Efficient Resource Allocation and Power Control for LTE-A D2D Communication with Pure D2D Model

Wei-Kuang Lai, You-Chiun Wang, He-Cian Lin, and Jian-Wen Li

Abstract—To support the rising number of user equipments (UEs), LTE-A allows some UEs directly talking with each other to facilitate spectrum reuse, which is known as *device-to-device (D2D) communication*. Since D2D UEs (DUEs) consume resources and bring out interference, how to allocate resources and power is important. Existing studies seek to make more DUEs reuse resources of cellular UEs (CUEs, which are the UEs talking with the eNB) to increase throughput. However, it is inefficient for some CUEs (e.g., those near cell edge) to share resources with others due to high interference. Thus, a new sharing paradigm, called the *pure D2D model*, is proposed to allow DUEs sharing resources without involving CUEs for flexibility. This new model is helpful to IoT (Internet of Things) applications, where the overwhelming majority of devices are usually DUEs. The paper defines an optimization problem to maximize links supported in the network, and proposes a *D2D resource allocation and power control (DRAPC)* framework. By vertex coloring, DRAPC gives a preliminary grouping of UEs for resource allocation. Then, each group is carefully reformed by exchanging members and adding new ones, so as to increase signal quality and degree of resource sharing. Simulation results show that DRAPC not only improves network performance but also guarantees fairness among links.

Index Terms—D2D communication, grouping, LTE-A, power control, resource allocation.

1 Introduction

LONG term evolution-advanced (LTE-A) has been the main standard for mobile networks and operated in many countries. To offer broadband wireless access, LTE-A adopts a number of efficient techniques like carrier aggregation [1], MIMO (multiple-input, multiple-output) [2], and heterogenous networking [3]. With LTE-A, people can enjoy the use of high-data-rate applications such as video downloads and multimedia streaming through mobile phones.

In LTE-A, the spectrum resource is divided into disjointed *resource blocks (RBs)*, each with 0.5 ms duration and 180 kHz bandwidth. The amount of data carried by one RB depends on its associated *modulation and coding scheme (MCS)*, which is decided by the channel quality. The eNB (i.e., base station) takes charge of allocating RBs to *user equipments (UEs)* in each *transmission time interval (TTI)*. If the downlink channel has bandwidth of 1.4, 3, 5, 10, 15, and 20 MHz, the eNB can allocate 6, 15, 25, 50, 75, and 100 RBs in a TTI.

Traditionally, even if two nearby UEs in the same cell want to talk with each other, two-hop transmission through the eNB is essential, which wastes RBs. To tackle the problem, LTE-A permits direct communication between a pair of UEs without relays by the eNB. This technology, known as *device-to-device (D2D) communication*, brings some benefits [4]. For example, spectral efficiency is improved, as RBs can be reused by D2D pairs. Moreover, the proximity of D2D UEs (DUEs) promises lower transmitted power, thereby increasing energy efficiency.

In general, RBs are considered as exclusive resources from the viewpoint of *cellular UEs (CUEs)* [5], which are the UEs talking with the eNB. D2D communication offers another way for resource sharing between CUEs and D2D pairs. There are four common sharing models proposed in the literature [6]: 1) one D2D pair with a CUE, 2) one D2D pair with multiple CUEs, 3) multiple D2D pairs with a CUE, and 4) multiple D2D pairs with multiple CUEs. Various approaches are developed for these models to improve network throughput.

The authors are with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung City, 80424, Taiwan. E-mail: wklai@mail.cse.nsysu.edu.tw; ycwang@mail.cse.nsysu.edu.tw; m043040095@student.nsysu.edu.tw; tom830001@gmail.com

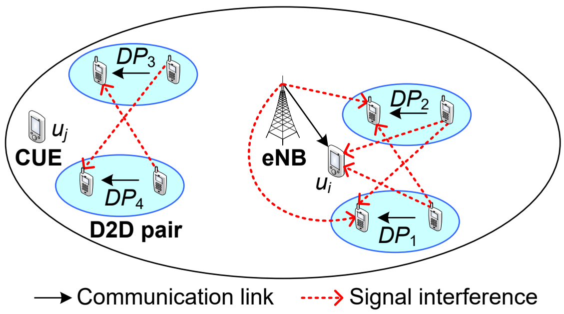


Fig. 1: Example of interference between CUEs and D2D pairs.

However, not every CUE is suitable to share resources with D2D pairs. In particular, it is difficult for D2D pairs to reuse the RBs allocated to a CUE close to cell edge. Fig. 1 gives an example. When the eNB communicates with CUE u_i, the eNB will interfere with D2D pairs DP_1 and DP_2, as their receivers are close to u_i. As a comparison, D2D pairs DP_3 and DP_4 can reuse u_i's RBs with very little interference. On the other hand, when the eNB talks with CUE u_j, it actually imposes high interference to all D2D pairs, since u_j is far away from the eNB and it has to emit large transmitted power to serve u_j. In this case, none of D2D pairs can reuse u_j's RBs due to the eNB's interference. To deal with this issue, we propose a new sharing paradigm, namely the *pure D2D model*, to let multiple D2D pairs reuse RBs without the intervention of CUEs.

The pure D2D model provides flexibility for resource allo-

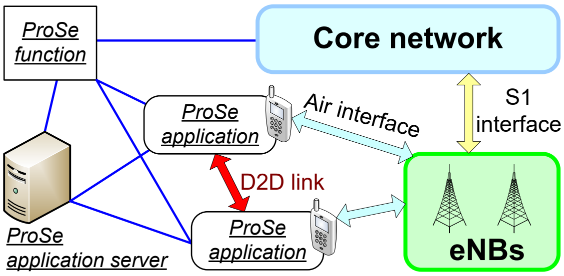


Fig. 2: LTE-A network structure for D2D communication.

cation, since some RBs can be shared by more DUEs with no need to care the constraint from CUEs. In many IoT (Internet of Things) applications, there usually exist numerous DUEs (i.e., IoT devices) but just few CUEs (e.g., monitoring or user devices) [7]. By relaxing the assumption that D2D pairs are allowed to reuse only the RBs allocated to CUEs, the pure D2D model facilitates resource sharing among UEs in these applications, thereby improving network performance.

With the pure D2D model, this paper defines an optimization problem whose objective is to maximize communication links supported in the network. The problem is formulated by mixed integer nonlinear programming (MINP), which is shown to be NP-hard [8]. An efficient solution, called the *D2D resource allocation and power control (DRAPC)* framework, is developed. It first gives an elementary grouping result of UEs by vertex coloring, such that all UEs in a group can share RBs without interference. Then, DRAPC well reforms each group by exchanging some members with better ones or adding new members, so as to improve signal quality of UEs in that group and let more UEs share the same RBs.

Our contributions are threefold. First, a new resource sharing model (i.e., the pure D2D model) is proposed to provide flexibility to resource and power allocation for D2D communication. This model is especially helpful to IoT applications where the majority of devices are comprised mainly of DUEs. Second, our DRAPC framework is not simply a graph-coloring based solution. Instead, it involves many sophisticated designs to carefully reform each group of UEs, such as eliminating bad members that impose high interference, adding new members to raise reusing degree of resources, and adaptively adjusting transmitted power to improve signal quality. Third, simulation results verify that DRAPC supports more D2D links, increases link throughput, and keeps fairness among links, as compared with other methods (including graph-coloring based ones).

The paper is outlined as follows: Section 2 introduces D2D communication in LTE-A and Section 3 surveys related work. The problem is defined in Section 4. Section 5 then proposes the DRAPC framework, followed by performance evaluation in Section 6. Section 7 finally concludes this paper.

2 Overview of D2D Communication in LTE-A

In 3GPP standard Release 12 [9], three new components are added to support D2D communication (referring to Fig. 2). Each DUE has a *ProSe (proximity-based services) application* to handle D2D discovery and linkage. A logical *ProSe function* is responsible for recognizing D2D applications, passing parameters, and supplying network-related functionality such as authorization and charging. This function also connects to the core network to retrieve subscriber information. Besides, third-party media (i.e., not subject to the standard), namely *ProSe application servers*, provide information of available functions to ProSe applications. These components can exchange data via the interfaces defined in [10]. Our DRAPC framework is installed in the ProSe function to manage D2D links.

D2D discovery is essential to communication, which has two strategies [11]. In the *direct discovery*, the eNB reserves some RBs for transmission and reception of discovery beacons. To search nearby devices, DUEs contend for these RBs and broadcast their beacons. In this way, D2D communication is initiated autonomously, and DUEs can cope with resource allocation and power control on their own. However, DUEs would impose high interference on CUEs [12]. On the other hand, the *fully network-assisted discovery* allows the eNB to control D2D communication, including device authentication, link maintenance, and allocation of RBs and power. Therefore, it reduces interference between CUEs and DUEs, which improves quality of service (QoS). Since the top priority of most operators is to support QoS for CUEs, the fully network-assisted discovery is popularly used in practice [4].

In the fully network-assisted discovery, the eNB needs to know channel conditions of all involved links for interference avoidance and allocation of RBs and power. Like CUEs, DUEs report their *channel state information (CSI)* to the eNB, which is the major signaling overhead for D2D communication. More details of the CSI reporting procedure can be found in [13].

3 Related Work

Several studies consider the simplest sharing model, where resources are shared by a CUE and one D2D pair. Specifically, [14] uses the Shannon formula to estimate network capacity, and picks a communication mode (i.e., cellular, dedicated, or underlay modes) for the D2D pair to maximize capacity. Yu et al. [15] find the optimal solution with maximum energy or power constraints. The work [16] selects the communication mode according to the location relationship between the CUE and the D2D pair. In [17], both mode selection and resource allocation is addressed to raise throughput and improve channel quality. Pan et al. [18] define an interference-bounded resource sharing problem, whose goal is to keep D2D SINR above a threshold. Li et al. [19] propose a rate-splitting sharing method, where each message is sliced into private and public parts. The former is decoded merely by an intended receiver, while the latter can be decoded by others.

The work [20] targets at the sharing model where a D2D pair partakes of resources with multiple CUEs. Only when the SINR of a CUE is above the given threshold, will it be able to share resources with that D2D pair. The study [21] deals with the same sharing model and proposes a distance-based approach to reduce cellular-to-D2D interference, so as to minimize the outage probability of the D2D link.

By contrast, a few studies address the model where a CUE shares resources with multiple D2D pairs. In particular, [22] adopts a gradient-descent method to increase admitted D2D links and their data rates. The work [23] proposes a greedy-based solution to the optimal power allocation problem, which asks how to assign subchannels to DUEs to keep interference among D2D pairs under a negligible level, such that the QoS demand of a CUE that uses the same subchannels can be met.

Recently, it attracts lots of attention on the sharing model of multiple CUEs with multiple D2D pairs, which is a general case of the above models. The work [24] formulates a sum-rate maximization problem with mode selection, partner as-

TABLE 1: Comparison between prior work and our DRAPC framework.

work	sharing model	uplink	downlink	pure D2D	technique
Reference [14]	1 D2D + 1 CUE	√	√		Shannon formula
Reference [15]	1 D2D + 1 CUE	√	√		optimization with maximum power/energy constraints
Reference [16]	1 D2D + 1 CUE	√	√		location relationship of CUEs and DUEs
Reference [17]	1 D2D + 1 CUE	√	√		mode selection + resource allocation
Reference [18]	1 D2D + 1 CUE		√		interference-bounded resource sharing
Reference [19]	1 D2D + 1 CUE		√		rate splitting
Reference [20]	1 D2D + multi-CUE	√	√		SINR threshold
Reference [21]	1 D2D + multi-CUE	√			distance-based approach
Reference [22]	multi-D2D + 1 CUE	√			gradient-descent method
Reference [23]	multi-D2D + 1 CUE	√			greedy-based solution
Reference [24]	multi-D2D + multi-CUE	√			sum-rate maximization
Reference [25]	multi-D2D + multi-CUE	√			maximization of energy efficiency
Reference [26]	multi-D2D + multi-CUE	√	√		model selection + transmission period + power control
Reference [27]	multi-D2D + multi-CUE	√			maximum-weight matching
Reference [28]	multi-D2D + multi-CUE	√			game-theoretic approach
Reference [29]	multi-D2D + multi-CUE	√			two-layer approach
Reference [30]	multi-D2D + multi-CUE		√		antenna arrays + interference alignment
Reference [31]	multi-D2D + multi-CUE		√		graph coloring
Reference [32]	multi-D2D + multi-CUE	√	√		graph coloring
Reference [33]	multi-D2D + multi-CUE		√		knapsack problem
Reference [34]	multi-D2D + multi-CUE	√			maximum independent set + Stackelberg game
Reference [35]	multi-D2D + multi-CUE		√		Gale-Shapley method
DRAPC	multi-D2D + multi-CUE		√	√	graph coloring + group reforming

signment, and power allocation, and proposes a greedy-based solution. Xu et al. [25] allocate channels to D2D pairs and find their transmitted power to improve energy efficiency of D2D links. The study [26] decides the communication mode, transmission period, and power allocation of DUEs to save energy and meet their demands. In [27], a bipartite graph is built for resource allocation, where the vertex set is composed of D2D pairs and RBs. Then, it finds the maximum-weight matching to achieve proportionally fair allocation of RBs. In [28], a game-theoretic approach is used to analyze correlations among UEs. D2D pairs are then matched with CUEs with the goal of improving energy efficiency. In [29], a two-layer approach is proposed, where the first layer finds the optimal power values for DUEs and the second layer then allocates RBs to them for maximizing energy efficiency.

