}_k$ denotes the amount of normalized throughput (based on the proportion of the BS's RBs owned by s_k , that is, $|\mathcal{R}_k|/|\mathcal{R}|$). The value of JFI is between $1/|\mathcal{S}|$ and 1, and a larger JFI value means that the system becomes fairer [40]. Evidently, DACS always has a significantly larger JFI value than JAPS. In other words, DACS makes the ratio of UE throughput under each SP closer to that of the BS's resources owned by the SP, as compared to JAPS. This result demonstrates that our DACS framework provides fairer UE throughput for SPs than JAPS.

As discussed in Section 1, another metric for qualifying SP fairness is whether the total resource amount that each SP borrows from others can be close to the total resource amount that the SP lends to others. Hence, Fig. 5 gives the aggregate resource amount that each SP borrows and lends, where the resource amount that an SP lends to another SP can be calculated by Eq. (12).

The RB lending mechanism in JAPS assigns a high priority to an SP that has lent more RBs to others. Since SP s_3 possesses the most RBs, it has a good chance of loaning out relatively more RBs in the early stage. This makes s_3 have a high priority for a long time. Eventually, s_3 will borrow many more RBs than it has lent. On the contrary, SP s_1 is forced to lend many more RBs than it borrows. These phenomena verify the unfairness of JAPS in RB lending.

In our DACS framework, the SP prioritization procedure considers two modes: 1) assigning a high priority to an SP that has lent more resources and 2) giving a high priority to an SP that has borrowed fewer resources. Through switching between these two modes periodically, DACS can efficiently address the above problem. Moreover, the lender selection procedure asks an SP that has borrowed more resources from SP s_k to first repay its RBs to s_k . As can be seen from Fig. 5, the resource amount that each SP borrows is similar to that it

lends, which verifies that DACS can maintain fairness among SPs.

6 CONCLUSION

In this paper, we propose the DACS framework to efficiently cope with resource allocation and power control for D2D communication in a RAN sharing scenario. Specifically, each SP first allots the owned RBs to its CUs and D2D pairs. To exploit the RAN sharing property, SPs are allowed to borrow RBs from one another using the RB lending mechanism. Then, DACS increases the power of senders to improve UE throughput while reducing the effect of interference. In DACS, the power control procedure can quickly find suitable power via a ternary search. Moreover, the RB lending mechanism takes account of both efficiency and SP fairness. Through simulations using MATLAB, we show that our DACS framework not only attains a high service ratio as well as high throughput but also improves energy efficiency as compared to the JMSRA, R-DACS, and JAPS methods. In addition, DACS is more effective than JAPS in maintaining fairness among SPs.

REFERENCES

- [1] M.S.M. Gismalla, A.I. Azmi, M.R.B. Salim, M.F.L. Abdullah, F. Iqbal, W.A. Mabrouk, M.B. Othman, A.Y.I. Ashyap, and A.S.M. Supa'at, "Survey on device to device (D2D) communication for 5G/6G networks: Concept, applications, challenges, and future directions," *IEEE Access*, vol.10, pp. 30792–30821, 2022.
- [2] P. Mach and Z. Becvar, "Device-to-device relaying: Optimization, performance perspectives, and open challenges towards 6G networks," *IEEE Comm. Surveys & Tutorials*, vol. 24, no. 3, pp. 1336–1393, 2022.
- [3] T. Islam and C. Kwon, "Survey on the state-of-the-art in device-to-device communication: A resource allocation perspective," *Ad Hoc Networks*, vol. 136, pp. 1–19, 2022.
- [4] Y.C. Wang and C.W. Chou, "Efficient coordination of radio frames to mitigate cross-link interference in 5G D-TDD systems," *Computer Networks*, vol. 232, pp. 1–13, 2023.
- [5] ETSI, "Network sharing; Architecture and functional description," 3GPP TS 23.251 V18.0.0, Mar. 2024.
- [6] X. Zhao, C. Hu, and Z. Li, "Multi-operator radio access network sharing for 5G SA network design and laboratory test," *Proc. Int'l Wireless Comm. and Mobile Computing Conf.*, 2021, pp. 187–193.
- [7] Y.C. Wang, "Resource and power management for in-band D2D communications," in *Horizons in Computer Science Research*, vol. 21, Nova Science Publishers, 2022, ch. 5, pp. 131–183.
- [8] F. Hussain, M.Y. Hassan, M.S. Hossen, and S. Choudhury, "An optimal resource allocation algorithm for D2D communication underlying cellular networks," *Proc. IEEE Consumer Comm. & Networking Conf.*, 2017, pp. 867–872.
- [9] T. Rathod and S. Tanwar, "Autoencoder-based efficient resource allocation in device-to-device communication," *Physical Comm.*, vol. 60, pp. 1–13, 2023.

