

## Research Article

# Sum Rate Maximization of D2D Communications in Cognitive Radio Network Using Cheating Strategy

Yanjing Sun , Qi Cao , Bowen Wang , and Song Li 

School of Information and Control Engineering, China University of Mining Technology, Xuzhou 221116, China

Correspondence should be addressed to Yanjing Sun; [yjsun@cumt.edu.cn](mailto:yjsun@cumt.edu.cn)

Received 6 November 2017; Accepted 19 February 2018; Published 22 March 2018

Academic Editor: Pierre-Martin Tardif

Copyright © 2018 Yanjing Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper focuses on the cheating algorithm for device-to-device (D2D) pairs that reuse the uplink channels of cellular users. We are concerned about the way how D2D pairs are matched with cellular users (CUs) to maximize their sum rate. In contrast with Munkres' algorithm which gives the optimal matching in terms of the maximum throughput, Gale-Shapley algorithm ensures the stability of the system on the same time and achieves a men-optimal stable matching. In our system, D2D pairs play the role of "men," so that each D2D pair could be matched to the CU that ranks as high as possible in the D2D pair's preference list. It is found by previous studies that, by unilaterally falsifying preference lists in a particular way, some men can get better partners, while no men get worse off. We utilize this theory to exploit the best cheating strategy for D2D pairs. We find out that to acquire such a cheating strategy, we need to seek as many and as large cabals as possible. To this end, we develop a cabal finding algorithm named RHSTLC, and also we prove that it reaches the Pareto optimality. In comparison with other algorithms proposed by related works, the results show that our algorithm can considerably improve the sum rate of D2D pairs.

## 1. Introduction

The ever-increasing demand of wireless communication services motivates system designers and researchers to seek the possibility of reusing frequency resources or enhancing the system capacity. Cognitive radio (CR), as a promising technology in future wireless communication, enables secondary users (usually unlicensed) to reuse the frequency band of primary users with the later ones' grants [1, 2]. The merits of CR technology are that first it can offload the communication overhead of the base station in a centralized system, and second it substantially improves the spectral efficiency (SS) and energy efficiency (ES). Specifically, there are two modes of sharing the spectral resources, that is, overlay and underlay. In the overlay mode, secondary users firstly need to sense spectrum holes, which means the temporarily unused channels, and then commence real transmissions [3]. On the other hand, if the channel gains between the primary and secondary users are sufficiently small so that both sides are able to successfully convey their signals, this situation is also

acceptable in CR networks, which is named underlay mode [4].

With the global explosive growth of services in nowadays wireless communication system pushing the concepts like massive machine type of communication (mMTC), ultra-dense networks (UDN), and so on, the communication environment tends to be saturated in terms of spectrum resources. An important relevant technology supporting CR networks is device-to-device (D2D) communication which can build direct link from end to end [5, 6]. In [7], the systematic capacity of relay assisted CR network is studied, where D2D users can both work in overlay and underlay modes. The paper [8] investigates energy efficiency maximization for the secondary users subject to a QoS requirement for the primary user, using precoder design and energy resource allocation.

To fully utilize the CR networks and D2D communication technology, it usually turns out that we need to solve an allocation problem. For example, assume we have  $N$  D2D pairs who hope to reuse the channel of  $M$  CUs; how to arrange the matching between the two sides for certain purposes (e.g.,

SS or ES maximization) becomes an interesting problem, since different allocation schemes would dramatically differ from others in terms of system performance. For one-to-one optimal matching, Munkres' algorithm, also known as Hungarian algorithm, is able to solve the problem in polynomial time [9, 10]. However, to find the matching by Munkres' algorithm, we need a centralized computation, which could cause some communication overhead, whereas the formation of D2D network is more like a natural process. Hence, we introduce another concept, namely, stability, which indicates the possibility that a system structure might change. For instance, if a secondary user prefers to be allocated to another primary user, and coincidentally this primary user prefers the secondary user to its partner, they might drop their current partners and organize a new channel-reusing relationship. Nevertheless, Gale and Shapley propose the celebrated GS algorithm based on the preference list of both sides [11] to solve the stable matching problem, which has been adopted in a broad studies [8, 12–15].

For a particular system, there could be more than one stable matching, while GS algorithm gives men side the best opportunities to get their partners that rank as high as possible in the men's preference lists. The paper [16] proposed a novel cheating strategy for the men side to further enhance their benefit. The "cheating" needs a group of men to falsify their preference lists exquisitely. The paper [13] firstly applies this cheating strategy in a CR network with secondary users being D2D pairs. However, compared with the existing studies, the main contributions of our work are as follows: first, in order to maximize the sum rate of D2D pairs, we need to search as many and as large cabals as possible, so we developed cabal finding algorithms that outperform existing algorithms greatly; second, for the proposed algorithm RHSTLC, we proved its Pareto optimality, whereas the algorithms in previous studies do not hold this property; third, we numerically demonstrated and evaluated the performance of our proposed algorithms.

The remainder of this paper is organized as follows. Section 2 elaborates the system mode of D2D communication involving cognitive radio networks. In Section 3, we detail the cheating strategy for D2D pairs including how to find a cabal, how to define the corresponding accomplice, and how to falsify their preference lists. The performance of our proposed algorithms and existing algorithms is shown and commented in Section 4. Eventually the study is concluded in Section 5.

