Caching resource management of mobile edge network based on Stackelberg game

Qiang Li a, Changlong Lu b, Bin Cao a,c,*, Qinyu Zhang a

a School of Electronic and Information Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen, Guangdong, 518055, China
b School of Mechanical and Electronic Engineering, Jingdezhen Ceramic Institute, Jingdezhen, Jiangxi, 333000, China
c Pengcheng Lab, Shenzhen, Guangdong, 518100, China

ABSTRACT

Mobile edge caching technology is gaining more and more attention because it can effectively improve the Quality of Experience (QoE) of users and reduce backhaul burden. This paper aims to improve the utility of mobile edge caching technology from the perspective of caching resource management by examining a network composed of one operator, multiple users and Content Providers (CPs). The caching resource management model is constructed on the premise of fully considering the QoE of users and the servicing capability of the Base Station (BS). In order to create the best caching resource allocation scheme, the original problem is transformed into a multi-leader multi-follower Stackelberg game model through the analysis of the system model. The strategy combinations and the utility functions of players are analyzed. The existence and uniqueness of the Nash Equilibrium (NE) solution are also analyzed and proved. The optimal strategy combinations and the best responses are deduced in detail. Simulation results and analysis show that the proposed model and algorithm can achieve the optimal allocation of caching resource and improve the QoE of users.

1. Introduction

With the rapid development of wireless communication and network technology, the number of mobile users has grown explosively. And as many mobile Internet services, such as high-definition video, online games and social communications have developed rapidly, the network traffic has also shown an exponential increase. It is estimated that the global mobile data traffic generated per month will reach 49 exabytes by 2021, of which 78% will be generated by video services [1]. The exponential proliferation of video service traffic and the limited network capacity pose an enormous challenge for mobile network operators.

The communication mode based on the master-slave network architecture makes a large amount of content repeatedly transmitted in the network, which leads to the low efficiency of content distribution, severe waste of bandwidth resource, and an increase in transmission delay of users. In order to solve the above problems, the academic and industry have proposed some content-oriented network solutions. In this context, mobile edge caching technology has received great attention. The caching resource is configured for mobile edge network nodes, and the current popular videos and other files are cached in advance, rendering contents closer to users. Hence, most of the popular files requested by users are directly processed by edge network nodes, which can not only reduce delay and improve the QoE of users effectively, but also reduce the load pressure of source servers and the network [2].

For popular files caching of edge network, an efficient caching mode can not only improve the utilization of caching resource, but also reduce the network transmission cost. Mobile edge caching technology can effectively reduce network traffic by up to 35%, which in turn greatly reduces network burden and transmission cost [3]. The caching resource is configured for popular files owned by the CPs. When the operator configures caching resource for mobile edge network, the CPs can purchase the caching resource to cache popular files and provide users with high-quality service. As the caching resource configured in the edge network nodes is limited, it may not be able to meet the requirements of all CPs. Therefore, for the operator, how to manage and allocate caching resource not only affects the interests of the operator and the CPs, but also has a significant impact on the QoE of users.

Related works dealing with the above issues are discussed in Section 2. In Section 3, we analyze different status and roles of the users, the operator and the CPs in the network and build the caching resource management model. To create the best caching resource management scheme, we analyze the system model and recognize that the strategies of...
users are to select the files of the best CP, the strategies of the CPs are to purchase the caching resource and to cache popular files, and the strategy of the operator is how to define the price of caching resource to gain the greatest benefit. Therefore, the problem of the caching resource management is transformed into a multi-leader multi-follower Stackelberg game problem with all users as leaders and all CPs as followers. Then we give the strategy combinations and utility functions of the users and the CPs, and the existence and uniqueness of the NE of the game are proved.

2. Related works

As the caching resource configured by the operator for the BS is limited, the number of popular files, such as videos on the Internet, is huge. To solve this problem, some literature have studied the collaborative management of caching resource at multi-edge network nodes. Tran et al. [4] studied how to cache popular files for multiple nodes by configuring caching resource to minimize the cost of the backhaul, and they also considered the transcoding among multiple bitrate versions of a video. As for the allocation of caching resource for multiple bitrate videos, Xie et al. [5] proposed to focus on energy efficiency and the QoE of users. Their optimization goal was to maximize both the QoE of users and energy cost saving. Jiang et al. [6] proposed an idea of combining multiple BS collaborations and delivery policies to improve content delivery and caching resource performance. Different optimization problems have been established [7–9] to manage edge node caching resource, with their optimization goals being caching utility, energy efficiency and backhaul cost, respectively.

