Learning Automaton based Distributed Caching for Mobile Social Networks

Chuan Ma, Zihuai Lin, Loris Marini, Jun Li, and Branka Vucetic
School of Electrical and Information Engineering, University of Sydney, 2006 Sydney, Australia
Email: {chuan.ma, zihuai.lin, loris.marini, jun.li1, branka.vucetic}@sydney.edu.au

***Abstract*—In this paper, a novel distributed caching strategy in mobile social networks based on device-to-device communications is proposed. The proposed approach combines the characters of social networks to handle some practical issues, e.g., the selfishness of users. In order to maximize the throughput of the whole system, a fast convergence learning automaton, called the discrete generalized pursuit algorithm is utilized. Incorporating with social characters, the algorithm not only optimizes the content placement problems in caching theory, but also satisfies the physical and social constraints appropriately. Simulation results show that, compared with other investigated caching strategies, the proposed algorithm has higher convergence speed and at the same time, it can reduce the transmission delay and improve the system throughput. Moreover, the proposed algorithm can get a better performance in higher density district.**

I. Introduction

The proliferation of wearable and hand-held smart devices is posing new challenges on existing wireless networks. High definition recording capabilities together with the widespread of social media are among key factors which are responsible for reducing uplink-downlink asymmetries. Moreover, with the bulk of wireless traffic dominated by video streaming, the network efficiency of centralized intra-cellular topologies will fall fast below the satisfactory levels [1]. Owing to the availability of high capacity and low-cost storage devices, data caching has been proposed to reduce the latency and increase the efficiency of heterogeneous networks.

In particular, there has been a demand for developing intelligent caching strategies in mobile cellular networks to enable data access through nearby content providers. Both in 3G mobile and 4G LTE-Advanced networks, caching have both been demonstrated its ability in reducing mobile traffic by one third to two thirds [2]. However, how to cache is always a difficult problem for researchers to solve. In [3], the authors use a distributed cache replacement strategy based on the Q-learning to replace the cache service data in the base station. The Q-learning is a well used reinforcement learning technique which can find an optimal action-selection policy from the Markov decision process, but the convergence time is much longer. In [4], the authors investigate a learning automaton denoted by the discrete generalized pursuit algorithm (DGPA) which helps caching in the small Base Stations.

However, in device-to-device (D2D) underlaying cellular networks, the caching scheme faces a nontrivial and practical problem, which is how to encourage users sacrifice their own storages to serve others. In order to solve this kind of problem, a cooperative game in [5] is proposed to set up rules for every player, which will bring relevant rewards or punishments. Nevertheless, since not everyone would like to join the game or follow the rules, the statement in [5] may be impractical in some sense. Considering this, combined with the notion of social relation among users, there is a way to use the social relationship to promote caching.

Since 1930s, social networks have been discovered and studied which have an obvious trend to become increasingly popular [6]. In social networks, there are four characters, i.e., tie, community, centrality and bridge [7]. Tie means two users have some relationship to be connected, such as real relationship (kinship, colleague) or virtual relation (two users are friends in the Facebook). Community can be formed according to the social relations, such as common interests. Centrality is a quantification of the relative structural importance of one user and bridge is the only way that connects different communities. In order to promote D2D communications, we intend to utilize a learning automaton combined with the characters of social networks to improve the caching strategy.

In this paper, we propose a fast convergence learning automaton (LA), whose environment feedbacks are determined by the social characters. This LA helps users to cache particular files, thus in return can improve the accurate rate of storing required files. Firstly, different communities are divided according to same locations, interests or background, which can be also called clusters. Then, important users (IUs) in each community are selected as *a prior*, which are used to cache files and provide downloading service for others via D2D links. Moreover, the fast-convergence discrete generalized pursuit algorithm with social characters (DGPA-SC) is used for each IU to complete the caching process. Simulation results show that the proposed learning algorithm convergence iterations reduce significantly compared with the Q-learning [3] (e.g., 123 iterations vs. 2000 iterations). The total transmission time is also shorter than other investigated caching strategies.