## 2. System Model and Problem Formulation

The device-to-device cognitive network system is illustrated in Figure 1, where  $M$  D2D pairs  $\mathcal{D} = \{d_1, \dots, d_m, \dots, d_M\}$ ,  $N$  CUs  $\mathcal{C} = \{c_1, \dots, c_n, \dots, c_N\}$ , and a base station are involved. In this study, we assume  $M = N$ , even though unbalanced cases seem more practical. Nevertheless, we can drop  $N - M$  poor CUs when  $N > M$  and do it likewise when  $M > N$ , which could always lead us to equalized situation. The CUs and the station are working on frequency division duplexing mode, which means each CU occupies an orthogonal frequency band to upload its data. On the

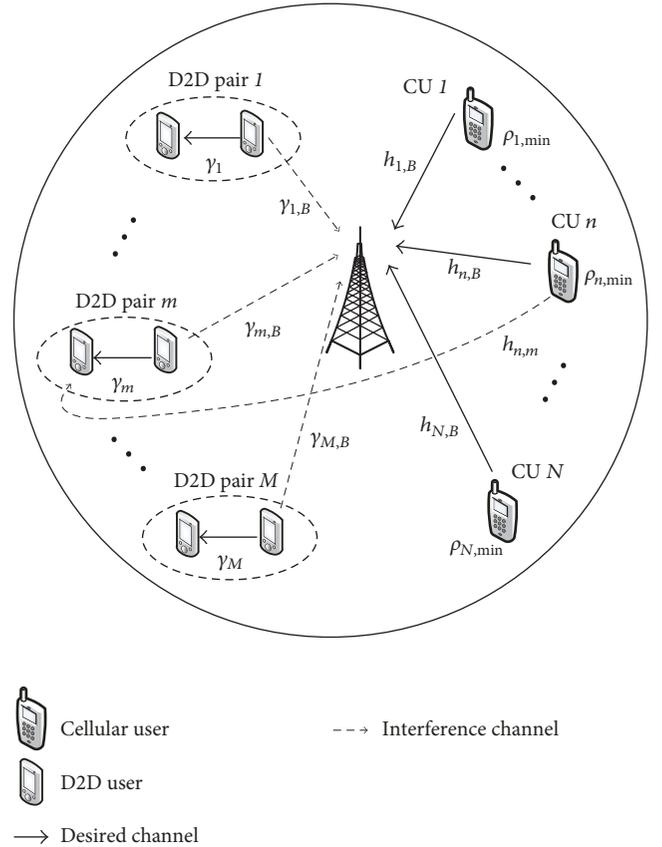


FIGURE 1: Device-to-device cognitive network.

meanwhile, D2D pairs expect to reuse the uplink channels of the CUs for fully utilizing the local spectral resources. The channel gain between the two devices in the D2D pair  $d_m$  is denoted by  $\gamma_m$ , and between the transmit device and base station is  $\gamma_{m,B}$ . On the other hand, the channel gain between the CU  $c_n$  and base station is denoted by  $h_{n,B}$  and between the CU and the D2D pair  $d_m$  is  $h_{n,m}$ . Each CU is constrained by a minimum signal to interference and noise ratio (SINR)  $\rho_{n,\min}$ ; otherwise, the transmission is unsuccessful. The maximum transmit power of CUs is  $P_{t,c}$  and of D2D pairs is  $P_{t,d}$ .

However, in our system, D2D users are not aware of CUs' SINR constraints, forcing themselves to always adopt the maximum transmit power. For an arbitrary D2D pair, it is only allowed to reuse at most one uplink channel, and the same goes to the other side; that is, each CU is only allowed to accept one D2D pair. For instance, if  $c_n$  agrees with the channel-reusing proposal from  $d_m$ , with  $N_0$  denoting the noise power, the SINR of the D2D pair is given by

$$\rho_m^d(n) = \frac{\gamma_m P_{t,d}}{h_{n,m} P_{t,c} + N_0}, \quad (1)$$

and correspondingly the SINR at the BS side from the CU  $c_n$  is written as

$$\rho_n^c(m) = \frac{h_{n,B} P_{t,c}}{\gamma_{m,B} P_{t,d} + N_0}. \quad (2)$$

(1) **Initialize:** Obtain the whole set of D2D pairs  $\mathcal{D}$  and the set of CUs  $\mathcal{C}$ . Obtain the preference lists of all the D2D pairs  $\mathcal{L}^d$  and all the CUs  $\mathcal{L}^c$ . Build a temporary set  $\mathcal{T}\mathcal{D} = \mathcal{D}$ .

(2) **while**  $\mathcal{T}\mathcal{D}$  is not empty **do**

(3) An arbitrary D2D pair  $d_m$  in  $\mathcal{T}\mathcal{D}$  proposes to the CU ranking highest in its preference list, say  $c_n$ .

(4) **if**  $c_n$  has not been proposed yet **then**

(5)  $c_n$  accepts the proposal of  $d_m$ .

(6) **else**

(7) **if**  $d_m \stackrel{c_n}{\leftarrow} d_{m'}$  **then**

(8)  $c_n$  accepts the proposal of  $d_m$ ;  $d_{m'}$  backs to  $\mathcal{T}\mathcal{D}$ .

(9) **else**

(10)  $d_m$  proposes to the next CU in its preference list.

(11) **end if**

(12) **end if**

(13) **end while**

ALGORITHM 1: Gale-Shapley algorithm.

A preference list  $\mathcal{L}$  of a participant (D2D pair or CU) is a set including all of its counterparts, sorted by the predicted SINR from the highest to the lowest. If  $c_n$  is located leftmost in  $d_m$ 's preference list  $\mathcal{L}_m^d$ ,  $c_n$  would be the most wanted counterpart of  $d_m$ , and we denote  $\text{rank}(d_m, c_n) = 1$ . In addition, we use  $c_n \stackrel{d_m}{\leftarrow} c_{n'}$  to indicate that  $d_m$  prefer  $c_n$  to  $c_{n'}$ , and mathematically we can define the expression by

$$c_n \stackrel{d_m}{\leftarrow} c_{n'} \triangleq \text{rank}(d_m, c_n) < \text{rank}(d_m, c_{n'}), \quad n \neq n'. \quad (3)$$

A matching  $\mathcal{M}$  arranges every D2D pair to be matched with a CU; however, it does not mean that the D2D pair can eventually commence a transmission, since the final say belongs to the CUs. In other words, as long as the D2D pairs satisfy the corresponding CU's SINR requirement, they can share the channel; otherwise the CU would solely occupy the channel. The collections of the counterparts of participants in  $S$  are given by  $\mathcal{M}(S)$ . For example,  $\mathcal{M}(d_m) = c_n$  indicates that  $d_m$  is allocated to  $c_n$  in matching  $\mathcal{M}$ , and conversely it must hold  $\mathcal{M}(c_n) = d_m$  for one-to-one matching. Now, we can introduce the concept of blocking pair and stable matching.

