This article has been accepted for publication in IEEE Transactions on Cloud Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCC.2022.3201544

AI-Bazaar: A Cloud-Edge Computing Power Trading Framework for Ubiquitous AI Services

Xiaoxu Ren, Chao Qiu, *Member, IEEE,* Xiaofei Wang, *Senior Member, IEEE,* Zhu Han, *Fellow, IEEE,* Ke Xu, *Senior Member, IEEE,* Haipeng Yao, *Senior Member, IEEE,* and Song Guo, *Fellow, IEEE*

Abstract—Driven by the burgeoning growth of the Internet of Everything and the substantial breakthroughs in deep learning (DL) algorithms, a booming of artificial intelligence (AI) applications keep emerging. Meanwhile, the advance in existing computing paradigms, i.e., cloud computing and edge computing, provide assorted computing solutions to satisfy the increasingly high requirements for ubiquitous AI services. Nevertheless, there are some non-trivial issues in the computing frameworks, including the underutilization of computing power, the self-interest of computing-power trading mechanism, and the inefficiency of AI services management. To tackle the above issues, we propose a computing-power trading framework based on blockchain, also named AI-Bazaar. In AI-Bazaar, the AI consumers play multiple roles and feel free to contribute the computing power rented from the computing-power provider (CPP) for blockchain mining and AI services. Accordingly, we formulate the computing trading problem as a Stackelberg game. Based on the win or learn fast principle (WoLF), we design a profit-balanced multi-agent reinforcement learning (PB-MARL) algorithm to search the AI-Bazaar equilibrium, while finding the balanced profits for AI consumers and CPP. Numerical simulations are carried out to demonstrate the satisfactory performance and effectiveness of the proposed framework.

Index Terms—Cloud computing, edge computing, computing-power trading, AI services, blockchain, Stackelberg game.

1 Introduction

Recently, driven by the burgeoning growth of the Internet of Everything and the prosperous development in deep learning (DL), a booming of artificial intelligence (AI) applications keep emerging [1]–[3]. Meanwhile, the advance in existing computing paradigms, i.e., cloud computing and edge computing, can satisfy the increasingly high requirements for numerous AI applications with low latency, high bandwidth, and strong computing power, making the AI services everywhere [4]–[6]. In turn, these ubiquitous AI services have given rise to the novel computing framework that requires comprehensive connectivity and efficient computing power [7]–[10].

Generally, the computing nodes in the computing frameworks are widely distributed, while the computing power resources are heterogeneous. The edge nodes are constrained by limited computing resources, which already cannot support the computing-intensive AI services. Meanwhile, cloud nodes provide a cost-effective solution for computing-power sharing among the resource-limited edge nodes, i.e., the AI consumers [11], [12]. To open up the computing power of different computing nodes, there is a pressing need to explore a trusted computing-power sharing and trading framework. Specifically, the computing-power provider (CPP) in this framework integrates the computing resources from the computing nodes to offer the computing power to the resource-constrained computing nodes, enabling a wealth of AI services to meet their business requirements.

Nevertheless, there are some non-trivial issues in the computing-power sharing and trading framework. For example, 1) **The computing power resources in CPP cannot be fully utilized** [13]. The computational resources in some computing nodes assigned to AI consumers may not be sufficient to support increasingly complex computational demands, while other nodes may have normal workloads, or even be idle. The above scenario enables the computing power underutilized, and further causes the computing supply contradiction. 2) **The computing power trading lacks a fair profit-balance mechanism**. Generally speaking, the CPP is self-serving and only focuses on its interests. As the number and types of AI consumers increase, the computing-power trading framework will enter a new era of the free-market economy. Therefore, it is necessary to break the egoism and monopoly of the CPP to provide a profit-balanced trading mechanism. 3) Currently, **the quality management of AI services is inefficient**. Coupled with the rise of AI services, the computing demands have been showing continuous and rapid growth. Nevertheless, subject to the resource-constrained computing environment, the existing computing framework cannot satisfy the accurate, personalized, and high-quality service requirements. In general, the

- *X. Ren, C. Qiu, X. Wang are with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China. C. Qiu, X. Wang are also with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518000, China. E-mails: xiaoxuren@tju.edu.cn, chao.qiu@tju.edu.cn, xiaofeiwang@tju.edu.cn.*
- *Z. Han is with the Department of Electrical and Computer Engineering at the University of Houston, Houston, TX 77004 USA, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea, 446-701. E-mail: e-mail: hanzhu22@gmail.com.*
- *K. Xu is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: xuke@tsinghua.edu.cn.*
- *H. Yao is with Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100088, China. E-mail: yaohaipeng@bupt.edu.cn.*
- *S. Guo is with the Department of Computing, The Hong Kong Polytechnic University, and the Hong Kong Polytechnic University Shenzhen Research Institute, Hong Kong, China. E-mail: song.guo@polyu.edu.hk.*

(Corresponding author: Chao Qiu)

above challenges have fueled the urgent need to develop a fully-utilized, profit-balanced, and efficient computing trading framework for ubiquitous AI services.

Notably, the CPP provides paid resources services for computing-intensive AI applications in the computing power trading framework. Blockchain, as an immutable, accessible, and tamper-proof ledger, promises several benefits for computing resource trading, including recording the scattered computing power, meeting contractual conditions of computing trading and pricing without trusted intermediaries, and so on [14], [15]. Specifically, with the development of consensus protocols, blockchain has become an AI-friendly system. For example, several types of research are focusing on the proof of useful work consensus protocols [16], such as proof of learning (PoL) [17], proof of training quality (PoQ) [18], proof of deep learning (PoDL) [19], etc. In general, these protocols mainly aim at replacing the hashing puzzle in the traditional proof of work (PoW) with the model training and then interconnect the whole learning models in a distributed blockchain manner to construct the lightweight intelligence system. These protocols conquer the problem of computing power waste while improving the learning abilities of the resource-poor computing nodes. Therefore, it is natural to integrate the proof of useful work consensus protocol-based blockchain with the computing-power trading framework to improve AI services quality.

