CCF Transactions on Networking (2019) 2:57–68
https://doi.org/10.1007/s42045-018-0004-z

REGULAR PAPER

Distributed resource allocation in caching-enabled heterogeneous cellular networks based on matching theory

Tingting Liu[1,2] · Jun Li[2] · Feng Shu[2] · Yu Du[2] · Zhu Han[3,4]

Received: 4 August 2018 / Accepted: 22 October 2018 / Published online: 10 December 2018
© China Computer Federation (CCF) 2018

Abstract
Caching mechanism has been identified as an effective solution in reducing mobile users' (MUs') transmission delay. In order to allocate proper files into small-cell base stations (SBSs) which are close to a certain group of MUs, distributed algorithms are proposed based on matching theory in minimizing MUs' transmission latency and maximizing their satisfactions. Specifically, we first investigate the file allocation and MUs association problems in minimizing MUs' transmission latency, and then we study the SBSs association problem to maximize MUs' satisfactions. Moreover, the performance of the proposed algorithms is analyzed from the perspectives of stability, overhead, and complexity. Simulation results demonstrate that our proposed algorithms have a better performance compared to the benchmarks.

Keywords Resource allocation · Matching theory · Caching mechanism · Heterogeneous networks

1 Introduction

Along with the developments of smart phones, mobile users (MUs) show growing interests in using various video entertainments. Evidences indicate that multimedia traffic has occupied a large percentage of the network traffic. However, the current communication networks cannot handle such massive traffic in an efficient way, especially in the peak hours (Bastug et al. 2014; Ge et al. 2014; Siddique et al. 2015). At the same time, the next generation of communication networks have to fulfill a much higher requirement, such as high throughput, high access amount, high data rate, low power consumption, and low latency (Andrews et al. 2014; Osseiran et al. 2014). In order to meet the aforementioned requirements, researches begin to find other alternatives. However, most attempts still rely on changing the hardware equipments or network protocols, leading to high cost and complexity to the current communication networks.

At the same time, it is reported that a large amount of data traffic is produced by a few popular files. Redundant transmissions can be greatly reduced by caching popular files into the network edge, known as content caching (Wang et al. 2014). Thus it is feasible to release backhaul pressures by caching popular files in the network edge facilities, such as SBSs, femto BS and end nodes (Li et al. 2015, 2017a, b; Shanmugam et al. 2013; Wang et al. 2016; Bai et al. 2016). In specific, in Li et al. (2015), the authors study a heterogeneous caching-enabled cellular network with embedded SBSs, where multiple mobile users want to download contents from the nearby SBSs. In Shanmugam et al. (2013), a femto-cell caching system is proposed for help placing files into the femto BSs, with the purpose to minimizing the network downloading latency. They consider two types of

✉ Tingting Liu
liutt@njit.edu.cn

Jun Li
jun.li@njust.edu.cn

Feng Shu
shufeng@njust.edu.cn

Yu Du
yudu@njust.edu.cn

Zhu Han
zhan2@uh.edu

1 School of Communication Engineering, Nanjing Institute of Technology, Nanjing 211167, China

2 School of Electronic and Optical Engineering Nanjing, Nanjing University of Science and Technology, Nanjing 210094, China

3 Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA

4 Department of Computer Science and Engineering, Kyung Hee University, Seoul 02447, South Korea

contents which are coded or uncoded. Wang et al. (2016) investigates an overall transmission cost minimization problem based on the coded contents. Users can download new content items or repair the lost content ones in device-to-device (D2D) based distributed storage systems while guaranteeing users' quality of services (QoSs). In Bai et al. (2016), the authors propose a novel hypergraph framework with caching capability by making use of the social relationship among users. The proposed algorithm can effectively release the traffic pressures on the concerned networks, and improve the overall energy and spectrum efficiency. Although different caching applications in heterogeneous networks have been discussed in the existing literatures, still there exists great challenges in designing an effective, high performance, and low complexity resource allocation scheme in this network.

Matching game is a mathematical tool which can be implemented in a semi distributed manner, and it has a smaller computation complexity compared to other methods (Zhu et al. 2017, 2018). In Bayat et al. (2016), the authors introduce a conventional classification of matching theory. Specifically, matching theory can be divided into three categories including one-to-one, many-to-one, and many-to-many matching. The simplest matching model is the marriage problem, i.e., the one-to-one matching type, which is first introduced by Gale and Shapley (1962). In the marriage problem, men have preferences over women, and also women have preferences over men. The outcome of the marriage problem is a set of one-to-one pairs such that there is no two persons on the opposite sides who both prefer each other over their current partners. In Namvar et al. (2014), authors utilize the many-to-one matching game in maximizing the satisfaction ratio and reducing the downloading latency, with a combined consideration of context-aware information about trajectory profile and QoS requirements of users. In their model, the preferences of the players are interdependent and contingent on the matching structure. They propose a novel algorithm that converges to a stable matching within a reasonable number of iterations. Many-to-many matching game has been used in Hamidouche et al. (2014) to reduce the backhaul loads and the experienced delay in small cell networks. In Bayat et al. (2014), the authors use matching theory to optimize the total satisfaction of users in the uplink of a femto-cell network. In this paper, we propose a two-tier matching framework to solve a resource allocation problem in caching-enabled heterogenous networks. We first propose a matching game to match SBSs who have cached certain contents to MUs, and then associate service providers (SPs) with SBSs. In specific, in the first stage, the SBSs need to decide the cached file set in minimizing the overall MUs' transmission latency. The first stage needs to solve two matching problems: (1) files and the SBSs matching problem; (2) the MUs and the SBSs association problem. In the second stage, the SPs want to rent certain SBSs to enhance the satisfaction level of their serving MUs, and further increase their loyalty. In return, SPs will provide monetary incentives for the SBSs. The relationships between the the players, the sub-problems and the two stages are depicted in Fig. 1. The key contributions of this paper can be summarized as follows.

1. We model a distributed caching system where the system objectives are to minimize the overall network transmission delay, and to maximize the satisfactory level of MUs.
2. We decouple the resource allocation problem into two-stage matching problems. In each stage, the matching algorithm converges to a high performance and stable matching outcome. The first matching problem is modeled as a many-to-many problem, while the second matching problem is modeled as a many-to-one problem.
3. The proposed algorithms are analyzed from the perspectives of stability, over-head, and complexity, and the two algorithms can be verified to be stable.
4. Finally, numerical results are provided to demonstrate that the proposed algorithms perform better than the benchmarks. The proposed matching algorithms can

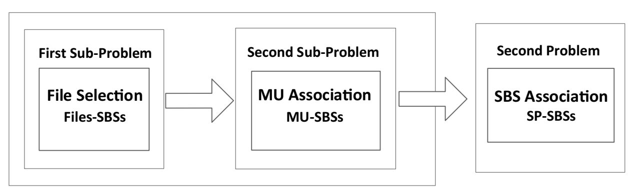


Fig. 1 The relationship between the sub-problems

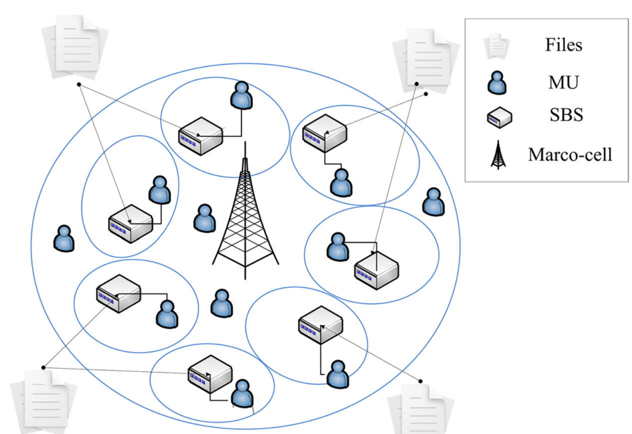


Fig. 2 SBSs based caching system model

achieve a comparable performance to the exhaustive searching method. The social welfare of the proposed matching algorithms is larger than the random al-location scheme.