*Definition 1.* A matching  $\mathcal{M}$  is unstable, if we can find a pair  $(d_m, c_n)$  such that  $c_n \stackrel{d_m}{\leftarrow} \mathcal{M}(d_m)$  and  $d_m \stackrel{c_n}{\leftarrow} \mathcal{M}(c_n)$ . Such a pair  $(d_m, c_n)$  is called a blocking pair. On the contrary, we say  $\mathcal{M}$  is stable, if it has no blocking pair.

Literally speaking, a stable matching  $\mathcal{M}$  of our network means that when a D2D pair  $d_m$  proposes to any CU ranking higher than its partner  $\mathcal{M}(d_m)$ , the CU will definitely refuse the D2D pair's proposal, so that the topology of the network remains constantly stable. To obtain a stable matching, the Gale-Shapley algorithm is adopted, which is a classical algorithm in graph theory solving the stable marriage problem. We demonstrate the way of using the GS algorithm for our system in Algorithm 1.

In GS algorithm, the proposers gain the most benefits; that is to say every D2D pair is assigned the best possible CU among all the stable matchings. Hence, the stable matching

obtained by GS algorithm is also called the men-optimal stable matching (in the original GS algorithm, men are the proposers). Consequently, the sum rate of D2D pair according to Shannon's theory can be written as

$$r_d(\mathcal{M}) = \sum_{m=1}^M B \log_2 \left( 1 + \rho_m^d(\mathcal{M}(d_m)) \right), \quad (4)$$

where  $B$  is the channel bandwidth. As we can see, the preference lists  $\mathcal{L}^c$ ,  $\mathcal{L}^d$  and the GS algorithm determine the men-optimal matching  $\mathcal{M}$  and hence determine the D2D pair sum rate. So, the problem in this study is how D2D pairs falsify their preference lists to maximize their average data rate: that is,

$$\begin{aligned} \max_{\mathcal{L}^{d*}} \quad & r_d(\mathcal{M}_s) \\ \text{s.t.} \quad & \text{rank}(d_m, \mathcal{M}_s(d_m)) \leq \text{rank}(d_m, \mathcal{M}_0(d_m)), \\ & \forall m, \end{aligned} \quad (5)$$

$$\rho_n^c(\mathcal{M}(c_n)) > R_0, \quad \forall n,$$

where  $\mathcal{M}_0$  is the original men-optimal stable matching without D2D users falsifying preference lists and  $\mathcal{M}_s$  is the new stable matching according to the falsified preference list  $\mathcal{L}^{d*}$ . The constraint indicates that every D2D pair must get a CU at least as good as the one in the original matching.

### 3. Cheating Strategy

*3.1. Coalition Strategy.* The study [16] introduced a novel cheating strategy for proposers, which requires some of them to falsify their preference lists. As a result, some of the proposers are better off, while none of them are worse off. In a more complicated system model, [13] adopted this strategy to achieve better throughput of D2D users. This study actually simplifies the model and focuses on how to improve the cheating strategy. Now, we introduce the cheating method.

First we need to find a cabal  $K = \{k_j \mid j = 1, 2, \dots, J\}$  whose members are all from the D2D pairs: that is,  $k_j \in$

```

(1) Initialize: Obtain a cabal  $K$  and its accomplices  $\mathcal{A}(K)$  in the original matching  $\mathcal{M}_0$ .
(2) for  $\forall a \in \mathcal{A}(K)$  do
(3)   if  $a \notin K$  then
(4)      $a$  falsifies its preference list in the form of  $\{\pi_r(\text{PL}(a) - X), \mathcal{M}_0(a), \pi_r(\text{PR}(a) + X)\}$ ,
       where  $X = \{c \mid c = \mathcal{M}_0(k_{j-1}), a \overset{c}{\leftarrow} k_j\}$ 
(5)   else
(6)      $a = k_s$ , falsifies its preference list in the form of  $\{\pi_r(\text{PL}(a) - X), \mathcal{M}_0(a), \pi_r(\text{PR}(a) + X)\}$ ,
       where  $X = \{c \mid c = \mathcal{M}_0(k_{j-1}), c \overset{k_s}{\leftarrow} \mathcal{M}_0(k_{s-1}), k_s \overset{c}{\leftarrow} k_j\}$ 
(7)   end if
(8) end for

```

ALGORITHM 2: Coalition strategy.

$\mathcal{D}$ . Also, in the cabal, each member has the property of  $\mathcal{M}_0(k_{j-1}) \overset{k_j}{\leftarrow} \mathcal{M}_0(k_j)$ . That is to say, the D2D pair  $k_j$  prefers  $k_{j-1}$ 's cellular partner to its own partner in the matching  $\mathcal{M}_0$  (except that  $k_1$  prefers  $k_j$ 's partner). If we could manage to let all the members in the cabal exchange their partners, they are better off in the new matching. However, we should keep two things in mind. First, the new matching must be also obtained by GS algorithm so that it remains stable. Second, besides the members in the cabal, the remaining D2D pairs should not be worse off or they lose the motivation helping the cabal.