With adaptive antenna arrays and application of interference alignment, [30] proposes a resource sharing scheme for multiple D2D groups to maximize throughput. Both [31] and [32] convert the resource allocation problem into a graph-coloring problem to eliminate interference among UEs. Besides, [33] adopts a solution to the knapsack problem to handle resource sharing between CUEs and D2D pairs. In [34], the maximum independent set and the Stackelberg game are used to solve the problems of RB allocation and power control. The work [35] employs the Gale-Shapley method for finding matches between CUEs and D2D pairs to share RBs, which is proved to be stable and weak Pareto optimal.

Table 1 gives a comparison between our DRAPC framework and prior work. As can be seen, none of the prior work considers sharing resources among D2D pairs without the intervention of CUEs. It thus motivates us to propose a new sharing model of pure D2D in this paper to improve resource utilization and also network performance, especially for IoT applications where the majority of devices are DUEs.

4 D2D Communication Problem

4.1 System Model

Let us make assumptions for the system model. First, the in-band communication is considered, where D2D communication takes place only in the licensed frequency band allocated to LTE-A CUEs. Second, the fully network-assisted discovery is used for an eNB to control D2D communication in its cell, as discussed in Section 2. Third, we adopt the underlay mode (also called the shared mode), where two DUEs talk to each other without the eNB's intervention. Besides, CUEs and D2D pairs can share RBs. Finally, we aim at the downlink scenario.

Fig. 1 shows the network topology, where a set $\hat{\mathcal{U}}_C$ of CUEs receive data from the eNB. A CUE does not partake in D2D communication (i.e., it is not also a D2D sender or receiver). A D2D pair comprises two UEs, and any two D2D pairs are non-overlapped. Let set $\hat{\mathcal{U}}_D$ contain the receiver of each D2D pair. It is obvious that $\hat{\mathcal{U}}_C \cap \hat{\mathcal{U}}_D = \emptyset$. Besides, the union of $\hat{\mathcal{U}}_C$ and $\hat{\mathcal{U}}_D$ is denoted by $\hat{\mathcal{U}}$. The eNB estimates the channel condition of each UE to handle interference between cellular and D2D communication, which will be discussed in Section 4.2.

The eNB allocates RBs to UEs based on three principles: 1) Two CUEs cannot use the same RBs. 2) Multiple D2D pairs may reuse the RBs given to a CUE. 3) Some RBs could be shared solely by D2D pairs (in other words, these RBs are not allocated to CUEs). The last principle is the pure D2D model. Note that this new model does not make CUEs out of the picture but provides more flexibility for RB allocation, especially when DUEs are much more than CUEs.

4.2 SINR Estimation

Suppose that a UE u_x receives data from its sender $t(x)$ via RB r_k. The strength of $t(x)$'s signal gotten by u_x is $\zeta^k_{t(x),x} = \tilde{g}^k_{t(x),x} \times \tilde{p}^k_{t(x),x}$, where $\tilde{g}^k_{t(x),x}$ is r_k's channel gain from $t(x)$ to u_x and $\tilde{p}^k_{t(x),x}$ is $t(x)$'s transmitted power on r_k for sending data to u_x. When the eNB allocates an RB to two UEs u_x and u_y, they could be interfered by $t(y)$ and $t(x)$, respectively. In particular, the amount of interference by $t(x)$ imposed on u_y (denoted by $t(x) \rightarrow u_y$) is calculated as follows:

- eNB (sending data to CUE u_x) $\rightarrow$ CUE u_y: $\zeta^k_{t(x),y} = 0$, as r_k cannot be shared by two CUEs.
- eNB (sending data to CUE u_x) $\rightarrow$ DUE u_y: $\zeta^k_{t(x),y} = \tilde{g}^k_{\text{eNB},y} \times \tilde{p}^k_{\text{eNB},x}$.
- DUE (sending data to DUE u_x) $\rightarrow$ UE u_y: $\zeta^k_{t(x),y} = \tilde{g}^k_{t(x),y} \times \tilde{p}^k_{t(x),x}$, where u_y is a CUE or DUE.

TABLE 2: Summary of notations.

notation	definition
$\hat{\mathcal{U}}_C, \hat{\mathcal{U}}_D$	sets of CUEs and D2D receivers ($\hat{\mathcal{U}}_C \cup \hat{\mathcal{U}}_D = \hat{\mathcal{U}}$)
$\hat{\mathcal{R}}$	the set of available RBs
$\hat{\mathcal{C}}$	the candidate set of UEs
$\hat{\mathcal{N}}_i$	the set of neighbors of UE u_i
$\hat{\mathcal{G}}_k$	the group of UEs that share RB r_k
τ_i^k	SINR of UE u_i on RB r_k
z_i^k	an indicator to check if UE u_i uses RB r_k
$\tilde{p}_{t(i),i}^k$	transmitted power of sender $t(i)$ for UE u_i
$\tilde{g}_{t(i),i}^k$	channel gain from sender $t(i)$ to UE u_i
$\zeta_{t(i),j}^k$	strength of sender $t(i)$'s signal gotten by UE u_j

For a CUE $u_j \in \hat{\mathcal{U}}_C$, its SINR on r_k is estimated by

$$\tau_j^k = \frac{z_j^k \times \zeta_{\mathrm{eNB},j}^k}{N_0^2 + \sum_{u_i \in \hat{\mathcal{U}}_D} z_i^k \zeta_{t(i),j}^k}, \tag{1}$$

where N_0 is the power of thermal noise and z_j^k is an indicator to reveal whether r_k is used by u_j to get data. Here, $z_j^k = 1$ if r_k is allocated to u_j, and $z_j^k = 0$ otherwise. On the other hand, the SINR of a DUE $u_i \in \hat{\mathcal{U}}_D$ on r_k is computed by

$$\tau_i^k = \frac{z_i^k \times \zeta_{t(i),i}^k}{N_0^2 + \sum_{u_j \in \hat{\mathcal{U}}_C} z_j^k \zeta_{\mathrm{eNB},i}^k + \sum_{u_l \in \hat{\mathcal{U}}_D \setminus \{u_i\}} z_l^k \zeta_{t(l),i}^k}, \tag{2}$$

where $t(i)$ is a DUE. One could add the amount of interference $\mathcal{I}$ from adjacent cells to the denominators of both Eqs. (1) and (2) to take into account inter-cell interference. However, as there have been many inter-cell interference coordination techniques developed [36], the effect of $\mathcal{I}$ can be negligible. Thus, we aim at the interference from those UEs that share the same RBs.

4.3 Problem Formulation

Given a set $\hat{\mathcal{R}}$ of available RBs, the D2D communication problem is formulated as an optimization problem:

$$\max \sum\nolimits_{r_k \in \hat{\mathcal{R}}} \sum\nolimits_{u_i \in \hat{\mathcal{U}}} z_i^k, \tag{3}$$

under the following constraints:

$$\tau_i^k \geq \tau_i^{\min} \text{ if } z_i^k = 1 \qquad \forall u_i \in \hat{\mathcal{U}}, \forall r_k \in \hat{\mathcal{R}}, \tag{4}$$

$$z_i^k \in \{0, 1\} \qquad \forall u_i \in \hat{\mathcal{U}}, \forall r_k \in \hat{\mathcal{R}}, \tag{5}$$

$$\tilde{p}_{t(i),i}^k \in [p_i^{\min}, p_i^{\max}] \qquad \forall u_i \in \hat{\mathcal{U}}, \forall r_k \in \hat{\mathcal{R}}, \tag{6}$$

$$\sum\nolimits_{u_i \in \hat{\mathcal{U}}_C} z_i^k \leq 1 \qquad \forall r_k \in \hat{\mathcal{R}}. \tag{7}$$

Specifically, the objective in Eq. (3) is to maximize the number of communication links supported by RBs in $\hat{\mathcal{R}}$, where both z_i^k (i.e., RB allocation) and $\tilde{p}_{t(i),i}^k$ (i.e., transmitted power) are the control parameters. Eq. (4) indicates that if an RB r_k is allocated to a UE u_i, then u_i's SINR on r_k should be above its minimum threshold $\tau_i^{\min}$ to provide QoS. Note that the calculation of τ_i^k (by Eqs. (1) and (2)) has taken interference into consideration. Besides, a link is viewed as *outage* if its UE is not given with any RB (due to the violation of Eq. (4)). That is why we need not add extra constraints for interference and outage. They are implicitly embedded in Eq. (4).

Eq. (5) means that a UE uses the same RB at most once. Eq. (6) mentions that the transmitted power of $t(i)$ (i.e., u_i's sender) should be between a minimum value $p_i^{\min}$ and a maximum value $p_i^{\max}$. If $t(i)$ is the eNB, u_i belongs to $\hat{\mathcal{U}}_C$. If $t(i)$ is a DUE, u_i belongs to $\hat{\mathcal{U}}_D$. Then, Eq. (7) points out that two CUEs cannot use the same RB. The above formulation is in the form of MINP, so the D2D communication problem is NP-hard.

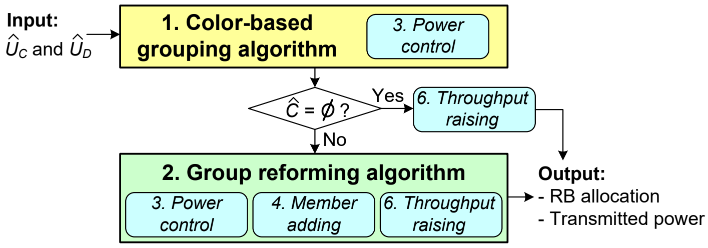


Fig. 3: DRAPC flowchart, where each number gives the index of its algorithm.

5 The DRAPC Framework

DRAPC adopts a graph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ to reveal interference among UEs, where the vertex set $\mathcal{V}$ includes every UE in $\hat{\mathcal{U}}$. If a UE u_i will be interfered by the sender of another UE u_j, there exists an edge (u_i, u_j) in $\mathcal{E}$. In this case, we say that u_i is a *neighbor* of u_j, and vice versa. Due to high interference, each RB cannot be allocated to two neighboring UEs. This is similar to the *vertex coloring problem* [37], where two vertices linking by the same edge cannot have the same color. Thus, when a group of vertices are painted with the same color, the corresponding UEs are interference-free and allowed to use the same RBs. Then, DRAPC seeks to exchange and add members in each group to help more UEs share resources, and adjusts transmitted power to raise their SINRs.

Fig. 3 presents the flowchart of DRAPC, which comprises two main algorithms. The *color-based grouping algorithm* first builds graph $\mathcal{H}$ and preliminarily groups UEs through vertex coloring. If there remain UEs unassigned in the candidate set $\hat{\mathcal{C}}$, the *group reforming algorithm* further improves every group to increase signal quality and support more links. Moreover, three auxiliary algorithms are developed: 1) the *power control algorithm* regulates transmitted power of each sender to meet the SINR requirement, 2) the *member adding algorithm* checks whether extra members can be added to a group to make the result better, and 3) the *throughput raising algorithm* seeks to increase transmitted power for throughput improvement. Next, we detail each algorithm, followed by a discussion.

5.1 Color-based Grouping Algorithm

This algorithm divides UEs into groups by vertex coloring such that all members in each group can share RBs without interference. Algo. 1 gives its pseudocode that contains three parts. Part 1 (lines 1–4) builds the graph, where $\hat{\mathcal{N}}_i$ is the set of neighbors of a UE u_i in $\hat{\mathcal{U}}$. If the power strength $\zeta_{t(i),j}^k$ on an RB r_k exceeds threshold ζ_{th}, it means that $t(i)$ (i.e., u_i's sender) imposes significant interference on u_j. Thus, u_i and u_j are neighbors, and both $\hat{\mathcal{N}}_i$ and $\hat{\mathcal{N}}_j$ are updated by line 4.

Part 2 includes lines 5 to 12, which adopts a greedy strategy to group UEs. Each RB r_k in $\hat{\mathcal{R}}$ is viewed as one color, and we use $\hat{\mathcal{G}}_k$ to record the set of UEs that share r_k (i.e., painted with the same color). Specifically, all UEs of $\hat{\mathcal{U}}$ are sorted by the number of neighbors (i.e., $|\hat{\mathcal{N}}_i|$) in a descending order. Let us use $\mathrm{SORT}(\hat{\mathcal{U}}, |\hat{\mathcal{N}}_i|)$ to denote this operation, and store the result in a *candidate set* $\hat{\mathcal{C}}$, as line 5 shows. Then, for each

Algorithm 1: Color-based Grouping Algorithm

1 **foreach** $u_i \in \hat{\mathcal{U}}$ **do**
2 **foreach** $u_j \in \hat{\mathcal{U}}$ **do**
3   **if** $u_i \neq u_j$ *and* $\max_{r_k \in \hat{\mathcal{R}}} \zeta^k_{t(i),j} > \zeta_{\mathrm{th}}$ **then**
4    $\hat{\mathcal{N}}_i \leftarrow \hat{\mathcal{N}}_i \cup \{u_j\}$ and $\hat{\mathcal{N}}_j \leftarrow \hat{\mathcal{N}}_j \cup \{u_i\}$;
5 $\hat{\mathcal{C}} \leftarrow \texttt{SORT}(\hat{\mathcal{U}}, |\hat{\mathcal{N}}_i|)$;
6 **foreach** $u_i \in \hat{\mathcal{C}}$ **do**
7 $\hat{\mathcal{R}}_i \leftarrow \texttt{SORT}(\hat{\mathcal{R}}, \zeta^k_{t(i),i})$;
8 **foreach** $r_k \in \hat{\mathcal{R}}_i$ **do**
9   **if** $\hat{\mathcal{G}}_k \cap \hat{\mathcal{N}}_i = \emptyset$ **then**
10   **if** $u_i \notin \hat{\mathcal{U}}_C$ *or* $\hat{\mathcal{G}}_k \cap \hat{\mathcal{U}}_C = \emptyset$ **then**
11     $\hat{\mathcal{G}}_k \leftarrow \hat{\mathcal{G}}_k \cup \{u_i\}$ and $\hat{\mathcal{C}} \leftarrow \hat{\mathcal{C}} \setminus \{u_i\}$;
12   break;
13 **foreach** $\hat{\mathcal{G}}_k \subseteq \hat{\mathcal{U}}$ **do**
14 Adjust transmitted power in $\hat{\mathcal{G}}_k$ by Algo. 3;
15 **while** $\min_{u_i \in \hat{\mathcal{G}}_k} \tau^k_i < \tau^{\min}_i$ **do**
16   $u_i \leftarrow \arg\min_{u_i \in \hat{\mathcal{G}}_k} \tau^k_i$;
17   $\hat{\mathcal{G}}_k \leftarrow \hat{\mathcal{G}}_k \setminus \{u_i\}$ and $\hat{\mathcal{C}} \leftarrow \hat{\mathcal{C}} \cup \{u_i\}$;

UE u_i in $\hat{\mathcal{C}}$, line 7 sorts each RB by its signal strength $\zeta^k_{t(i),i}$ decreasingly. Thus, we can greedily pick an RB for u_i such that it gets the largest signal strength from its sender $t(i)$ by two conditions. Line 9 gives the 1st condition, where the RB is not allocated to any neighbor of u_i (to meet the constraint of vertex coloring). Line 10 gives the 2nd condition, where the RB cannot be allocated to multiple CUEs (for the constraint of Eq. (7)). Once both conditions are satisfied, u_i is added to $\hat{\mathcal{G}}_k$ and removed from $\hat{\mathcal{C}}$ by line 11.

