QoE-Oriented Resource Optimization for Mobile Cloud Gaming: A Potential Game Approach

Dongyu Guo[1], Yiwen Han[1], Wei Cai[2], Xiaofei Wang[1], and Victor C. M. Leung[3 4]
[1]Tianjin Key Laboratory of Advanced Networking, College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong, China
[3]College of Computer Science & Software Engineering, Shenzhen University, Shenzhen, China
[4]Dept. of Electrical & Computer Engineering, The University of British Columbia, Vancouver, Canada

***Abstract*—Cloud gaming is a novel service provisioning paradigm, which hosts video games in the cloud and transmits interactive game streaming to players via the Internet. In this model, the cloud is required to consume tremendous resources for video rendering and streaming, especially when the number of concurrent players reaches a certain level. On the other hand, different players may have distinct requirements on Quality-of-Experience, such as high video quality, low delay, etc. Under this circumstance, to ensure an overall satisfaction for all players with finite cloud resources becomes a major challenge to existing cloud services. This paper employs game theory to the cloud gaming scenario and proposes a model to meet players' overall requirements with low cost. This game is proved to be a potential game with determining a devised potential function. Our experiment has shown that, with our algorithm, players can achieve a mutually satisfactory steady state, and the system will reduce the overhead up to 50% within the time complexity of $\mathcal{O}(M\log M)$, where M is the number of physical servers.**

***Index Terms*—Mobile Cloud Gaming, Potential Game, Resource Optimization**

I. Introduction

The market for mobile gaming is becoming increasingly popular, and it is reported that mobile revenues account for more than 50% of the global gaming market as it reaches $137.9 billion in 2018 [1]. However, state-of-the-art mobile terminals are still lack of support in increasing needs in gaming storage and computational power. To this end, cloud gaming [2] has been proposed to help relieve the burden on terminal devices by hosting game engine in the cloud. In cloud gaming paradigm, players' commands are sent to a remote server, while the server renders gaming scenes and streams the real-time video frames back to the players [3]. Such architectures can help players experience resource-hungry video games on their light-weight devices such as phones or tablets.

Nevertheless, there are many remaining issues for cloud gaming industrialization, since cloud gaming is resource-intensive and delay-sensitive. Virtualization, which allows multiple players to share one physical machine, is an essential technique in supporting the elastic feature of cloud. However, with respect to commercial scenes with massive players, if players with different requirements and networking conditions are not scheduled properly [4], cloud resources will be not sufficiently utilized, and the players' Quality-of-Experience (QoE) will be downgraded. Therefore, with different interests of players, how to meet each gaming session while guaranteeing the overall good quality of gaming becomes a challenge.

In this paper, we employ game theory to solve this challenge and to help service providers reduce maintenance costs. The adoption of game theory is natural due to the decentralized [5] characteristic of the system: players are competing for the limited cloud resources, thus, they are self-organized into a mutually satisfactory state. Moreover, optimization based on game theory can help data center ease the burden of complex centralized management such as collecting players' network information and dealing with large scale of data for resource allocation. Furthermore, as players may want to play different games and get a better experience in terms of QoE, game theory can be utilized to analyze the resource competition among different players with diverse targeting games.

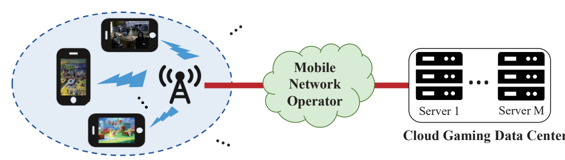

Fig. 1. Mobile cloud gaming architecture

The major contributions of our work are summarized below:

- We propose a mobile cloud gaming architecture along with resource optimization and show it is NP-hard to find optimal solutions for the resource allocation problem.
- We study the inner property of the resource competing game, and show that the game always possesses a Nash Equilibrium (NE) and give the solution coupled with the distributed algorithm brought by potential game theory.
- Besides, the efficiency ratio of our solution with respect to the optimal solution is quantified. Results show that the proposed algorithm can achieve efficient performance and scales well with the number of players.

The remainder of paper is organized as follows. Related works are discussed in Sec. II and Sec. III elaborate the proposed cloud gaming architecture. Then, multi-player resource competing problem is introduced in Sec. IV. Performance analysis is derived in Sec. V along with experiment results given in Sec. VI. Finally, Sec. VII concludes the paper.

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

II. Related Work

A previous literature survey [6] investigated the development and challenges of cloud gaming and particularly mentioned that using virtualization can reduce the cost of service providers. A number of studies [7] [8] used Virtual Machines (VMs) to support cloud gaming, and they proposed specific algorithms to optimize the interference between VMs. Besides, some research groups [9] [10] proposed optimization methods to improve cloud gaming performance, focusing on the prediction of the game session end time and dynamic wireless network conditions. A game theory based algorithm [11], [12] addressed the problem of resource management, however, concerning the geography distributed cloud computing.