In order to further reduce the network burden, some researchers have studied the caching resource management based on Device-to-Device (D2D). Malak et al. introduced the mobile caching architecture. And the proposed mobile devices can cache popular files locally and share them with other neighboring devices via D2D networks, which greatly reduce the network burden [10]. Wang et al. [11] and Gregori et al. [9] analyzed the combination of edge nodes and D2D caching. The difference is that the former model only has one BS serving the users, while the latter uses multiple BSs to serve the users. Their common point is the use of D2D network. These techniques may meet some challenges. For example, the user will consume his own energy when sharing files, and he may be interfered by other users and then faced with the danger of privacy leakage. As a result, users are often reluctant to share with others.

Although the above literature have deeply studied the caching resource allocation of mobile edge network and D2D network, these research works are all based on the premise of giving popular files collection but not considering the source of the files. The benefits of the CPs are directly related to files caching. Therefore, the status of the CP needs to be considered in caching resource management. Shen et al. [12] applied the Stackelberg game theory to allocate caching resource to the CPs. Xie et al. [13] used a non-cooperative game to rationally allocate caching resource to service providers. The principles of proportional, maximum and minimum fairness have been applied by Hoteit to achieve a reasonable allocation of caching resource [14]. Least Recently Used (LRU) caching replacement algorithm has been used to update the replacement files by Dehghan et al., which takes the sum of utility functions as the goal to achieve a reasonable allocation of caching resource among CPs [15].

The main idea of existing literature is to cache popular files in edge nodes and process users’ requests, which can minimize the network burden. Nevertheless, the edge node servicing capability is limited, and the increase of the number of users’ requests will lead to an increase of the transmission delay, thus would greatly lower the QoE of users. Therefore, the QoE of users and the servicing capability of edge nodes need to be fully considered in the caching resource management.

3. System model and Stackelberg game

3.1. System model

Our model can be mainly divided into three parts, namely, the users, the CPs and the operator, as depicted in Fig. 1. An in-depth analysis of the three is as follows:

Users: Suppose \( J = \{1, \ldots, I\} \) is a set of \( I \) different users. The users are served by the operator and the CPs, so the QoE of the users is affected by them. Wired and wireless transmission services are provided by the operator, and the time delay cannot be reduced because of the existence of distance. In this paper, we suppose that the operator can provide the best service for the users. The users can receive popular file services provided by the CPs. And they will receive the best CP service when they want to obtain a popular file. However, the choices of users are affected due to limited servicing capability of CPs.

CPs: The demands of users are met by the CPs, and we suppose that all requested popular files are cached in the servers of the CPs. We define \( J = \{1, \ldots, \} \) as a set of CPs, and \( \{1, \ldots, K\} \) is the files set of CP \( j \). The same file may be owned by multiple CPs, and its size is different in CPs. In order to provide better service and reduce cost, the CPs can greatly enhance the quality of service through caching popular files in the edge BS configured caching resource. Since the requests of users are directly served by the BS, the data of these requests do not need to be transmitted through path 1 and path 2 in Fig. 1 when files needed by users are cached in the BS. As a result, the transmission delay of users is greatly reduced, the QoE is improved, and the pressure and cost of CP servers are greatly reduced. However, the CPs need to purchase caching resource from the operator. As the popular files are determined by users, the caching operations by the CPs can be performed only after the users have made their decisions.

Operator: Data transmission services of the users and the CPs are provided by the operator. We assume that the data transmission rate is fixed because the issue of caching resource management is the focus of this research. The operator can also provide popular files caching service for the CPs. Considering the limited caching resource and the large demands of the CPs, the operator will increase the price as the demand for caching resource increases, and vice versa. In this regard, the CPs will influence each other because they all need caching resource. At the same time, the BS cannot meet a large number of requests due to the limited servicing capability. The requests of users may be sent to CPs’ servers when the BS cannot deal with all of them. Therefore, the requests of users will be handled by both the BS and different CPs.

In summary, the operator determines the price of caching resource and sells it to the CPs. The users and the CPs are in a competitive relationship. The users’ decision-making dominates the CPs’ decision-making. In order to obtain the best caching resource management scheme, the problem can be transformed into a multi-leader multi-follower Stackelberg game model, in which the users are leaders and the CPs are followers.