Then, part 3 calls the power control algorithm in Algo. 3 to estimate transmitted power. If a group has some UEs that cannot satisfy the SINR constraint of Eq. (4) in line 15, we iteratively pick a UE with the smallest SINR, remove it from the group, and add it to the candidate set $\hat{\mathcal{C}}$. Lemma 1 presents the time complexity of Algo. 1, and Theorem 1 analyzes the minimum number of groups found by the algorithm.

Lemma 1. *Given n_U UEs in $\hat{\mathcal{U}}$ and n_R RBs in $\hat{\mathcal{R}}$, Algo. 1 takes time of $O(n_U^2 n_R) + f_3(n_U)$, where $f_3(\cdot)$ is the computation time of Algo. 3.*

Proof: In lines 1–4, double for-loops are used to check if any two UEs are neighbors. Line 3 takes $O(n_R)$ time, as it checks each RB in $\hat{\mathcal{R}}$. Thus, part 1 takes $n_U^2 O(n_R)$ time. Then, line 5 spends $O(n_U \lg n_U)$ time to sort UEs. Let us take a look at the for-loop in lines 6–12, which repeats n_U times. Line 7 spends $O(n_R \lg n_R)$ time to sort RBs. As $\hat{\mathcal{N}}_i \subseteq \hat{\mathcal{U}}$ and $\hat{\mathcal{U}}_C \subseteq \hat{\mathcal{U}}$, the inner for-loop in lines 8–12 takes $n_R O(n_U)$ time. Thus, part 2 spends time of $O(n_U \lg n_U) + n_U(O(n_R \lg n_R) + n_R O(n_U))$. It is expected that $n_U > \lg n_R$, so the equation can be simplified to $O(n_U^2 n_R)$. For part 3, since any two groups are disjointed, line 14 (along with its for-loop) uses Algo. 3 to compute the transmitted power of each sender once, so it takes $f_3(n_U)$ time. Then, the while-loop in lines 15–17 checks every UE once in the worst case, which spends $O(n_U)$ time. Thus, part 3 spends time of $f_3(n_U) + O(n_U)$. To sum up, the total time complexity of Algo. 1 is $n_U^2 O(n_R) + O(n_U^2 n_R) + f_3(n_U) + O(n_U) = O(n_U^2 n_R) + f_3(n_U)$. □

Algorithm 2: Group Reforming Algorithm

foreach $\hat{\mathcal{G}}_k \subseteq \hat{\mathcal{U}}$ **do**
 foreach $u_i \in \hat{\mathcal{C}}$ **do**
  if $u_i \in \hat{\mathcal{U}}_C$ *and* $\hat{\mathcal{G}}_k \cap \hat{\mathcal{U}}_C \neq \emptyset$ **then**
   $u_j \leftarrow \hat{\mathcal{G}}_k \cap \hat{\mathcal{U}}_C$;
   if $\tilde{p}^k_{\mathrm{eNB},i} < \tilde{p}^k_{\mathrm{eNB},j}$ **then**
    $\hat{\mathcal{G}}_{\mathrm{tmp}} \leftarrow (\hat{\mathcal{G}}_k \setminus \{u_j\}) \cup \{u_i\}$;
  else if $u_i \in \hat{\mathcal{U}}_D$ **then**
   $u_j \leftarrow \arg\max_{u_j \in \hat{\mathcal{G}}_k} \tilde{p}^k_{t(j),j}$;
   $\hat{\mathcal{G}}_{\mathrm{tmp}} \leftarrow (\hat{\mathcal{G}}_k \setminus \{u_j\}) \cup \{u_i\}$;
  Adjust transmitted power in $\hat{\mathcal{G}}_{\mathrm{tmp}}$ by Algo. 3;
  if $\min_{u_x \in \hat{\mathcal{G}}_{\mathrm{tmp}}} \tau^k_x > \tau^{\min}_x$ *and* $\max_{u_x \in \hat{\mathcal{G}}_{\mathrm{tmp}}} \tilde{p}^k_{t(x),x} < \max_{u_y \in \hat{\mathcal{G}}_k} \tilde{p}^k_{t(y),y}$ **then**
   $\hat{\mathcal{G}}_k \leftarrow \hat{\mathcal{G}}_{\mathrm{tmp}}$ and $\hat{\mathcal{C}} \leftarrow (\hat{\mathcal{C}} \setminus \{u_i\}) \cup \{u_j\}$;
 Add extra members to $\hat{\mathcal{G}}_k$ by Algo. 4;
 Raise throughput of UEs in $\hat{\mathcal{G}}_k$ by Algo. 6;

Theorem 1. *Supposing that $|\hat{\mathcal{U}}| \geq \max\{n_R, N_{\max}+1\}$, the minimum number of groups that can be found by Algo. 1 is $\min\{n_R, N_{\max}\}$, where n_R is the number of RBs in $\hat{\mathcal{R}}$ and $N_{\max} = \max_{u_i \in \hat{\mathcal{U}}}\{|\hat{\mathcal{N}}_i|\}$.*

Proof: Since every RB in $\hat{\mathcal{R}}$ is considered as one color, the worst case is to find out n_R groups of UEs. However, the Brooks' theorem [38] indicates that if each vertex in a connected graph has no more than $N_{\max}$ neighbors, all vertices can be colored with just $N_{\max}$ colors, except for two cases, complete graphs or cycle graphs of odd length, which require $(N_{\max}+1)$ colors. These cases occur in Algo. 1 when only $N_{\max}$ or $(N_{\max}+1)$ RBs in $\hat{\mathcal{R}}$ are selected (depending on the graph) to be allocated to all UEs in $\hat{\mathcal{C}}$ by lines 6–12, where $N_{\max}+1 \leq n_R$, and no UEs in any group $\hat{\mathcal{G}}_k$ are removed by lines 13–17. Thus, the minimum number of groups found by Algo. 1 will be $\min\{N_{\max}, n_R\}$. □

According to the Brooks' theorem, at least $\min\{n_R, N_{\max}\}$ groups of UEs will be formed by *any* graph-coloring based method. In a dense network where some UEs have more than n_R neighbors, the optimal solution must divide UEs into n_R groups because of $n_R < N_{\max}$. In this case, Algo. 1 performs as better as other graph-coloring based methods in terms of the number of groups found. In other words, Algo. 1 is also feasible in dense networks. On the other hand, when $n_R > N_{\max}$ (i.e., a relatively sparse network), it is possible that fewer than n_R groups are found based on Theorem 1. In this case, we can allocate the residual RBs in $\hat{\mathcal{R}}$ by the rules in Algo. 1 again. Consequently, some groups of UEs each can share multiple RBs.

5.2 Group Reforming Algorithm

Algo. 1 groups UEs for RB allocation by their relationship of interference, but some UEs may be still left unallocated in the set $\hat{\mathcal{C}}$. Thus, the group reforming algorithm aims to improve each group by exchanging its members with UEs in $\hat{\mathcal{C}}$ that have better conditions. Besides, this algorithm finds extra members from $\hat{\mathcal{C}}$ for some groups to achieve the objective in Eq. (3), and also raises transmitted power to increase network throughput.

Algo. 2 presents the pseudocode of the group reforming algorithm, which uses a for-loop to adjust each group $\hat{\mathcal{G}}_k$. The inner for-loop iteratively picks a UE u_i from $\hat{\mathcal{C}}$ and checks if it can replace a member u_j in $\hat{\mathcal{G}}_k$ to better the result. Specifically, lines 3–6 handle the case when u_i is a CUE and $\hat{\mathcal{G}}_k$ also has a CUE u_j. If the eNB uses less transmitted power for u_i than for u_j (i.e., $\tilde{p}^k_{\mathrm{eNB},i} < \tilde{p}^k_{\mathrm{eNB},j}$), u_i actually causes less interference to $\hat{\mathcal{G}}_k$. In this case, replacing u_j by u_i could improve SINRs of most DUEs in $\hat{\mathcal{G}}_k$. Thus, a temporary group $\hat{\mathcal{G}}_{\mathrm{tmp}}$ is used to store the modified result of $\hat{\mathcal{G}}_k$ in line 6. Then, lines 7–9 deal with the case when u_i is a DUE. From $\hat{\mathcal{G}}_k$, we pick the DUE u_j whose sender emits the largest transmitted power $\tilde{p}^k_{t(j),j}$, and replace u_j by u_i. The result is then stored in $\hat{\mathcal{G}}_{\mathrm{tmp}}$.

Line 10 uses Algo. 3 to compute transmitted power for UEs in $\hat{\mathcal{G}}_{\mathrm{tmp}}$. If both conditions in line 11 are satisfied, a better group $\hat{\mathcal{G}}_{\mathrm{tmp}}$ is found for $\hat{\mathcal{G}}_k$. First, every UE in $\hat{\mathcal{G}}_{\mathrm{tmp}}$ meets the SINR constraint of Eq. (4) (i.e., $\min_{u_x\in\hat{\mathcal{G}}_{\mathrm{tmp}}} \tau^k_x > \tau^{\min}_x$). Second, the largest transmitted power in $\hat{\mathcal{G}}_{\mathrm{tmp}}$ is smaller than that in $\hat{\mathcal{G}}_k$ (i.e., $\max_{u_x\in\hat{\mathcal{G}}_{\mathrm{tmp}}} \tilde{p}^k_{t(x),x} < \max_{u_y\in\hat{\mathcal{G}}_k} \tilde{p}^k_{t(y),y}$), so interference in $\hat{\mathcal{G}}_{\mathrm{tmp}}$ can be reduced as compared with $\hat{\mathcal{G}}_k$. Thus, line 12 replaces $\hat{\mathcal{G}}_k$ by $\hat{\mathcal{G}}_{\mathrm{tmp}}$ and updates $\hat{\mathcal{C}}$ accordingly.

After the above adjustment, the member adding algorithm in Algo. 4 finds extra UEs from $\hat{\mathcal{C}}$ and adds them to group $\hat{\mathcal{G}}_k$. Then, the throughput raising algorithm in Algo. 6 increases transmitted power to improve throughput of UEs. Lemma 2 gives the time complexity of Algo. 2, and Theorem 2 proves that it can better the grouping result of Algo. 1.

Lemma 2. *Let n_U, n_R, and n_C be the sizes of $\hat{\mathcal{U}}$, $\hat{\mathcal{R}}$, and $\hat{\mathcal{C}}$, respectively. Algo. 2 takes time of $n_C(O(n_U + f_3(n_U))) + n_R f_4(n_C) + f_6(n_U)$, where $f_3(\cdot)$, $f_4(\cdot)$, and $f_6(\cdot)$ denote the computation time of Algos. 3, 4, and 6, respectively.*

Proof: We first analyze computation time of the inner for-loop in lines 2–12, which repeats n_C times. The codes in lines 3–6 and 7–9 are mutual exclusive, but both of them search each UE in a group $\hat{\mathcal{G}}_k$ to find u_j. Thus, lines 3–9 takes $O(|\hat{\mathcal{G}}_k|)$ time. Then, line 10 runs Algo. 3, which spends time of $f_3(|\hat{\mathcal{G}}_{\mathrm{tmp}}|) = f_3(|\hat{\mathcal{G}}_k|)$. Line 11 searches UEs in $\hat{\mathcal{G}}_{\mathrm{tmp}}$ twice and in $\hat{\mathcal{G}}_k$ once, which takes $3O(|\hat{\mathcal{G}}_k|)$ time. Thus, the inner for-loop spends time of $n_C(O(|\hat{\mathcal{G}}_k|) + f_3(|\hat{\mathcal{G}}_k|) + 3O(|\hat{\mathcal{G}}_k|)) = n_C(O(|\hat{\mathcal{G}}_k|) + f_3(|\hat{\mathcal{G}}_k|))$.