The rest of this paper is organized as follows. The system model is provided in Sect. 2. Matching-related definitions and resource allocation problems in heterogeneous networks are presented in Sect. 3. Then, the matching algorithms are proposed in Sect. 4. The performances of the proposed matching algorithms are analyzed in Sect. 5. Numerical results are illustrated in Sect. 6, and at last conclusions are drawn in Sect. 7.

2 System model

We consider a heterogenous cellular network consisting of a single marco-cell base station (MBS), N SBSs, and M MUs, as shown in Fig. 2. The set of N SBSs is represented as $\mathcal{N} = \{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_N\}$, and the set of M MUs is denoted by $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_M\}$. SBSs and MUs are assumed to be randomly a allocated in this network. MBS places certain files into SBSs during off-peak time in order to releaset he traffic pressures during peak hour, and further to enhance MUs' satisfactions.

There exists two stages resource allocation problems in the concerned network. In the first stage, files are allocated to SBSs, and then MUs will be associated to certain SBSs. In the second stage, SPs rent certain SBSs. The two stages are interrelated in a way that the matching result of the first stage will be transferred to the second stage to influence the SPs' decisions. In what follows, we will introduce some key parameters in formulating this problem.

2.1 File popularity

We denote the file set by $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_V\}$ consisting of V popular files. Generally, popular files are more preferable than other files. The popularity q_v of $\mathcal{V}_v$ is modeled as the Zipf distribution (Li et al. 2016). q_v is defined as

$$q_v = \frac{1/v^{\beta}}{\sum_{j=1}^{V} 1/j^{\beta}}, \quad \forall v, \tag{1}$$

where the exponent β is a positive skewness parameter. Also, it can be seen that the file with a smaller index v corresponds to a higher popularity. We assume that the MBS' storage capacity is sufficiently large to accommodate the entire file set, while SBSs have a uniform and limited storage size of $Q_{\mathcal{V}}$ files, and $Q_{\mathcal{V}} < V$. All the files are assumed to have the same size of S bits.

2.2 Transmission rate of MUs

We define the transmission rate from each SBSs to MUs as.

$$R_{mn} = W \log_2 \left(1 + \frac{P_n d_{mn}^{-\alpha} h_{mn}^2}{\sum_{n' \in \mathcal{N} \setminus n} P_{n'} d_{mn'}^{-\alpha} h_{mn'}^2 + \sigma^2} \right), \tag{2}$$

where W means the transmission bandwidth, and P_n is the transmission power of SBS $\mathcal{N}_n$. d_{mn} represents the distance between SBS $\mathcal{N}_n$ and an MU $\mathcal{M}_m$. α is a path-loss exponent. The channel between $\mathcal{N}_n$ and $\mathcal{M}_m$ is assumed to follow the Rayleigh distribution. σ^2 is the noise power.

For simplicity, we assume that MBS can support a fixed downloading rate for each MU, denoted by R_m. The channels from MBS to MUs are assumed to be orthogonal to those spanning from the SBSs to MUs. We also assume that

$$R_{mn} > R_m. \tag{3}$$

This assumption makes sure that an MU will prefer downloading files from the SBSs than from the MBS.

2.3 Caching related issue

There are two steps in caching procedures. In the first step, each SBS wants to cache a file set. This set should be determined by its associated MUs. It is clear that MUs show different preferences towards different files. Thus, we define the preference from MU $\mathcal{M}_m$ to file $\mathcal{V}_v$ as

$$p_{mv} = \alpha_{mv} q_v, \tag{4}$$

where α_{mv} is a factor that represent MUs' preferences to file $\mathcal{V}_v$, and q_v is the popularity of the file $\mathcal{V}_v$. Then, each SBS will download a certain file set relying on the preferences of its associated MU set C_n. We define the preference p_{nv} of SBS $\mathcal{N}_n$ to file $\mathcal{V}_v$ as

$$p_{nv} = \frac{1}{|C_n|} \sum_{k \in C_n} p_{kv}, \tag{5}$$

where k is the k th serving MU of SBS $\mathcal{N}_n$.

In the second step, MUs request certain files. Generally, an MU can be covered by multiple SBSs. When an MU $\mathcal{M}_m$ requires a file $\mathcal{V}_v$, we consider the 'SBS-first' strategy. That is to say each MU firstly try to connect to the nearest SBSs which cached the requested file. If such an SBS exists, MU will download files directly from this SBS. If such an SBS doesn't exist, the MU $\mathcal{M}_m$ will download the requested file from an MBS instead. The transmission delay of an MU is shown as follows

$$\tau_{ml}^v = \begin{cases} \tau_m^v = \frac{S}{R_m} & \text{if MU connects to an MBS,} \\ \tau_{mn}^v = \frac{S}{R_{mn}} & \text{if MU connects to an SBS,} \end{cases} \tag{6}$$

where S is the file size.

3 Resource allocation problems in heterogeneous cellular networks

In this paper, two matching problems will be formulated. The first matching problem is constructed between SBSs and MUs. The second matching problem is constructed between SPs and SBSs. Some matching-related definitions are provided in the following.

3.1 Matching-related definition

3.1.1 First matching problem-related definition

The first matching problem can be further divided into two sub-matching problems:

– *File selection*: A sub-matching game between files and the SBSs.
– *MU association*: A sub-matching game between MUs and the SBSs.