In our previous work [20], by jointly considering the perspectives of AI consumers, networking, and CPP, we design a computing-power networking framework for ubiquitous AI. However, the proposed framework is not concerned with the comprehensive computing-power trading mechanism. In this paper, assisted by the proof of useful work consensus protocol in blockchain (i.e., PoL consensus protocol), we propose a computing trading framework for ubiquitous AI services, named AI-Bazaar. For the three issues raised above, AI-Bazaar can address them from several mechanisms. Specifically, the AI consumers play multiple roles, and they feel free to contribute the computing power rented from the CPP for blockchain mining and AI service, thus solving the first and third issues. Meanwhile, the computing trading problem can be formulated as a Stackelberg game to ensure the balance of interests in AI-Bazaar, which in turn can solve the second issue. The key contributions of this work are as follows.

- Based on blockchain, we propose a computing-power trading framework with a multi-role feature for ubiquitous AI services.
- In order to reach the balance of the profits between CPP and AI consumers, we formulate the computing trading problem as a single-leader-multiple-follower Stackelberg game. Moreover, the Nash equilibrium (NE) of each stage and the AI-Bazaar equilibrium in the Stackelberg game are given with proof of existence and uniqueness.
- Based on the win or learn fast (WoLF) principle, we design a profit-balanced multi-agent reinforcement learning (PB-MARL) algorithm to help the CPP and AI consumers find balanced profits. Experimental results demonstrate the satisfactory performance and effectiveness of the proposed framework.

This article is organized into seven sections. The related work is shown in Section 2. Section 3.1 describes the proposed computing trading framework. In Section 4,we present the system model and problem formulation. The PB-MARL algorithm is designed for searching the AI-Bazaar equilibrium of the proposed game model in Section 5. Simulation results are presented and discussed in Section 6. Finally, concluding remarks are given in Section 7.

2 RELATED WORK

2.1 Traditional Computing-Power Network Frameworks

As a novel computing framework, the computing-power network comes into existence. It aims to provide more flexible and high-quality AI services by connecting resources between the cloud, edge, and end to form a computing-power resources pool.

There are several efforts in the computing-power network frameworks, which considered the computing power can be regarded as the information to share among the computing devices [21]–[23]. For example, the authors in [21] studied a computing first network. In this network, computing power and network states were treated as the routing information and published to the whole nodes. By this means, computing power was connected as a systematic grid. Likewise, the authors in [22] argued that computing power is as easily traded and shared as matter, energy, and information. Further, the work in [23] presented the computing network can seamlessly discover the computing power, and identify the server closest to the user for efficient processing of the AI tasks.

However, all of that works didn't consider the usage of computing power as a paid service, which made the proposed approach un-generalized.

2.2 Blockchain-Assisted Computing Frameworks

With the development of blockchain, its transparency and traceability can offer significant benefits to computing trading systems [24]–[26]. Moreover, aided by cloud computing and blockchain, the authors in [27] developed a resource trading platform for Industrial IoT. Similarly, the authors in [28] established a blockchain-based toll collection system to motivate the heterogeneous edge platforms to share their vacant resources. To satisfies the desirable economic properties of the computing-power trading, the authors in [29] set up an auction-based model for high-efficiency resources trading. The above works probed into a paid service model of computing trading via blockchain.

Nevertheless, on the one hand, the consensus mechanism used in these works was traditional, i.e., PoW protocol, and it needed large amounts of computing power to break the hashing puzzle through brute force, which leaded to resource wastage. On the other hand, the AI consumers were single-role, meaning that they only devoted their computing power to blockchain mining. None of the above considered the comprehensive scenario that consumers could either perform blockchain mining or AI service tasks, or both.

This article has been accepted for publication in IEEE Transactions on Cloud Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCC.2022.3201544

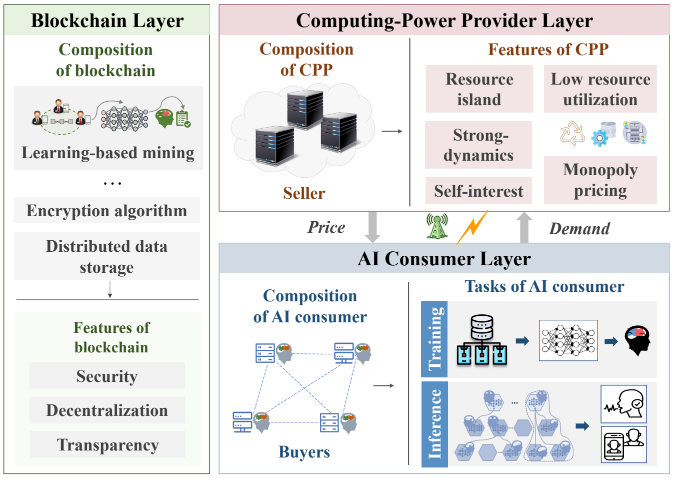


Fig. 1. The proposed computing-power trading framework for AI services.

3 The AI-Bazaar Computing Framework

3.1 Framework Description

In this paper, we propose a computing-power trading framework, i.e., AI-Bazaar, with multi-role (i.e., consumers feel free to be a miner or an AI servicer, or both), and profit-balanced (i.e., the computing profits are satisfied by both sides) features for AI services.

As shown in Fig. 1, AI-Bazaar is committed to balancing the allocation of computing power, connecting the scattered computing resources, motivating the computing nodes to share their computing power, as well as allowing the multifarious customization needs of AI consumers. There are two types of entities in the AI-Bazaar, including AI consumers and CPP. Particularly, the resource-constrained AI consumers request computing power from the CPP via blockchain for satisfying their requirements. Concretely, the AI-Bazaar computing framework consists of the following layers.

3.2 AI Consumers Layer

In AI-Bazaar, there exists a group of AI consumers, denoted by $i \in \mathcal{N} = \{1, 2, \ldots, N\}$, as shown in the AI consumers layer of Fig. 1. Specifically, the tasks of AI consumers are mainly concentrated in two classes: AI training and AI inference.