The rest of this paper is organized as follows. In Section II, the system model is presented and the methods to divide different communities are described. The IUs are chosen and then the transmission delay which need to optimize is evaluated. In Section III, the process of the DGPA-SC is presented and we propose a corresponding algorithms to solve the caching problem. Simulation results are presented in

978-1-4673-9814-5/16/$31.00 ©2016 IEEE

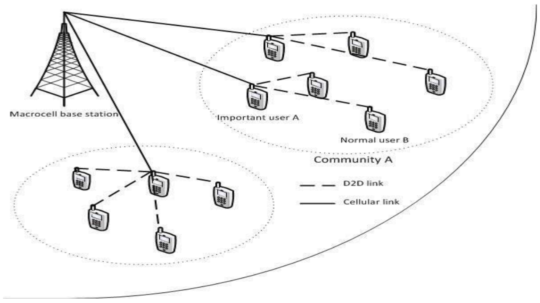


Fig. 1. Illustrative example of the considered network

Section IV. Finally, conclusions are drawn in Section V.

II. System Model and Problem Formulation

A. System Overview

We focus on the scenario of one cell involving all users (cellular and D2D users). The network is artificially divided into different communities according to same location, interests or background. In one community, each user can choose to communicate with the base station via the cellular link or to communicate with other users directly via the D2D links. The users in this situation also have two constraints: the social constraint (they have stable social relations with each other so that they are glad to communicate) and the physical constraint (the actual distance between a transmitter and a receiver is not larger than the maximum D2D transmission distance). Fig. 1 gives a simple example of the considered network. There are two communities illustrated in the network which are formed by social and physical constraints. In Community A, one important user A receives date from the macro cell base station via the cellular link, then transmits the data to a normal user B via the D2D link.

B. System model

In one community, we consider the downlink transmission of a single carrier macro cell network. There are $N = \{1, 2, ..., N\}$ users overlaid on this macro cell network. Among the users, $M_c = \{1, 2, ..., M_c\}$ cellular users are chosen by the base station to store files in their caches and $M_d = \{1, 2, ..., M_d\}$ D2D users tend to download required files from the IUs via the D2D links. The cellular users are also defined as the IUs. To take the social and physical constraints into account, we introduce one graph $G_c^{s,p} \triangleq (M_d, \xi_c^{s,p})$, where $\xi_c^{s,p} \triangleq \{(a, b) : e_{ab}^{s,p} = 1, \forall a \in M_c, b \in M_d\}$. $e_{ab}^{s,p} = 1$ if and only if user a in M_c has stable and positive social relation with user b in M_d, and their physical distance allows them to form a D2D link.

Then we give a method to choose the IUs. In the process of selecting the IUs, the betweenness centrality **B** and available storage capacity **C** are chosen to determine the social importance **I**, such as

$$\mathbf{I} = \mu{\cdot}\mathbf{B} + \nu{\cdot}\mathbf{C}, \tag{1}$$

where μ and ν are tunable parameters which satisfy $\mu + \nu = 1$.

Betweenness centrality **B** measures the extent of a node which lies on the paths linking other nodes, and it can be regarded as a measure that a node control information flowing between others. Thus, betweenness centrality represents the importance of one node. According to [8], the betweenness centrality of node i can be calculated as

$$\mathbf{B}(p_i) = \sum_{j=1}^{N} \sum_{j<k} \frac{g_{jk}(p_i)}{g_{jk}}, \tag{2}$$

where g_{jk} is the number of geodesic (shortest) paths linking node j and node k, and $g_{jk}(p_i)$ is the number of those geodesic paths that include or pass node i.

After detecting every user equipment's available storage capacity **C**, the base station choose a number of important users M_c according to their social importance **I**.