In the first sub-matching game, SBSs and files are regarded as players. This game can be modeled as a many-to-many matching problem, since a file can be cached into multiple SBSs, and the SBSs can cache many different files. In many-to-many matching game, each player has a preference list over the members of the opposite set. To construct the preference lists in this model, the symbol $\succ$ is used to represent that a player prefers one player over another player. For example, when an SBS $\mathcal{N}_n$ shows $\mathcal{V}_1 \succ \mathcal{V}_2$ in its preference list, it means that $\mathcal{N}_n$ prefers file $\mathcal{V}_1$ over file $\mathcal{V}_2$. For $\mathcal{V}_v \in \mathcal{V}$ and $\mathcal{N}_n \in \mathcal{N}$, a matching $\mu_{1.1}$ is $\mathcal{V} \bigcup \mathcal{N} \rightarrow 2^{\mathcal{V} \cup \mathcal{N}}$, which satisfies

1. $\mu_{1.1}(\mathcal{N}_n)$ are contained in $\mathcal{V}$ and $\mu_{1.1}(\mathcal{N}_n)$ are contained in $\mathcal{N}$,
2. $\mu_{1.1}(\mathcal{N}_n) \in \mathcal{V}$ and $|\mu_{1.1}(\mathcal{N}_n)| \leq Q_{\mathcal{V}}$,
3. $\mu_{1.1}(\mathcal{V}_v) \in \mathcal{N}$ and $|\mu_{1.1}(\mathcal{V}_v)| \leq Q_{\mathcal{N}}$,
4. $\mu_{1.1}(\mathcal{V}_v) = \mathcal{N}_n \Leftrightarrow \mu_{1.1}(\mathcal{N}_n) = \mathcal{V}_v$,

where item 1 represents that the matching players of $\mathcal{N}_n$ are contained in $\mathcal{V}$ and matching players of $\mathcal{V}_v$ are contained in $\mathcal{N}$, item 2 means that the SBSs can cache $Q_{\mathcal{V}}$ files due to its capacity constraint, and item 3 implies that one file can only be cached in $Q_{\mathcal{N}}$ SBSs. By this constraint, file diversity in this network will be enhanced. Item 4 represents if $\mathcal{V}_v$ is matched to $\mathcal{N}_n$, then $\mathcal{N}_n$ is matched to $\mathcal{V}_v$.

In the second sub-matching game, the players are SBSs and MUs. This game can be modeled as a many-to-one matching problem. The strategy of MUs is to select the best SBS that satisfies their delay tolerance. The SBSs' strategy is to decide whether to serve this MU. An SBS can serve many MUs, but an MU can only be associated to one SBS. We use $\mu_{1,2}$ to represent the matching result of the second sub-matching game.

For $\mathcal{M}_m \in \mathcal{M}$ and $\mathcal{N}_n \in \mathcal{N}$, a matching $\mu_{1.2}$ is $\mathcal{M} \bigcup \mathcal{N} \rightarrow 2^{\mathcal{M} \cup \mathcal{N}}$, which satisfies

1. $\mu_{1.2}(\mathcal{M}_m) \in \mathcal{N}$ and $|\mu_{1.2}(\mathcal{M}_m)| \leq 1$,
2. $\mu_{1.2}(\mathcal{N}_n) \in \mathcal{M}$ and $|\mu_{1.2}(\mathcal{N}_n)| \leq Q_{\mathcal{M}}$,
3. $\mu_{1.2}(\mathcal{M}_m) = \mathcal{N}_n \Leftrightarrow \mu_{1.2}(\mathcal{N}_n) = \mathcal{M}_m$,

where item 1 means an MU can be served by one SBS, item 2 represents an SBS can connect at most $Q_{\mathcal{M}}$ MUs, and item 3 indicates that if $\mathcal{N}_n$ is matched to $\mathcal{M}_m$, then $\mathcal{M}_m$ is matched to $\mathcal{N}_n$.

3.1.2 Second matching problem-related definition

In the second matching game, the players are SBSs and SPs. This game can be modeled as a many-to-one matching problem. The strategy of an SP is to select its most preferring set of the SBSs. The strategy of the SBSs is to gain the most profits through their cooperations with the SP. We use μ_2 to represent the matching result of the second matching game. It has the following properties.

For $\mathcal{S}_s \in \mathcal{S}$ and $\mathcal{N}_n \in \mathcal{N}$, a matching μ_2 is $\mathcal{N} \bigcup \mathcal{S} \rightarrow 2^{\mathcal{N} \cup \mathcal{S}}$, which satisfies

1. $\mu_2(\mathcal{N}_n) \in \mathcal{S}$ and $|\mu_2(\mathcal{N}_n)| \leq 1$,
2. $\mu_2(\mathcal{S}_s) \in \mathcal{N}$ and $|\mu_2(\mathcal{S}_s)| \leq Q_{\mathcal{S}}$,
3. $\mu_2(\mathcal{S}_s) = \mathcal{N}_n \Leftrightarrow \mu_2(\mathcal{N}_n) = \mathcal{S}_s$,

where item 1 means an SBS can be occupied by only one SP, item 2 represents an SP can rent $Q_{\mathcal{S}}$ SBSs, and item 3 states that if $\mathcal{S}_s$ is matched to $\mathcal{N}_n$, then $\mathcal{N}_n$ is matched to $\mathcal{S}_s$.

The relationship between the proposed problems are illustrated in Fig. 1. The resource allocation problems in the concerned heterogeneous cellular networks are constructed as follows.

3.2 First matching problem formulation: file selection and MU association

At first, we formulate the first matching problem to minimize the overall transmission delay of MUs. The problem can be formulated as

$$
\begin{aligned}
&\min_{X,Y} \sum_{n\in\mathcal{N}} \sum_{m\in\mathcal{M}} \sum_{v\in\mathcal{V}} x_{nv} y_{nm} \tau^{v}_{ml}, \\
&\text{s.t.}(a) \sum_{v\in\mathcal{V}} x_{nv} \leq Q_{\mathcal{V}}, \\
&\quad (b) \sum_{n\in\mathcal{N}} x_{nv} \leq Q_{\mathcal{N}}, \\
&\quad (c) \sum_{n\in\mathcal{N}} y_{nm} \leq 1, \\
&\quad (d) \sum_{m\in\mathcal{M}} y_{nm} \leq Q_{\mathcal{M}}, \\
&\quad (e) x_{nv}, y_{nm} \in \{0,1\},
\end{aligned}
\tag{7}
$$

where x_{nv} is the element of matrix X. $x_{nv}=1$ represents that the SBS $\mathcal{N}_n$ caches the file $\mathcal{V}_v$, otherwise $x_{nv}=0$. y_{nm} is the element of matrix Y, and $y_{nm}=1$ denotes that the SBS $\mathcal{N}_n$ serves the MU $\mathcal{M}_m$, otherwise $y_{nm}=0$. Constraint (a) guarantees that each SBS can cache at most $Q_{\mathcal{V}}$ files. Constraint (b) is to make sure that file $\mathcal{V}_v$ can only be cached $Q_{\mathcal{N}}$ times in this network. Constraint (c) limits that one user can be only served by one SBS. Constraint (d) assures that each SBS can serve $Q_{\mathcal{M}}$ MUs at most, and constraint (e) indicates that the values of x_{nv} and y_{nm} can only be 0 or 1.

This optimization problem in (7) is a generalized knapsack problem which is an NP-hard combinatorial problem (Boyd and Vandenberghe 2004). Hence, in order to solve (7), we resort to a suboptimal approach and split the optimization problem into two independent subproblems, i.e., file selection problem and MU association problem.