- AI training: An AI consumer requires intensive computation to support the AI training service, such as image recognition, speech recognition, natural language processing, transportation control, and augmented reality, sustained by neural networks (NNs), reinforcement learning (RL), federated learning (FL) algorithms, etc.
- AI inference: In addition to the AI training tasks, an AI consumer also requires computation to process the AI inference leveraging the training results to manage the realistic tasks. This task retains the learning ability, and can be applied to the new datasets for quick inference response.

To meet the AI demands, bandwidth must also be acquired in AI-Bazaar. Nevertheless, it is provided by the network operator and is therefore not included in the computing-power trading framework. Next, the computing-power provider layer is introduced as follows.

3.3 Computing-Power Provider Layer

With the acceleration of IoT and emerging computing frameworks constructions, the proliferation of computing power from the centralized cloud node to the network node is a matter of momentum. Generally, the computing nodes in the computing frameworks are widely distributed, while the computing power resources are heterogeneous. The CPP are committed to integrating computing-power resources from the computing nodes by resource pooling, providing a cost-effective solution for resource-limited edge devices, i.e., the AI consumers.

Hence, a trusted computing-power trading framework is of great necessity to support credible computing sharing between CPP and AI consumers. Blockchain, as an open, cryptographic, and decentralized system, provides a range of related computing solutions. Then we present the blockchain layer as follows.

3.4 Blockchain Layer

Blockchain includes a series of technologies, such as consensus mechanism, encryption algorithm, distributed data storage, and so on. These technologies enable blockchain to possess the characteristics of decentralization, transparency, security, etc., to help with establishing the secure trusted computing-power trading framework. Specifically, this paper adopts the PoL-based consensus mechanism in response to the huge computing power waste of work-based protocols. PoL regards the NNs training as a working puzzle, which is viewed as the learning-based mining service, rather than meaningless hashes. It is for this reason that the AI consumers in the computing-power trading framework can play multiple roles (i.e. the miner or AI servicer), and we will give a detailed description in Section 4.

From Fig. 1, it is also observed that the NNs in AI-Bazaar are multiplexed. On the one hand, NNs can be treated as a puzzle in the learning-based blockchain. On the other hand, they serve as the underlying support for the DL algorithms. As a result, multiplexed NNs provide strong support for the sustainable development of computing-power trading and AI services at a higher level.

4 AI-Bazaar Model and Problem Formulation

We first present the computing-power trading model in AI-Bazaar in section 4.1 and 4.2. Then the computing-power pricing and purchase demand problem between the CPP and AI consumers can be formulated as a Stackelberg game in section 4.3 . Further, we give the proof of existence and uniqueness of the Stackelberg game in 4.4 .

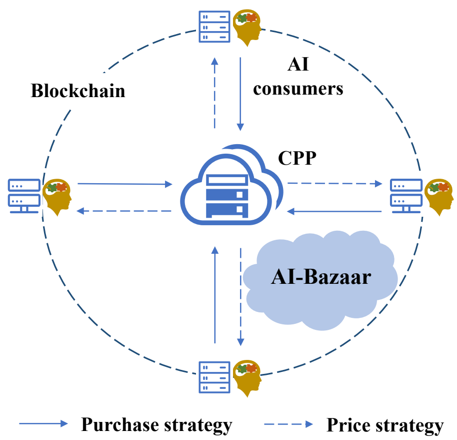


Fig. 2. The AI-Bazaar model.

4.1 AI-Bazaar Model Description

4.1.1 Model Assumption

The AI-Bazaar model is illustrated in Fig. 2. Noted that different computing-power members placed in the framework have widely various strategies, divided into the purchase strategy from AI consumers and the pricing strategy from CPP. Specifically, in AI-Bazaar, CPP concentrates on selling computing power, while AI consumers submit computing-power purchase demands to rent computing power from the CPP for managing diversiform tasks and running blockchain applications for AI services.

Assume that there exists the CPP and N AI computing-power consumers. We formulate the computing trading problem between CPP and AI consumers as a single-leader-multiple-follower Stackelberg game. At first, CPP proclaims a uniform computing-power unit price for all AI consumers. Then each AI consumer determines the purchase strategy to get computing power for supporting a great variety of tasks. The uniform computing-power unit price is represented as $p \in [p_{min}, p_{max}]$, where p_{min} and p_{max} are the minimum and maximum computing-power unit price, respectively. The decided computing-power purchase demands of AI consumers are denoted as $F = \{F_1, F_2, \cdots, F_n, \cdots, F_N\}$, where $F_i \in [F_{min}, F_{max}]$. Specifically, F_{min} and F_{max} are the minimum and maximum computing resources requested by the consumer, respectively.

4.1.2 Multi-Role AI Consumers in AI-Bazaar

In AI-Bazaar, the tasks of AI consumers include AI training and AI inference, resulting in the two roles shown in Fig. 3. As the mining puzzle is solved, the training tasks for AI consumers will be accelerated. At this point, the AI consumer acts as a miner to fight for the bookkeeping right of the blockchain, to receive the block reward. If the CPP returns the required training result to the AI consumer for inference, the AI consumer can be an AI servicer to satisfy their personalized demands. In addition, AI consumers have the flexibility to switch roles between AI services and miners. That is, they can choose a portion of computing power for business needs and another portion for blockchain mining.

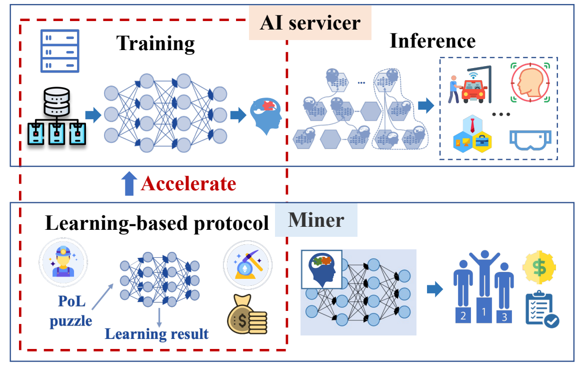


Fig. 3. Multiple roles for the consumer.