C. Problem Analysis

The transmission channel is modeled as the Rayleigh fading channel [9]. A certain MAC layer protocol, e.g., Time-Division Multiple Access (TDMA), is assumed to be employed by the base station and we assume one IU can only communicate with one normal user simultaneously. Let ζ_{cd} represent the distance between an IU and a normal user and let ζ_{bd} represent between the base station with a normal user, respectively. We further denote the path loss exponents of these two links (the D2D and cellular links) by λ_{cd} and λ_{bd}, respectively. The additive white Gaussian noise is denoted by N_0. According to [9], the transmission rates of D2D and cellular links can be respectively expressed as

$$r_{c_i,d_j} = BW \cdot \log_2(1 + \frac{P_{c_i,d_j}|h_{cd}|^2}{\sum\limits_{c_{i'} \in M_c, d_{j'} \in M_d} P_{c_{i'},d_{j'}}|h_{cd}|^2 + P_{b,d_{j'}}|h_{bd}|^2 + N_0}), \tag{3}$$

and

$$r_{b,d_j} = BW \cdot \log_2(1 + \frac{P_{b,d_j}|h_{bd}|^2}{\sum\limits_{c_{i'} \in M_c, d_{j'} \in M_d} P_{c_{i'},d_{j'}}|h_{cd}|^2 + N_0}), \tag{4}$$

where BW denotes the bandwidth which is allocated in this community. $|h_{cd}|^2 = (\zeta_{cd}^{-\lambda_{cd}}) \cdot |h_0|^2$ and $|h_{bd}|^2 = (\zeta_{bd}^{-\lambda_{bd}}) \cdot |h_0|^2$ denote the channel gain of these two links, and $|h_0|^2$ is the second-order statistics of the Rayleigh fading. P_{c_i,d_j} and P_{bd_j} represent the transmission power of the important user c_i and the base station b, respectively. The expressions (3) and (4) are generic, including the situation with inactive D2D pairs. In fact, if a D2D pair $(c_{i'}, d_{j'})$ is inactive, we simply set the corresponding transmission power $P_{c_{i'},d_{j'}}$ to zero. From (3) we can see that a D2D link suffers interferences from other D2D links and the occurrent cellular link as they share the same radio resources.

The different communication demands for the jth receiver are denoted by Y_j. Combining all the transmission rates together, we can get the total transmission time according to different Y_j :

$$T_{trans} = \sum_{j=1}^{N_{bd}} \frac{Y_j}{r_{b,d_j}} + \sum_{j=1}^{N_{cd}} \frac{Y_j}{r_{c,d_j}}. \tag{5}$$

Here, N_{bd} denotes the number of the cellular links and N_{cd} denotes the number of available D2D links. The expression (6) is generic, including the situation with simultaneous active D2D pairs. N_{bd} and N_{cd} satisfy the condition:

$$N_{bd} + 2 \cdot N_{cd} \leq N. \tag{6}$$

After determining the number of important users, the base station should distribute files into the IUs in off-peak hours. Let P_{b,c_j} denote the transmission power of the base station when the base station transmits information to the IUs, so the caching rate is

$$r_{b,c_i} = B \cdot \log_2(1 + \frac{P_{b,c_j}|h_{bc}|^2}{N_0}), \tag{7}$$

where $|h_{bc}|^2 = (\zeta_{bc}^{-\lambda_{bc}}) \cdot |h_0|^2$, and ζ_{bc} and λ_{bc} denote the distance and the path loss exponent of this link, respectively. N_0 represents the additive white Gaussian noise. So the final transmission delay including caching stage and transmission stage can be expressed by:

$$T_{total} = T_{trans} + \sum_{i=1}^{M_c} \frac{Y_i}{r_{b,c_i}}, \tag{8}$$

Our purpose is to maximize the system throughput, which is equivalent to minimize the total transmission delay. As D2D links can communicate simultaneously, the delay is reduced. Thus, increasing the number of available D2D links as well as improving caching accuracy are critical in our system.