III. System Model

A. Architecture of Cloud Gaming

Generally, in mobile cloud gaming architecture, we assume that there are players $\mathcal{N} = \{1, 2, ..., N\}$ and physical servers $\mathcal{M} = \{1, 2, ..., M\}$ performing intensive computation in the data center. With respect to the wireless communication supporting such architecture, we consider a comprehensive scenario including B Base Stations (BSs). Within each BS, each player can enjoy mobile gaming via the wireless link. Following the conventional settings, one VM will host only one game in our architecture. Specifically, for each player, the data center will create a corresponding VM allocated with the devised configuration and files of the requested game.

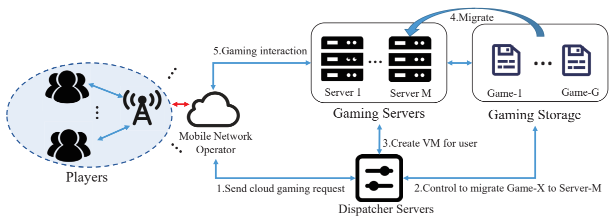

Fig. 2. Processes on the perspective of Data Center

B. Resource Allocation Modeling

Players of mobile cloud gaming are actually contenders competing for finite wireless resources and such completion game are well investigated in previous study [15]. Specifically, the authors formulate that within each BS there are interference-free wireless channels $\mathcal{C} = \{1, 2, ..., C\}$ supporting H mobile devices denoted by $\mathcal{H}$ and each channel is allocated with ω Hz bandwidth.

For the wireless resource competition game, we denote c_n as the allocated channel for the player n, and on the data center side, a_n is denoted as the associated physical server that shall create VM for the gaming session of player n. Thus, for different decision profiles $\boldsymbol{c} = (c_1, c_2, ..., c_H)$ of players associated with the same BS and $\boldsymbol{a} = (a_1, a_2, ..., a_N)$ of all cloud gaming players, i.e., with different resource allocation strategies, introduced gaming experiences may be quite different. In particular, three representative metrics of gaming experience are taken for analysis in our proposed architecture. Nonetheless, the proposed algorithm could incorporate more customized metrics.

*1) **Quality of Gaming Video**:* Video resolution $\mathcal{V}_n$ is one important metric to judge the quality of cloud gaming. As shown in Fig. 1, for the same BS, it will admit the player's gaming request and assign a channel according to the resolution of gaming request. We denote $\mathcal{V}_n$ as the actual gaming resolution player n brought by imperfect wireless transmission and V_n as its required resolution. In general, $\mathcal{V}_n$ is proportional to the transmission rate in the wireless network [14], i.e., the transmission rate $b_n = \mu \mathcal{V}_n$, and then we can compute the actual resolution of player n as

$$\mathcal{V}_n(\boldsymbol{c}) = \frac{\omega}{\mu} \log_2\Big(1 + \frac{\mu \mathrm{V}_n}{\varpi_0 + \sum\limits_{i \in \mathcal{H} \setminus \{n\}: c_i = c_n} \mu \mathrm{V}_n}\Big). \quad (1)$$

Herein, ϖ_0 is the background noise and μ is the proportional constant. Since the transmission rate on backbone network is much better than the wireless transmission, the resolution of mobile cloud gaming is determined by the transmission rate of wireless communication. Therefore, on the side of BS, players are competing for higher $b_n = \mu \mathcal{V}_n$ and finally reach a satisfying NE, which are proved in [15].

*2) **Delay Model in Data Center**:* The proposed cloud gaming architecture is presented in Fig. 2. First, the player chooses one game to play. Then, the dispatcher server will take all requests of players into consideration and allocate an appropriate physical server along with creating a customized VM for each of them. Next, the dispatcher server will admit the request and copy files of the target game to this VM. Finally, the dispatcher server returns the IP address of the created VM to the player for a gaming session.

Since the VMs with game instances are created and destroyed dynamically, there shall exist a clone delay for each player n, which represents the service initialization delay the player must wait for [13]. In the game storage of the data center, the writing speed of the hard disk is denoted as W for simplicity. Note that the clone delay introduced here can also be extended to the whole gaming delay by incorporating the computing delay and data transmission delay without violating the general model. Therefore, if one player chooses to play a game with the file size S_n, the total delay can be represented by (2) concerning the VM initialization time P in work [16].

$$\mathcal{D}_n = \frac{S_n}{W} + P \quad (2)$$

*3) **Experienced FPS**:* Frame per Second (FPS) experienced by cloud gaming players is also a pivotal experience metric. In the scenario of cloud gaming, FPS is determined mainly by the CPU, GPU and RAM usage in the physical server.

Our cloud gaming architecture adopts virtualization techniques. When the sum of resources allocated for all VMs exceeds the resource limit of the physical machine, certain interference and performance degradation will occur. [7] makes great efforts to investigate the relationship between the number of VMs and the gaming FPS, and the experimental results show that it can be approximated by a Sigmoid function.