1. In file selection problem, we match files to SBSs by the following sub-problem:

$$\begin{aligned} \min_X \ & \sum_{n\in\mathcal{N}}\sum_{v\in\mathcal{V}} x_{nv}\tau_{ml}^{v}, \\ \text{s.t.}(a)\ & \sum_{v\in\mathcal{V}} x_{nv} \le Q_{\mathcal{V}}, \\ (b)\ & \sum_{n\in\mathcal{N}} x_{nv} \le Q_{\mathcal{N}}, \\ (c)\ & x_{nv}\in\{0,1\}. \end{aligned} \tag{8}$$

2. In MU association problem, we match the MUs to SBSs by the following sub-problem:

$$\begin{aligned} \min_Y \ & \sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{M}} y_{nm}\tau_{ml}^{v}, \\ \text{s.t.}(a)\ & \sum_{n\in\mathcal{N}} y_{nm} \le 1, \\ (b)\ & \sum_{m\in\mathcal{M}} y_{nm} \le Q_{\mathcal{M}}, \\ (c)\ & y_{nm}\in\{0,1\}. \end{aligned} \tag{9}$$

3.3 Second matching problem formulation: SBS association

We assume that the SBSs are owned by a rational and selfish operator who wants to increase its profits by leasing SBSs to SPs. At the same time, the SPs want to improve the satisfactions of their subscribers. Sigmoid function is used to represent the satisfaction of an MU as (Bayat et al. 2014)

$$\mathfrak{F}^{SP_s} = \frac{1}{1+e^{-\upsilon[\tau_{max}^{SP_s}-\tau_{mn}^{SP_s}]}}, \tag{10}$$

where υ is the steepness of satisfactory curve, and τ_{max} is the maximum endurable latency that an MU can accept.

The second matching problem is formulated as

$$\begin{aligned} \max_Z \ & \sum_{n\in\mathcal{N}}\sum_{s\in\mathcal{S}} z_{ns}(\mathfrak{F}^{SP_s}-c_{ns}), \\ \text{s.t.}(a)\ & \sum_{s\in\mathcal{S}} z_{ns} \le 1, \\ (b)\ & \sum_{n\in\mathcal{N}} z_{ns} \le Q_{\mathcal{S}}, \\ (c)\ & z_{ns}\in\{0,1\}, \end{aligned} \tag{11}$$

where z_{ns} is the element of matrix Z. $z_{ns}=1$ represents the SBS $\mathcal{N}_n$ is rented by SP $\mathcal{S}_s$, otherwise $z_{ns}=0$. c_{ns} is the payment from $\mathcal{S}_s$ to $\mathcal{N}_n$. Constraint (a) guarantees that each SBS could only associates with one SP. Constraint (b) states that at most $Q_{\mathcal{S}}$ SBSs can be allocated to one SP, and constraint (c) states that the values of z_{ns} can only be 0 or 1.

4 The proposed distributed matching algorithms

In this section, we propose three different matching algorithms to solve the established problems. The proposed algorithms are all based on matching theory.

4.1 File selection algorithm

We define files' and SBSs' preference lists as

Definition 1 SBS's preference over file $\mathcal{V}_v\in\mathcal{V}$ is

$$\mathbf{\Gamma}^{v} = p_{nv}, \tag{12}$$

where $\mathbf{\Gamma}^{v}\in\mathbb{C}^{N*V}$ is SBSs' preference matrix over files.

Also, the files have certain preferences towards different SBSs considering their average transmission delay. Assuming, an SBS can serve C_n MUs so that we take the average transmission delay of all the serving MUs as this files' preference over this SBS. We define it as follows.

Algorithm 1 First Sub-matching Algorithm: File Selection Matching Algorithm

Input: $\mathbf{\Gamma}^v$ $\mathbf{\Gamma}^n$
Output: $\mu_{1.1}$.
Steps:
1: **Initialize:** Set $Q_{\mathcal{V}}$, $Q_{\mathcal{N}}$.
2: Construct the unmatched SBSs as a set of $Unmatch_{1.1} = \{\mathcal{N}_1, \mathcal{N}_2, \cdots, \mathcal{N}_n\}$.
3: **Main Process:** Each SBS sends offers to the first file in their preference lists.
4: **If** the total number of offers from SBSs is more than $Q_{\mathcal{V}}$.
5: The file chooses itd most preferred SBSs. The other SBSs will be rejected and stay in the unmatched set.
6: **If** the accepted SBSs still have storage space, it will stay in the unmatched set.
7: **Else** the accepted SBSs will be removed from the unmatched set.
8: **End if**
9: **Else** All of the offers from SBSs will be accepted.
10: **End if**
11: The unmatched SBSs will update the most preferring files list.
12: Return to step 3.
13: The algorithm continues until $Unmatch_{1.1}$ is empty.
14: We arrive at a stable matching $\mu_{1.1}$.

Definition 2 For $\mathcal{V}_v \in \mathcal{V}$, its preference over SBS $\mathcal{N}_n \in \mathcal{N}$ can be given as

$$\mathbf{\Gamma}^n = \frac{1}{|C_n|} \sum_{k \in C_n} \tau_{kn}, \tag{13}$$

where $\mathbf{\Gamma}^n \in \mathbb{C}^{V*N}$, is the files' preference matrix over the SBS.

The detailed algorithm of the first sub-problem is shown in **Algorithm 1**. We propose a distributed algorithm where both files and SBSs interact in a way such that the MUs' sum transmission delay can be minimized. We use $\mathbf{\Gamma}^v$ and $\mathbf{\Gamma}^n$ as preference lists. We assume that the number of files is much larger than the number of SBSs. At first, the SBSs send their first choices according to their preference lists. Since a file can be cached in $Q_{\mathcal{N}}$ SBSs, we need to judge whether the requests from SBSs for a file are more than the quota $Q_{\mathcal{N}}$. If there are more than $Q_{\mathcal{N}}$ SBSs requesting for the same file, this file will select its most preferred $Q_{\mathcal{V}}$ SBSs according to its preference list. The accepted SBSs need to judge whether they have free memory space for other files. If it has, it will be kept on the unmatched list, otherwise it will be removed from the unmatched list. If the requests for a file are less than or equal to the quota $Q_{\mathcal{N}}$, this file may accept these requests all. The remaining SBSs in unmatched list will update its most preferring files and then continue to take part in next round of matching until each SBS gets $Q_{\mathcal{V}}$ files. Finally, we will get a stable matching result $\mu_{1.1}$.

Algorithm 2 Second Sub-matching Problem: MU Association Matching Algorithm

Input: $\mathbf{\Gamma}^m$ $\mathbf{\Gamma}^{n'}$
Output: $\mu_{1.2}$.
Steps:
1: **Initialize:** Set $Q_{\mathcal{M}}$.
2: Construct lists of the nmatched MUs as a set of $Unmatch_{1.2} = \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_m\}$.
3: **Main Process:** Each MU sends an offer to the SBS which is in first place on MUs' preference lists.
4: **If** the number of the offers is more than $Q_{\mathcal{M}}$.
5: the SBS chooses the most preferable $Q_{\mathcal{M}}$ MUs and remove them from $Unmatch_{1.2}$.
6: The others are rejected and still stay in $Unmatch_{12}$.
7: **Else** All of the offers will be accepted by SBSs.
8: **End if**
9: The MUs who are still in $Unmatch_{1.2}$ update their preference lists.
10: Return to step 3.
11: The process repeats until $Unmatch_{1.2}$ is empty or all the SBSs cannot are occupied.
12: The unmatched MUs will establish connections with the MBS.
13: We arrive a stable matching $\mu_{1.2}$.