3.2. Stackelberg game

There is a master-subordinate relationship between decision-makers.
in the Stackelberg game, that is, the status of two decision-making layers are not equal, with the leader being more influential than the follower. The leaders are always the first to make decisions, and the followers make the best decisions on the basis of the strategies of the leaders and other followers at the same level. We define the players, strategies and utilities of the game according to the system model as follows:

Leaders: Users

Followers: CPs

Strategies: Users' strategies are to select one CP that can provide the greatest utility. Users make decisions on the basis of utility functions, whose strategy combination is defined as \( x = \{x_1, \ldots, x_t\} \). In order to maximize the utility, CPs purchase appropriate caching resource from the operator, and then cache popular files based on the strategy combination of the users. The strategy combinations of the CPs are defined as \( c = \{c_1, \ldots, c_j\} \).

Utilities: The utility of the user \( i \) is defined as \( q_i \), and the utility of the CP \( j \) is defined as \( q_{ij} \), where \( i \in \mathcal{I} \), \( j \in \mathcal{J} \).

In order to maximize their own utilities, the CPs make a non-cooperative game in the situation where the strategy combination of the users is known. Therefore, once the strategy combination of the users is determined, the best responses of CPs satisfy

\[
c' = \{c'_1, \ldots, c'_j\} : q_i(x, c'_j) \geq q_i(x, c_j), \quad i \in \mathcal{I}, j \in \mathcal{J} \quad (1)
\]

where \( c'_j \) is the best response of CP \( j \), and \( c''_j \) are the best responses of all CPs except CP \( j \), i.e.,

\[
c''_j = \left( c''_1, \ldots, c''_{j-1}, c''_{j+1}, \ldots, c'_j \right) \quad (2)
\]

The best response is called the NE point of the non-cooperative game among CPs when the strategy combination of the users is known. And the strategies of all CPs satisfy (1). At this point, each CP obtains the best response, because it will not obtain more benefits by unilaterally adjusting its own strategy.

In this paper, the Stackelberg game consists of two decision making phases: the users’ decisions and the CPs’ decisions. Firstly, the users make decisions to obtain strategy combination \( x \), then the CPs make the best responses \( c \) according to \( x \). After that the users adjust \( x \) based on \( c \), and the two phases alternate with each other. Neither the users’ nor CPs’ strategies change, the NE of the Stackelberg game is reached and satisfies

\[
p_i(x', x'', c') = p_i(x, x'', c'), \quad i \in \mathcal{I} \quad (3)
\]

The following section analyzes the utility functions of the users and the CPs. We assume that each user is concerned about the QoE to be obtained and the cost to be consumed. The QoE of the user is related to file quality, terminal equipment, network condition, and so on. It is assumed that the data transmission rate provided by the operator is fixed, and file quality and terminal equipment are not the research focus. Therefore, the caching locations of files have a great influence on the transmission delay of the requests. The closer it is to the user, the shorter the delay is, and the better the QoE is. Based on the above analysis, the QoE \( q \) is inversely proportional to the transmission delay \( t \), so the relationship between them is defined as follows:

\[
q = \frac{1}{t} \quad (4)
\]

No matter for the BS or the CPs, the servicing capabilities are limited. That is, as the number of requests increases, the transmission delay also increases. Therefore, the transmission delay should be proportional to the number of the requests. In summary, the QoE is inversely proportional to the number of the requests. In order to compare different QoE provided by the CPs and the BS, we assume that the users always know the number of requests serviced by the BS and the CPs, so the relationship between the QoE and the number of requests \( z \) is presented as

\[
q = \frac{1}{t} \equiv \frac{1}{\rho + \rho'} \quad (5)
\]

where \( \rho \) and \( \rho' \) are constant, respectively. And \( z \) is the total number of requests.

Suppose some users have made the best choices before user \( i \), and user \( i \) needs the file \( k \), which is owned by multiple CPs. For file \( k \), we calculate the total requests quantity \( z_k \) served by BS and the total requests quantity \( z_{k,j} \) served by CP \( j \), where 1 and 2 indicate that the request of the user is serviced by the BS and CP \( j \), respectively.