Similarly, if each VM as a unit is allocated with only a unit of virtualized resource, the relationship between the gaming performance provided by this unit and the FPS can be also be naturally established as well as in [7]. Therefore, the number of resource units that the player n requires to support perfect gaming is denoted as $\mathcal{R}_n$. On condition that all physical servers in the data center are equipped with the same configuration, i.e., κ units of comprehensive hardware resources (related to CPU cores, memories of GPU and RAM), the FPS of gaming can be quantified as

$$\mathcal{F}_n\left(\boldsymbol{a}\right) = \frac{\omega_1}{1 + \mathrm{e}^{\omega_2\left[\frac{1}{\kappa}\left(\sum_{i\in\mathcal{N}\setminus\{n\}:a_i=a_n}\mathcal{R}_i + \mathcal{R}_n\right)\right]+\omega_3}}\mathcal{R}_n, \quad (3)$$

where ω_1, ω_2 and ω_3 are parameters for approximation.

C. Quantified Experience of Players

By comprehensively combining delay, FPS and resolution, the gaming experience loss $\mathcal{L}_n\left(\boldsymbol{a}\right)$ of player n is introduced as (4), which is the target optimization goal of each player.

$$\mathcal{L}_n(\boldsymbol{a}) = \lambda_1\mathcal{D}_n - \lambda_2\mathcal{F}_n(\boldsymbol{a}) - \lambda_3\mathcal{V}_n \quad (4)$$

Herein, λ_1, λ_2 and λ_3 are control parameters catering for different requirement of players. It can be directly inferred that the experience loss $\mathcal{L}_n\left(\boldsymbol{a}\right)$ conceived by player n relates to the size of the playing game, the hardware burden of the associated server and the target game hardware requirement.

IV. Potential Game-based Resource Competition

In this section, how to achieve a lower experience loss among all players is investigated. According to Sec. III, it can be found that players' decisions on associating appropriate servers are tightly coupled. If too many players associate with the same physical server, the interference among VMs shall lead to much more experience loss than $\sum_{i=1}^{N}\mathcal{L}_n\left(\boldsymbol{a}\right)$ in total.

For clear description, the initial associated server and the other server are represented by $\psi \in \{1\}$ and $\overline{\psi} \in \{2, ..., M\}$, respectively. Thus, the initialization strategy adopted in this architecture is first admitting all requests from players and associating them with an initialization server ψ, noted that associating with ψ is only for quantifying the experience of players rather than actually start gaming session on it.

Obviously, the dispatch server should offload the burden from ψ to $\overline{\psi}$. Therefore, the concept of beneficial offloading, indicating that a player chooses associating $\overline{\psi}$ rather ψ and achieves better gaming experience, is introduced as

$$\mathcal{L}_n^{\psi}\left(1, \boldsymbol{a}_{-n}\right) > \mathcal{L}_n^{\overline{\psi}}\left(a_n, \boldsymbol{a}_{-n}\right),\ a_n \in \{2, ..., M\}\ , \quad (5)$$

where $\boldsymbol{a}_{-n} = (a_1, \ldots, a_{n-1}, a_{n+1}, \ldots, a_N)$ is the association choice of players except player n.

A. Centralized Optimization Objective

Maximizing the number of beneficial offloading players is one optimization objectives, and it can be formulated as

$$\begin{aligned} &\max_{\boldsymbol{a}} \sum\nolimits_{n\in\mathcal{N}} I_{\{a_n=\overline{\psi}\}} \\ &\text{s.t. } \mathcal{L}_n^{\psi}\left(1, \boldsymbol{a}_{-n}\right) > \mathcal{L}_n^{\overline{\psi}}\left(a_n, \boldsymbol{a}_{-n}\right), \forall a_n = \overline{\psi}, n \in \mathcal{N}, \\ &\quad a_n \in \{0, 1, \ldots, M\}, \forall n \in \mathcal{N} \end{aligned} \quad (6)$$

where $I_{\{\cdot\}} \in \{0, 1\}$ indicates whether the formula in $\{\cdot\}$ is true ($I = 1$) or not ($I = 0$). Nonetheless, the optimization problem in (6) can be converted into the maximum cardinality bin packing problem [17], and thus proved to be NP-hard.

Another important optimization objective is the overall experience loss of players, i.e., $\min\limits_{\boldsymbol{a}} \sum_{n=1}^{N}\mathcal{L}_n(\boldsymbol{a})$. This optimization problem is also NP-hard, because it is naturally a combinatorial optimization over the dimensional discrete space. Particularly, in the scenario of multi-player cloud gaming, if the time complexity of optimization algorithms is too high, it will be impractical to be employed in large-scale scenarios. Therefore, a potential game based optimization algorithm is proposed for optimizing the aforementioned two objectives.

B. Game Formulation

With perceiving association strategies $\boldsymbol{a}_{-n}$ of other players, player n can thus associate with a new server a_n by considering the load of all servers. In other words, player n can keep associating with the original initialization server ψ, or choose to migrate to another server $\overline{\psi}$ to ease the burden on the initial server while improving their cloud gaming experience.