4.2 MU association algorithm

Next, the Deferred Acceptance (DA) algorithm (Sotomayor and Oliveira 1992) is proposed to solve sub-problem 2. After the file allocation problem solved by **Algorithm 1**, we turn to handle the MU association problem. The transmission delay is an important criteria in judging the provided quality of service of a network. So the transmission delay from the SBS to each serving MU is employed to construct the preference lists of this SBS to MUs. The preference of SBS over MUs is defined as

Definition 3 For an SBS $\mathcal{N}_n \in \mathcal{N}$, its preference over MU $\mathcal{M}_m \in \mathcal{M}$ can be given by

$$\boldsymbol{\Gamma}^{m} = \tau^{v}_{mn}, \tag{14}$$

where $\boldsymbol{\Gamma}^{m} \in \mathbb{C}^{N*M}$ is the SBSs' preference matrix over MUs.

It can be noted that the preferences over SBSs by each MU is affected by the outcome of the first sub-problem. The file selection matching algorithm decides which file set will be cached into a specific SBS. Thus, in the second sub-problem, MUs' preferences over SBS is based on the cached files' popularity p_{mv}. The MU's preference over SBSs is defined as

Definition 4 For an MU $\mathcal{M}_m \in \mathcal{M}$, its preference over an SBS $\mathcal{N}_n \in \mathcal{N}$ can be given by

$$\boldsymbol{\Gamma}^{n'} = \tau^{v}_{mn} p_{mv}, \tag{15}$$

where $\boldsymbol{\Gamma}^{n'} \in \mathbb{C}^{M*N}$ is the MUs' matrix preference over SBSs.

The specific algorithm of the second sub-problem is shown in **Algorithm 2**. MUs send their offers to their most preferable SBSs firstly. Then, if the total number of the requests from MUs for an SBS is larger than the quota $Q_{\mathcal{M}}$, the SBS will decide to accept the most $Q_{\mathcal{M}}$ preferable MUs according to their own preference lists, and reject the others. Otherwise, The SBSs will accept all the requests from MUs. The MUs who are in the unmatched set will continue to take part in the matching progress until all the SBSs are occupied or the unmatched set is empty. If there still exists some MUs, they will connect to the MBS. Finally, we get a stable matching $\mu_{1,2}$.

Algorithm 3 Second Matching Problem: SBS Association Matching Algorithm

Input: $\boldsymbol{\Gamma}^{s}$ $\boldsymbol{\Gamma}^{n''}$
Output: μ_2.
Steps:
1: **Initialize:** Set $Q_{\mathcal{S}}$, t=1.
2: Construct lists of unmatched SBSs as a set of $Unmatch_2 = \{\mathcal{N}_1, \mathcal{N}_2, \cdots, \mathcal{N}_n\}$.
3: **SPs propose to SBSs**: SBSs announce their price c_{ns} based on its serving MUs and the cached files.
4: SPs bid for the SBSs and guarantee that (17) ≥ 0.
5: **SBSs make decisions:**
6: An SBS receives more than one bid, it will increase its price as $c_{ns}(t+1) = c_{ns}(t) + \delta$.
7: An SBS receives one offer from only one SP, and it will match with this SP immediately..
8: An SBS receives nothing, and it will wait for the next round with the same offering price.
9: The remaining SPs continue to bid for the SBS, and assure that (17) ≥ 0.
10: The process repeats until the $Unmatch_2$ is empty.
11: We arrive at a stable matching μ_2.

4.3 SBS association algorithm

When the first matching problem including two sub-matchings is solved, we move on to solve the second matching problem. Firstly, we need to define the preference lists of SPs and SBSs. The preference of an SBS over SPs is defined as follows.

Definition 5 For an SBS $\mathcal{N}_n \in \mathcal{N}$, its preference over an SP $\mathcal{S}_s \in \mathcal{S}$ is given as:

$$\boldsymbol{\Gamma}^{n''} = c_{ns} - b_n, \tag{16}$$

where $\boldsymbol{\Gamma}^{n''} \in \mathbb{C}^{N*I}$ is the SBS's preference matrix over SPs. c_{ns} is the maximum payment that the SP pays to the SBS, while b_n represents the basic cost of each SBS.

Then, the preference of an SP over SBSs is defined as follows.

Definition 6 For an SP $\mathcal{S}_s \in \mathcal{S}$, its preference over the SBSs can be given as:

$$\boldsymbol{\Gamma}^{s} = \zeta \sum_{k \in \mathcal{G}} R_{nk} \frac{1}{p_{ns}} - \sum_{n \in N_s} c_{ns}, \tag{17}$$

where $\boldsymbol{\Gamma}^{s} \in \mathbb{C}^{S*N}$ is the SPs' preferences over SBS. $\zeta \in \mathfrak{R}^{+}$ is a fixed coefficient.$\sum_{n \in N_s} c_{ns}$ is the payment that SPs paid for all its associated SBSs.

As shown in **Algorithm 3**, after initializing the two preference lists $\boldsymbol{\Gamma}^{\mathbf{n}''}$ and $\boldsymbol{\Gamma}^{\mathbf{s}}$, there may have three possibilities that

1. An SBS receives more than one offer, and then this SBS will increase its price with a increment factor δ and wait for next round's bid.
2. An SBS receives one offer from only one SP, and it will match with this SP immediately.
3. An SBS receives nothing, and it will wait for the next round with the same offering price.

An SP is willing to rent as many SBSs as possible when it has a positive utility. The algorithm repeats until the SPs' demand sets are empty. Finally, we get a stable matching μ_2.

5 Algorithm performance analysis

In order to show that the proposed algorithms have noticeable performance, algorithm performance is investigated in this section.

5.1 Stability of the proposed matching algorithms

Firstly, we will discuss the stability of the proposed matching algorithms. Stable performance is a game theory concept which guarantees the reliable behavior of the matching results.

5.1.1 Stability of a many-to-many matching algorithm

A stability theory for the matching problem is developed in (Echenique and Oviedo 2006; Sotomayor 1999). In line with (Sotomayor 1999), we use the concept of pairwise-stability to verify algorithm stability. A matching algorithm is pairwise-stable if there is no agents $\mathcal{V}$ or $\mathcal{N}$ who are not partners. As file selection problem is solved by a many-to-many matching algorithm, we need to prove that the stability of the many-to-many matching result $\mu_{1.1}$ is pairwise stable.

Theorem 1 *Matching result $\mu_{1.1}$ is a pairwise stable one, such that there exists no blocking pair. A pair $(\mathcal{V}_b, \mathcal{N}_b)$ is a blocking pair if both $\mathcal{V}_b$ and $\mathcal{N}_b$ prefer being together to their assignments under $\mu_{1.1}$.*

Proof Assume there is a blocking pair $(\mathcal{V}_1, \mathcal{N}_1)$, $\mathcal{V}_1 \notin \mu_{1.1}(\mathcal{N}_1)$, $\mathcal{N}_1 \notin \mu_{1.1}(\mathcal{V}_1)$

We have the following conclusions:
1. $\mathcal{V}_v \in \mu_{1.1}(\mathcal{N}_1) : \mathcal{V}_1 \succcurlyeq \mathcal{V}_v$.
2. $\mathcal{N}_n \in \mu_{1.1}(\mathcal{V}_1) : \mathcal{N}_1 \succcurlyeq \mathcal{N}_n$.
Then, we have two inferring:

1. *First inferring* $\mathcal{N}_1$ never proposed to $\mathcal{V}_1$. It means $\mathcal{V}_1$ is not better than any matched players of $\mathcal{N}_1$ which contradicts the first conclusion. Thus, $(\mathcal{V}_1, \mathcal{N}_1)$ cannot be a blocking pair.
2. *Second inferring* $\mathcal{N}_1$ proposed to $\mathcal{V}_1$ but it was rejected. It means that $\mathcal{V}_1$ preferred other SBSs $\mathcal{N}_n$ which contradicts the second conclusion. Thus, $(\mathcal{V}_1, \mathcal{N}_1)$ cannot be a blocking pair.