According to (5), the QoE obtained after user \( i \) choosing the two cases are

\[
q_i^1 = \frac{1}{t_i^1} = \frac{1}{a(z_i^1 + 1) + \beta} \quad (6)
\]

where \( a, \beta, b_i \), and \( h_i \) are constants, respectively. As the user’s request is serviced by the BS in the first case, the coefficients are not related to the CPs in (6). And the user’s request is serviced by the CP in the other case, the coefficients are related to the CP in (7).

Let the size of file \( k \) of CP \( j \) is \( s_k \) and the unit cost consumed by users who obtain files is \( f \), the total cost of user \( i \) is

\[
B_k = r s_k \quad (8)
\]

Therefore, we define that the utility function of user \( i \) is

\[
p_i = \lambda q_i^1 - (1 - \lambda B_k), \quad \forall i \in \{1, 2\} \quad (9)
\]

where \( \lambda \in (0, 1) \) is the weight coefficient. It will tend to 1 when user \( i \) pays more attention to the QoE. And it will tend to be zero when user \( i \) pays more attention to the cost. The users would not just care about cost but not the QoE, so \( \lambda \) is not equal to zero.

The users make decisions based on their own utility functions and obtain services from the BS or CPs. With the increase of the number of requests serviced by the BS or the CP, the utility provided decreases continuously due to their limited servicing capabilities. At this time, it may not be the best choice for the next user, so he will obtain service from other CPs. In this way, requests of users are assigned to different CPs in the process. Although the network burden will be increased, the QoE of the user is effectively improved. Finally we can obtain the strategy combination composed of all users and it satisfies

\[
p_i(x'_i, x''_i, c_i) \geq p_i(x, x'', c_i), \quad i \in \mathcal{I} \quad (10)
\]

Each CP counts the number of different file requests based on the users’ strategies, and it also calculates its own utility and chooses the best caching resource. For file \( k \), CP \( j \) will only count the number of requests serviced by the BS if it has cached in the BS. Otherwise, the number of requests of file \( k \) in the server of CP \( j \) is counted. Finally the total number of requests \( M_k \) is counted. We can calculate different request rates for all files and sort them in descending order, and it can be approximated as a Zipf-like distribution with the parameter \( \theta \). Therefore, the rate of file \( k \) is written as

\[
p_k = \frac{k^{-\theta}}{\sum\limits_{k=1}^{M_k} k^{-\theta}} \quad (11)
\]

Suppose the average size of all files of CP \( j \) is \( s_j \), and popular files will be cached in the BS according to popularity. Let caching resource of CP \( j \) be \( c_j \) and the number of caching files is
The ratio of the number of requests $M_j^{cache}$ serviced by the BS to the total number of requests $M_j$ is

$$\frac{M_j^{cache}}{M_j} = \frac{\sum_{i=1}^N \frac{N_i^{cache}}{\Omega_i}}{(1 - \theta_j)\Omega_j} \approx \frac{N_j^{cache}}{(1 - \theta_j)\Omega_j} \tag{13}$$

The caching resource is owned by the operator. According to the analysis of the system model, the price of caching resource is related to the demand, which means that the price increases when the demands increase, and vice versa. So we define the unit price as

$$\eta_{cache} = \frac{\alpha C}{C - \sum c_j} \tag{14}$$

where $C$ is the total caching resource, $\alpha$ is a constant.

The cost of each CP to transmit data over wired links should be considered. Let the unit cost of transmission files be $\eta_{file}$. There is no caching cost when the caching resource $c_j$ of CP $j$ is zero, the link cost can be expressed as

$$W_j^{cache} = \eta_{cache} c_j \tag{15}$$

When the caching resource $c_j$ purchased by CP $j$ is not zero, the caching cost can be expressed as

$$W_j^{cache} = \eta_{cache} c_j \tag{16}$$

The link cost can be expressed as

$$W_j^{cache} = \left( M_j - M_j^{cache} \right) \eta_{link} s_j \tag{17}$$

In summary, the utility function of CP $j$ is defined as the cost saving from link cost consumption to storage cost consumption, which is

$$\pi_{sj} = W_j^{cache} - W_j^{cache} - E_j^{cache} \tag{18}$$

$$= \eta_{link} s_j \left( M_j - M_j^{cache} \right) \eta_{cache} c_j - \eta_{cache} c_j \tag{19}$$

$$= M_j^{cache} \eta_{link} s_j - \eta_{cache} c_j \tag{20}$$

$$= M_j^{cache} \eta_{link} s_j - \eta_{cache} c_j \tag{21}$$

$$= M_j^{cache} \frac{c_j^{1-\theta_j}}{(1 - \theta_j) \Omega_j} - \frac{\alpha C c_j}{C - \sum c_j} \tag{22}$$