Since Algo. 4 finds new members for a group from $\hat{\mathcal{C}}$, it takes $f_4(n_C)$ time. Algo. 6 improves throughput of each UE in $\hat{\mathcal{G}}_k$, which spends $f_6(|\hat{\mathcal{G}}_k|)$ time. By taking the outer for-loop into consideration, the overall time complexity is

$$\sum\nolimits_{k=1}^{n_R} n_C(O(|\hat{\mathcal{G}}_k|) + f_3(|\hat{\mathcal{G}}_k|)) + f_4(n_C) + f_6(|\hat{\mathcal{G}}_k|). \quad (8)$$

As any two groups are disjointed, $\sum_{k=1}^{n_R} O(|\hat{\mathcal{G}}_k|) \approx O(n_U)$ holds, and the worst case occurs when we do the operation for each UE in $\hat{\mathcal{U}}$. Similarly, we derive that $\sum_{k=1}^{n_R} f_3(|\hat{\mathcal{G}}_k|) \approx f_3(n_U)$ and $\sum_{k=1}^{n_R} f_6(|\hat{\mathcal{G}}_k|) \approx f_6(n_U)$. Thus, Eq. (8) can be simplified to $n_C(O(n_U + f_3(n_U))) + n_R f_4(n_C) + f_6(n_U)$. □

Theorem 2. *Algo. 2 must increase signal quality or support more links if it changes the grouping result of Algo. 1.*

Proof: Observing from Algo. 2, there are two parts to change the grouping result. One is the inner for-loop in lines 2–12, where we exchange a member in group $\hat{\mathcal{G}}_k$ with a UE in the candidate set $\hat{\mathcal{C}}$. Thus, the size of $\hat{\mathcal{G}}_k$ does not change. However, the 2nd condition in line 11 ensures that the largest transmitted power in the modified group $\hat{\mathcal{G}}_{\mathrm{tmp}}$ must be smaller than that of $\hat{\mathcal{G}}_k$. Therefore, SINRs of worse UEs in $\hat{\mathcal{G}}_{\mathrm{tmp}}$ are improved, since the overall interference in $\hat{\mathcal{G}}_{\mathrm{tmp}}$ is reduced as compared with $\hat{\mathcal{G}}_k$. If $\hat{\mathcal{G}}_k$ is replaced by $\hat{\mathcal{G}}_{\mathrm{tmp}}$, signal quality of UEs must increase. The other part lies in line 13, where Algo. 4 picks UEs from $\hat{\mathcal{C}}$ to be added to $\hat{\mathcal{G}}_k$. If the grouping result is changed by this part, some groups will be enlarged by adding new members to them. In this case, Algo. 2 must support more links than Algo. 1. □

Based on Theorem 2, we show that DRAPC is not simply a graph-coloring based method by Algo. 1 but involves more sophisticated reforming of groups by Algo. 2. With the help of auxiliary algorithms in Algos. 3, 4, and 6, our group reforming algorithm allocates RBs to UEs and adjusts transmitted power such that not only signal quality of UEs will be improved but also the number of supported links can increase.

Algorithm 3: Power Control Algorithm

1 **foreach** $u_i \in \hat{\mathcal{G}}_k$ **do**
2 $\quad\tilde{p}^k_{t(i),i} \leftarrow \tau^{\min}_i \times N_0/\tilde{g}^k_{t(i),i}$;
3 $\quad$**if** $\tilde{p}^k_{t(i),i} > p^{\max}_i$ **then**
4 $\quad\quad\tilde{p}^k_{t(i),i} \leftarrow p^{\max}_i$;
5 $\quad$**else if** $\tilde{p}^k_{t(i),i} < p^{\min}_i$ **then**
6 $\quad\quad\tilde{p}^k_{t(i),i} \leftarrow p^{\min}_i$;
7 $\quad p_{\mathrm{adj}} \leftarrow p^{\max}_i/\sigma$;
8 $\quad$**while** $p_{\mathrm{adj}} \geq p^{\max}_i/\sigma^{\alpha}$ **do**
9 $\quad\quad$**if** $\tau^k_i < \tau^{\min}_i$ *and* $\tilde{p}^k_{t(i),i} + p_{\mathrm{adj}} \leq p^{\max}_i$ **then**
10 $\quad\quad\quad\tilde{p}^k_{t(i),i} \leftarrow \tilde{p}^k_{t(i),i} + p_{\mathrm{adj}}$;
11 $\quad\quad$**else if** $\tau^k_i > \tau^{\min}_i \times (1 + (p_{\mathrm{adj}}/\tilde{p}^{\min}_{t(i),i}))$ *and* $\tilde{p}^k_{t(i),i} - p_{\mathrm{adj}} \geq p^{\min}_i$ **then**
12 $\quad\quad\quad\tilde{p}^k_{t(i),i} \leftarrow \tilde{p}^k_{t(i),i} - p_{\mathrm{adj}}$;
13 $\quad\quad p_{\mathrm{adj}} \leftarrow p_{\mathrm{adj}}/\sigma$;

5.3 Power Control Algorithm

Both Algos. 1 and 2 invoke the power control algorithm to adjust the transmitted power of the sender of each UE u_i in a group $\hat{\mathcal{G}}_k$, such that u_i's SINR τ^k_i can be above threshold $\tau^{\min}_i$ (i.e., Eq. (4)'s constraint) but would not cause large interference. Algo. 3 presents its pseudocode. For each u_i, line 2 sets the default transmitted power of its sender by

$$\tau^k_i = \frac{\tilde{g}^k_{t(i),i} \times \tilde{p}^k_{t(i),i}}{N_0} \geq \tau^{\min}_i \Rightarrow \tilde{p}^k_{t(i),i} \geq \tau^{\min}_i \times N_0/\tilde{g}^k_{t(i),i}.$$

Here, only thermal noise N_0 is considered to simplify calculation. Thus, u_i's SINR is at least $\tau^{\min}_i$ in terms of N_0. Then, lines 3–6 do the boundary check for the constraint in Eq. (6).

In lines 8–13, we adopt a while-loop to adjust power $\tilde{p}^k_{t(i),i}$ by a *shrinking* variable p_{adj}. Specifically, p_{adj} is initially set to $p^{\max}_i/\sigma$, where $\sigma \in \mathbb{N}$ and $\sigma \geq 2$. In the end of the while-loop (i.e., line 13), p_{adj} is divided by σ. In this way, we can fine tune $\tilde{p}^k_{t(i),i}$ iteration by iteration. For example, p_{adj} will be exponentially decreased by setting $\sigma = 2$.

In an iteration of the while-loop, line 9 indicates that if u_i's SINR is below its minimum requirement (i.e., $\tau^k_i < \tau^{\min}_i$) and adding p_{adj} to $\tilde{p}^k_{t(i),i}$ will not exceed the maximum threshold $p_i^{\max}$, $\tilde{p}^k_{t(i),i}$ can be increased by p_{adj}. On the other hand, line 11 handles the case when $\tilde{p}^k_{t(i),i}$ is too large. Let $\tilde{p}^{\min}_{t(i),i}$ denote the minimum transmitted power for $t(i)$ to make u_i's SINR be equal to $\tau_i^{\min}$. Suppose that u_i encounters an amount I_i of interference from other senders. To ensure that u_i's SINR is still above $\tau_i^{\min}$ after decreasing $\tilde{p}^k_{t(i),i}$ by p_{adj}, the following condition should be met:

$$\frac{\tilde{g}^k_{t(i),i} \times (\tilde{p}^k_{t(i),i} - p_{\text{adj}})}{I_i + N_0} > \tau_i^{\min} = \frac{\tilde{g}^k_{t(i),i} \times \tilde{p}^{\min}_{t(i),i}}{I_i + N_0}$$

$$\Rightarrow \frac{\tilde{g}^k_{t(i),i} \times \tilde{p}^k_{t(i),i}}{I_i + N_0} > \frac{\tilde{g}^k_{t(i),i} \times \tilde{p}^{\min}_{t(i),i} + \tilde{g}^k_{t(i),i} \times p_{\text{adj}}}{I_i + N_0}$$

$$\Rightarrow \frac{\tilde{g}^k_{t(i),i} \times \tilde{p}^k_{t(i),i}}{I_i + N_0} > \frac{\tilde{g}^k_{t(i),i} \times \tilde{p}^{\min}_{t(i),i}}{I_i + N_0} \times \left(1 + \frac{p_{\text{adj}}}{\tilde{p}^{\min}_{t(i),i}}\right)$$

$$\Rightarrow \tau_i^k > \tau_i^{\min} \times \left(1 + \frac{p_{\text{adj}}}{\tilde{p}^{\min}_{t(i),i}}\right).$$

Moreover, we have to check if the condition of $\tilde{p}^k_{t(i),i} \geq p_i^{\min}$ holds after adjustment. If both conditions are satisfied, $\tilde{p}^k_{t(i),i}$ is decreased by p_{adj} in line 12.

We remark that the objective of the power control algorithm in Algo. 3 is to find the smallest transmitted power for a node $t(i)$ to let the SINR of its receiver u_i overtake $\tau_i^{\min}$. As mentioned in Section 4.3, $\tau_i^{\min}$ is a given input to indicate the minimum SINR threshold for u_i to satisfy its QoS requirement (i.e., traffic demand). Thus, this algorithm is suitable regardless of traffic models. Theorem 3 proves the convergence of Algo. 3 and Lemma 3 then analyzes its time complexity.

Theorem 3. *Given a finite number of UEs in group $\hat{\mathcal{G}}_k$, Algo. 3 must converge.*

Proof: Algo. 3 adopts a for-loop to check each UE u_i in $\hat{\mathcal{G}}_k$. Obviously, the number of iterations repeated in the for-loop is equal to the number of UEs in $\hat{\mathcal{G}}_k$. Then, we use an inner while-loop to iteratively adjust transmitted power $\tilde{p}^k_{t(i),i}$, whose termination condition is $p_{\text{adj}} < p_i^{\max}/\sigma^{\alpha}$. The variable p_{adj} is initially set to $p_i^{\max}/\sigma$ in line 7, and it will be divided by σ whenever an iteration of the while-loop ends. Thus, the while-loop must be repeated at most α times based on lines 7, 8, and 13. In other words, the while-loop runs a finite number of times for each UE in $\hat{\mathcal{G}}_k$, so Algo. 3 must converge. □

Lemma 3. *Suppose that group $\hat{\mathcal{G}}_k$ contains n_G UEs. Algo. 3 spends $O(\alpha n_G)$ time, where $\alpha \in \mathbb{N}$ decides the number of iterations used to adjust transmitted power $\tilde{p}^k_{t(i),i}$.*

Proof: Algo. 3 uses a for-loop that repeats n_G times. Each statement in lines 2–7 spends constant time. As discussed in Theorem 3, the while-loop repeats no more than α times, where each statement in lines 9–13 takes constant time. Thus, the time complexity is $n_G(O(1) + \alpha O(1)) = O(\alpha n_G)$. □

5.4 Member Adding Algorithm

This algorithm picks extra members for a group $\hat{\mathcal{G}}_k$ from the set $\hat{\mathcal{C}}$, which contains the UEs that have not yet been assigned to any group. Algo. 4 presents its pseudocode, where we copy $\hat{\mathcal{C}}$ to a temporary set $\hat{\mathcal{C}}_{\text{tmp}}$. If $\hat{\mathcal{G}}_k$ contains a CUE, all CUEs should be removed from $\hat{\mathcal{C}}_{\text{tmp}}$ due to the constraint of Eq. (7) (i.e., two CUEs cannot be put in the same group to share RBs), as shown in lines 2–3. Then, for each UE u_i in $\hat{\mathcal{C}}_{\text{tmp}}$, we check if adding

Algorithm 4: Member Adding Algorithm

1 $\hat{\mathcal{C}}_{\text{tmp}} \leftarrow \hat{\mathcal{C}}$;
2 **if** $\hat{\mathcal{G}}_k \cap \hat{\mathcal{U}}_C \neq \emptyset$ **then**
3     $\hat{\mathcal{C}}_{\text{tmp}} \leftarrow \hat{\mathcal{C}}_{\text{tmp}} \setminus \hat{\mathcal{U}}_C$;
4 **foreach** $u_i \in \hat{\mathcal{C}}_{\text{tmp}}$ **do**
5   Adjust transmitted power in $\hat{\mathcal{G}}_k \cup \{u_i\}$ by Algo. 3;
6   **if** $\min_{u_j \in \hat{\mathcal{G}}_k \cup \{u_i\}} \tau_j^k < \tau_j^{\min}$ **then**
7         $\hat{\mathcal{C}}_{\text{tmp}} \leftarrow \hat{\mathcal{C}}_{\text{tmp}} \setminus \{u_i\}$;
8 Use BnB to select new members from $\hat{\mathcal{C}}_{\text{tmp}}$ and add them to $\hat{\mathcal{G}}_k$; the result is stored in $\hat{\mathcal{G}}_{\text{new}}$;
9 $\hat{\mathcal{G}}_k \leftarrow \hat{\mathcal{G}}_{\text{new}}$ and $\hat{\mathcal{C}} \leftarrow \hat{\mathcal{C}} \setminus \hat{\mathcal{G}}_{\text{new}}$;

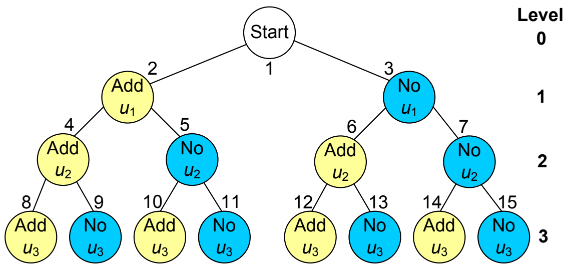


Fig. 4: Example of a state space tree.

u_i to $\hat{\mathcal{G}}_k$ can still meet the SINR constraint in Eq. (4). If not, u_i will not be a candidate for $\hat{\mathcal{G}}_k$ and it is removed from $\hat{\mathcal{C}}_{\text{tmp}}$. The code is given in lines 4–7.