4.2 Learning-Based Mining Model

In this paper, we adopt the learning-based consensus protocol, i.e., PoL [17], as shown in Fig. 4. It regards the local training process as the puzzle and believes an NNs with the smaller loss function can efficiently provide high-quality AI services. The winner will issue a block and further get returns. After reaching the consensus, the other nodes then change their parameters in the NNs according to the winner's transaction. To perform the PoL process between AI consumers, we need to utilize the same standard dataset only including normal data. The model can only be brought online if its results on this standard dataset are up to par, which also prevents poisoning attacks.

Assume consumer i decides to allocate the part of computing power β_i for mining, named the role-playing ratio. Accordingly, the proportion of consumer' computing power devoted to mining i for mining in the whole blockchain network is α_i, which is

$$\alpha_i = \frac{\beta_i F_i}{\sum_{j=1}^{N} \beta_j F_j}. \tag{1}$$

Similar to PoW [30], we consider the occurrence of addressing the learning-based puzzle can be formulated as a random variable subject to a Poisson distribution. Analogously, the probability that miner i successfully solves this type of puzzle and reaches consensus is ρ_i, which can be expressed as

$$\rho_i = \alpha_i e^{-\varepsilon T_i^p}, \tag{2}$$

where ε refers to a constant parameter associated with learning puzzle, and T_i^p means the propagation time required for a block to reach consensus. Specifically, T_i^p represents the liner function with respect to the block size B_i, the propagation factor τ and the evaluation metric of each miner solution for the learning-based puzzle δ_i, i.e. $T_i^p = \tau\delta_i B_i$. For example, in the PoL, ε is related to the given training time T_{Train}, which means $\varepsilon = 1/T_{Train}$. Without loss of generality, we assume each block contains the equal number of transactions, that is $B_i = B$. Additionally, consider the loss function value of trained NNs from the miner i as the evaluation indicator δ_i.

If the miner successfully addresses the learning-based puzzle, it will broadcast its solution to the whole network, while other nodes throughout the network validate the

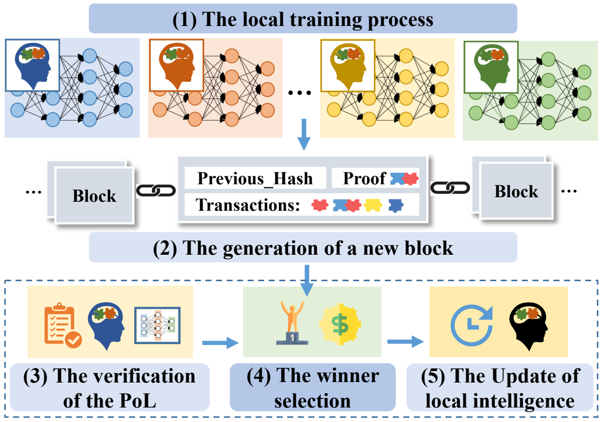


Fig. 4. The process of the PoL blockchain.

correctness of this solution and reach a consensus. The first miner that successfully wins the bookkeeping right will receive the reward. Likewise, the miners could gain rewards form two ways. The reward R is obtained from mining successfully, and the reward R_p is the performance reward, defined as the product between a performance reward factor λ and block size B, i.e. $R_p = \lambda B$. Thus, the profit that an AI consumer derives from mining is defined as follows:

$$U_i^m = (R + \lambda B)\alpha_i e^{-\varepsilon\tau\delta_i B}. \tag{3}$$

4.3 Profit-Balanced Stackelberg Game Model

We formulate the computing-power trading between computing-power members as a two-stage Stackelberg game model. Their profit functions in each stage are given as follows.

Consider that computing-power members have quasi-linear utilities [31]. Specifically, the CPP profit gained in AI-Bazaar is defined by the AI consumers' payment minus the service cost. In the first stage, the CPP decides the computing-power pricing strategy within strategy space $P = \{p \mid p_{min} \leq p \leq p_{max}\}$ based on the demands provided by the AI consumers to maximize its profit U_{cpp}, it can be described as:

$$\begin{aligned} P1: \max_{p} \ & \sum_{i=1}^{N} (p - C)F_i \\ \text{s.t. } & p \in [p_{min}, p_{max}], \end{aligned} \tag{4}$$

where C is the electricity cost per computing-power unit. This revenue-incentive mechanism will incentivize selfish CPP to join in the AI-Bazaar, while utilizing the idle computing power for AI services.

Furthermore, the AI consumer's profit received in the AI-Bazaar is defined by the reward from blockchain minus the mining risk cost and payment for renting computing power from CPP. In the second stage, the AI consumer decides the purchase amount of computational power with the purchase strategy space $F = \{F_i \mid_{i\in\mathcal{N}}: F_{min} \leq F_i \leq F_{max}\}$ based on the computing-power unit price set in the first stage. As such, for consumer i, it can flexibly act multi-role to maximize its own profit U_i, which can be formulated as:

$$\begin{aligned} P2: \max_{F_i} \ & F_i((1-\beta_i)C_b - u_i p) + m_i U_i^m \\ \text{s.t. } & F_i \in [F_{min}, F_{max}], \end{aligned} \tag{5}$$

where U_i^m is the profit of the AI consumer from mining. Specifically, u_i is a weight factor indicating the tendency of the ith AI consumer to think about how much the payment cost is more important to the total profit, and the m_i is the token effect parameter denoting the equivalent monetary worth of mining. C_b denotes the profits per computing-power unit caused by realizing business needs.

Overall, both CPP and AI consumers will like to acquire the greatest profits [20], [24]. In the following, we investigate the equilibrium property and tradeoff analysis of the two parties' profits.

4.4 Model Analysis

To verify the formulated Stackelberg game exists the equilibrium, we bring the following definitions and theorems.