Thus, our file selection algorithm assures that $\mu_{1.1}$ is pairwise stable. This completes the proof.

5.1.2 Stability of a many-to-one matching algorithm

Since the matching $\mu_{1.2}$ obeys a similar matching rule with μ_2, we analyze the stability of the second sub-matching $\mu_{1.2}$ for brevity. We prove the stability of many-to-one matching $\mu_{1.2}$ by contradiction.

Theorem 2 *Matching $\mu_{1.2}$ and μ_2 are stable matching, such that there exists no blocking pair.*

Proof Given a blocking pair $(\mathcal{M}_1, \mathcal{N}_1)$ for $\mu_{1.2}$. In this case, $\mathcal{N}_1$ prefers $\mathcal{M}_1$ to $\mu_2(N_1)$, and $\mathcal{M}_1$ prefers $\mathcal{N}_1$ to at least one element in $\mu_{1.2}(\mathcal{M}_1)$, i.e., the MU matches with another SBS which is in a higher order in preference list than its previous matched player. In this way both opposite sides will improve their outcomes. But it is contrary to the original definition of matching, i.e., the opposite two sides will change its order in preference list firstly in order to get a better outcome. Thus, there exists no blocking pair in matching games and $\mu_{1.2}$ is proved to be a stable many-to-one matching, This completes the proof.

5.2 Overhead and complexity

The proposed matching algorithms are semi-distributed, so it will cause less overhead and complexity compared to the centralized one. The number of communication packets of the algorithms is hard to analyse because the system parameter are set independently. Nevertheless, we can evaluate the communication packets bound for the proposed matching algorithm. Regarding the time scale of the proposed algorithm, the signaling packet number required for the communication between the players until the algorithm converges is very small. In particular, each player is only required to send one bit to the resource-owner to ask for occupying the resources. In return, the resource-owner will only send one bit back to the players indicating its decisions. The total amount of overhead of the proposed algorithms thus can be quite small.

Theorem 3 *The number of communication packets between the files and the SBSs required in the first matching is upper bounded by*

$$\mathfrak{N}_{max} = Q_{\mathcal{N}} N(N-1)(N-1). \tag{18}$$

Proof The worst situation is that the preference lists of MUs and files are totally opposite, i.e., the most popular file from the perspective of an SBS is least interested by this file. Thus, in each round, $N-1$ SBSs will send the rejection information and it will ask $N-1$ files that whether to accept or reject. This completes the proof.

Theorem 4 *The number of communication packets between the MUs and the SBSs required in the first matching is upper bounded by*

$$\mathfrak{N}_{max} = M(M-N)(N-1). \tag{19}$$

Proof In each round, $M-N$ SBSs will be rejected and it will ask $N-1$ round SBSs that whether to accept or reject. This completes the proof.

Theorem 5 *The number of communication packets between the SPs and the SBSs required in the second matching is upper bounded by*

$$\mathfrak{N}_{max} = \frac{1}{\delta} max(c_{max}^{SP} - b_n)(NS + N + F), \tag{20}$$

where $F = min\{N, S\}$.

Proof All SBSs announce their price and it will require NS communication packets. In the worst situation, the SPs will bid for all the SBSs. So it will need N communication

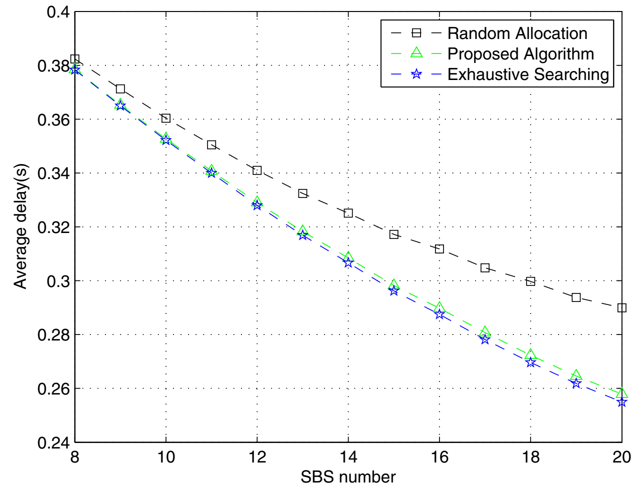


Fig. 4 Average delay vs number of SBSs after applying the second sub-matching algorithm

packets. In the next, some SBSs will accept the bids, it needs F communication packets. Some SPs will cancel their bids, and thus a maximum number of N communication packets are needed. The proof thus follows by noting that the maximum number of communication packets at each iteration is $NS + N + F$.

This completes the proof.

Also, we compare our proposed matching algorithm to the exhaustive searching method and random allocation algorithm. Particularly, for the second sub-problem, the complexity of the different algorithms are listed as follows. The complexity of the algorithms shows as follow:

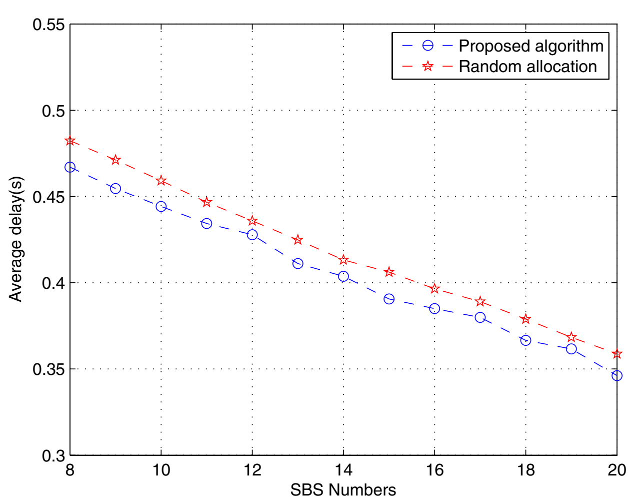


Fig. 3 Average delay vs number of SBSs after applying the first sub-matching algorithm