However, there are two sorts of D2D pairs that get in the cabal's way of getting the new better partners. So, we need them to falsify their preference lists, and this set of D2D pairs is called the accomplice of the cabal  $\mathcal{A}(K)$ , specifically

$$\begin{aligned}
\mathcal{A}(K) &= \{a \in \mathcal{D} \mid \text{s.t. } \textcircled{1} \text{ or } \textcircled{2}, \forall k_j \in K\}, \\
\textcircled{1} \quad &a \notin K, \mathcal{M}_0(k_{j-1}) \overset{a}{\leftarrow} \mathcal{M}_0(a), a \overset{\mathcal{M}_0(k_{j-1})}{\leftarrow} k_j, \\
\textcircled{2} \quad &a \in K, a = k_s, j \neq s, \mathcal{M}_0(k_{j-1}) \\
&\overset{k_s}{\leftarrow} \mathcal{M}_0(k_{s-1}), k_s \overset{\mathcal{M}_0(k_{j-1})}{\leftarrow} k_j.
\end{aligned} \tag{6}$$

We assume that the accomplices are willing to help the cabals to get their new better partners under one crucial condition that they are not getting a worse partner in the new matching. To satisfy the accomplices' condition, we find the following Theorem 2 proposed and proved by [16] to be important. Before entering the theorem, we have to define some notations.  $\pi_r(\cdot)$  stands for the random permuting operation that disorders a set.  $\text{PL}(m)$  and  $\text{PR}(m)$  are the CUs located on the left and right side of  $\mathcal{M}(m)$  in  $d_m$ 's preference list, respectively, so that the preference list of  $d_m$  could be expressed as  $\{\text{PL}(m), \mathcal{M}(m), \text{PR}(m)\}$ .

**Theorem 2.** For a subset of D2D pairs  $S \in \mathcal{D}$ , if every member  $m \in S$  falsifies their preference list in the form of  $\{\pi_r(\text{PL}(m) - X), \mathcal{M}(m), \pi_r(\text{PR}(m) + X)\}$ ,  $X \in \text{PL}(m)$ , no D2D pairs get worse cellular partners in the new men-optimal stable matching.

According to this theorem, [13, 16] give the cheating strategy for the *men* side, which is briefed in Algorithm 2.

Submitting the falsified lists, the accomplices make the cabal successfully exchange their partners in the new matching, and hence the average data rate of D2D pairs could be enhanced.

**3.2. Searching Large Cabals.** Generally speaking, the larger the cabal is, the more improvement in data rate the D2D pairs can achieve. Therefore, the maximization problem in the last section results in a classical problem in graph theory, namely, finding the largest circle in a directed graph, which is a NP hard problem. In detail, each vertex is a D2D pair, and each directed edge shows a preference relationship. Hence, [13] suggests a random cabal search (RCS), which starts from an arbitrary D2D pair, and stops as a cycle is found. In order to obtain a relatively larger cabal, [13] conducts the RCS for each D2D pair and adopt the largest one during the process. However, in this study, inspired by the method in [17], we develop a heuristic search for the largest cabal (HSTLC) that is able to possibly find the largest circle.

Before entering the HSTLC algorithm, we have to build the directed graph; hence, we need the following things: (i) the whole set of D2D pairs  $\mathcal{D}$ , (ii) D2D pairs' preference lists  $\mathcal{L}^d$ , and (iii) the original men-optimal stable matching  $\mathcal{M}_0$ . For each parent vertex  $d_m$ , we find all its child vertices by the definition  $\{d_j \in \mathcal{D} \mid \mathcal{M}_0(d_j) \overset{d_i}{\leftarrow} \mathcal{M}_0(d_i)\}$ . We draw a directed edge from  $d_i$  to every vertex in its child vertices set; therefore, the directed map  $\mathbb{G}$  is constructed.

The pseudo code of HSTLC is given in Algorithm 3, which mainly consists of two parts: the main function ensures that each vertex is reached at least once; the `visit_next` function defines a dyeing scheme that contributes to finding a large cabal. In detail, we start from an arbitrary vertex  $d_i$ , dyeing it gray. Then we exploit  $d_i$ 's child vertices, the D2D pairs whose cellular partner is preferred by  $d_i$  to its partner in  $\mathcal{M}_0$ . Next the algorithm chooses a white child vertex to move on, and during the process if any exploited child vertex is found dyed gray, a cabal is then formed. Each time we find a cabal, it is compared with the largest cabal ever obtained, that is,  $K_{\max}$ . If it is larger than  $K_{\max}$ , we update it as the new largest cabal and drop the preceding one; otherwise, we drop it and move on. Eventually, the algorithm ceases only when all vertices have been exploited.

```

HSTLC( $\mathbb{G}$ ):
(1) Initialize: Obtain the directed map  $\mathbb{G}$ ; Build a set  $\text{Color}(m) = \text{white}$ ,  $m = 1, 2, \dots, M$ ; Denote  $\text{Child}(d_m)$ 
    as the collection of all the child vertices of  $d_m$  in  $\mathbb{G}$ . Set the largest cabal  $K_{\max} = \emptyset$ .
(2) while  $\exists d_i \in \mathbb{G} \ \& \ \text{Color}(i) = \text{white}$  do
(3)    $\text{visit\_next}(d_i)$ ;
(4) end while
(5) return  $K_{\max}$ .
 $\text{visit\_next}(d_i)$ :
(1)  $\text{Color}(i) = \text{gray}$ ;
(2) for  $\forall d_j \in \text{Child}(d_i)$  do
(3)   if  $\text{Color}(j) = \text{white}$  then
(4)      $\text{visit\_next}(d_j)$ ;
(5)   else if  $\text{Color}(j) = \text{gray}$  then
(6)     A cabal  $K$  is found;
(7)     if  $\text{size}(K) > \text{size}(K_{\max})$  then
(8)        $K_{\max} = K$ ;
(9)     end if
(10)  else
(11)    continue;
(12)  end if
(13) end for
(14)  $\text{Color}(i) = \text{black}$ 