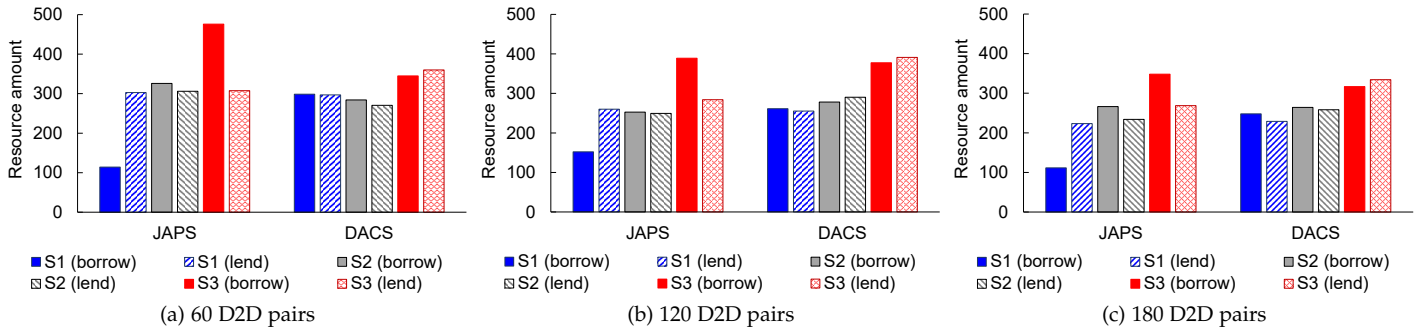


Fig. 5: Comparison of resource amount that each SP borrows and lends in the JAPS and DACS methods.