The first and second derivatives of the utility function $\pi_{sj}$ with respect to $c_j$ are

$$\frac{\partial \pi_{sj}}{\partial c_j} = \frac{M_j^{cache} \eta_{link} s_j}{(1 - \theta_j) \Omega_j} \left( \frac{1}{\theta_j} \right) c_j^{\beta_j - 1} - \frac{\alpha C}{C - \sum c_j} \tag{23}$$

$$\frac{\partial^2 \pi_{sj}}{\partial c_j^2} = \frac{M_j^{cache} \eta_{link} s_j}{(1 - \theta_j) \Omega_j} \left( \frac{1}{\theta_j} \right) c_j^{\beta_j - 2} - 2\alpha C \left( \frac{C - \sum c_j}{C - \sum c_j} \right)^2 < 0 \tag{24}$$

The second derivative of $\pi_{sj}$ is negative, thus the function is a concave function. When the strategy combination of the users are known, each CP can obtain the maximum utility and the best response, which satisfies Eq. (1). Hence, the NE of the non-cooperative game exists and is unique. For the best response, we can make the first derivative of the utility function of each CP zero and solve these equations. However, it is too hard to get the optimal solution by using the above method. To reduce complexity, Newton iteration method can be used to solve the problem.

In summary, for any strategy combination of the users, the NE of the caching resource allocation among CPs exists. There must be a best strategy combination of the users in which the best response of the CPs can be found so that the Stackelberg solution exists.

4. Proposed algorithm

Suppose all users at time $T$ have made the best choices and obtained strategy combination $x$, then the CPs can obtain the best responses based on their utility functions after getting the number of requests. According to the analysis in the previous section, we will use the Newton iteration method to solve the problem. The iteration formula for caching resource allocation is

$$c_j(t + 1) = c_j(t) + \delta \frac{\partial \pi_{sj}}{\partial c_j(t)} \tag{25}$$

where $\frac{\partial \pi_{sj}}{\partial c_j(t)}$ is the gradient of the utility function of CP $j$, $\delta$ is the search step length, time from $t$ to $t + 1$ is defined as one iteration cycle. In the actual system, the value of $\frac{\partial \pi_{sj}}{\partial c_j(t)}$ is estimated by CP $j$, because CP $j$ does not know the specific strategies of other CPs. The specific method is that CP $j$ sends two caching resource requests $c_j \pm \delta$ to the operator every time, where $\delta$ is a very small number (such as $\delta = 0.001$). The operator calculates the corresponding price in Eq. (14) according to CPs’ demands and return the price to CP $j$ and others. CP $j$ can obtain its marginal utility $\frac{\partial \pi_{sj}}{\partial c_j(t)}$ according to different prices. It is necessary for CP $j$ to reduce the amount of requests when the marginal utility is less than zero, and vice versa. It has been proved that the utility function is concave function for each CP. After a finite number of iterations, the best responses must converge to the NE.

At time $T + 1$, the CPs obtain the best responses $c^*$. Based on $c^*$ and users’ utility functions, they will update strategies and obtain a new strategy combination $x^*$. The users and the CPs alternately obtain the optimal strategy combinations. That is, in the users’ decision making phase, the best responses of the CPs should remain unchanged, and the same is the CP. The strategy combination of the users and the best responses of the CPs are continuously updated. Ultimately, all users achieve the optimal strategy combination $x^*$, and the best responses of CPs obtained do not change any more. At this time, no individual among leaders and followers has the motivation to change their strategies, since even if the strategies are changed, the utility no longer increases. Finally, all the leaders and the followers would obtain the NE solution $(x^*, c^*)$, and the best caching resource allocation scheme are obtained.