```

ALGORITHM 3: Heuristic search for the largest cabal (HSTLC).

```

(1) Initialize: Obtain the original directed map  $\mathbb{G}^0$ ; Build a cabal set  $\mathcal{K} = \emptyset$ . Define  $K = \text{HSTLC}(\mathbb{G})$  as the
    cabal find by algorithm HSTLC according to the directed graph  $\mathbb{G}$ . Set the iteration index  $i = 0$ ;
(2) while 1 do
(3)   Find the cabal  $K_i = \text{HSTLC}(\mathbb{G}^i)$ ;
(4)   if  $K_i$  is not empty then
(5)     Add  $K_i$  to the cabal list  $\mathcal{K} = \mathcal{K} \cup K_i$ ;
(6)     Remove vertices in the cabal  $K_i$  from the directed graph  $\mathbb{G}^{i+1} = \mathbb{G}^i - K_i$ ;
(7)   else
(8)     break;
(9)   end if
(10) end while
(11) return  $\mathcal{K}$ .

```

ALGORITHM 4: Recurrent HSTLC (RHSTLC).

Finding a large cabal is sufficiently able to improve the average data-rate of D2D pair; however, after removing the found cabal by HSTLC, the remaining graph may still include some cabals, which means there are possibilities to further improve the system performance. For example, in Figure 2, the first run of HSTLC might find a large cabal  $K_1 = \{d_1, d_2, d_3\}$ ; however, apart from  $K_1$ , we can see that  $\{d_4, d_5\}$  is also a cabal. Based on this scenario, we propose a recurrent HSTLC (see Algorithm 4) that runs HSTLC over and over again until no more cabals can be found.

## 4. Further Analysis

*4.1. Computational Complexity.* To obtain the men-optimal stable matching, a computational complexity by GS algorithm  $\mathcal{O}(M^2)$  is needed. Also, after determining an

appropriate cabal, we need to change the preference lists of D2D pairs by the coalition strategy, which costs a complexity of  $\mathcal{O}(MJ)$ , where  $J$  is the size of the cabal. Since the size of a cabal cannot exceeds the size of  $\mathcal{D}$ , the complexity of the coalition strategy is at most  $\mathcal{O}(M^2)$ . To search a random circle from a given vertex, it costs  $\mathcal{O}(M + V)$  computations, where  $V$  is the amount of edges in the graph  $\mathbb{G}$ . Hence, the ergodic RCS (ERCS) algorithm given by [13], which runs RCS upon every vertex, is conceived as with the complexity order of  $\mathcal{O}(M(M + V))$ . On the other hand, HSTLC reaches out to each vertex once, leaving us with a complexity order of  $\mathcal{O}(M + V)$ . The worst case that RHSTLC can only remove a smallest cabal (2 D2D pairs) from the graph  $\mathcal{G}$  gives us the complexity of  $\mathcal{O}(M(M + V))$ . To sum up, we conclude the above discussion into Table 1. Besides, since the entire process of falsifying preference lists follows the procedure:

TABLE 1: Computational complexity comparison.

Process	Complexity order (flops)
GS	$\mathcal{O}(M^2)$
CS	$\mathcal{O}(MJ) \leq \mathcal{O}(M^2)$
ERCS	$\mathcal{O}(M(M+V))$
HSTLC	$\mathcal{O}(M+V)$
RHSTLC	$\mathcal{O}(M(M+V))$
GS + ERCS + CS	$\mathcal{O}(M(M+V))$
GS + HSTLC + CS	$\mathcal{O}(M^2)$
GS + RHSTLC + CS	$\mathcal{O}(M(M+V))$

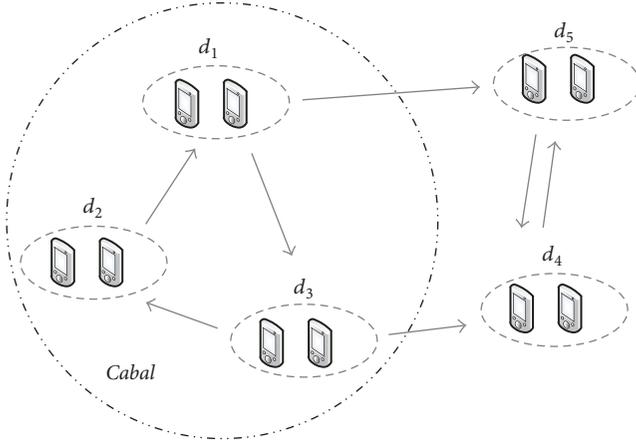


FIGURE 2: Directed graph based on D2D pairs.

GS (to find the original stable matching  $\mathcal{M}_0$ )  $\rightarrow$  cabal finding algorithm  $\rightarrow$  CS, we also compute the complexity order of the overall complexities throughout in the table.

**4.2. Pareto Optimality.** In this section, we solely view the optimality from the D2D pair side, so we have the following theorem.

**Theorem 3.** *If the D2D pairs follow the RHSTLC to find cabals and change their preference lists according to the coalition strategy, then after the GS algorithm, D2D pairs reach Pareto optimality in the final stable matching  $\mathcal{M}_F$ .*

*Proof.* Suppose not, there is a subset of D2D pairs  $\mathcal{S} = \{s_i\} \subseteq \mathcal{D}$  that can improve their own data rates without reducing other D2D pairs' data rates, which means all the D2D pairs in  $\mathcal{S}$  get better off while the D2D pairs outside remain the same. Since  $s_i$  gets better off, it must be allocated to some CU  $c_j$  that is located on the left hand side of  $\mathcal{M}_F(s_i)$  in  $s_i$ 's preference list. Therefore, the D2D pair  $s_j$  who was originally allocated to  $c_j$  must seek for a new CU  $c_k$  that ranks higher than  $\mathcal{M}_F(s_j)$  in  $s_j$ 's preference list. Eventually some D2D pair  $s_g$  must be allocated to  $\mathcal{M}_F(s_i)$ . The D2D pairs  $\{s_i, s_j, \dots, s_g\}$  swap their partners and everyone gets better off, so this group is a cabal. However, as we can see from RHSTLC, the terminating condition is when no more cabals can be found; therefore, a contradiction occurs and hence we complete the proof.  $\square$

TABLE 2: Simulation parameters.