III. DGPA-SC for Caching

Caching can be performed by the IUs. This enables other users to download files directly from the IUs rather than to ask the base station for help. According to [4], the DGPA is a good tool which help IUs to learn caching strategies. We combine this tool with the Social Characters, denoted by the DGPA-SC, and apply it in the IUs. It will increase the probability of caching required files. As a result, the number of available D2D links will increase.

A. Discrete Generalized Pursuit Algorithm

The goal of learning automaton (LA) is to determine an optimal action out of a set of allowable actions $\mathbf{A} = [\alpha_1, \alpha_2, ..., \alpha_s]$. The LA has a probability vector $\mathbf{P(t)} = [p_1(t), p_2(t), ..., p_s(t)]$, where $p_i(t)$ is the probability that the automaton will select the action α_i at time t. $\sum_1^s p_i(t) = 1$, and the probabilities satisfy a reward estimation $\mathbf{d(t)} = [d_1(t), d_2(t), ..., d_s(t)]$ [10]. In our case, the action is the process that one learner (one IU) chooses one specific file from the file library to cache. This action is taken following the probability vector and will get a positive reward from the social network environment if the file is the optimal one.

A type of variable structure stochastic automaton is considered and we specifically focus on the DGPA, which is proven to be the fastest and most accurate algorithm for LA [10]. This algorithm generalizes the concepts of the pursuit algorithm by "pursuing" all the actions that have higher reward estimates than the current chosen actions.

At each iteration, by applying the algorithm, the number of actions which have higher reward estimates than the current chosen one are counted. Let $K(t)$ denote this number. By using the DGPA, the probability of all the actions with higher estimates will increase with the amount of $\Delta/K(t)$, and the probability of all the other actions except the chosen one will decrease with the amount of $\Delta/(r - K(t))$. $\Delta = 1/rH$ is a resolution step and H is a resolution parameter.

The algorithm recursively updates the action probability vector $\mathbf{P}(t)$ by the following equation:

$$\mathbf{P}(t+1) = \mathbf{P}(t) + \frac{\Delta}{K(t)} \cdot \mathbf{e}(t) - \frac{\Delta}{r - K(t)} \cdot [\mathbf{u} - \mathbf{e}(t)], \tag{9}$$

where $\mathbf{e}$ is a unit vector and can be expressed as:

$$\begin{aligned} e_i(t) &= \begin{cases} 1, & \text{if } \mathbf{d}_i(t) = \max\{\mathbf{d}_j(t)\}; \\ 0, & \text{otherwise.} \end{cases} \\ e_j(t) &= \begin{cases} 0, & \text{if } \mathbf{d}_j(t) \leq \mathbf{d}_i(t); \\ 1, & \text{if } \mathbf{d}_j(t) > \mathbf{d}_i(t). \end{cases} \end{aligned} \tag{10}$$

According to (10), the probability of different action is displayed as following:

$$\begin{cases} p_j(t+1) = \min\{p_j(t) + \frac{\Delta}{K(t)}, 1\}, \text{such that } \mathbf{d}_j(t) > \mathbf{d}_i(t); \\ p_j(t+1) = \max\{p_j(t) - \frac{\Delta}{r-K(t)}, 0\}, \text{such that } \mathbf{d}_j(t) < \mathbf{d}_i(t); \\ p_i(t+1) = 1 - \sum_{j \neq i} p_j(t+1). \end{cases} \tag{11}$$

B. Environment and Feedback with Social Characters

In our model, we assume the transmission time among different D2D users is almost equal. This is because users share the same channel bandwidth, and the difference of physical distances among them is small within one community. Therefore, other differences among users need to be considered, e.g., the social influence.

As social networks display a high degree of transitivity, there is a heightened probability of two users being acquainted if they have one or more other neighbors in common. The degree of similarity among users also has an important effect in terms of information dissemination. For example, when the degree of similarity between two users is lower, more time would be expected to take while transmitting the same length of information. As a result, we use the degree of similarity to determine the closeness of two D2D users. A new method to calculate the environment feedback according to different similarities is given below.