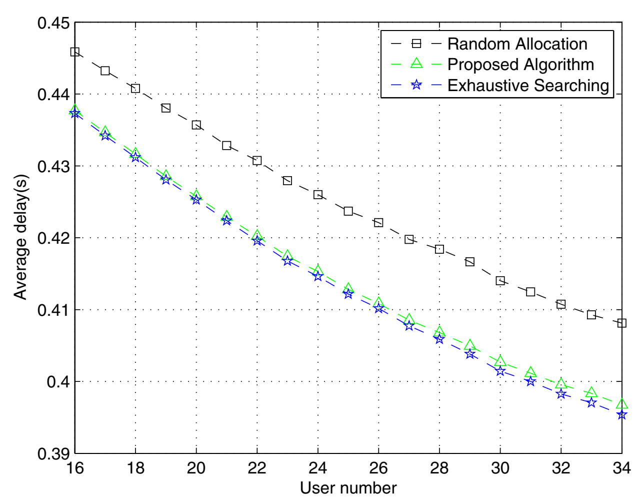


Fig. 5 Average delay vs number of MUs by applying the second sub-matching algorithm

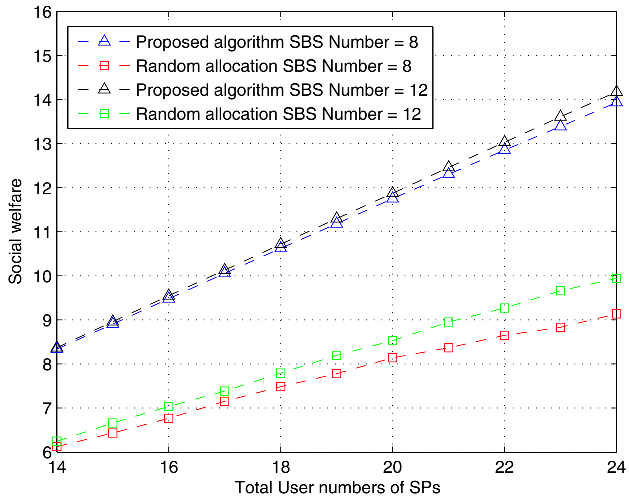


Fig. 6 Social welfare vs number of MUs

$$
\begin{aligned}
&O(\frac{M!}{M-N!}2^{2M+N}) \text{ Exhaustive Searching} \\
&O(M(M-N)(N-1)) \text{ Proposed,} \\
&O(M) \quad \text{Random Allocation.}
\end{aligned}
\tag{21}
$$

It can be observed that the complexity of the exhaustive searching method increases exponentially with the number of players. We can get all pairs of matching by $\frac{M!}{M-N!}$ time, and the complexity of solving each liner programming is 2^{2M+N} (Rinnooykan and Telgen 2008). In contrast, the complexity of the proposed algorithm is $O(M(M-N)(N-1))$, which is significantly lower than that of the exhaustive searching method. The random allocation scheme is proportional to M.

6 Numerical results

In this section, we evaluate the performance of the proposed three matching algorithms by numerical simulations. We assume that the quota of $\mathcal{V}$ is 50, and $Q_{\mathcal{N}} = 2$. The quota of $\mathcal{M}$ is $\in [16, 34]$, and the quota of $\mathcal{M}$ is $\in [8, 20]$. In this simulations, we assume that all MUs and SBSs are randomly located. The transmission power of an SBS is typically 2 w, the pass loss parameters α is 4, and the noise power is set to be 10^{-10} w.

In the following figures, we compare our proposed algorithms with random allocation and exhaustive searching algorithms. In the random allocation algorithm, files are randomly cached. In the exhaustive searching algorithm, the problems are solved in a centralized manner with a very high complexity.

In Fig. 3, the number of files is 50, and the number of SBSs varies from 8 to 20. Each SBS can cache two files at most. As shown in Fig. 3, with the increasing of the number of SBS, the average delay curves of both the random allocation algorithm and proposed algorithm have a declining trend. The proposed algorithm shows a better performance than the random one.

In Fig. 4, the number of MUs is 60 and the number of SBS varies from 8 to 20 with each SBS serving at most 2 MUs. Observing Fig. 4, with the increasing of the number of SBSs, the average delay curves of both the exhaustive searching algorithm and proposed algorithm have a decreasing trend. Though the exhaustive searching algorithm shows a better performance compared to the proposed one, the proposed algorithm has a less computation complexity, while the complexity of the exhaustive searching algorithm increases exponentially over the network size. It is easy to verify that with low complexity, the proposed algorithm can save considerable time. Also, we find that the random allocation algorithm exhibits an inferior delay performance compared with the proposed one.

In Fig. 5, we fix the SBS number to be 10, and vary the MUs' number from 16 to 34. Figure 5 displays that the proposed one has a similar delay performance with the exhaustive searching algorithm, and they both descend as the MUs' numbers increase. The random allocation algorithm shows a inferior performance compared with the proposed algorithm.

In Fig. 6, we fix the number of SBSs to be 8 or 12, and vary the number of MUs from 14 to 24. Figure 6 shows that the proposed algorithm and random allocation algorithm all increase when the total number of MUs increases. It can be seen that our proposed algorithm performs much better than the random allocation one. For example, when the number of users is 20, the proposed algorithm shows 138% better than the random allocation one. Also, we can see that the increase of the number of SBSs will produce more social welfare. The social welfare is the sum of the utilities of all the players in the concerned networks.

7 Conclusion

In this paper, the resource allocation problem in a caching-enabled heterogeneous cellular networks is proposed. A series of distributed matching algorithms are proposed to solve the resource allocation problem. In specific, the matching algorithms include the file allocation, MU association, and SBSs association algorithm. The algorithm performances including the stability, overhead, and complexity are also analyzed, and the two algorithms can be verified to be stable. At last, simulation results are provided to demonstrate that the proposed algorithms have a highly comparable performance with the exhaustive searching algorithm in reducing the average system transmission delay. Moreover, the social welfare of the proposed matching algorithms is larger than the random allocation scheme.

Acknowledgements This work is supported in part by the National Natural Science Foundation of China under Grant nos. 61702258, 61727802, 61771244, 61472190, 61501238, in part by the National Key R&D Program under the grant number 2018YFB1004802, in part by the Jiangsu Provincial Science Foundation under Project BK20150786, in part by the Specially Appointed Professor Program in Jiangsu Province, 2015, in part by the Fundamental Research Funds for the Central Universities under Grant no. 30916011205, in part by the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University, under Grant no. 2017D04, in part by the China Postdoctoral Science Foundation under Grant no. 2016M591852, in part by Postdoctoral research funding program of Jiangsu Province under Grant no. 1601257C, in part by the China Scholarship Council Grant no. 201708320001, in part by US MURI18RT0073, US NSF CNS-1717454, CNS-1731424, CNS-1702850, CNS-1646607.