Parameter name	Symbol	Value
Cell radius	$R_{\text{cell}}$	200~800 m
D2D communication range	$R_{\text{D2D}}$	10~30 m
Cellular user transmit power	$P_{t,c}$	24 dbm
D2D user transmit power	$P_{t,d}$	20 dbm
Noise floor	$N_0$	104 dbm
Base station noise figure	$\delta_b$	5 db
D2D user noise figure	$\delta_d$	10 db
Number of D2D pairs	$M$	10~40
Number of CUs	$N$	10~40
SNR constraint of CUs	$R_0$	5 db

However, as can be easily seen in the Pareto optimality of ERc HSTLC, the optimality in terms of the average data rate of D2D pairs cannot be guaranteed.

## 5. Performance Evaluation

In this section, we investigate the performance of the proposed algorithms. The default simulation environment is elaborated in Table 2. All the CUs and D2D pairs are uniformly distributed in the cell, and we assume that the D2D communication distance is uniformly over 10~30 m. Munkres' algorithm (MK) and original GS algorithm are benchmarks, which, respectively, show the upper bound and lower bound of performance of our proposed algorithm. Particularly, the MK curves indicate the optimal matching that can achieve the maximal throughput of the system, regardless of stability. GS curves show the status of D2D pairs without falsifying their preference lists. In addition, the outcomes by ERCS algorithm are also plotted to validate the efficacy of HSTLC and RHSTLC.

In our simulation, the channel gain between CUs (or D2D users) and the base station is given by

$$h_{n,B} = 128 + 37.6 \log_{10}(l_n) - 10 \log_{10}(\lambda) + \chi, \quad (7)$$

where  $l_n$  is the distance from the CU to the base station.  $\lambda$  follows the exponential distribution  $\text{Exp}(1)$ , which models the Rayleigh fading effect.  $\chi$  is normally distributed by  $\mathcal{N}(0, 3)$ , modeling the shadowing effect. This equation also works for  $\gamma_{m,B}$ , that is, the channel gain of D2D users. In addition, we believe the path loss between users should be larger than in (7) due to the complex indoor environment, which leads us to

$$h_{n,m} = 148 + 40 \log_{10}(l_{n,m}) - 10 \log_{10}(\lambda) + \chi, \quad (8)$$

where  $l_{n,m}$  is the distance from the CU  $c_n$  to the D2D pair  $d_m$  and the same equation goes to  $\gamma_m$ , that is, the channel gain between the two devices in the D2D pair  $d_m$ .

In Figure 3, we present the relationship between the systematic sum rate of all users including CUs as well as D2D pairs and the number of users. We recall that in our system the number of CUs equalizes the number of D2D pairs,  $M = N$ . As can be seen clearly in the figure, RHSTLC locates above

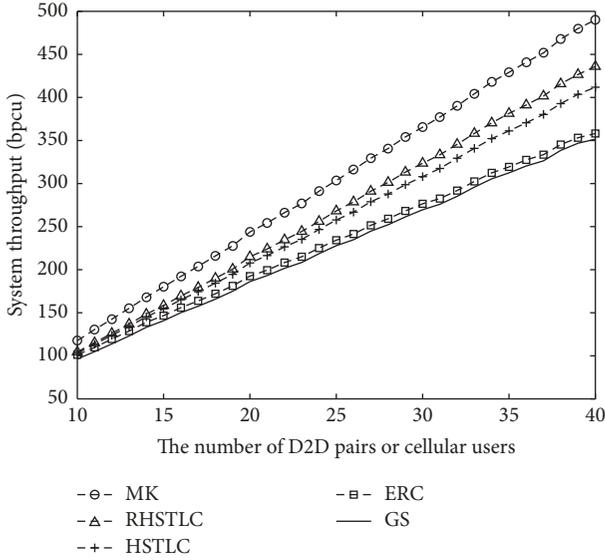


FIGURE 3: System throughput comparison among different algorithms versus user size.

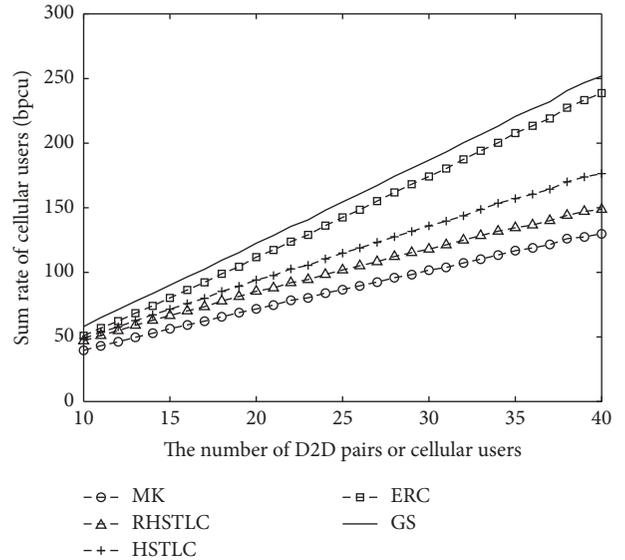


FIGURE 5: Comparison of cellular users' sum rate among different algorithms versus user size.