Consequently, the optimization goal of each player is to lower its own experience loss, i.e.,

$$\min\nolimits_{a_n\in\mathcal{A}_n\triangleq\{0,1,\ldots,M\}} \mathcal{L}_n\left(a_n, \boldsymbol{a}_{-n}\right), \forall n \in \mathcal{N}. \quad (7)$$

According to (5), the experience loss function $\mathcal{L}_n$ of player n can be further denoted as (8), where $\mathcal{L}_n^{\psi}(\boldsymbol{a})$ represents player n associates with the initial server ψ, and $\mathcal{L}_n^{\overline{\psi}}(\boldsymbol{a})$ means that it associates with other servers $\overline{\psi}$.

$$\mathcal{L}_n\left(a_n, \boldsymbol{a}_{-n}\right) = \begin{cases} \mathcal{L}_n^{\psi}(\boldsymbol{a}), & \text{if } a_n = \psi \\ \mathcal{L}_n^{\overline{\psi}}(\boldsymbol{a}), & \text{if } a_n = \overline{\psi} \end{cases} \quad (8)$$

Hence, the competition game above can be formulated as

$$\Gamma = \left(\mathcal{N}, \{\mathcal{A}_n\}_{n\in\mathcal{N}}, \{\mathcal{L}_n\}_{n\in\mathcal{N}}\right), \quad (9)$$

where $\mathcal{N}$ is the set of players, $\mathcal{A}_n$ is the available set of association strategies for player n. Experience loss $\mathcal{L}_n$ is the target to be minimized by player n and Γ represents the whole game process. Next, the concept of Nash Equilibrium (NE) is introduced to address this resource optimization problem.

Definition 1. *The NE for multi-player resource competition game is defined as $\boldsymbol{a}^* = (a_1^*, ..., a_n^*)$ which is also a given profile of player strategies. If the game has reached NE $\boldsymbol{a}^*$, it signifies no players can further reduce their experience loss by changing their associated server unilaterally, i.e.,*

$$\mathcal{L}_n\left(a_n^*, \boldsymbol{a}_{-n}^*\right) \leq \mathcal{L}_n\left(a_n, \boldsymbol{a}_{-n}^*\right),\ \forall a_n \in \mathcal{A}_n,\ n \in \mathcal{N}\ . \quad (10)$$

With the concept of NE, a corollary can be deduced:

Corollary 1. *If the player n at NE choose to associate with $\overline{\psi}$ (i.e., $a_n^* \in \{2, 3, ..., M\}$), then the player n must be a beneficial offloading player.*

This can be proved by contradiction as follows. If a player at NE chooses a server in $\overline{\psi}$ and the server is not a beneficial offloading server, according to the definition of a beneficial offloading, the player can easily keep associating with ψ to reduce its experience loss and get into a new NE. And such new equilibrium strategy $a_n^* = 1$ obviously contradicts with the fact that $a_n^* \in \{2, 3, ..., M\}$.

C. Discussions On Potential Game-based Optimization

Potential game [19] is introduced in our model to prove the presence of NE in resource competition game.

Definition 2. *If player n select a_n and then choose a_n', i.e.,*

$$\mathcal{L}_n(a_n', \boldsymbol{a}_{-n}) \leq \mathcal{L}_n(a_n, \boldsymbol{a}_{-n}), \boldsymbol{a}_{-n} \in \prod\nolimits_{i \neq n} \mathcal{A}_i \ . \tag{11}$$

There exists a global function $\Phi(\boldsymbol{a})$ called potential function and (12) holds, the NE of the competition game can be proved and reached within a finite number of iterations.

$$\Phi(a_n', \boldsymbol{a}_{-n}) \leq \Phi(a_n, \boldsymbol{a}_{-n}) \tag{12}$$

That is if one player reduces its $\mathcal{L}_n$ by changing strategy, the update in potential function will be finite and contribute to a NE. This implies the threshold $\mathcal{T}_n$ for player n choosing beneficial offloading association from ψ to $\overline{\psi}$ shall meet

$$\mathcal{T}_n = \sum_{i \in \mathcal{N} \setminus \{n\} : a_i = \overline{\psi}} \mathcal{R}_i \quad \leq \sum_{i \in \mathcal{N} \setminus \{n\} : a_i = \psi} \mathcal{R}_i \quad . \tag{13}$$

In the following, a potential function (14) is devised to show that the proposed multi-player resource competing game is potential game.

$$\Phi(\boldsymbol{a}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i} \mathcal{R}_i \mathcal{R}_j I_{\{a_i = a_j\}} I_{\{a_i = \overline{\psi}\}} + \sum_{i=1}^{N} \mathcal{R}_i \mathcal{T}_i I_{\{a_i = \psi\}} \tag{14}$$

Theorem 1. *The proposed competition game in cloud gaming is a potential game, and it will converge to a NE in limited times with finite experience improvement of players.*

Proof: We assume that multi-player resource competing game is a potential game with the potential function $\Phi(\boldsymbol{a})$. The player n updates its strategy from a_n to a_n' and reduce its experience loss, i.e., $\mathcal{L}_n(a_n, \boldsymbol{a}_{-n}) > \mathcal{L}_n(a_n', \boldsymbol{a}_{-n})$. According to Definition 2, this will lead to the decrease of potential function $\Phi(\boldsymbol{a})$, i.e., $\Phi(a_n, \boldsymbol{a}_{-n}) > \Phi(a_n', \boldsymbol{a}_{-n})$. Three cases should be considered:

1) $a_n = \overline{\psi}$ and $a_n' = \overline{\psi}$. It means the player n is changing from one beneficial server to another. According to (4), since the hardware requirement of each player and the resolution determined by wireless transmission are fixed within one gaming session, with $\mathcal{L}_n(a_n, \boldsymbol{a}_{-n}) > \mathcal{L}_n(a_n', \boldsymbol{a}_{-n})$, we can get

$$\sum\nolimits_{i \in \mathcal{N} \setminus \{n\} : a_i = a_n} \mathcal{R}_i \quad > \sum\nolimits_{i \in \mathcal{N} \setminus \{n\} : a_i = a_n'} \mathcal{R}_i . \tag{15}$$

In addition, by combining (14) and (15), (16) holds.

$$\begin{aligned} &\Phi(a_n, \boldsymbol{a}_{-n}) - \Phi(a_n', \boldsymbol{a}_{-n}) \\ &= \mathcal{R}_n \sum_{i \neq n} \mathcal{R}_i I_{\{a_i = a_n\}} - \mathcal{R}_n \sum_{i \neq n} \mathcal{R}_i I_{\{a_i = a_n'\}} > 0. \end{aligned} \tag{16}$$

2) $a_n = \overline{\psi}$ and $a_n' = \psi$. It means the player n is changing to the initialization server ψ. In this situation, the reason player n selects to migrate is that the load in server $\overline{\psi}$ is above the threshold $\mathcal{T}_n$, i.e., $\sum_{i \in \mathcal{N} \setminus \{n\} : a_i = a_n} \mathcal{R}_i > \mathcal{T}_n$. This can implies that

$$\begin{aligned} &\Phi(a_n, \boldsymbol{a}_{-n}) - \Phi(a_n', \boldsymbol{a}_{-n}) \\ &= \mathcal{R}_n \sum\nolimits_{i \neq n} \mathcal{R}_i I_{\{a_i = a_n\}} - \mathcal{R}_n \mathcal{T}_n > 0 \ . \end{aligned} \tag{17}$$

3) $a_n = \psi$ and $a_n' = \overline{\psi}$ means the player n judges and associates with the server $\overline{\psi}$ providing better gaming experience. According to the definition of beneficial offloading, $\sum_{i \in \mathcal{N} \setminus \{n\} : a_i = a_n'} \mathcal{R}_i < \mathcal{T}_n$ holds and further the result (18) can be obtained.

$$\begin{aligned} &\Phi(a_n, \boldsymbol{a}_{-n}) - \Phi(a_n', \boldsymbol{a}_{-n}) \\ &= \mathcal{R}_n \mathcal{T}_n - \mathcal{R}_n \sum\nolimits_{i \neq n} \mathcal{R}_i I_{\{a_i = a_n'\}} > 0 \ . \end{aligned} \tag{18}$$

At last, an algorithm based on game theory is presented as Algorithm 1. The time complexity of it is $\mathcal{O}(M \log M)$ which makes agile resource optimization possible. ■

Algorithm 1 Beneficial Offloading Algorithm

Initialization:
Each player associates with ψ, i.e., $a_n(0) = \psi$.
Iteration:

```
repeat
  Each player at time slot t transmits its target physical
  server a_n(t) to the data center dispatcher
  Receive decisions of other players from dispatcher
  Each player computes its beneficial server set Δ_n(t)
  if Δ_n(t) ≠ ∅ then
    Each player sends its target server decision to the
    cloud to contend for decision update token
    if receive the update message from the dispatcher
    then
      Choose one from the beneficial server set Δ_n(t)
      and update the decision
    else
      Keep original decision a_n(t + 1) = a_n(t)
    end if
  else
    Keep original decision a_n(t + 1) = a_n(t)
  end if
until END message is received from the dispatcher
```

V. The Performance Of Distributed Game

To quantify the performance of our solution, we use Price of Anarchy (PoA) [18] in game theory to quantify the effectiveness. γ is denoted as the strategy set of players at a NE. By defining $\boldsymbol{a}^*$ as the optimal solution minimizing the overall experience loss, i.e., $\boldsymbol{a}^* = \arg\min_{a \in \prod_{n=1}^{N} \mathcal{A}_n} \sum_{n \in \mathcal{N}} \mathcal{L}_n(\boldsymbol{a})$, PoA can be obtained as

$$\text{PoA} = \frac{\max_{a \in \gamma} \sum_{n \in \mathcal{N}} \mathcal{L}_n(\boldsymbol{a})}{\sum_{n \in \mathcal{N}} \mathcal{L}_n(\boldsymbol{a}^*)} . \tag{19}$$

For the overall experience loss, a smaller PoA is better. With first introducing (20), we can show the result as (21).