We first give the updating equations of reward estimation $\mathbf{d}(t)$ for the chosen action:

$$\begin{cases} Z_i(t+1) = Z_i(t) + 1; \\ W_i(t+1) = W_i(t) + \beta(t); \\ \hat{d}_i(t+1) = \frac{W_i(t+1)}{Z_i(t+1)}, \end{cases} \tag{12}$$

where $Z_i(t)$ and $W_i(t)$ respectively represent the number of times the action i has been chosen and has been rewarded. $\beta(t) \in [0,1]$ is a binary factor reflecting the positive or negative feedback. If the feedback is positive ($\beta = 1$), this action is rewarded.

One IU is pursued to cache one specific file in the library F at each learning iteration. The cached file would produce different social influences according to different similarities among users. If one normal user can not find its required file from its most similar neighbor, it will choose other less familiar IUs or the base station to download.

The degree of similarity can be measured by the ratio of common neighbors between individuals. According to [11], if one IU c, and one normal user d is connected, let $V(c)$, $V(d)$ denote the set of neighbors of user c and d, respectively. Let z be the common neighbors of user c and d and let $V(z)$ denote the number of node $z's$ neighbor. $K(z)$ is denoted by the degree of z and $K(z) = |V(z)|$. Since every IUs can only communicate with one normal user at one time, so we should calculate every normal user and their corresponding IUs' similarity by,

$$q_{d,c} = \sum_{z \in V(d) \cap V(c)} \frac{1}{K(z)}. \tag{13}$$

If d and c have no common neighbors , then $q_{d,c} = 0$. Next we normalize the similarity $\mathbf{S}_{d,c}$:

$$S_{d,c} = \frac{q_{d,c}}{\sum\limits_{c \in M_c} q_{d,c}}. \tag{14}$$

The matrix $\mathbf{S}$ represents the unified similarity between one normal user d and its important neighbors. We assume that a normal user would like to choose the IU with the highest similarity with it if this IU has the required file f. This action is determined to get a positive reward Ψ_P. Otherwise, it would get a negative reward Ψ_N. The reward functions are given by,

$$\begin{cases} \Psi_P = l_f \cdot S_{d,c} \\ \Psi_N = -l_f \cdot S_{d,c'}, \end{cases} \tag{15}$$

where l_f is the probability that file f is required. We use $Zip-f$ distribution, which is commonly used in the file popularity, to model this probability l_f.

The negative reward Ψ_N means that this IU is chosen by other normal users while their similarity is not the highest. So we rewrite the reward equation:

$$\begin{cases} \Psi_P = l_f \cdot S_{c,d} \\ \Psi_N = -l_f \cdot S_{c,d'}. \end{cases} \tag{16}$$

Thus, for one IU c, the aggregated reward function is expressed as :

$$R_c^f = \sum \Psi_P + \sum \Psi_N. \tag{17}$$

If $R_c^f > 0$, then $\beta = 1$ and this action can get a positive reward. In this case the estimation vector $\mathbf{d}(t)$ can be updated.

The proposed algorithm is guaranteed to converge [10]. Because of the paper limit, we only show the proof in two steps. First, the DGPA-SC possesses the moderation property. This is because the magnitude of decrement of any action probability is bounded by the value $1/rH$ at any iteration of the algorithm. Second, the DGPA-SC possesses the monotone property. Due to these properties, $p(t)$ is a submartingale. Thus, according to the submartingale convergence theorem, $p(t)$ will converge to 1 with probability one. Therefore, the algorithm is proven to converge.

C. Game Initialization

The vector probability of each action **P(t)** is initially set to be equal. The vector of estimates for the reward probabilities $\mathbf{d}(t)$ is initially set to be zero for any learner in this game, which can be referred to game initialization. Learners keep on selecting files randomly until each file is selected over a minimum number of times. At the same time **P(t)** and $\mathbf{d}(t)$ are updated according to (9) and (12), respectively. It will guarantee the impartiality of the game. If $\mathbf{d}(t) \neq 0$ for all actions, the game initialization is successful.