Definition 1. *An AI-Bazaar equilibrium point of Stackelberg Game in this paper is a group of strategies $\{\{F_i^*\}_{i\in N}, P^*\}$ meeting the following conditions:*

$$\begin{aligned} U_{cpp}(F^*, P^*) \geq & U_{cpp}(F^*, P), \\ U_i(F^*, P^*) \geq & U_i(F, P^*). \end{aligned} \tag{6}$$

An AI-Bazaar equilibrium refers to that no player has an incentive to change its strategy after taking into account the opponent's decisions. Therefore, it is the optimal strategy in this profit-balanced game. Next, we will prove the equilibrium property of each stage and the AI-Bazaar equilibrium, and further analyze the profit-balanced problems of the two parties in detail.

4.4.1 AI Consumers' Game

The self-interest behaviors of computing-power members in AI-Bazaar create a competitive environment, and form a noncooperative game on the consumer side.

Lemma 1. *The strategy space for each consumer F is a non-empty, convex, and compact set, and U_i is an apparently continuous function in F.*

Proof. In the second stage, the AI consumers have N purchase strategies, and the domain of these strategies is $[F_{min}, F_{max}]$. It's easy to observe this strategy space is a non-empty convex set, and the profit function u_i of each consumer is continuous. Moreover, the domain of the strategy space has the upper bound, meaning F is a compact set. The proof is completed. □

Theorem 1. *The NE in the AI consumer's game exists.*

Proof. From lemma 1, we know U_i is apparently continuous in F. We set $R_i^c = m_i(R + \lambda B)e^{-\varepsilon\tau\delta_i B}$ and take the first and second derivatives of (5) with respect to F_i respectively, which are given by:

$$\frac{\partial U_i}{\partial F_i} = -\left((\beta_i - 1)C_b + u_i p\right) + R_i^c \frac{\beta_i \sum\limits_{j\neq i} \beta_j F_j}{\left(\sum\limits_{j=1}^{N} \beta_j F_j\right)^2}, \tag{7}$$

$$\frac{\partial^2 U_i}{\partial^2 F_i} = -2R_i^c \frac{\beta_i^2 \sum_{j\neq i} \beta_j F_j}{\left(\sum_{j=1}^{N} \beta_j F_j\right)^3} < 0. \tag{8}$$

Therefore, U_i is proved to be a strictly concave function with respect to F_i. As such, the Nash equilibrium of the game on the AI consumers side exists [32]. □

Theorem 2. *The profit function of the consumer u_i has the fixed point, and the NE of AI consumers definitely is the fixed point of the profit function.*

Proof. According to Lemma 1 and the similar proofs in [33], [34], we could arrive this theorem. □

Then, under certain conditions, the uniqueness of NE is proved by leveraging the standard function method [32]. We then introduce the definition of the standard function as follows.

Definition 2. *If a general function $f(x)$ satisfies the following conditions, it could be defined as a standard function: (1) Positivity: $\forall x \in X, f(x) > 0$; (2) Monotonicity: $\forall x_1, x_2 \in X, x_1 < x_2, f(x_1) < f(x_2)$; (3) Scalability: $\forall \rho > 1, x \in X, f(\rho x) < \rho f(x)$.*

As we proved in Theorems 1 and 2, there is at least one NE, which can be shown to be a fixed point. Then we get

$$(F^*) = \left(f(F_1^*), f(F_2^*), \cdots, f(F_N^*)\right), \tag{9}$$

where $f(F_i)$ is the purchase strategy function for the consumer i.
Let $\frac{\partial U_i}{\partial F_i} = 0$, and we have

$$\sum_{j=1}^{N} \beta_j F_j = \sqrt{\frac{R_i^c \beta_i \sum_{j\neq i} \beta_j F_j}{pu_i + (\beta_i - 1)C_b}}, \tag{10}$$

while we know that

$$\sum_{j=1}^{N} \beta_j F_j = \sum_{j\neq i} \beta_j F_j + \beta_i F_i. \tag{11}$$

Therefore, the strategy function satisfies the following equality by substituting (11) into (10):

$$F_i = \sqrt{\frac{R_i^c \sum_{j\neq i} \beta_j F_j}{\beta_i \left(pu_i + (\beta_i - 1)C_b\right)}} - \frac{1}{\beta_i} \sum_{j\neq i} \beta_j F_j. \tag{12}$$

Overall, the strategy function is shown as follows:

$$F_i^* = f(F_i) = \begin{cases} F_{min}, & F_i < F_{min}, \\ F_i, & F_{min} \le F_i \le F_{max}, \\ F_{max}, & F_i > F_{max}. \end{cases} \tag{13}$$

In this paper, we mainly focus on the second case. Then, the Theorem 3 proves the uniqueness of NE in the consumer's game.

Theorem 3. *When the role-playing ratio of each consumer is same, i.e., β, there must be only one NE in consumer's game given the following condition*

$$\frac{2(N-1)pu_i + (\beta - 1)C_b}{R_i^c} < \sum_{j=1}^{N} \frac{pu_j + (\beta - 1)C_b}{R_j^c} \tag{14}$$

is satisfied.

Proof. (1) When the role-playing ratio of each consumer is same, it's easy to obtain that $\sum_{j\neq i} F_j < \frac{R_i^c}{4(pu_i + (\beta-1)C_b)} < \frac{R_i^c}{(pu_i + (\beta-1)C_b)}$ on the basis of the condition (14) and Lemma 2, then we get

$$\sum_{j\neq i} F_j < \sqrt{\frac{R_i^c \sum_{j\neq i} F_j}{(pu_i + (\beta - 1)C_b)}}. \tag{15}$$

Therefore, we can prove $f(F_i) > 0, \forall F_i \in F$.
(2) Let $\bar{F}, F^{'} \in F$ and $\bar{F} < F^{'}$, we can obtain the following conclusion by substituting it into (12),

$$\begin{aligned} & f(\bar{F}) - f(F^{'}) = \\ & \left(\sqrt{\sum_{j\neq i} \bar{F}_j} - \sqrt{\sum_{j\neq i} F_j^{'}}\right) \sqrt{\frac{R_i^c}{pu_i + (\beta - 1)C_b}} - \\ & \left(\sqrt{\sum_{j\neq i} \bar{F}_j} - \sqrt{\sum_{j\neq i} F_j^{'}}\right) \left(\sqrt{\sum_{j\neq i} \bar{F}_j} + \sqrt{\sum_{j\neq i} F_j^{'}}\right). \end{aligned} \tag{16}$$