$$\begin{aligned}\mathcal{L}_{n,\min} &\triangleq \lambda_1(\frac{S_n}{W}+P)-\frac{\lambda_2\omega_1\mathcal{R}_n}{1+\mathrm{e}^{\omega_2\frac{\mathcal{R}_n}{\kappa}+\omega_3}}-\lambda_3\mathcal{V}_n \\ \mathcal{L}_{n,\max} &\triangleq \lambda_1(\frac{S_n}{W}+P)-\frac{\lambda_2\omega_1\mathcal{R}_n}{1+\mathrm{e}^{\omega_2(\mathcal{T}_n+\frac{\mathcal{R}_n}{\kappa})+\omega_3}}-\lambda_3\mathcal{V}_n\end{aligned} \tag{20}$$

$$1 \leq \mathrm{PoA} \leq \frac{\sum_{n=1}^{N}\min\{\mathcal{L}_n^{\psi},\mathcal{L}_{n,\max}^{\overline{\psi}}\}}{\sum_{n=1}^{N}\min\{\mathcal{L}_n^{\psi},\mathcal{L}_{n,\min}^{\overline{\psi}}\}} \tag{21}$$

Proof: We denote $\widetilde{a} \in \gamma$ as a NE of potential game. Since the optimal solution is inevitably better than any other solutions, $\mathrm{PoA} \geq 1$ naturally holds.

We find the interference on the same server can at most be $\frac{1}{M}\sum_{i\in\mathcal{N}\backslash\{n\}}\mathcal{R}_i$. Counter-evidence method is used to show this result. We let the interference sensed by the player n at NE $\hat{\boldsymbol{a}}$ over $\frac{1}{M}\sum_{i\in\mathcal{N}\backslash\{n\}}\mathcal{R}_i$, i.e.,

$$\sum\nolimits_{i\in\mathcal{N}\backslash\{n\}:\hat{a}_i=\hat{a}_n}\mathcal{R}_i > \frac{1}{M}\sum\nolimits_{i\in\mathcal{N}\backslash\{n\}}\mathcal{R}_i \ . \tag{22}$$

Because players at NE can not change the decision to get lower experience loss, so there exists

$$\sum_{i\in\mathcal{N}\backslash\{n\}:\hat{a}_i=\hat{a}_n}\mathcal{R}_i \ \leq \sum_{i\in\mathcal{N}\backslash\{n\}:\hat{a}_i=m}\mathcal{R}_i, \ \forall m\in\mathcal{M}. \tag{23}$$

We can get that

$$M\sum_{i\in\mathcal{N}\backslash\{n\}:\hat{a}_i=\hat{a}_n}\mathcal{R}_i \ \leq \sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{N}\backslash\{n\}:\hat{a}_i=m}\mathcal{R}_i \ . \tag{24}$$

According to (22) and (24), we can get the contradiction

$$\begin{aligned}\sum_{i\in\mathcal{N}\backslash\{n\}}\mathcal{R}_i &< M\sum_{i\in\mathcal{N}\backslash\{n\}:\hat{a}_i=\hat{a}_n}\mathcal{R}_i \\ &\leq \sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{N}\backslash\{n\}:\hat{a}_i=m}\mathcal{R}_i \ \leq \sum_{i\in\mathcal{N}\backslash\{n\}}\mathcal{R}_i \ .\end{aligned} \tag{25}$$

This contradiction means the interference experienced by player n must be less than or equal to $\frac{1}{M}\sum_{i\in\mathcal{N}\backslash\{n\}}\mathcal{R}_i$. According to that and (3), if we let $\hat{a}_n=\overline{\psi}$, we can have

$$\mathcal{F}_n(\hat{\boldsymbol{a}}) \geq \frac{\omega_1\mathcal{R}_n}{1+\mathrm{e}^{\omega_2[\frac{1}{\kappa}(\sum_{i\in\mathcal{N}\backslash\{n\}}\frac{\mathcal{R}_i}{M}+\mathcal{R}_n)]+\omega_3}} = \mathcal{F}_n'(\boldsymbol{a}) \ . \tag{26}$$

With combining (4), the experience loss of player n equals

$$\mathcal{L}_n^{\overline{\psi}}(\hat{\boldsymbol{a}}) \leq \ \lambda_1\mathcal{D}_n-\lambda_2\mathcal{F}_n'(\boldsymbol{a})-\lambda_3\mathcal{V}_n=\mathcal{L}_{n,\max}^{\overline{\psi}} \ . \tag{27}$$

Besides, if $\mathcal{L}_n^{\psi}(\boldsymbol{a}) < \mathcal{L}_n^{\overline{\psi}}(\boldsymbol{a})$ and $\hat{a}_n=\overline{\psi}$, then player can always choose server ψ to maintain lower experience loss, i.e., $\hat{a}_n=\psi$. So we can get

$$\mathcal{L}_n(\hat{\boldsymbol{a}})=\min\{\mathcal{L}_n^{\psi},\mathcal{L}_n^{\overline{\psi}}\} \leq \min\{\mathcal{L}_n^{\psi},\mathcal{L}_{n,\max}^{\overline{\psi}}\}. \tag{28}$$

In addition, for centralized optimum $\overline{\boldsymbol{a}}$, if $\overline{a}_n=\overline{\psi}$, we have

$$\mathcal{F}_n(\overline{\boldsymbol{a}}) \leq \frac{\omega_1\mathcal{R}_n}{1+\mathrm{e}^{\omega_2\mathcal{R}_n/\kappa+\omega_3}} \ . \tag{29}$$