After eliminating non-candidates from $\hat{\mathcal{C}}_{\text{tmp}}$, we adopt the *branch-and-bound (BnB)* method [39] to search new members for $\hat{\mathcal{G}}_k$ (the updated set is denoted by $\hat{\mathcal{G}}_{\text{new}}$), and remove these UEs from $\hat{\mathcal{C}}$. The BnB method builds a *state space tree*, which is a binary tree whose nodes indicate whether to add each UE in $\hat{\mathcal{C}}_{\text{tmp}}$ to $\hat{\mathcal{G}}_{\text{new}}$. Fig. 4 shows an example, where $\hat{\mathcal{C}}_{\text{tmp}} = \{u_1, u_2, u_3\}$. The root is a starting node (with level 0). A left child in level i means to add u_i, while a right child in level i means not to add u_i. Then, BnB will find a branch (from the root to one leaf) to make $\hat{\mathcal{G}}_{\text{new}}$ contain the most UEs.

To do so, BnB uses the bread-first search to check each node in the state space tree. Since the tree size could be very large, a *bound function* is adopted to do the check. If a node cannot be passed by the bound function, it means that including this node will make the solution worse (i.e., the node is not a part of the optimal solution). In this case, its subtree is pruned, so we need not check all of its descendants to save the computational cost. Fig. 4 gives an example. Suppose that node 5 fails to pass the bound function. Then, both nodes 10 and 11 will be pruned.

Algo. 5 gives the pseudocode of our bound function, where p_{low} and $n_{\max}$ record the maximum transmitted power and the number of UEs in the best solution $\hat{\mathcal{G}}_{\text{new}}$ that we have found, respectively. If the visiting node is a left child in level i, it means to add $u_i \in \hat{\mathcal{C}}_{\text{tmp}}$ to the solution. By computing power for UEs in $\hat{\mathcal{G}}_{\text{new}}$, line 4 checks if they meet the SINR constraint. If not, adding u_i to $\hat{\mathcal{G}}_k$ will lead some members to violate Eq. (4), so the visiting node cannot pass the bound function. Then, the if-statement in line 6 indicates that a better solution with more UEs is found. Thus, line 7 updates both $n_{\max}$ and p_{low}. Moreover, if we find a solution whose number of members is $n_{\max}$ whereas it has less transmitted power (i.e.,

Algorithm 5: Bound Function for BnB

Data: $p_{\text{low}} \leftarrow \infty$, $n_{\max} \leftarrow |\hat{\mathcal{G}}_k|$, $\hat{\mathcal{G}}_{\text{new}} \leftarrow \hat{\mathcal{G}}_k$
1 **if** *the visiting node is a left child in level* i **then**
2 $\quad \hat{\mathcal{G}}_{\text{new}} \leftarrow \hat{\mathcal{G}}_{\text{new}} \cup \{u_i\}$;
3 Adjust transmitted power in $\hat{\mathcal{G}}_{\text{new}}$ by Algo. 3;
4 **if** $\min_{u_j \in \hat{\mathcal{G}}_{\text{new}}} \tau_j^k < \tau_j^{\min}$ **then**
5 $\quad$ **return** false;
6 **if** $|\hat{\mathcal{G}}_{\text{new}}| > n_{\max}$ **then**
7 $\quad n_{\max} \leftarrow |\hat{\mathcal{G}}_{\text{new}}|$ and $p_{\text{low}} \leftarrow \max_{u_j \in \hat{\mathcal{G}}_{\text{new}}} \tilde{p}^k_{t(j),j}$;
8 $\quad$ **return** true;
9 **else if** $|\hat{\mathcal{G}}_{\text{new}}| = n_{\max}$ *and* $\max_{u_j \in \hat{\mathcal{G}}_{\text{new}}} \tilde{p}^k_{t(j),j} < p_{\text{low}}$ **then**
10 $\quad p_{\text{low}} \leftarrow \max_{u_j \in \hat{\mathcal{G}}_{\text{new}}} \tilde{p}^k_{t(j),j}$;
11 $\quad$ **return** true;
12 **return** false;

$\max_{u_j \in \hat{\mathcal{G}}_{\text{new}}} \tilde{p}^k_{t(j),j} < p_{\text{low}}$), this solution is also a better one. In this case, the visiting node can pass the bound function.

The BnB method is implemented by using a queue to store tree nodes to be visited. Due to the space limitation, interested readers can find the detail in [39]. Lemma 4 analyzes the time complexity of Algo. 4. In Theorem 4, we show that Algo. 4 must maximize the size of a group $\hat{\mathcal{G}}_k$.

Lemma 4. *Let* $|\hat{\mathcal{C}}| = n_C$ *and* $|\hat{\mathcal{G}}_k| = n_G$. *Algo. 4 spends* $O(n_C^3 f_3(n_G))$ *time, where* $f_3(\cdot)$ *is Algo. 3's time complexity.*

Proof: The if-statement in lines 2–3 removes all CUEs from $\hat{\mathcal{C}}$ if $\hat{\mathcal{G}}_k$ has a CUE, which takes $O(n_C)$ time. Then, the for-loop in lines 4–7 spends time of $n_C \cdot O(f_3(n_G) + n_G) = O(n_C f_3(n_G))$, as it computes transmitted power in $\hat{\mathcal{G}}_k \cup \{u_i\}$ by Algo. 3 and then finds the minimum SINR, where each UE u_i is picked from $\hat{\mathcal{C}}_{\text{tmp}}$. Line 8 calls the BnB method to search new members from $\hat{\mathcal{C}}_{\text{tmp}}$. There are some efficient methods proposed to reduce visiting nodes in the state space tree. For example, [40] shows that it visits at most $O(n_C^3)$ tree nodes in BnB. Besides, the bound function in Algo. 5 takes $O(f_3(n_G))$ time due to invoking Algo. 3. Since line 8 checks each visiting node by the bound function, it will take $O(n_C^3 f_3(n_G))$ time. Finally, those UEs that appear in $\hat{\mathcal{G}}_{\text{new}}$ are removed from $\hat{\mathcal{C}}$. This operation takes $O(n_C)$ time, as we search each UE in $\hat{\mathcal{C}}$. Thus, the total time complexity is $O(n_C) + O(n_C f_3(n_G)) + O(n_C^3 f_3(n_G)) + O(n_C) = O(n_C^3 f_3(n_G))$. □

Theorem 4. *Algo. 4 is optimal in the sense that it can find out the most new members for group* $\hat{\mathcal{G}}_k$ *from* $\hat{\mathcal{C}}$.

Proof: Without loss of generality, suppose that a UE $u_i \in \hat{\mathcal{C}}$ is not selected by Algo. 4 but it can actually be added to $\hat{\mathcal{G}}_k$. This situation will occur in one of three parts in Algo. 4: 1) the if-statement in lines 2–3, 2) the for-loop in lines 4–7, and 3) the BnB method in line 8. If u_i is excluded by part 1, it will be a CUE and $\hat{\mathcal{G}}_k$ must contain a CUE (not u_i). Thus, a contradiction occurs since two CUEs cannot be added to the same group. Then, let us consider that u_i is excluded by part 2. In this case, adding u_i to $\hat{\mathcal{G}}_k$ will violate the SINR constraint in Eq. (4) due to the check in line 6, so a contradiction occurs. Finally, if u_i is excluded by part 3, it means that u_i cannot pass the bound function in Algo. 5. Since adding u_i makes the condition of $|\hat{\mathcal{G}}_{\text{new}}| > n_{\max}$ hold in line 6 of Algo. 5, the only way to exclude u_i in Algo. 5 must lie in the if-statement of lines 4–5. In this case, if u_i is added to $\hat{\mathcal{G}}_k$, the SINRs τ_j^k of some members in $\hat{\mathcal{G}}_k$ must be below their thresholds $\tau_j^{\min}$. Obviously, a contradiction also occurs.

Based on the above argument, such a UE u_i should not exist in $\hat{\mathcal{C}}$. In other words, any UE of the optimal solution in $\hat{\mathcal{C}}$ will not be excluded by the check of Algo. 4. Since the BnB method can find out the optimal solution from $\hat{\mathcal{C}}$, Algo. 4 can thus add the maximum number of new members to $\hat{\mathcal{G}}_k$. □

Algorithm 6: Throughput Raising Algorithm

1 $\hat{\mathcal{G}}_{\text{tmp}} \leftarrow \hat{\mathcal{G}}_k$;
2 **while** *true* **do**
3 $\quad$ **foreach** $u_i \in \hat{\mathcal{G}}_{\text{tmp}}$ **do**
4 $\quad\quad \tilde{p}_i^{\text{new}} \leftarrow \tilde{p}^k_{t(i),i} + \delta$;
5 $\quad\quad$ **if** $\tilde{p}_i^{\text{new}} > p_i^{\max}$ *or* $\min_{u_j \in \hat{\mathcal{G}}_k} \tau_j^k < \tau_j^{\min}$ **then**
6 $\quad\quad\quad \hat{\mathcal{G}}_{\text{tmp}} \leftarrow \hat{\mathcal{G}}_{\text{tmp}} \setminus \{u_i\}$;
7 $\quad\quad$ **else**
8 $\quad\quad\quad \xi_i^{\text{inc}} \leftarrow \sum_{u_j \in \hat{\mathcal{G}}_k} (\xi_j^{\text{new}} - \xi_j)$;
9 $\quad\quad\quad$ **if** $\xi_i^{\text{inc}} \leq 0$ **then**
10 $\quad\quad\quad\quad \hat{\mathcal{G}}_{\text{tmp}} \leftarrow \hat{\mathcal{G}}_{\text{tmp}} \setminus \{u_i\}$;
11 $\quad$ **if** $\hat{\mathcal{G}}_{\text{tmp}} = \emptyset$ **then**
12 $\quad\quad$ break;
13 $\quad u_i \leftarrow \arg\max_{u_i \in \hat{\mathcal{G}}_{\text{tmp}}} \xi_i^{\text{inc}}$;
14 $\quad \tilde{p}^k_{t(i),i} \leftarrow \tilde{p}_i^{\text{new}}$;

5.5 Throughput Raising Algorithm

After finding all members of a group $\hat{\mathcal{G}}_k$, the last algorithm seeks to raise transmitted power for some UEs to improve the overall throughput in $\hat{\mathcal{G}}_k$. Algo. 6 gives its pseudocode, where $\hat{\mathcal{G}}_{\text{tmp}}$ is a subset of UEs in $\hat{\mathcal{G}}_k$ whose senders are capable of raising power. Initially, $\hat{\mathcal{G}}_{\text{tmp}}$ is set to $\hat{\mathcal{G}}_k$. For the sender $t(i)$ of each UE u_i in $\hat{\mathcal{G}}_{\text{tmp}}$, we add an amount δ of power to its transmitted power and record the result in $\tilde{p}_i^{\text{new}}$. Then, line 5 checks if $\tilde{p}_i^{\text{new}}$ exceeds the upper bound $p_i^{\max}$ or using $\tilde{p}_i^{\text{new}}$ makes some UEs in $\hat{\mathcal{G}}_k$ violate the SINR constraint (i.e., $\min_{u_j \in \hat{\mathcal{G}}_k} \tau_j^k < \tau_j^{\min}$). If so, it is infeasible to increase $t(i)$'s power and u_i is removed from $\hat{\mathcal{G}}_{\text{tmp}}$. Otherwise, we compute the amount of *throughput gain* in $\hat{\mathcal{G}}_k$ (denoted by ξ_i^{inc}) when $t(i)$ uses the new power $\tilde{p}_i^{\text{new}}$ by line 8. Let ξ_j^{new} and ξ_j denote the amount of throughput of a UE u_j in $\hat{\mathcal{G}}_k$ when $t(i)$'s power is $\tilde{p}_i^{\text{new}}$ and $\tilde{p}^k_{t(i),i}$, respectively[1]. If ξ_i^{inc} is not positive, which implies that there is no throughput gain by using the new power $\tilde{p}_i^{\text{new}}$, u_i is removed from $\hat{\mathcal{G}}_{\text{tmp}}$.

In case that all UEs in $\hat{\mathcal{G}}_{\text{tmp}}$ are removed by the above check (i.e., line 11), the current value of transmitted power is the best one, so the algorithm finishes in line 12. When $\hat{\mathcal{G}}_{\text{tmp}} \neq \emptyset$, we pick a UE u_i in $\hat{\mathcal{G}}_{\text{tmp}}$ that has the largest throughput gain ξ_i^{inc} in line 13, and then update $\tilde{p}^k_{t(i),i}$ by $\tilde{p}_i^{\text{new}}$ in line 14.

Although Algo. 6 contains a while-loop that seems to repeat forever (i.e., the checking condition is set to *true* in line 2), Theorem 5 verifies that this algorithm never diverges. Besides, Lemma 5 analyzes the time complexity of Algo. 6.

1. Both ξ_j^{new} and ξ_j can be calculated by the Shannon formula.

Theorem 5. *Algo. 6 must converge if group $\hat{\mathcal{G}}_k$ has a finite number of UEs.*

Proof: Observing from Algo. 6, the termination condition of the while-loop is $\hat{\mathcal{G}}_{\text{tmp}} = \emptyset$ in line 11, where $\hat{\mathcal{G}}_{\text{tmp}}$ is initially set to $\hat{\mathcal{G}}_k$. Then, we use a for-loop in lines 3–10 to iteratively pick one UE u_i from $\hat{\mathcal{G}}_{\text{tmp}}$ and raise the transmitted power of its sender by δ. Specifically, u_i may be removed from $\hat{\mathcal{G}}_{\text{tmp}}$ due to the conditions in either lines 5 or 9. This implies that the size of $\hat{\mathcal{G}}_{\text{tmp}}$ only decreases and never increases.