D. Caching In Advance

After an IU choosing one file from the library, which would be the most beneficial to its neighbors, the base station starts to offer this service to the IU during the off-peak hours. After that, the IU can repeat the learning process to store another file until its available storage is full. Then the learning process comes to another IU. After finishing caching, the base station can get the total caching time T_{cache}.

IV. Performance Evaluation

A. Simulation Scenario

A wireless network consisting of one macro-cell base station is considered. In this section, we will present the MATLAB simulation results of our proposed scheme. The base station is designed as an omniscience, with 1000 meters' coverage [5]. In this range, there are a number of communities which are formed and divided by the social connections and physical distances. We assume that users are movable in one community. Each community consists of the maximum of 800 randomly distributed user equipments. The community radius is about 200 meters. The D2D communication distance constraint is no longer than 25 meters. Both the macro-cell base station and user equipments share the same frequency bandwidth where the non-orthogonal downlink transmission is assumed. The simulation is carried out in one isolate community. Users can only communicate with others in the same community by pre-establishing and pre-setting social characters by some incentives [7]. The tunable variables μ and ν are set to 0.5. The system parameters are listed in Table I.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Number of Macro-cell BS	1
Coverage radius os BS	1000 m
Transmission power of BS	46 dBm
Coverage radius of one Community	200 m
Number of User Equipments	800
Distance Constraint of D2D	25 m
Transmission power of D2D	24 dBm
System Bandwidth	5 MHZ
Noise figure	7 dB
D2D pathloss	$(d)^{-\lambda}, \lambda = 3$
Requirement demands	100MB

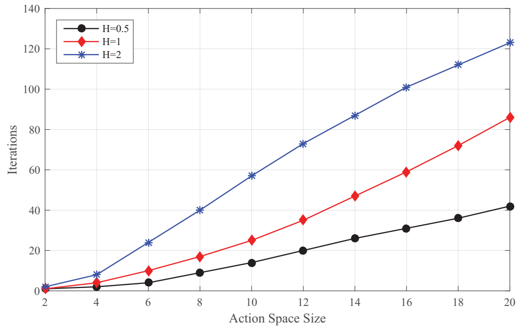


Fig. 2. The average iterations for convergency of different resolution parameter H

B. Convergency and Accuracy

We first test and verify the convergence of the DGPA-CS in a Normal environment, in which the probability of requested files follows a discrete additive gaussian-like random distribution with the expectation $E[\hat{\alpha}(t)] = \alpha_0$ and variance $\delta = 1$. The social similarities among users are pre-determined according to the power law distribution. α_0 is the optimal action belonging to the available action set **A**. Although it is impractical for many applications, a Normal environment can provide a lower bound of iterations for the performance of the DGPA-CS, which is a strong evidence to support the convergence. As shown in Fig. 2, to guarantee the universality, the average number of iterations for different space action size is more than 10 times. As the space action size grows, the average iterations increase. For example, when the resolution parameter $H = 1$, the number of iterations of 20 actions is about 1.8 times larger than the number of 10 actions. As to a smaller resolution parameter H, learning will take less time to converge. For example, when $H = 0.5$, the number of average iterations is almost half (48.9%) of the number when $H = 1$ (42 vs. 86). Compared with the results in [3], in which simulations are run for 32 actions using the Q-learning algorithm and over 2,000 iterations are taken to converge, our algorithm only requires 123 iterations on the condition of $H = 2$. It clearly show that our DGPA-CS algorithm is much faster than the Q-learning.