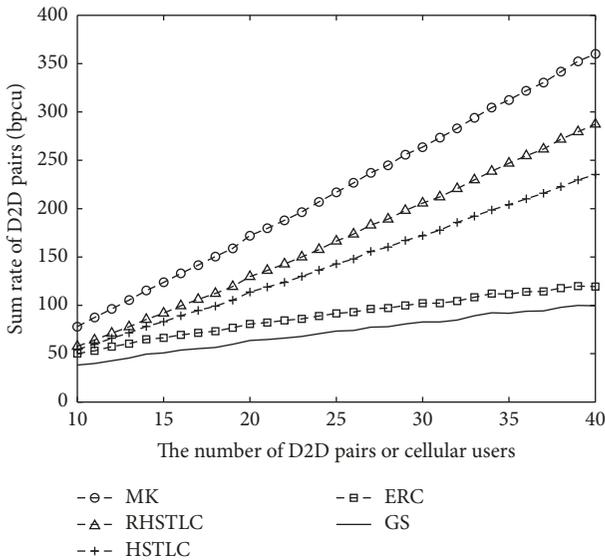


FIGURE 4: Comparison of D2D pairs' sum rate among different algorithms versus user size.

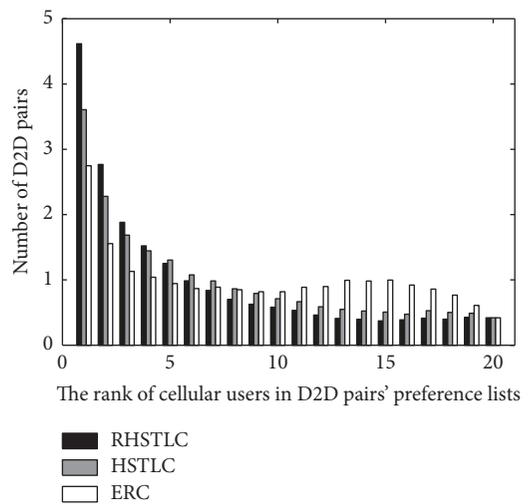


FIGURE 6: Histogram of the rank of cellular users in D2D pairs' preference lists.

all other stable matchings, where RHSTLC achieves 20.03% higher system throughput than GS when  $N = 30$ . Though being a little shy to RHSTLC, HSTLC still outperforms ERCs and original GS algorithm. According to this result, we can understand that our proposed algorithms are conducive to the enhancement of system throughput, even though they are designed to solely improve the sum rate of D2D pairs.

Figure 4 illustrates the sum rate of D2D pairs versus the number of D2D pairs or CUs. Due to more D2D pairs participating into cabals, the effect is strong and clear: that is, RHSTLC outperforms other stable matching algorithms. As we can see that the improvement in sum rate of D2D pairs is larger than the improvement in throughput of the whole

system, which means that, on the other side, the sum rate of CUs is reduced. This phenomenon is understandable, since D2D pairs and CUs are two competitive sides in terms of communication resources. In comparison, we plot the sum rate of CUs in Figure 5. It can be clearly seen that the curves are upside down compared with Figure 4, which actually verified what we mentioned. With D2D pairs falsifying their preference lists and extracting more communication resources, the performance of cellular degrades.

In Figure 6, we show averagely how many D2D pairs get their favorite CU, that is, the one that ranks the first in their preference lists, and how many get their second favorite one and so on. We set the cell range  $r = 300$  m and user number  $N = 20$  in the simulation, and, respectively, 23.08%, 18.03%, and 13.75% of the total D2D pairs get the channel reuse

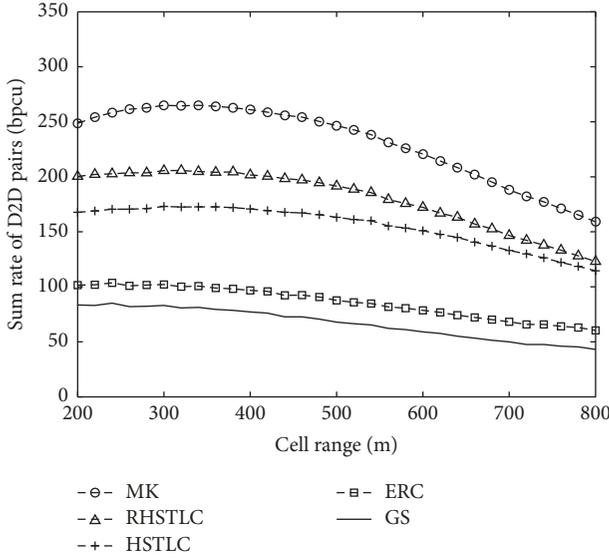


FIGURE 7: Device-to-device cognitive network.

granted from their favorite CUs in RHSTLC, HTLC, and ERCS. Generally speaking, D2D pairs in RHSTLC are most likely to get partners that rank high in their preference list. From the simulation data, we calculate the expected values of the rank of CUs in RHSTLC, HTLC, and ERCS which are 6.252, 7.123, and 8.986 correspondingly.

We demonstrate the sum rate of D2D pairs versus cell range in Figure 7, where the simulation result is averaged over 2000 system realizations with CUs  $N = 30$ . Still the RHSTLC is higher than other algorithms, while a little shy to the MK algorithm which is unstable. An interesting phenomenon is found that the sum rate of D2D pairs increases at first but decreases later on instead as the range continues to rise. It can be explained by the fact that when the cell range is short, as the range increases, the interference caused by CUs decreases so that the D2D pairs gain more data rate. However, when the cell range is beyond some point, more and more CUs are likely to refuse D2D pairs' proposal due their own SNR constraints; therefore, less D2D pairs can eventually reuse the channel and the sum rate starts to fall. Also because of the same reason, when the cell range is large, the optional CUs for each D2D pair are shrunk, which means the edges in the in the map  $\mathbb{G}$  are shrunk as well. Even if we recurrently search for cabals, few more of them can be found, so the RHSTLC and HSTLC curves tend to converge together as the cell range becomes larger. We can also observe the phenomenon from Figure 8 which shows the throughput versus cell ranges. The overall system performance manifests a monotonous decreasing property with the cell range.