Suppose that a UE u_j always stays in $\hat{\mathcal{G}}_{\text{tmp}}$, which makes the while-loop run forever. According to lines 4, 13, and 14, the transmitted power of u_j's sender will be kept increasing in some iterations of the while-loop. The power will eventually exceed $p_j^{\max}$, making u_j be removed from $\hat{\mathcal{G}}_{\text{tmp}}$ by line 5. This causes a contradiction, which shows that $\hat{\mathcal{G}}_{\text{tmp}}$ will be empty and the while-loop terminates accordingly. Thus, Algo. 6 must converge when $\hat{\mathcal{G}}_k$ contains a finite number of UEs. □

Lemma 5. *Assume that group $\hat{\mathcal{G}}_k$ has n_G UEs. Algo. 6 takes $O(\beta n_G^2)$ time, where $\beta = \max_{u_i \in \hat{\mathcal{G}}_k}(p_i^{\max} - p_i^{\min})/\delta$.*

Proof: The inner for-loop repeats at most n_G times. Both lines 5 and 8 take $O(n_G)$ time, as they search all UEs in $\hat{\mathcal{G}}_k$ once. Thus, the for-loop spends time of $n_G \times O(n_G) = O(n_G^2)$. Since line 13 also takes $O(n_G)$ time, one iteration of the while-loop takes time of $O(n_G^2) + O(n_G) = O(n_G^2)$. Then, the worst case occurs when $\tilde{p}_{t(i),i}^k$ is initially set to $p_i^{\min}$ and $\tilde{p}_i^{\text{new}}$ is iteratively increased to $p_i^{\max}$. In this case, the while-loop will be repeated $\max_{u_i \in \hat{\mathcal{G}}_k}(p_i^{\max} - p_i^{\min})/\delta$ times. Thus, the time complexity of Algo. 6 is $O(\beta n_G^2)$. □

5.6 Discussion

By building graph $\mathcal{H}$ to display interference relation of UEs, DRAPC uses Algo. 1 to group UEs based on three rules: 1) if a group contains a CUE, only DUEs are allowed to join the group; 2) a group can comprise just DUEs (i.e., the pure D2D model); 3) each UE in a group meets its SINR demand (i.e., $\min_{u_i \in \hat{\mathcal{G}}_k} \tau_i^k \geq \tau_i^{\min}$). In this way, every group of UEs can share RBs without interference. From the flowchart in Fig. 3, if the candidate set $\hat{\mathcal{C}}$ becomes empty, which means that every UE in $\hat{\mathcal{U}}$ has been allocated with RBs by Algo. 1, Algo. 6 is called to improve throughput by raising transmitted power.

Nevertheless, DRAPC is not just a graph-coloring based method. Instead, it adopts vertex coloring to get a *rudimentary* result of groups. When $\hat{\mathcal{C}}$ is not empty after the execution of Algo. 1, the sophisticated Algo. 2 then reforms each group by carefully exchanging its members and the unassigned UEs in $\hat{\mathcal{C}}$ to improve signal quality in the group. This involves the adjustment of transmitted power to satisfy SINR demands of UEs by Algo. 3. Then, Algo. 4 checks if extra members (from $\hat{\mathcal{C}}$) can be included in each group to support more links (i.e., the objective function in Eq. (3)), which makes the result closer to the optimal solution. Though Algo. 4 uses the time-consuming BnB method (i.e., Algo. 5) to search extra members, it checks only residual UEs in $\hat{\mathcal{C}}$ rather than all UEs in $\hat{\mathcal{U}}$. Thus, the problem space for BnB is greatly reduced, and Algo. 4 would not spend much time in practice. Afterwards, Algo. 6 is used to maximize network throughput. These special designs in Algo. 2 (together with auxiliary algorithms in Algos. 3, 4, and 6) distinguish our DRAPC framework from other graph-coloring based methods. Theorem 6 analyzes the time complexity of DRAPC.

TABLE 3: Simulation parameters, where 'SD' is standard deviation.

eNB's parameters:	
channel bandwidth	5 MHz (supports 25 RBs per TTI)
cell range	500 meters
transmitted power	min: -40 dBm, max: 46 dBm
frame structure	frequency division duplexing (FDD)
modulation	QPSK, 16QAM, 64QAM
UE's parameters:	
number of CUEs	10 and 25
number of D2D pairs	small-scale: 5–25, large-scale: 25–75
transmitted power	min: -40 dBm, max: 23 dBm
SINR threshold $\tau_i^{\min}$	10 dB and 15 dB
channel's parameters:	
propagation loss	urban macro-cell model
path loss	eNB to CUE: $128.1 + 37.6 \log L$
	D2D pair: $148 + 40 \log L$
slow fading	shadowing fading:
	log-normal distribution (mean: 0 dB, SD: 8 dB)
fast fading	the Rayleigh fading model
thermal noise N_0	-174dBm/Hz

Theorem 6. *Given n_U UEs and n_R RBs, DRAPC spends time of $O(n_U^2(n_R + \beta) + \alpha n_U n_R (n_U - n_R)^3)$, where α and β are the parameters used in Algos. 3 and 6, respectively.*

Proof: DRAPC is composed of Algos. 1 and 2. Based on Lemmas 1 and 2, its time complexity is $O(n_U^2 n_R) + f_3(n_U) + n_C(O(n_U + f_3(n_U))) + n_R f_4(n_C) + f_6(n_U)$, where $|\hat{\mathcal{C}}| = n_C$. By Lemmas 3, 4, and 5, the above equation can be calculated by $O(n_U^2 n_R + \alpha n_U + n_C(n_U + \alpha n_U) + \alpha n_U n_R n_C^3 + \beta n_U^2) = O(n_U^2(n_R + \beta) + \alpha n_U n_R n_C^3)$. Since each group must contain at least one UE (or the corresponding RB will be wasted), we have $n_C \leq \{n_U - n_R, 0\}$. Therefore, the time complexity of DRAPC is $O(n_U^2(n_R + \beta) + \alpha n_U n_R (n_U - n_R)^3)$. □

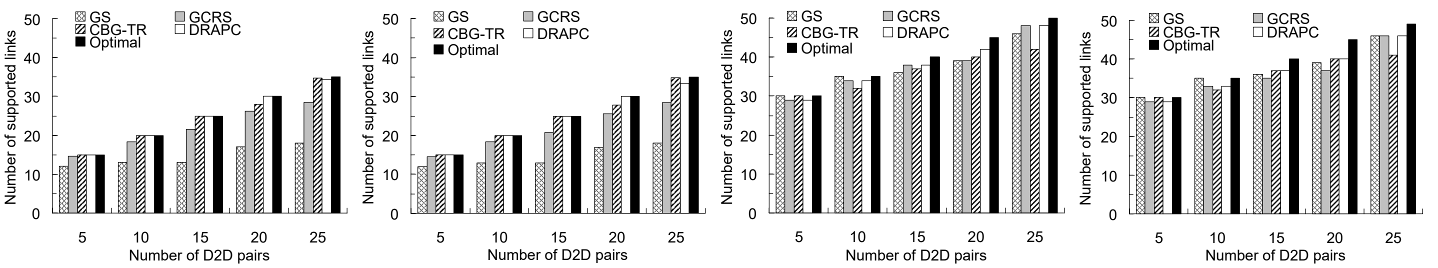
In Algo. 3, parameter α decides the number of iterations used to adjust power $\tilde{p}_{t(i),i}^k$. It can be set as a small constant (e.g., $\alpha \leq 10$). According to Lemma 5, parameter β is equal to $\max_{u_i \in \hat{\mathcal{G}}_k}(p_i^{\max} - p_i^{\min})/\delta$, where $p_i^{\max}$, $p_i^{\min}$, and δ are given inputs and not large, so β will be also a small constant. As discussed earlier in Section 1, the number of RBs provided by the eNB depends on the bandwidth of the downlink channel, and n_R is within [6, 100]. However, the number of UEs could be pretty large in some scenarios (e.g., dense cities [41] or IoT applications [7]), so it is expected that $n_R \ll n_U$. Based on the above observation, Corollary 1 gives the time complexity of DRAPC after simplification.

Corollary 1. *Let α and β be constants. The time complexity of DRAPC in Theorem 6 can be simplified to $O(n_U n_R (n_U - n_R)^3)$ if $n_R \ll n_U$.*

From Corollary 1, n_U obviously dominates the time complexity of DRAPC. To save the computational cost of DRAPC, one can limit the maximum number of links to share each RB (denoted by λ). However, this solution may also degrade its performance. In Section 6, we will investigate the effect of λ and give a suitable value in practical implementation.

6 Performance Evaluation

Our simulation is conducted by MATLAB, where Table 3 lists its parameters [42]. We consider a macrocell with 10 or 25 CUEs, and vary the number of D2D pairs. The effect of path loss depends on the distance L between two nodes. For a D2D pair, L ranges from 1 to 20 meters. As discussed in Section 3, the *Gale-Shapley (GS) method* [35] can achieve the weak Pareto optimality, so it is taken for comparison. Except for GS, we



(a) 10 CUEs, $\tau_i^{\min}$: 10dB (b) 10 CUEs, $\tau_i^{\min}$: 15dB (c) 25 CUEs, $\tau_i^{\min}$: 10dB (d) 25 CUEs, $\tau_i^{\min}$: 15dB

Fig. 5: Comparison on the number of supported links between each method and the optimal solution in a small-scale network.

compare DRAPC with two graph-coloring based methods. One is the *graph coloring based resource sharing (GCRS) method* [32] mentioned in Section 3. The other is the *color-based grouping and throughput raising (CBG-TR) method*, which allocates RBs and power by Algos. 1 and 6, respectively. Here, CBG-TR is also used to study the effect of our group reforming algorithm (in Algo. 2) on DRAPC. As discussed in Section 5.6, limiting the number λ of links that share each RB helps reduce the computational cost of DRAPC but degrades its performance. Thus, we set λ to 2, 3, 4, and ∞ (i.e., the original DRAPC) for comparison.

6.1 Optimality Gap

We first observe the gap between the optimal solution and each method, where the optimal solution is derived by solving the MINP problem defined in Section 4.3. Fig. 5 gives the experimental data. Specifically, the gap grows when there are more D2D pairs. Besides, as $\tau_i^{\min}$ increases, the stricter SINR constraint in Eq. (4) makes resource sharing become more difficult, so the number of supported links decreases. On the other hand, increasing CUEs also raises the number of supported links. The reason is that the overall links (i.e., cellular and D2D communication) increase, and each D2D pair has more candidate CUEs to choose for resource sharing.

Let $\mathcal{N}_{\text{opt}}$ and $\mathcal{N}_x$ be the number of supported links in the optimal solution and a method x, respectively. We define the *optimality gap (OG) ratio* for method x as follows:

$$\frac{\mathcal{N}_{\text{opt}} - \mathcal{N}_x}{\mathcal{N}_{\text{opt}}} \times 100\%. \tag{9}$$

A smaller OG ratio implies that the solution found by method x is closer to the optimal one. In particular, when there are 10 CUEs, the average OG ratios of GS, GCRS, CBG-TR, and DRAPC are 48.6%, 18.9%, 0.6%, and 2.9%, respectively. The average OG ratios of GS, GCRS, CBG-TR, and DRAPC reduce to 7.1%, 5.2%, 16.2%, and 5.0%, respectively, as there are 25 CUEs. These results show that our DRAPC framework can find a solution closer to the optimal one in a small-scale network, as compared with other methods.

6.2 Network Performance

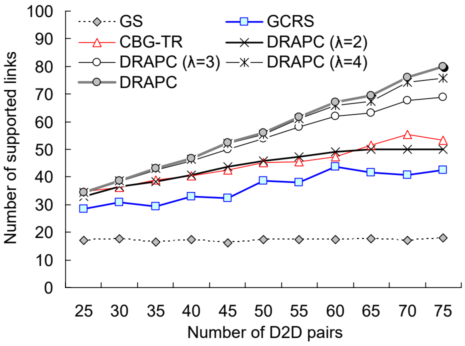
Next, we evaluate the performance of different methods in a large-scale network (with 25–75 D2D pairs). Since solving the MINP problem for more D2D pairs spends too much time, we do not show the optimal solution in the following experiments.

Fig. 6(a)–(d) present the number of links supported by each method. As mentioned earlier, raising CUEs or reducing $\tau_i^{\min}$ helps increase the number of supported links. Except for GS, the number of links supported by each other method increases significantly as there are more D2D pairs. This result shows the superiority of graph-coloring based methods over GS. Since GCRS adopts a more complex approach for graph coloring, it performs better than CBG-TR in the case of 25 CUEs. However, when there are fewer CUEs (i.e., 10 CUEs), GCRS cannot find more D2D pairs to share the RBs allocated to CUEs, so it performances worse than CBG-TR. To conquer this problem, DRAPC allows some RBs to be shared merely by DUEs (i.e., the pure D2D model). Thus, DRAPC provides more flexibility in resource allocation and supports the most D2D links in both cases of 10 and 25 CUEs.