5. Simulation analysis

This section presents the simulations to verify and analyze the caching resource allocation algorithm. There are a certain number of users in the coverage of the edge BS that owns the caching resource. The users can select different files from different CPs according to their requirements and each one selects 10 files under average conditions. For the convenience of analysis, it is assumed that the files requested by the users are only between two CPs. CP1 has 500 popular files, and their average size is 100 MB; CP2 has 500 popular files, and their average size is 120 MB. At the same time, most of files are shared by CP1 and CP2. Therefore, when the user needs a file, he may select that file either in CP1 or in CP2 according to his own utility function. The parameters set in the user utility function are showed as: $a = 0.005$, $b_1 = 0.001$, $b_2 = 0.001$, $b_1 = 0.1$, $b_2 = 0.12$. The total caching resource in the BS is 10 GB, the unit link cost $\eta_{file} = 1$, the price parameter $a = 1$, the unit cost of users.
Fig. 2. Best responses convergence of CP1 and CP2.

Fig. 3. Effect of number of users on caching resource requirements.

Fig. 4. Effect of caching price parameter on caching resource demands.

Fig. 5. Effect of link cost on caching resource demands.

Fig. 6. Comparison of QoE for the two algorithms.

That CP1, CP2, and the total request resource gradually converge after a certain number of iterations. It shows that the total caching resource of the requests are slightly smaller than that of the operator. Because the more resource is requested, the more expensive the resource is, and the smaller the effectiveness of the CPs. In order to obtain the maximum utility, each CP will formulate its own strategy based on the strategies of other CPs and the price of the operator. Finally, all CPs can receive the best responses.

Fig. 3 shows how the number of caching resource requested by CP1 and CP2 changes as the number of requesting users increases. It can be seen from the figure that the CPs' best responses increase with the increase of the total number of users. As more users will send more requests, the CPs naturally need more caching resource to cache files in the BS to save cost. The rate of increase is slower and slower, for the more the requirements of caching resource, the higher the price, and there would be a relative decrease in CPs' demands.

Fig. 4 shows how the CPs' caching resource requests change with the operator's price parameter. The larger \( \alpha \) causes the higher caching price for the same demand. For maximum utility, the best responses of the CPs will continue to decrease as the price increases. Fig. 5 shows the change of the best responses of the CPs along with the unit link cost \( \eta_{bh} \). It can be seen that the higher the link cost \( \eta_{bh} \), the larger the requested caching resource is. The reason is that the CPs would choose to request more...
caching resource to reduce the cost due to the higher link cost. By doing that, the cost of files transmission on the backbone network will be reduced.

In order to minimize the link cost, files requested by all users are served by the BS when they are cached in the BS, and then the result data, i.e., the minimum link cost algorithm, is returned to the user. Due to the limited processing capability of the BS, the increasing number of users' requests will increase the transmission delay continuously. Fig. 6 compares the proposed algorithm with the minimum link cost algorithm. It can be seen that when the number of users is small, the QoE of the two algorithms are equal. As the number of users increases, the QoE of both algorithms decreases. However, the declining of the minimum link cost algorithm is faster than the proposed algorithm, as the former algorithm does not consider the servicing capability of the BS and the QoE of users while the proposed algorithm does. With the increase of the number of users, some users may choose to request files from the CP servers, thereby reducing the burden of the BS, because although the link cost is increased, the QoE of the user is guaranteed.

6. Conclusion

As a key technology for mobile edge caching, the caching resource management not only affects the utilization of caching resource, but also has a significant impact on the users, the operator, and CPs. In order to achieve the best allocation of caching resource, this paper analyzes from the perspectives of the users, the operator, and the CPs. We fully consider the BS's servicing capability and the users' QoE, and then build the system model. After the analysis, we examine the relationship between the users and the CPs. And the caching resource management problem is transformed into a multi-leader multi-follower Stackelberg game problem with all users as leaders and all CPs as followers. The existence and uniqueness of the NE solution for the Stackelberg game are analyzed and proved. The optimal strategy combination of the users and the best responses of the CPs are deduced in detail. The final simulation results and analysis show that this paper achieves a reasonable allocation of caching resource and improves the QoE of users.

Acknowledgement

This work has been supported by the National Natural Sciences Foundation of China (NSFC) under Grant No. 61501211, the Basic Research Project of Shenzhen under Grant No. JCYJ20160531192013063 and JCYJ20170307151148585, the Natural Sciences Foundation of Guangdong under Grant No. 2017A030313372, the Natural Scientific Research Innovation Foundation in Harbin Institute of Technology, the Natural Sciences Foundation of Jiangxi under Grant No. 20151BAB217001 and 20151BAB217018, and S&T Foundation of Jingdezhen.

References