Fig. 3 depicts the different average accuracy rates of different resolution parameter H. The accuracy rate represents

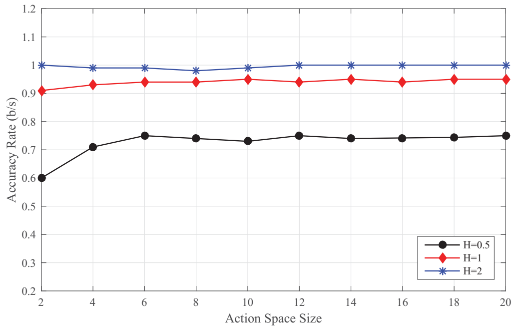


Fig. 3. The average accuracy rate of different resolution parameter H

that the learning process results in taking the optimal action α_0 without error. We run simulations over 50 times for each action size to calculate the average accurate rate. It is shown that though the action space size grows, the average accuracy rate remains no change. For large H, the accuracy rate is high but its convergence speed is slow. We found that when $H = 2$, its accuracy rate is almost 100%, which would produce a great advantage when applying this algorithm into caching strategy.

C. Transmission Rate and Time

The transmission rate and time are simulated to present the advantage of our algorithm. Fig. 4 represents the transmission rate for both D2D links and cellular links. The two rates both demonstrate descending trends with different decreasing speeds. In addition, when the number of IUs (N_{IU}) increases, the decrement of the rate of the D2D links is more obvious than the cellular links. When N_{IU} is smaller, more users will ask the base station for help and the rate of the D2D links is much larger than the cellular links in this situation, e.g., 1.57×10^5 bps vs. 1.26×10^4 bps when $N_{IU} = 25$. However, as N_{IU} grows, more D2D links are established. The gap between these two becomes smaller, viz., 2.28×10^4 bps vs. 1.24×10^4bps when $N_{IU} = 300$. Note that, we assume the transmission power is unchanged. So the transmission rate is only affected by the interference from other links. If N_{IU} is large, the D2D link will get serious interference. However, as the base station has higher transmission power compared to the D2D transmitter, the rate of the cellular links will not get severe influence.

Fig. 5 (a) and (b) show the transmission time for different caching strategies. The two figures describe a comparison of our DGPA-CS caching algorithm with random caching (random choosing files from a 10 size of library), the MRU (caching most recent used files) [3], and no caching. According to the $Zip-f$ distribution, the file popularity are preset with discounted rate $\gamma = 0.5$.

We first simulate the scenario that D2D links happen simultaneously. In this case, the interference from other D2D links is the largest. From Fig. 5 (a) we can see that our DGPA-SC caching strategy has better performances than others.

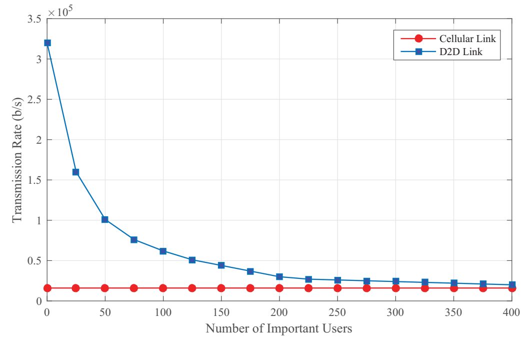
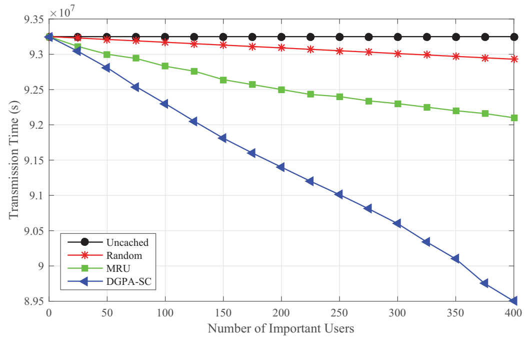
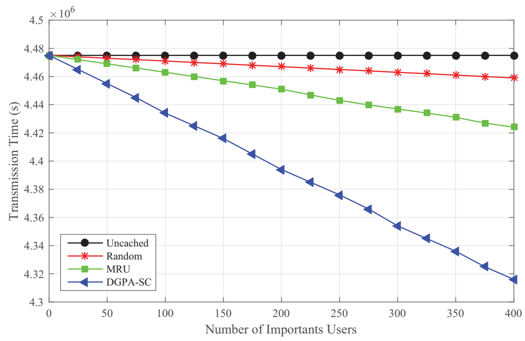