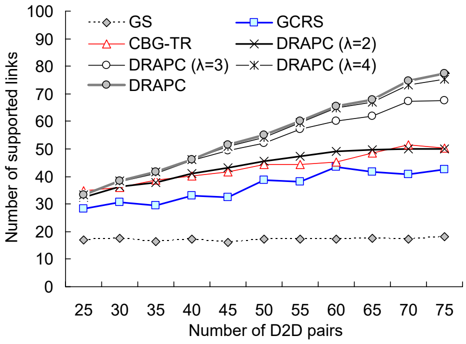
Fig. 6(e)–(h) measure the average throughput of each link. Evidently, adding more D2D pairs degrades the throughput, because more UEs share the fixed resources and cause higher interference. For GS, increasing CUEs significantly raises its throughput. GCRS outperforms CBG-TR when there are 25 CUEs, but the result is reversed if there are only 10 CUEs. Comparing with these methods, our DRAPC framework always keeps the highest throughput. Moreover, increasing $\tau_i^{\min}$ from 10 dB to 15 dB helps improve DRAPC's throughput. The reason is that it becomes stricter to put more UEs in a group to share the same RB. In this case, each group may contain fewer members that impose less interference on each other, so they can use more complex MCSs for data transmissions.

Fig. 6(i)–(l) evaluate the link outage ratio, where a link is considered as outage if it gets no RB to transmit data. When the number of D2D pairs grows, the outage ratio increases, since the eNB has to allocate the constant RBs to more links. However, increasing CUEs reduces the outage ratio in most methods. The reason is that the eNB has more choices of CUEs when finding D2D pairs to reuse their RBs. Besides, the maximum outage ratios of GS, GCRS, and CBG-TR are 78.7%, 49.9%, and 44.9%, respectively. By contrast, DRAPC always keeps the outage ratio below 9.2%, which is much lower than the other methods.

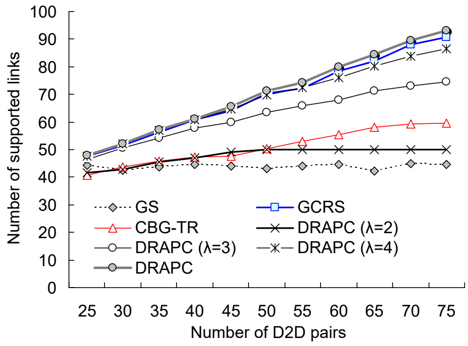
Three observations are obtained from the experimental data in Fig. 6. First, unlike GCRS whose performance substantially degrades in the case of 10 CUEs, our DRAPC framework can still perform well by taking advantage of the pure D2D model when there are fewer CUEs. In fact, DRAPC supports the most links, keeps the highest throughput, and has the smallest link outage ratio, under different combinations of CUE number and $\tau_i^{\min}$ value. Second, the gap in performance between CBG-TR and DRAPC is pretty large. This result verifies that the group reforming algorithm in Algo. 2 plays a critical role in DRAPC, which carefully finds better members for each group of UEs and thereby distinguishes DRAPC from other graph-coloring based methods. Third, limiting the number λ of links to share



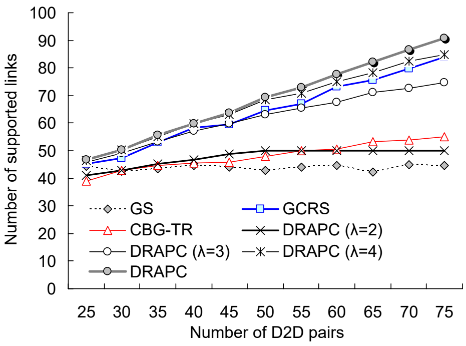
(a) supported links (10 CUEs, $\tau_i^{\min}$: 10dB)



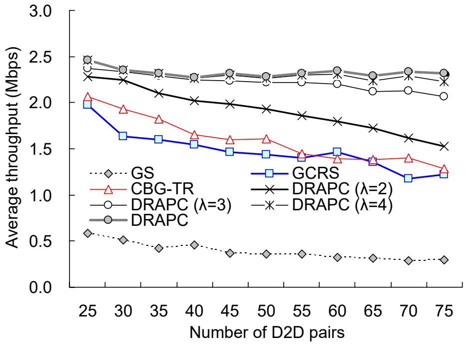
(b) supported links (10 CUEs, $\tau_i^{\min}$: 15dB)



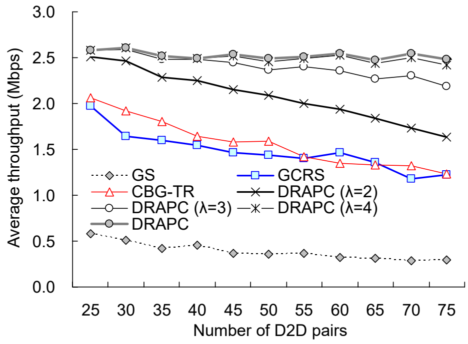
(c) supported links (25 CUEs, $\tau_i^{\min}$: 10dB)



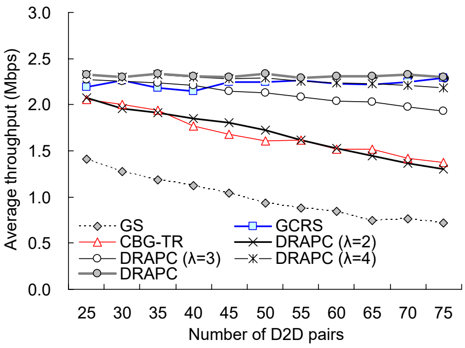
(d) supported links (25 CUEs, $\tau_i^{\min}$: 15dB)



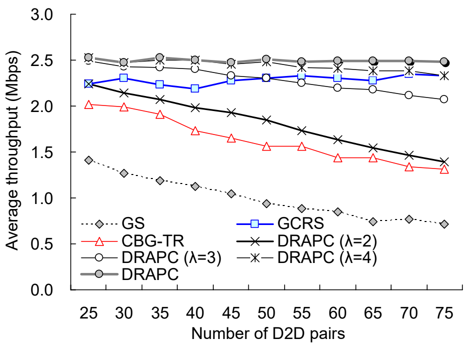
(e) throughput (10 CUEs, $\tau_i^{\min}$: 10dB)



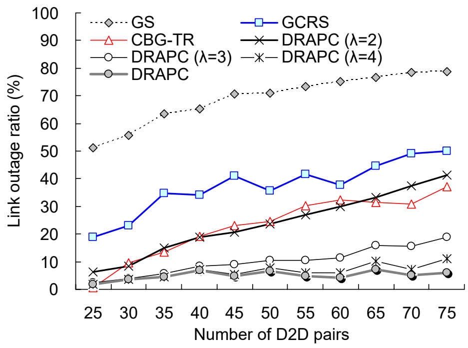
(f) throughput (10 CUEs, $\tau_i^{\min}$: 15dB)



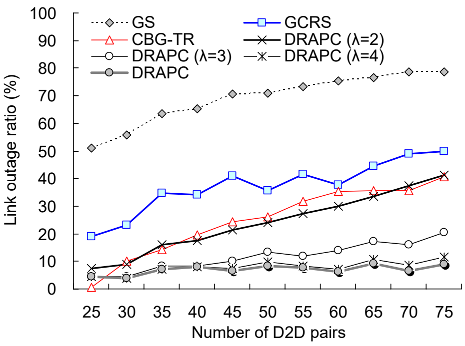
(g) throughput (25 CUEs, $\tau_i^{\min}$: 10dB)



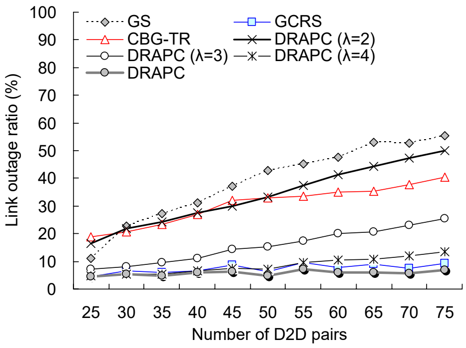
(h) throughput (25 CUEs, $\tau_i^{\min}$: 15dB)



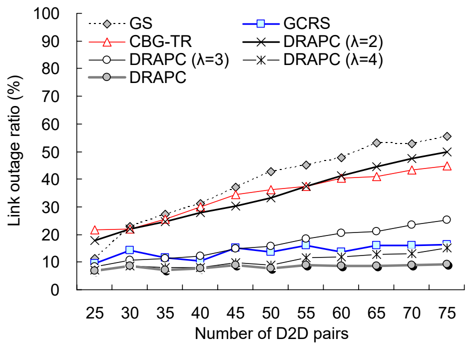
(i) link outage (10 CUEs, $\tau_i^{\min}$: 10dB)



(j) link outage (10 CUEs, $\tau_i^{\min}$: 15dB)



(k) link outage (25 CUEs, $\tau_i^{\min}$: 10dB)



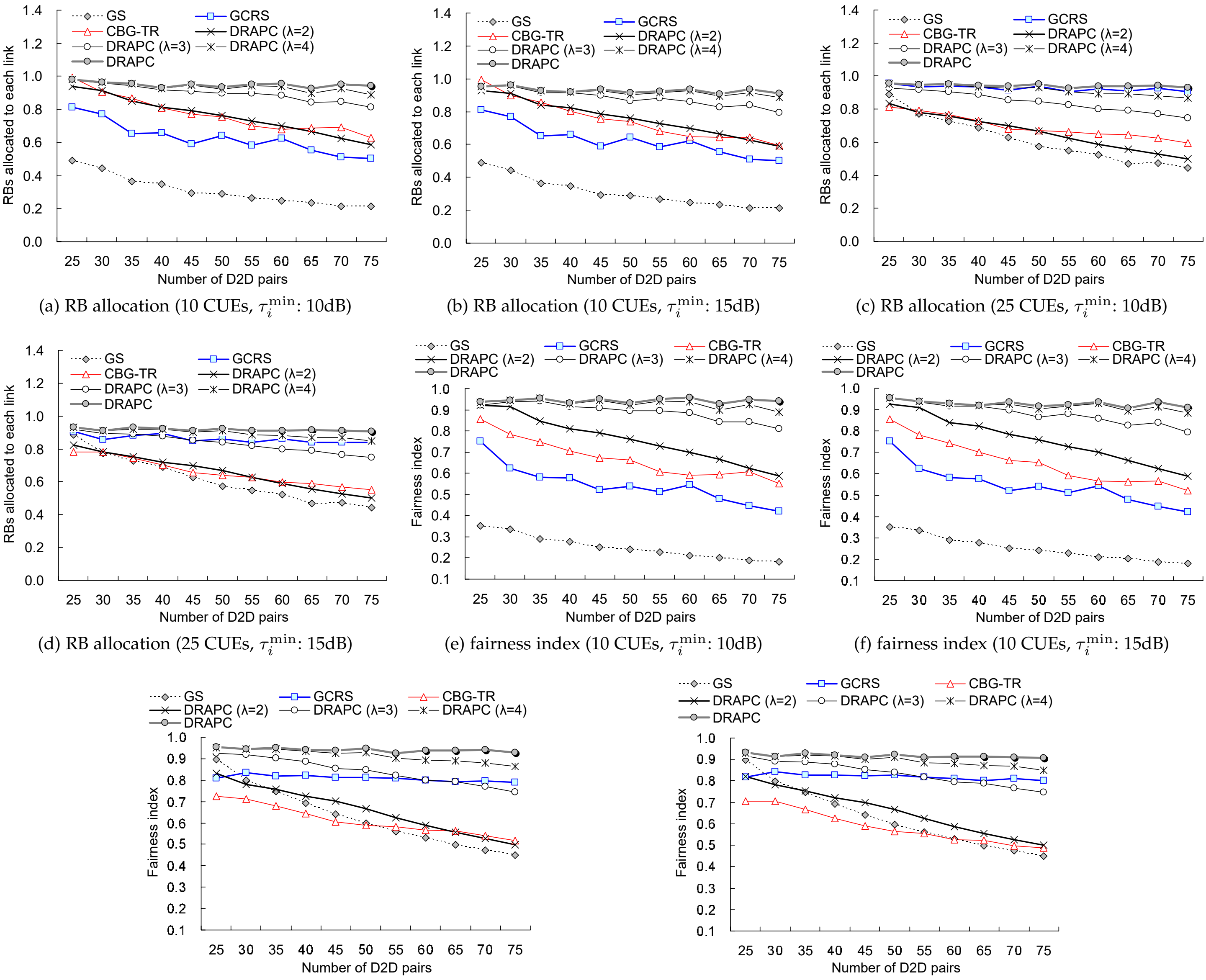
(l) link outage (25 CUEs, $\tau_i^{\min}$: 15dB)

Fig. 6: Comparison on network performance, including the number of supported links, the average throughput, and the link outage ratio.

each RB reduces the computational overhead of DRAPC but degrades its performance. Based on the experimental data, we show that the *restricted* DRAPC framework still wins most methods in the case of $\lambda = 4$. In other words, we can balance between the computational cost and performance of DRAPC by setting λ to 4.

6.3 Fairness

D2D links may experience different levels of interference caused by their surrounding environment. When the interference source is kept for a long while but nearby DUEs have low or even no mobility, these DUEs would suffer from bad channel quality for a long time. If one method seeks to raise throughput by merely giving most resources to the D2D links



(a) RB allocation (10 CUEs, $\tau_i^{\min}$: 10dB)
(b) RB allocation (10 CUEs, $\tau_i^{\min}$: 15dB)
(c) RB allocation (25 CUEs, $\tau_i^{\min}$: 10dB)
(d) RB allocation (25 CUEs, $\tau_i^{\min}$: 15dB)
(e) fairness index (10 CUEs, $\tau_i^{\min}$: 10dB)
(f) fairness index (10 CUEs, $\tau_i^{\min}$: 15dB)
(g) fairness index (25 CUEs, $\tau_i^{\min}$: 10dB)
(h) fairness index (25 CUEs, $\tau_i^{\min}$: 15dB)

Fig. 7: Comparison on fairness, including the average number of RBs allocated to each link and the Jain's fairness index.

with good channel quality, there is a high possibility that other links will be starved. Thus, we evaluate the degree of fairness that each method can achieve in terms of resource allocation in this section.