References

Andrews, J.G., Buzzi, S., Choi, W., Hanly, S.V., Lozano, A., Soong, A.C.K., Zhang, J.C.: What will 5G be? IEEE J. Sel. Areas Commun. **32**(6), 1065–1082 (2014)

Bai, B., Wang, L., Han, Z., Chen, W., Svensson, T.: Caching based socially-aware D2D communications in wireless content delivery networks: a hypergraph framework. IEEE Wirel. Commun. **23**(4), 74–81 (2016)

Bastug, E., Bennis, M., Debbah, M.: Living on the edge: the role of proactive caching in 5G wireless networks. IEEE Commun. Mag. **52**(8), 82–89 (2014)

Bayat, S., Li, Y., Song, L., Han, Z.: Matching theory: applications in wireless communications. IEEE Sign Process Mag. **33**(6), 103–122 (2016)

Bayat, S., Louie, R.H.Y., Han, Z., Vucetic, B., Li, Y.: Distributed user association and femtocell allocation in heterogeneous wireless networks. IEEE Trans. Commun. **62**(8), 3027–3043 (2014)

Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, Cambridge (2004)

Echenique, F., Oviedo, J.: A theory of stability in many-to-many matching markets. Theor. Econ. **1**, 233–273 (2006)

Gale, D., Shapley, L.S.: College admissions and the stability of marriage. Am Math Mon **69**, 1 (1962)

Ge, X., Cheng, H., Guizani, M., Han, T.: 5G wireless backhaul networks: challenges and research advances. IEEE Netw. **28**(6), 6–11 (2014)

Hamidouche, K., Saad, W., Debbah, M.: Many-to-many matching games for proactive social- caching in wireless small cell networks. In: 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Tunisia, pp. 569–574 (2014)

Li, J., Chen, H., Chen, Y., Lin, Z., Vucetic, B., Hanzo, L.: Pricing and resource allocation via game theory for a small-cell video caching system. IEEE J. Sel. Areas Commun. **34**(8), 2115–2129 (2016)

Li, J., Chen, Y., Lin, Z., Chen, W., Vucetic, B., Hanzo, L.: Distributed caching for data dissemination in the downlink of heterogeneous networks. IEEE Trans. Commun. **63**(10), 3553–3568 (2015)

Liu, T., Li, J., Shu, F., Tao, M., Chen, W., Han, Z.: Design of contract-based trading mechanism for a small-cell caching system. IEEE Trans. Wireless Commun. **16**(10), 6602–6617 (2017)

Liu T., Li J., Shu F., Han Z.: Resource trading for a small-cell caching system: A contract-theory based approach. In: 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA (2017)

Namvar, N., Saad, W., Maham, B., Valentin, S.: A context-aware matching game for user association in wireless small cell networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italia, pp. 439–443 (2014)

Osseiran, A., Boccardi, F., Braun, V., Kusume, K.: Scenarios for 5G mobile and wireless communications: the vision of the METIS project. IEEE Commun. Mag. **52**(5), 26–35 (2014)

Rinnooykan, A.H.G., Telgen, J.: The complexity of linear programming. Stat. Neerl. **35**(2), 91–107 (2008)

Shanmugam, K., Golrezaei, N., Dimakis, A.G., Molisch, A.F., Caire, G.: Femtocaching: wireless content delivery through distributed caching helpers. IEEE Trans. Inf. Theory **59**(12), 8402–8413 (2013)

Siddique, U., Tabassum, H., Hossain, E., Kim, D.I.: Wireless backhauling of 5G small cells: challenges and solution approaches. IEEE Wirel. Commun. **22**(5), 22–31 (2015)

Sotomayor, M.: Three remarks on the many-to-many stable matching problem. Math. Soc. Sci. **38**(1), 55–70 (1999)

Sotomayor, M., Oliveira, M.A.: Two-sided matching. Cambridge University Press, Cambridge (1992)

Wang, X., Chen, M., Taleb, T., Ksentini, A., Leung, V.C.M.: Cache in the air: exploiting content caching and delivery techniques for 5G systems. IEEE Commun. Mag. **52**(2), 131–139 (2014)

Wang, L., Wu, H., Ding, Y., Chen, W., Poor, H.V.: Hypergraph-based wireless distributed storage optimization for cellular D2D underlays. IEEE J. Sel. Areas Commun. **34**(10), 2650–2666 (2016)

Zhu, H., Dusit, N., Walid, S., Tamer, B.: Game theory for next-generation wireless and communication networks: modeling, analysis, and design. Cambridge University Press, Cambridge (2018)

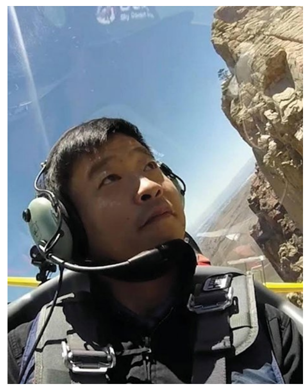
Zhu, H., Yunan, G., Walid, S.: Matching theory for wireless networks. Springer, Berlin (2017)

Tingting Liu received the B.S. degree in Communication Engineering from Nanjing University of Science and Technology, Nanjing, China, and the Ph.D. degree in Information and Communication Engineering from Nanjing University of Science and Technology, Nanjing, China. Now, she is with the school of Communication Engineering in Nanjing Institute of Technology, China. She is also a post-doctor in Nanjing University of Science and Technology. Her research interests include game theory, caching-enabled systems, mobile edge computing, and cognitive radio networks.

Jun Li received Ph. D degree in Electronic Engineering from Shanghai Jiao Tong University, Shanghai, P. R. China in 2009. From January 2009 to June 2009, he worked in the Department of Research and Innovation, Alcatel Lucent Shanghai Bell as a Research Scientist. From June 2009 to April 2012, he was a Postdoctoral Fellow at the School of Electrical Engineering and Telecommunications, the University of New South Wales, Australia. From April 2012 to June 2015, he is a Research Fellow at the School of Electrical Engineering, the University of Sydney, Australia. From June 2015 to now, he is a Professor at the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include network information theory, channel coding theory, wireless network coding and cooperative communications.

Feng Shu received the Ph.D., M.S., and B.S. degrees from the Southeast University, Nanjing, in 2002, XiDian University, Xian, China, in 1997, and Fuyang teaching College, Fuyang, China, in 1994, respectively. From Oct. 2003 to Oct. 2005, he is a post-doctor researcher with the National Key Mobile Communication Lab at the Southeast University. From Sept. 2009 to Sept. 2010, he is a visiting post-doctor at the University of Texas at Dallas. In Oct. 2005, he joined the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China, where he is currently a Professor and supervisor of Ph.D and graduate students. He is also with Fujian Agriculture and Forestry University and awarded with Mingjian Scholar Chair Professor in Fujian Province. His research interests include wireless networks, wireless location, and array signal processing. He has published about 200 scientific and conference papers, of which more than 100 are in archival journals including more than 50 papers on IEEE Journals and 80 SCI-indexed papers. He holds six Chinese patents. Now, Dr. Shu is serving as an Editor for IEEE Access, and is also a Member of IEEE. He has served as session chair or technical program committee member for various international conferences, such as IEEE WCSP 2016, IEEE VTC 2016, etc.

Yu Du received Bachelor's degree from Nanjing University of Science and Technology, P. R. China in 2015. From September 2015 to now, he is a postgraduate at the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include mobile edge computing and matching theory.



Zhu Han received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1999 and 2003, respectively. From 2000 to 2002, he was a Research and Development Engineer with JDSU, Germantown, MD, USA. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an assistant professor with Boise State University, Boise, ID, USA. He is currently John and Rebecca Moores Professor with the Department of Electrical and Computer Engineering and with the Computer Science Department, University of Houston, Houston, TX, USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea. His current research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, the IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in the IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. He currently is an IEEE Communications Society Distinguished Lecturer. He is 1% highly cited researcher according to Web of Science 2017.