Then, we can get experience loss relation as

$$\mathcal{L}_n^{\overline{\psi}}(\overline{\boldsymbol{a}}) \geq \lambda_1\mathcal{D}_n-\frac{\lambda_2\omega_1\mathcal{R}_n}{1+\mathrm{e}^{\omega_2\mathcal{R}_n/\kappa+\omega_3}}-\lambda_3\mathcal{V}_n=\mathcal{L}_{n,\min}^{\overline{\psi}}. \tag{30}$$

Besides, if $\mathcal{L}_n^{\psi}(\boldsymbol{a}) < \mathcal{L}_n^{\overline{\psi}}(\boldsymbol{a})$ and $\overline{a}_n=\overline{\psi}$, then player can choose the initialization server ψ to minimize its experience loss, i.e., $\overline{a}_n=\psi$. The reason is that player n's decision changed to ψ will not affect the other server. So we can get

$$\mathcal{L}_n(\overline{\boldsymbol{a}})=\min\{\mathcal{L}_n^{\psi},\mathcal{L}_n^{\overline{\psi}}\} \geq \min\{\mathcal{L}_n^{\psi},\mathcal{L}_{n,\min}^{\overline{\psi}}\} \ . \tag{31}$$

According to (28) and (31), we can obtain (21). ■

We can easily learn from (21) that when the physical server of the cloud gaming increases the total resources, the computing performance allocated by each player will be improved overall, and the maximum player experience loss $\mathcal{L}_{n,\max}^{\overline{\psi}}$ will decrease. The lower interference player experiences from others, the better cloud gaming experience will be. Besides, the result of NE will be closer to the central optimal algorithm.

VI. Experiment Result

In our experiments, $M=10$ physical servers are employed. Each server is equipped with a CPU with 18 cores, 4 GTX 1080 graphics card, and 128 GB RAM. We select 40 popular games with different hardware requirements. Each players target game is initialized randomly. To acquire the average result, all experiments are repeated 100 times.

A. Convergence of the Resource Competition Game

For portraying the convergence of the resource competition game, we collect all the statics in experiments and present the experience variation of 30 players chosen at random. During the competition process, as shown in Fig. 3, the experience loss of each player is high at the beginning when all players are associated with the initial server ψ. Nonetheless, with performing the proposed algorithm, every player updates its strategy at each time of iteration. By virtue of this, the gaming experience of everyone is improving over iterations and eventually reaching equilibrium, which means the resource competition game shall converge to a Nash Equilibrium.

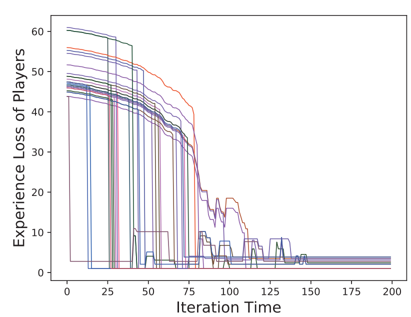

Fig. 3. Experience loss of players throughout the game.

Further, in order to demonstrate the overall performance of the proposed algorithm, two metrics in terms of the overall experience loss and the number of beneficial offloading players are investigated. As shown in Fig. 4, both metrics are finally reaching the state point. In addition, these results demonstrate the effectiveness of the proposed algorithm especially on improving the gaming experience of players.

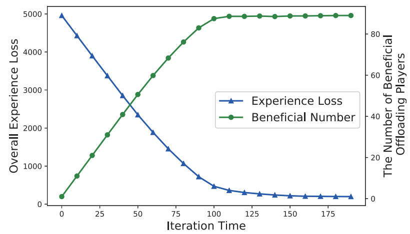

Fig. 4. Overall experience loss and beneficial player numbers.

B. *Performance of the Proposed Algorithm*

To evaluate the performance of our algorithm, three common strategies are introduced for comparison: 1) **Polling Placement Strategy**, players associate with server from 1 to M in sequence repeatedly; 2) **Random Greedy Placement Strategy**, players waiting in queue associate with the server which can provide the best gaming experience according to others' association strategies; 3) **Sorted Greedy Placement Strategy**, players are sorted by the hardware requirement and the least resource-requiring player is first admitted to associate with a server providing the best gaming experience.

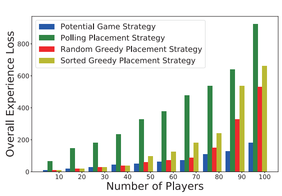

Fig. 5. Performance comparison on experience losses.

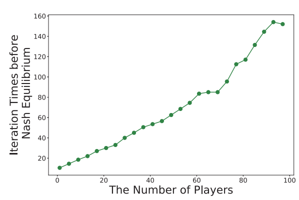

Fig. 6. Iteration times for convergence.

Comparison results of different algorithms are given in Fig. 5. For the metric of overall experience loss, the proposed algorithm can achieve up to 75%, 54%, and 66% improvement compared with the polling placement strategy, the random greedy placement strategy and the sorted greedy placement strategy, respectively. In addition, with more players, our algorithm can bring very good placement strategies and effects when resources are relatively sufficient.