Fig. 7(a)–(d) give the average number of RBs allocated to each link. Intuitively, if a method lets every link get more RBs (by resource sharing), it can improve the resource utilization and alleviate starvation (as those links with bad channel quality have a relatively high opportunity to get RBs). Since the total number of RBs is constant (i.e., $|\hat{\mathcal{R}}| = 25$ per TTI), the result decreases when there are more D2D pairs. Although GS achieves the weak Pareto optimality in theory, it actually allocates the fewest RBs to each link. For graph-coloring based methods, GCRS outperforms CBG-TR in the case of 25 CUEs, but its performance drops significantly when there are only 10 CUEs in the network. Comparing with these methods, DRAPC always allocates the most number of RBs to each link, which also shows its outstanding performance in resource allocation. When applying the λ restriction to DRAPC, the experimental data verify that it can still allocate more RBs to each link than most other methods by setting $\lambda = 4$.

Then, we adopt the Jain's fairness index [41] to evaluate the degree of fairness in each method, which is defined as follows:

$$FI = \frac{\left(\sum_{i=1}^{m} \tilde{r}_i\right)^2}{m \sum_{i=1}^{m} \tilde{r}_i^2}, \tag{10}$$

where $\tilde{r}_i$ denotes the normalized throughput of a link and m is the number of total links. According to Eq. (10), the condition of $0 < FI \leq 1$ must hold, and a larger FI value means that the method achieves more fair transmissions. Fig. 7(e)–(h) give the experimental data. On the whole, the fairness index decreases when there are more D2D pairs, because more UEs compete for the limited resources and some of them may lose the competition (i.e., getting fewer or even no RBs). When there are only 10 CUEs, both GS and GCRS have evidently lower fairness indices than CBG-TR and DRAPC. The reason is that CBG-TR and DRAPC allow some RBs to be shared solely by D2D pairs (i.e., the pure D2D model). In this case, there

TABLE 4: Ratio of pure D2D pairs.

number of CUEs	5	10	15	20	25
ratio of pure D2D pairs	28%	22.7%	16%	8%	4%

is a large opportunity for DUEs with bad channel quality to get RBs, thereby increasing the fairness index. Thanks to the sophisticated group reforming algorithm in Algo. 2, DRAPC puts more UEs in each group to share resources, so it always keeps the highest fairness index among all methods.

6.4 Pure D2D Model

Finally, we study how many D2D pairs are allocated with RBs based on the pure D2D model in DRAPC. For ease of presentation, such D2D pairs are called *pure D2D pairs*. In this experiment, there are 75 D2D pairs and the λ restriction is not applied. Table 4 gives the experimental data. As the number of CUEs increases, the ratio of pure D2D pairs significantly decreases. The reason is that more CUEs ask for resources, so the eNB prefers letting D2D pairs share RBs with CUEs. The result in Table 4 verifies that DRAPC can take advantage of the pure D2D model to facilitate RB allocation, especially when there are fewer CUEs in the network.

7 Conclusion and Future Work

D2D communication improves LTE-A performance by letting UEs share resources. Many studies are devoted to finding DUEs to reuse RBs of CUEs to raise throughput. This paper proposes a pure D2D model to provide flexibility for resource allocation, which allows some RBs to be shared solely by DUEs. Based on the new model, an efficient DRAPC framework is proposed with the objective of maximizing the number of supported links. It groups UEs by vertex coloring, reforms each group through BnB, and raises transmitted power to better SINRs. Simulation results show that DRAPC well exploits the pure D2D model to increase throughput, reduce link outage, raise resource utilization, and improve fairness, as compared with GS and graph-coloring based methods. Furthermore, we can save the computational cost of DRAPC while achieving a similar performance by setting $\lambda = 4$.

Some expansions are worth elaborating in future work. First, like most other schemes, DRAPC relies on correct CSI to get channel conditions of all links for power control and RB allocation. Thus, we will seek to improve the robustness of DRAPC for the case of imperfect CSI. One similar issue is when some UEs have high mobility, which could bring about inaccuracy in CSI. Second, we use the SINR constraint in Eq. (4) to provide QoS for UEs. It deserves further investigation to take various traffic patterns into account. One possible manner is to adaptively change the minimum SINR threshold $\tau_i^{\min}$ based on the traffic pattern of each UE. Finally, DRAPC can be applied to a *heterogeneous network (HetNet)* by considering the interference caused by small-cell eNBs in the calculation of SINR (i.e., Eqs. (1) and (2)). It is interesting to address how to share traffic loads of some eNBs in a HetNet with the help of D2D communication. In this case, some extensions of DRAPC can be developed for the above scenario.

References

[1] R. Zhang, Z. Zheng, M. Wang, X. Shen, and L.L. Xie, "Equivalent capacity in carrier aggregation-based LTE-A systems: a probabilistic analysis," *IEEE Trans. Wireless Comm.*, vol. 13, no. 11, pp. 6444–6460, 2014.

[2] W. Guo, J. Fan, G.Y. Li, Q. Yin, X. Zhu, and Y. Fu, "MIMO transmission with vertical sectorization for LTE-A downlink," *IEEE Wireless Comm. Letters*, vol. 5, no. 4, pp. 372–375, 2016.

[3] Y.C. Wang and K.C. Chien, "EPS: energy-efficient pricing and resource scheduling in LTE-A heterogeneous networks," *IEEE Trans. Vehicular Technology*, vol. 67, no. 9, pp. 8832–8845, 2018.

[4] P. Mach, Z. Becvar, and T. Vanek, "In-band device-to-device communication in OFDMA cellular networks: a survey and challenges," *IEEE Comm. Surveys & Tutorials*, vol. 17, no. 4, pp. 1885–1922, 2015.

[5] Y.C. Wang and T.Y. Tsai, "A pricing-aware resource scheduling framework for LTE networks," *IEEE/ACM Trans. Networking*, vol. 25, no. 3, pp. 1445–1458, 2017.

[6] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-device communication in LTE-advanced networks: a survey," *IEEE Comm. Surveys & Tutorials*, vol. 17, no. 4, pp. 1923–1940, 2015.

[7] P.K. Wali, A. Aadhithan, and D. Das, "Optimal time-spatial randomization techniques for energy efficient IoT access in LTE-advanced," *IEEE Trans. Vehicular Technology*, vol. 66, no. 8, pp. 7346–7359, 2017.

[8] J. Lee and S. Leyffer, *Mixed Integer Nonlinear Programming*. Berlin, Germany: Springer, 2012.

[9] 3GPP, "Universal Mobile Telecommunications System (UMTS); LTE; Proximity-based services (ProSe)," ETSI TS 123 303 V12.2.0, 2014.

[10] 3GPP, "Technical Specification Group Core Network and Terminals; Proximity services (ProSe) User Equipment (UE) to Proximity-services (ProSe) Function Protocol Aspects (Release 12)," 3GPP TS 24.334 V1.1.0, 2014.

[11] Z. Kaleem, N.N. Qadri, T.Q. Duong, and G.K. Karagiannidis, "Energy-efficient device discovery in D2D cellular networks for public safety scenario," *IEEE Systems J.*, vol. 13, no. 3, pp. 2716–2719, 2019.

[12] Z. Kaleem, Y. Li, and K. Chang, "Public safety users' priority-based energy and time-efficient device discovery scheme with contention resolution for ProSe in third generation partnership project long-term evolution-advanced systems," *IET Comm.*, vol. 10, no. 15, pp. 1873–1883, 2016.

[13] 3GPP, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," ETSI TS 136 213 V14.2.0, 2017.

[14] K. Doppler, C.H. Yu, C.B. Ribeiro, and P. Janis, "Mode selection for device-to-device communication underlaying an LTE-advanced network," *Proc. IEEE Wireless Comm. and Networking Conf.*, 2010, pp. 1–6.

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

[16] Z. Liu, T. Peng, S. Xiang, and W. Wang, "Mode selection for device-to-device (D2D) communication under LTE-advanced networks," *Proc. IEEE Int'l Conf. Comm.*, 2012, pp. 5563–5567.

[17] G. Yu, L. Xu, D. Feng, R. Yin, G.Y. Li, and Y. Jiang, "Joint mode selection and resource allocation for device-to-device communications," *IEEE Trans. Wireless Comm.*, vol. 62, no. 11, pp. 3814–3824, 2014.

[18] J.Y. Pan and Y.Y. Liu, "Device-to-device interference avoidance underlaying cellular downlink transmission," *Proc. IEEE Asia-Pacific Conf. Comm.*, 2015, pp. 464–469.

[19] X. Li, X. Liu, Y. Li, L. Xiao, and J. Wang, "A cooperative rate splitting scheme for device-to-device underlay cellular communication," *Proc. IEEE Asia-Pacific Conf. Comm.*, 2017, pp. 1–5.

[20] B. Wang, L. Chen, X. Chen, X. Zhang, and D. Yang, "Resource allocation optimization for device-to-device communication underlaying cellular networks," *Proc. IEEE Vehicular Technology Conf.*, 2011, pp. 1–6.

[21] Q. Duong, Y. Shin, and O.S. Shin, "Distance-based resource allocation scheme for device-to-device communications underlaying cellular networks," *Int'l J. Electronics and Comm.*, vol. 69, no. 10, pp. 1437–1444, 2015.

[22] P. Liu, C. Hu, T. Peng, and W. Wang, "Distributed cooperative admission and power control for device-to-device links with QoS protection in cognitive heterogeneous network," *Proc. Int'l Conf. Comm. and Networking in China*, 2012, pp. 712–716.

[23] W. Zhao and S. Wang, "Resource sharing scheme for device-to-device communication underlaying cellular networks," *IEEE Trans. Comm.*, vol. 63, no. 12, pp. 4838–4848, 2015.

[24] Y. Li, M. Sheng, Y. Zhu, T. Jiang, and J. Li, "Sum rate maximization in underlay SCMA device-to-device networks," *Proc. IEEE Global Comm. Conf.*, 2016, pp. 1–6.

[25] H. Xu, W. Xu, Z. Yang, Y. Pan, J. Shi, and M. Chen, "Energy-efficient resource allocation in D2D underlaid cellular uplinks," *IEEE Comm. Letters*, vol. 21, no. 3, pp. 560–563, 2017.

[26] D.D. Penda, L. Fu, and M. Johansson, "Energy efficient D2D communications in dynamic TDD systems," *IEEE Trans. Comm.*, vol. 65, no. 3, pp. 1260–1273, 2017.
[27] I. Mondal, A. Neogi, P. Chaporkar, and A. Karandikar, "Bipartite graph based proportional fair resource allocation for D2D communication," *Proc. IEEE Wireless Comm. and Networking Conf.*, 2017, pp. 1–6.
[28] 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.
[29] Y. Jiang, Q. Liu, F. Zheng, X. Gao, and X. You, "Energy-efficient joint resource allocation and power control for D2D communications," *IEEE Trans. Vehicular Technology*, vol. 65, no. 8, pp. 6119–6127, 2016.
[30] Y. Li, Z. Kaleem, and K. Chang, "Interference-aware resource-sharing scheme for multiple D2D group communications underlaying cellular networks," *Wireless Personal Comm.*, vol. 90, no. 2, pp. 749–768, 2016.
[31] X. Cai, J. Zheng, and Y. Zhang, "A graph-coloring based resource allocation algorithm for D2D communication in cellular networks," *Proc. IEEE Int'l Conf. Comm.*, 2015, pp. 5429–5434.
[32] T. Yang, R. Zhang, X. Cheng, and L. Yang, "Graph coloring based resource sharing (GCRS) scheme for D2D communications underlaying full-duplex cellular networks," *IEEE Trans. Vehicular Technology*, vol. 66, no. 8, pp. 7506–7517, 2017.
[33] F. Hajiaghajani and M. Rasti, "Downlink resource reuse for device-to-device communication underlaying cellular networks using a generalized knapsack framework," *Proc. IEEE Annual Consumer Comm. & Networking Conf.*, 2016, pp. 171–176.
[34] K.Y. Chen, J.C. Kao, S.A. Ciou, and S.H. Lin, "Joint resource block reuse and power control for multi-sharing device-to-device communication," *Proc. IEEE Vehicular Technology Conf.*, 2016, pp. 1–6.
[35] W. Chang, Y.T. Jau, S.L. Su, and Y. Lee, "Gale-Shapley-algorithm based resource allocation scheme for device-to-device communications underlaying downlink cellular networks," *Proc. IEEE Wireless Comm. and Networking Conf.*, 2016, pp. 1–6.
[36] A.S. Hamza, S.S. Khalifa, H.S. Hamza, and K. Elsayed, "A survey on inter-cell interference coordination techniques in OFDMA-based cellular networks," *IEEE Comm. Surveys & Tutorials*, vol. 15, no. 4, pp. 1642–1670, 2013.
[37] R.M.R. Lewis, *A Guide to Graph Colouring: Algorithms and Applications*. Berlin, Germany: Springer, 2015.
[38] R.L. Brooks, "On colouring the nodes of a network," *Mathematical Proc. the Cambridge Philosophical Society*, vol. 37, no. 2, pp. 194–197, 1941.
[39] J. Clausen, "Branch and bound algorithms–principles and examples," Department of Computer Science, University of Copenhagen, Tech. Rep., 1999.
[40] S. Jukna and G. Schnitger, "Yet harder knapsack problems," *Theoretical Computer Science*, vol. 412, no. 45, pp. 6351–6358, 2011.
[41] 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.
[42] 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.