As $\bar{F} < F^{'}$, then $\sqrt{\sum_{j\neq i} \bar{F}_j} - \sqrt{\sum_{j\neq i} F_j^{'}} < 0$. Moreover,

$$\begin{aligned} & \sqrt{\frac{R_i^c}{pu_i + (\beta - 1)C_b}} - \sqrt{\sum_{j\neq i} \bar{F}_j} - \sqrt{\sum_{j\neq i} F_j^{'}} \in \\ & (\sqrt{\frac{R_i^c}{pu_i + (\beta - 1)C_b}} - 2\sqrt{\sum_{j\neq i} \bar{F}_j}, \\ & \sqrt{\frac{R_i^c}{pu_i + (\beta - 1)C_b}} - 2\sqrt{\sum_{j\neq i} F_j^{'}}). \end{aligned}$$

Based on the Lemma 2, we have $f(\bar{F}) < f(F^{'})$. (3) Consider $\forall \rho > 1$, it can be proved that

$$\begin{aligned} & \rho f(F_i) - f(\rho F_i) = \\ & (\rho - \sqrt{\rho}) \sqrt{\frac{R_i^c \sum_{j\neq i} F_j}{pu_i + (\beta - 1)C_b}} > 0. \end{aligned} \tag{17}$$

Hence, we can verify (13) is the standard function. According to [32], it can be concluded that there is definitely one NE in the strategy space set. □

Specifically, the unique NE is given by the following.

Theorem 4. *If each AI consumer i chooses the same role-playing ratio β and the condition (14) holds, the unique NE in the AI* *consumer's game is provided as follows:*

$$F_i^* = \frac{N-1}{\sum_{j=1}^{N}\left(\frac{pu_j+(\beta-1)C_b}{R_j^c}\right)} - \left(\frac{N-1}{\sum_{j=1}^{N}\left(\frac{pu_j+(\beta-1)C_b}{R_j^c}\right)}\right)^2 \cdot \frac{pu_i+(\beta-1)C_b}{R_i^c}. \tag{18}$$

Proof. Due to the equal role-playing ratio of each AI consumer, from (12), we have

$$\frac{\sum_{j\neq i} F_j}{\left(\sum_{i=1}^{N} F_i\right)^2} = \frac{pu_i+(\beta-1)C_b}{R_i^c}, \tag{19}$$

Specifically, we sum the two sides of (19) separately, and obtaining the left part of (19)

$$\begin{aligned}\frac{\sum_{i=1}^{N}\sum_{j\neq i} F_i}{\left(\sum_{j=1}^{N} F_j\right)^2} &= \frac{\overbrace{\sum_{j\neq 1} F_j + \sum_{j\neq 2} F_j + \cdots + \sum_{j\neq N} F_j}^{N}}{\left(\sum_{j=1}^{N} F_j\right)^2} \\ &= (N-1)\frac{\sum_{j=1}^{N} F_j}{\left(\sum_{j=1}^{N} F_j\right)^2}.\end{aligned} \tag{20}$$

Then we have

$$(N-1)\frac{\sum_{j=1}^{N} F_j}{\left(\sum_{j=1}^{N} F_j\right)^2} = \sum_{j=1}^{N}\left(\frac{pu_j+(\beta-1)C_b}{R_j^c}\right). \tag{21}$$

Next, it can be obtained that

$$\sum_{j=1}^{N} F_j = \frac{N-1}{\sum_{j=1}^{N}\left(\frac{pu_j+(\beta-1)C_b}{R_j^c}\right)}. \tag{22}$$

After simplifying and applying (10) into (22), we have

$$\sqrt{\frac{R_i^c \sum_{j\neq i} F_j}{pu_i+(\beta-1)C_b}} = \frac{N-1}{\sum_{j=1}^{N}\left(\frac{pu_j+(\beta-1)C_b}{R_j^c}\right)}. \tag{23}$$

By (22) and (23), we can derive (18) and the proof is completed. □

Lemma 2. *If the AI consumer has the same role-playing ratio β, while the condition (14) is ensured, then*

$$\sum_{j\neq i} F_j < \frac{R_i^c}{4\left(pu_i+(\beta-1)C_b\right).} \tag{24}$$

Proof. From the (18) and (22), we can obtain that

$$\sum_{j\neq i} F_j = \left(\frac{N-1}{\sum_{j=1}^{N}\left(\frac{pu_j+(\beta-1)C_b}{R^c}\right)}\right)^2 \cdot \frac{pu_i+(\beta-1)C_b}{R_i^c}. \tag{25}$$

By substituting (14) into (25), we can easily get (24). □

4.4.2 CPP' Profit Optimization

In the AI consumers' stage, the pricing strategy of CPP related to the computing-power purchase amount F_i. After substituting (18) into (4), we can arrive

$$U_{cpp} = \frac{(p-C)(N-1)}{\sum_{j=1}^{N}\left(\frac{pu_j+(\beta_j-1)C_b}{R_j^c}\right)}. \tag{26}$$

Correspondingly, we analyze the profit maximization of the CPP under the scenario that the AI consumer achieves the optimal purchase strategy.