Finally, the relationship between the number of players and the iterations times required for the competing game is studied. The average required iteration times is shown in Fig. 6, which demonstrates that as the number of players increases, the iteration time increases almost linearly. Hence, our algorithm scales well with the number of players and could be a potential method for optimizing resources in cloud gaming.

VII. Conclusion

In this paper, we devise a distributed algorithm by means of potential game. First, optimization problems in multi-player cloud gaming scenario are proved to be NP-hard. Then, the potential function is constructed for proving the presence of Nash Equilibrium. Finally, we analyze and quantify the performance of our algorithm in term of PoA, and demonstrate that our algorithm can converge to a stable point quickly. Besides, numerical results demonstrate that the proposed algorithm achieves superior performance than other strategies and scales well as the increasing number of players.

Acknowledgment

This work is partially supported by the National Key R&D Program of China (2018YFC0809803), China NSFC (Youth) through grant 61702364, China NSFC GD Joint fund U1701263, Shenzhen Fundamental Research Fund under grants No. KQTD2015033114415450, No. ZDSYS201707251409055, and by grant No. 2017ZT07X152.

References

[1] "Global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half."[Online], https://newzoo.com/insights/articles/global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half/

[2] W. Cai, R. Shea, C. Y. Huang, K. T. Chen, J. C. Liu, V. C. Leung, and C. H. Hsu,"The Future of Cloud Gaming", in *Proceedings of the IEEE*, vol. 104, no. 4, pp. 687-691, Apr. 2016.

[3] C. Y. Huang, K. T. Chen, D. Y. Chen, H. J. Hsu, and C. H. Hsu, "GamingAnywhere: The First Open Source Cloud Gaming System," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 10, no. 1, Jan. 2014.

[4] Y. T. Lee, K. T. Chen, H. Su, and C. L. Lei, "Are All Games Equally Cloud-Gaming-Friendly? An Electromyographic Approach" *Workshop on Network and Systems Support For Games*, 2012.

[5] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng and V. C. M. Leung, "Decentralized Applications: The Blockchain-Empowered Software System," in IEEE Access, vol. 6, pp. 53019-53033, 2018.

[6] W. Cai, R. Shea, C. Y. Huang, K. T. Chen, J. C. Liu, V. C. Leung, and C. H. Hsu, "A Survey on Cloud Gaming: Future of Computer Games," in *IEEE Access*, vol. 4, pp. 7605-7620, Aug. 2016.

[7] H. J. Hong, D. Y. Chen, C. Y. Huang, K. T. Chen, and C. H. Hsu, "Placing Virtual Machines to Optimize Cloud Gaming Experience," in *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 42-53, Mar. 2015.

[8] Q. Zhu and T. Tung, "A performance interference model for managing consolidated workloads in QoS-aware clouds," in *2012 IEEE Fifth International Conference on Cloud Computing*, pp. 170-179, Jun. 2012.

[9] Y. S. Li, X. Y. Tang, and W. T. Cai, "Play Request Dispatching for Efficient VM Usage in Cloud Gaming," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, pp. 2052-2063, Dec. 2015.

[10] S. X. Wang, Y. Liu, and S. Dey, "Wireless Network Aware Cloud Scheduler for Scalable Cloud Mobile Gaming," in *IEEE International Conference on Communications (ICC)*, pp. 2081-2086, Jun. 2012.

[11] R. Yu, J. F. Ding, S. Maharjan, S. Gjessing, Y. Zhang, D. Tsang, "Decentralized and Optimal Resource Cooperation in Geo-Distributed Mobile Cloud Computing," in *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 72-84, Jan. 2018.

[12] H. K. Liu, H. Jin, C. Z. Xu, and X. F. Liao, "Performance and energy modeling for live migration of virtual machines" in *The Journal of Networks Software Tools and Applications*, vol. 16, no. 2, pp. 249-264, Jun. 2013

[13] M. Claypool and K. Claypool, "Latency and player actions in online games," in *Communications of the ACM*, vol. 49, no. 11, pp. 40-45, Nov. 2006.

[14] T. S. Rappaport, "Wireless communications: principles and practice," in *New Jersey: prentice hall PTR*, 1996

[15] X. Chen, L. Jiao, W. Z. Li, and X. M. Fu, "Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing," in *IEEE Transactions on Networking*, vol. 24, no. 5, pp. 2827-2840, Oct. 2016

[16] K. T. Chen, Y. C. Chang, H. J. Hsu, D. Y. Chen, C. Y. Huang, C. H. Hsu, "On the Quality of Service of Cloud Gaming", in *IEEE Transactions on MultiMedia*, vol. 16, no. 2, pp. 480-495, Feb. 2014

[17] K. H. Loh, B. Golden, and E.Wasil, "Solving the Maximum Cardinality Bin Packing Problem with a Weight Annealing-Based Algorithm," in *Operations Research and Cyber-Infrastructure*, *Springer*, 2009

[18] T. Roughgarden, Selfish Routing and the Price of Anarchy. in *The MIT Press*, 2005.

[19] D. Monderer and L. S. Shapley, "Potential Games", in *Games and Economic Behavior*, vol. 14, no. 1, pp. 124-143, May 1996.