## 6. Conclusion

We considered a cognitive radio system, where D2D pairs could reuse the uplink channels of CUs. To build a stable system, we adopted GS algorithm, whereas D2D pairs are able to falsify their preference lists to gain more data rate. With

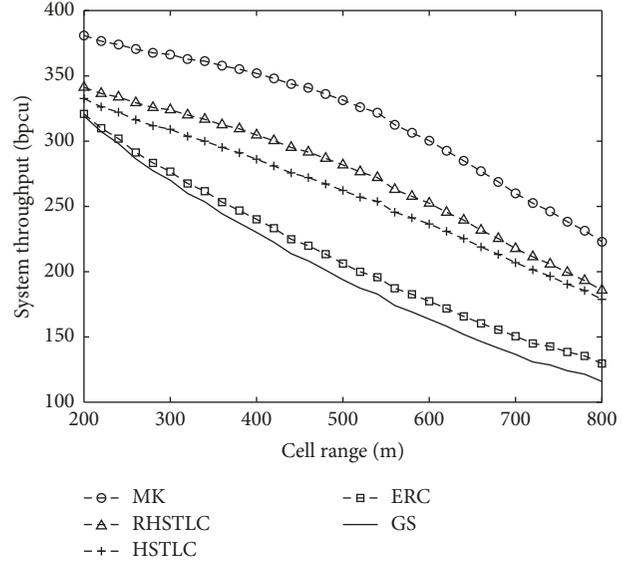


FIGURE 8: System throughput comparison among different algorithms versus cell range.

the fact that the more D2D pairs join the cabal, the more they gain, we developed algorithms to search cabal as large as possible. We computed and compared the computational complexity of all elaborated cabal finding algorithms and proved the Pareto optimality of RHSTLC. In the simulation results, with respect to all stable matchings, the one obtained by RHSTLC shows superiority over others. Also the D2D pairs following the RHSTLC are more likely to be matched to the CUs who rank as high as possible in D2D pairs' preference lists.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grants nos. 61771417, 51504214, and 51504255).

## References

- [1] E. Z. Tragos, S. Zeadally, A. G. Fragkiadakis, and V. A. Siris, "Spectrum assignment in cognitive radio networks: a comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1108–1135, 2013.
- [2] X. Hong, J. Wang, C.-X. Wang, and J. Shi, "Cognitive radio in 5G: a perspective on energy-spectral efficiency trade-off," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 46–53, 2014.
- [3] S. Peng, W. Zheng, R. Gao, and K. Lei, "Fast cooperative energy detection under accuracy constraints in cognitive radio networks," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 3984529, 8 pages, 2017.
- [4] L. Wang, H. Tang, H. Wu, and G. L. Stüber, "Resource Allocation for D2D Communications Underlay in Rayleigh Fading

- Channels,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1159–1170, 2017.
- [5] F. Jiang, Y. Liu, C. Li, and C. Sun, “Energy-Efficient Multicast Transmission for Underlay Device-to-Device Communications: A Social-Aware Perspective,” *Mobile Information Systems*, vol. 2017, Article ID 7148564, 17 pages, 2017.
- [6] J. Chen, C. Liu, H. Li, X. Li, and S. Li, “A Categorized Resource Sharing Mechanism for Device-to-Device Communications in Cellular Networks,” *Mobile Information Systems*, vol. 2016, Article ID 5894752, 10 pages, 2016.
- [7] Y. Yang, Y. Zhang, L. Dai, J. Li, S. Mumtaz, and J. Rodriguez, “Transmission Capacity Analysis of Relay-Assisted Device-to-Device Overlay/Underlay Communication,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 380–389, 2017.
- [8] Z. Zhou, K. Ota, M. Dong, and C. Xu, “Energy-efficient matching for resource allocation in D2D enabled cellular networks,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5256–5268, 2017.
- [9] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the Society For Industrial & Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
- [10] F. Bourgeois and J.-C. Lassalle, “An extension of the Munkres algorithm for the assignment problem to rectangular matrices,” *Communications of the ACM*, vol. 14, pp. 802–804, 1971.
- [11] D. Gale and L. S. Shapley, “College Admissions and the Stability of Marriage,” *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.
- [12] Y. Gu, Y. Zhang, M. Pan, and Z. Han, “Cheating in matching of device to device pairs in cellular networks,” in *Proceedings of the 2014 IEEE Global Communications Conference, GLOBECOM 2014*, pp. 4910–4915, USA, December 2014.
- [13] Y. Gu, Y. Zhang, M. Pan, and Z. Han, “Matching and cheating in device to device communications underlying cellular networks,” *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 2156–2166, 2015.
- [14] Z. Zhou, G. Ma, M. Dong, K. Ota, C. Xu, and Y. Jia, “Iterative Energy-Efficient Stable Matching Approach for Context-Aware Resource Allocation in D2D Communications,” *IEEE Access*, vol. 4, pp. 6181–6196, 2016.
- [15] Z. Zhou, C. Gao, C. Xu, T. Chen, D. Zhang, and S. Mumtaz, “Energy-Efficient Stable Matching for Resource Allocation in Energy Harvesting-Based Device-to-Device Communications,” *IEEE Access*, vol. 5, pp. 15184–15196, 2017.
- [16] C.-C. Huang, “Cheating by men in the Gale-Shapley stable matching algorithm,” in *Algorithms—ESA 2006*, vol. 6 of *Lecture Notes in Comput. Sci.*, pp. 418–431, Springer, Berlin, Germany, 2006.
- [17] P. Kumar and N. Gupta, “A heuristic algorithm for longest simple cycle problem,” in *Proceedings of the International Conference on Wireless Networks (ICWN)*, *The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*, 2014.