Fig. 4. Transmission rate of D2D links and Cellular links for different numbers of important users



(a) Simultaneously Transmitting



(b) Asynchronous Transmitting

Fig. 5. The transmitting time of D2D links for different numbers caching strategies

For example, compared with the MRU and random caching, the total transmission time of the DGPA-SC is $1.5 \times 10^6 s$ and $1.7 \times 10^6 s$ shorter, respectively, on the condition of $N_{IU} = 200$. As the number of IUs increases, the advantages become more obvious. When $N_{IU} = 300$, the gaps between the DPGA-SC and the MRU and between the DGPA-SC and random caching are $2.1 \times 10^6 s$ and $2.5 \times 10^6 s$, respectively. This reveals that our strategy has better performance in higher population density district. The simulation result in Fig. 5 (b) also depicts a similar trend. We set up D2D links happening asynchronously in this case and the interference impact can be reduced. For example, in comparison with the MRU and random caching, the transmission time of the DGPA-SC is reduced by $1.18 \times 10^5 s$ and $2.25 \times 10^5 s$, respectively, when $N_{IU} = 400$. We also find that the total transmission time in Fig. 5 (b) is always shorter than Fig. 5 (a) ($4.36 \times 10^6 s$ vs. $9.1 \times 10^7 s$ when $N_{IU} = 250$). This is because the interference among D2D links is a main factor affecting the transmission rate. If there are more available D2D links, more transmission time will be taken.

After a detailed analysis of the above figures, we can deduce that in this scenario, although more IUs will produce more interferences, they can still reduce more transmission delay in total. Our DGPA-SC caching strategy is proven to be an outstanding way to ensure the accuracy rate, and it can improve the throughput of the whole system.

V. Conclusion

In this paper, we proposed a DGPA-SC algorithm to solve the cache placement problem in D2D underlaying cellular networks. In the process of learning, we combined the characters of social network to solve some practical issues, such as the selfishness of the users. Simulation results showed our DGPA-SC algorithm has fast convergence speed compared with the Q-learning and obvious advantages in decreasing transmission delay compared with other caching strategies. We concluded that our algorithm is more suitable for high density district.

References

[1] U. Niesen, D. Shah, and G. W. Wornell, "Caching in wireless networks," *Information Theory, IEEE Transactions on*, vol. 58, no. 10, pp. 6524–6540, 2012.

[2] C. V. N. I. Cisco, "Global mobile data traffic forecast update, 2010-2015," *White Paper, February*, 2011.

[3] J. Gu, W. Wang, A. Huang, H. Shan, and Z. Zhang, "Distributed cache replacement for caching-enable base stations in cellular networks," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2648–2653.

[4] L. Marini, J. Li, Y. Li, and B. Vucetic, "Distributed caching based on decenteralized learning automata," in *IEEE International Conferencce on Communications (ICC)*. IEEE, 2015.

[5] L. Song, D. Niyato, Z. Han, and E. Hossain, "Game-theoretic resource allocation methods for device-to-device communication," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 136–144, 2014.

[6] B. Zhang, Y. Li, D. Jin, P. Hui, and Z. Han, "Social-aware peer discovery for d2d communications underlaying cellular networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 5, pp. 2426–2439, May 2015.

[7] Y. Li, T. Wu, P. Hui, D. Jin, and S. Chen, "Social-aware d2d communications: qualitative insights and quantitative analysis," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 150–158, 2014.

[8] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[9] C. Xu, L. Song, Z. Han, Q. Zhao, X. Wang, X. Cheng, and B. Jiao, "Efficiency resource allocation for device-to-device underlay communication systems: A reverse iterative combinatorial auction based approach," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 348–358, 2013.

[10] M. Agache and B. J. Oommen, "Generalized pursuit learning schemes: new families of continuous and discretized learning automata," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 6, pp. 738–749, 2002.

[11] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 71, no. 4, pp. 623–630, 2009.