Theorem 5. *The CPP can achieve profit maximization under the unique optimal price.*

Proof. From (26), we can obtain the first order and second order derivatives of profit U_{cpp} with respect to computing-power unit price p, i.e.

$$\frac{\partial U_{cpp}}{\partial p} = \frac{(N-1)\sum_{j=1}^{N} R_j^c\left(\sum_{j=1}^{N}(\beta_j-1)C_b + C\sum_{j=1}^{N} u_j\right)}{\left(p\sum_{j=1}^{N} u_j + \sum_{j=1}^{N}(\beta_j-1)C_b\right)^2}, \tag{27}$$

and

$$\frac{\partial^2 U_{cpp}}{\partial^2 p} = -2\sum_{j=1}^{N} u_j \frac{(N-1)\sum_{j=1}^{N} R_j^c\left(\sum_{j=1}^{N}(\beta_j-1)C_b + C\sum_{j=1}^{N} u_j\right)}{\left(p\sum_{j=1}^{N} u_j + \sum_{j=1}^{N}(\beta_j-1)C_b\right)^3}. \tag{28}$$

Due to the negativity of (28), we can prove that U_{cpp} is strictly concave function with respect to p. Thus, the CPP can maximize profit with the unique optimal price. The conclusion arrives. □

As such, there is definitely a p^* enabling CPP to acquire the optimal profits. That is, both AI consumers and CPP can achieve optimal profits, which is essentially the AI-Bazaar equilibrium point of the game.

5 Profit-Balanced Multi-agent Reinforcement Learning Algorithm Based Computing-power Trading

In this section, we develop the PB-MARL algorithm for learning optimal strategies and searching the AI-Bazaar equilibrium. Concretely, we introduce the multi-agent

Algorithm 1. The PB-MARL Algorithm for the CPP.

Input: α_{cpp}, γ_{cpp}, δ_{cpp}^{win}, δ_{cpp}^{lose}.

Initialization: $t=1$, $Q_{cpp}(s_{cpp}^t, p^t)=0$, $\pi_{cpp}(s_{cpp}^t, p^t)=\frac{1}{|\mathcal{A}_{cpp}|}$, $\bar{\pi}_{cpp}(s_{cpp}^t, p^t)=\frac{1}{|\mathcal{A}_{cpp}|}$, $\delta_{cpp}^{win}<\delta_{cpp}^{lose}$, $C(s_{cpp}^t)=0$.

for $t=1,2,3,\cdots$

1. Observe the state s_{cpp}^t.
2. Select action p^t at the probability policy $\pi_{cpp}(s_{cpp}^t, p^t)$.
3. Observe the next reward R_{cpp} and the state s_{cpp}^{t+1}.
4. Update $Q_{cpp}(s_{cpp}^t, p^t)$:
   $Q_{cpp}(s_{cpp}^t, p^t) \leftarrow (1-\alpha_{cpp})Q_{cpp}(s_{cpp}^t, p^t)+\alpha_{cpp}\cdot$
   $(R_{cpp}+\gamma_{cpp}\max_{p\in\mathcal{A}_{cpp}} Q_{cpp}(s_{cpp}^{t+1}, p))$.
5. Update average policy $\bar{\pi}_{cpp}(s_{cpp}^t, p)$:
   $C(s_{cpp}^t)=C(s_{cpp}^t)+1$
   $\bar{\pi}_{cpp}(s_{cpp}^t, p) \leftarrow \bar{\pi}_{cpp}(s_{cpp}^t, p)+\frac{1}{C(s_{cpp}^t)}(\pi_{cpp}(s_{cpp}^t, p)$
   $-\bar{\pi}_{cpp}(s_{cpp}^t, p)), \quad \forall p \in A_{cpp}$.
6. Update current strategy $\pi_{cpp}(s_{cpp}^t, p)$:
   $\pi_{cpp}(s_{cpp}^t, p) \leftarrow \pi_{cpp}(s_{cpp}^t, p)+\Gamma_{s_{cpp}^t, p}, \quad \forall p \in A_{cpp}$.

end for

until

model of AI-Bazaar in section 5.1, and introduce the PB-MARL algorithm in section 5.2.

5.1 Profit-Balanced Multi-Agent System Model

The RL framework enables players to learn the optimal policy by trial and error leveraging feedback from their own actions and experiences in an interactive environment. In this paper, let $s_u^t=p^t$ and $s_{cpp}^t=[F_i^{t-1}]_{i\in N}$ signify the state of AI consumers and CPP, respectively. Consider $F^t\in\mathcal{A}_u$ and $p^t\in\mathcal{A}_{cpp}$ indicate the submitted computing-power purchase action of AI consumers and the computing-power unit price policy determined by CPP, respectively. $\mathcal{A}_u$ expresses the AI consumer's action space, and $\mathcal{A}_{cpp}$ denotes the CPP action space.

At the beginning of the time slot t, the CPP sets the uniform computing-power unit price p^t based on the state $F_i^t=[F_i^{t-1}]_{i\in N}$ observed from underlying game model, and F_i^{t-1} means the submitted purchase computing-power amounts of each AI consumer in time slot $(t-1)$. According to (4), we define the Markov Decision Process (MDP) of the CPP, $s_{cpp}^t=[F_i^{t-1}]_{i\in N}$ signifies the state space, action space is denoted by $p^t\in\mathcal{A}_{cpp}$, and the reward function is shown as follows:

$$R_{cpp}=\sum_{i=1}^{N}(p^t-C)F_i^t. \tag{29}$$

For the AI consumers, after observing the computing-power unit price action of the CPP in time slot t, then each of them determines the submitted purchase action F_i^t on account of its state. Similarly, according to (5), we define the MDP of the AI consumer, where $s_u^t=p^t$ represents state space, action space is defined as $F^t\in\mathcal{A}_u$, whereas the reward function is given by:

$$R_u=F^t((1-\beta_i)C_b-u_ip^t)+m_i(R+\lambda B)\alpha_i e^{-\varepsilon\tau\delta_i B}. \tag{30}$$

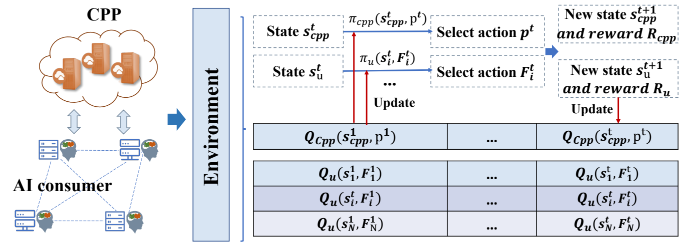


Fig. 5. The architecture of the PB-MARL algorithm.