- [10] Z. Zhou, K. Ota, M. Dong, and C. Xu, "Energy-efficient matching for resource allocation in D2D enabled cellular networks," *IEEE Trans. Vehicular Technology*, vol. 66, no. 6, pp. 5256–5268, 2017.
- [11] R. Gour and A. Tyagi, "Semi-distributed resource management for underlay D2D communication with user's cooperation," *Int'l J. Comm. Systems*, vol. 33, no. 4, pp. 1–18, 2020.
- [12] Z. Liu, J. Su, Y.A. Xie, K. Ma, Y. Yang, and X. Guan, "Resource allocation in D2D enabled vehicular communications: A robust Stackelberg game approach based on price-penalty mechanism," *IEEE Trans. Vehicular Technology*, vol. 70, no. 8, pp. 8186–8200, 2021.
- [13] D. Shi, L. Li, T. Ohtsuki, M. Pan, Z. Han, and H.V. Poor, "Make smart decisions faster: Deciding D2D resource allocation via Stackelberg game guided multi-agent deep reinforcement learning," *IEEE Trans. Mobile Computing*, vol. 21, no. 12, pp. 4426–4438, 2022.
- [14] Z. Liu, Z. Liu, Y. Xie, K.Y. Chan, Y. Yuan, and Y. Yang, "Power allocation in D2D enabled cellular network with probability constraints: A robust Stackelberg game approach," *Ad Hoc Networks*, vol. 133, pp. 1–10, 2022.
- [15] M. Piming, Z. Peng, B. Zhiquan, D. Xu, Y. Xinghai, and K. Kyungsup, "Coalitional game based resource allocation in D2D-enabled V2V communication," *J. Systems Engineering and Electronics*, vol. 34, no. 6, pp. 1508–1519, 2023.
- [16] S. Gowri and S. Vimalanand, "Game theory based resource allocation in D2D communication with traffic aware beam selection in 5G networks," *Int'l J. Intelligent Systems and Applications in Engineering*, vol. 12, no. 20, pp. 354–362, 2024.
- [17] J. Akhyani, V. Desai, R. Gupta, N.K. Jadav, T. Rathod, S. Tanwar, and S. Malhotra, "AI and game-based efficient resource allocation and interference mitigation scheme for D2D communication," *Physical Comm.*, vol. 66, pp. 1–12, 2024.
- [18] T. Yang, R. Zhang, X. Cheng, and L. Yang, "Graph coloring based resource sharing (GCRS) scheme for D2D communications underlying full-duplex cellular networks," *IEEE Trans. Vehicular Technology*, vol. 66, no. 8, pp. 7506–7517, 2017.
- [19] W.K. Lai, Y.C. Wang, H.C. Lin, and J.W. Li, "Efficient resource allocation and power control for LTE-A D2D communication with pure D2D model," *IEEE Trans. Vehicular Technology*, vol. 69, no. 3, pp. 3202–3216, 2020.
- [20] S. Guo, B.J. Hu, and Q. Wen, "Joint resource allocation and power control for full-duplex V2I communication in high-density vehicular network," *IEEE Trans. Wireless Comm.*, vol. 21, no. 11, pp. 9497–9508, 2022.
- [21] Y. Wang, J. Guo, D. Xu, and L. Meng, "Research on D2D communication resource allocation algorithm based on graph coloring-SGADIA," *Proc. Int'l Conf. Electronic Information Engineering and Computer Science*, 2024, pp. 855–859.
- [22] C. Sun, G. Huang, J. Shu, Y. Yang, and B. Wu, "Joint mode selection and resource allocation based on many-to-many reuse in D2D-aided IoT cellular networks," *Internet of Things*, vol. 25, pp. 1–19, 2024.
- [23] T.V. Raghu and M. Kiran, "An iterative-based optimum power and resource allocation in application-dependent scenarios for one-to-one D2D communication," *IEEE Access*, vol. 12, pp. 38972–38984, 2024.
- [24] H.M.F. Noman, K. Dimyati, K.A. Noordin, E. Hanafi, and A. Abdrabou, "FeDRL-D2D: Federated deep reinforcement learning-empowered resource allocation scheme for energy efficiency maximization in D2D-assisted 6G networks," *IEEE Access*, vol. 12, pp. 109775–109792, 2024.
- [25] C.H.R. Babu and S. Nandakumar, "Towards energy-efficient joint relay selection and resource allocation for D2D communication using hybrid heuristic-based deep learning," *Scientific Reports*, vol. 15, pp. 1–28, 2025.
- [26] Q. Alghazali, H. Al-Amaireh, and T. Cinkler, "Mobility-aware resource allocation in D2D communications using genetic algorithms," *IEEE Access*, vol. 13, pp. 144591–144606, 2025.
- [27] A. Moubayed, A. Shami, and H. Lutfiyya, "Wireless resource virtualization with device-to-device communication underlying LTE network," *IEEE Trans. Broadcasting*, vol. 61, no. 4, pp. 734–740, 2015.
- [28] Y. Cheng and L. Yang, "Cost-oriented virtual resource allocation for device-to-device communications underlying LTE networks," *Proc. IEEE Int'l Conf. Comm. Software and Networks*, 2017, pp. 580–583.
- [29] S. Gendy and Y. Gadallah, "A resource trading scheme in slice-based M2M communications," *Proc. IEEE Vehicular Technology Conf.*, 2019, pp. 1–5.
- [30] L. Nadeem, Y. Amin, J. Loo, M.A. Azam, and K.K. Chai, "Quality of service based resource allocation in D2D enabled 5G-CN with network slicing," *Physical Comm.*, vol. 52, pp. 1–10, 2022.
- [31] W.K. Lai, Y.C. Wang, and H.B. Lei, "Joint resource and power management for D2D communication across multiple service providers," *IEEE Systems J.*, vol. 16, no. 3, pp. 3488–3499, 2022.
- [32] G. Tseliou, K. Samdanis, F. Adelantado, X.C. Perez, and C. Verikoukis, "A capacity broker architecture and framework for multi-tenant support in LTE-A networks," *Proc. IEEE Int'l Conf. Comm.*, 2016, pp. 1–6.
- [33] H.L. Bertoni, "Radio propagation," in *Encyclopedia of Physical Science and Technology (Third Edition)*, R.A. Meyers, Ed., Cambridge, Massachusetts: Academic Press, 2003, pp. 769–792.
- [34] Y.C. Wang and C.C. Huang, "Efficient management of interference and power by jointly configuring ABS and DRX in LTE-A HetNets," *Computer Networks*, vol. 150, pp. 15–27, 2019.
- [35] B.K. Cetin, "IoT underlying cellular uplink through D2D communication principle," *IEEE Access*, vol. 11, pp. 102103–102114, 2023.
- [36] J. Ma and H. Gao, "Joint resource allocation for high-mobility HCNs with D2D communications," *Computer Networks*, vol. 270, pp. 1–13, 2025.
- [37] ETSI, "Radio Frequency (RF) system scenarios," 3GPP TR 36.942 V18.0.0, Mar. 2024.
- [38] Y.C. Wang and C.A. Chuang, "Efficient eNB deployment strategy for heterogeneous cells in 4G LTE systems," *Computer Networks*, vol. 79, pp. 297–312, 2015.
- [39] A. Marinescu, I. Macaluso, and L.A. DaSilva, "System level evaluation and validation of the ns-3 LTE module in 3GPP reference scenarios," *Proc. ACM Symp. QoS and Security for Wireless and Mobile Network*, 2017, pp. 59–64.
- [40] Y.C. Wang and D.R. Jhong, "Efficient allocation of LTE downlink spectral resource to improve fairness and throughput," *Int'l J. Comm. Systems*, vol. 30, no. 14, pp. 1–13, 2017.