5.2 Profit-Balanced Multi-Agent Reinforcement Learning Algorithm

We design the PB-MARL algorithm for learning optimal policies in the non-stationary environment constructed by multi-agents. This algorithm is the extension of Q-learning [35], and it can avoid suffering the seriously oscillatory problem compared to the single-agent RL. Concretely, the PB-MARL algorithm selects the suitable learning parameter δ_{cpp}^{win} and δ_{cpp}^{lose} for updating the learning rate δ_{cpp} according to the WoLF principle. And the architecture of the PB-MARL algorithm is shown in Fig. 5.

Algorithm 1 shows the details of the PB-MARL algorithm for the CPP in the multiagent system. Specifically, $\alpha_{cpp}\in(0,1]$ represents the learning rate, and $\gamma_{cpp}\in(0,1]$ denotes the discount factor. At the beginning of the algorithm, some parameters need to be initialized. By observing the relevant state, action, and reward, the Q-function of the CPP with the computing-power unit price p can be formulated by $Q_{cpp}(s_{cpp}^t, p^t)$. Similar to Q-learning, the updating rule of the Q-table can be given in step 4. Then, an average policy $\bar{\pi}_{cpp}(s_{cpp}^t, p)$ is introduced to determine the 'win' or 'failure' of the current strategy. It can be updated by step 5, where $C(s_{cpp}^t)$ represents the occurrences count of the state.

The CPP then can update its submitted price policy, interacting with the environment and other agents to maximize the cumulative reward. And the update of the current price strategy $\pi_{cpp}(s_{cpp}^t, p)$ is given by step 6, where

$$\Gamma_{s_{cpp}^t, p}=\begin{cases} -\min\left(\pi_{cpp}(s_{cpp}^t, p), \frac{\delta_{cpp}}{|\mathcal{A}_{cpp}|-1}\right), & \Omega, \\ \sum\limits_{p'\neq p}\min\left(\pi_{cpp}(s_{cpp}^t, p'), \frac{\delta_{cpp}}{|\mathcal{A}_{cpp}|-1}\right), & \text{otherwise.} \end{cases}$$

Furthermore, we define $\Omega : p \neq \arg\max_{p'\in\mathcal{A}_{cpp}} Q_{cpp}(s_{cpp}^t, p')$, $|\ \mathcal{A}_{cpp}\ |$ is the size of the CPP's action set, and the δ_{cpp} is given by:

$$\delta_{cpp}=\begin{cases} \delta_{cpp}^{win}, & \Omega', \\ \delta_{cpp}^{lose}, & \text{otherwise,} \end{cases}$$

where Ω' represents the case that the current value is greater than the average value, i.e.,

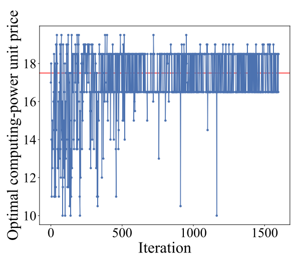
$$\begin{aligned} &\sum_{p\in\mathcal{A}_{cpp}} \pi_{cpp}(s_{cpp}^t, p)Q_{cpp}(s_{cpp}^t, p) \geq \\ &\sum_{\bar{p}\in\mathcal{A}_{cpp}} \bar{\pi}_{cpp}\left(s_{cpp}^t, \bar{p}\right)Q_{cpp}\left(s_{cpp}^t, \bar{p}\right). \end{aligned} \tag{31}$$

As for the complexity of the PB-MARL, the agent updates its strategy according to step 4 in Algorithm 1. Therefore,

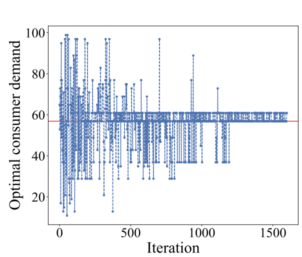
This article has been accepted for publication in IEEE Transactions on Cloud Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCC.2022.3201544

TABLE 1
Parameter values

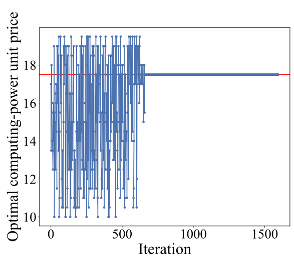
Parameter	Value	Description
N	10	Number of AI consumers
ε	20	Training Parameter
τ	0.01	Propagation factor
δ_i	0.8	Evaluation metric
C	5	Electricity cost
C_b	220	AI service profits
u_i	0.5	Weight factor
m_i	1	Token effect parameter
R	20000	Mining reward
B	500	Block size
λ	2.5	Performance reward factor
β_i	0.2	Role-playing ratio of consumer i
$\beta_{\mathcal{N}\backslash\{i\}}$	0.1	Role-playing ratio of $\mathcal{N}\backslash\{i\}$



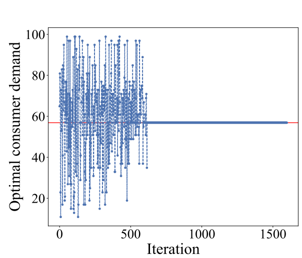
(a) P^* in PGA-APP



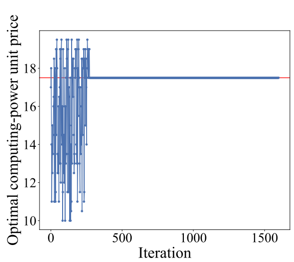
(b) F_i^* in PGA-APP



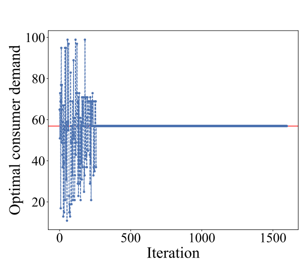
(c) P^* in MiniMax Q-learning



(d) F_i^* in MiniMax Q-learning



(e) P^* in PB-MARL



(f) F_i^* in PB-MARL

Fig. 6. Convergence performance of distinct algorithms.

the complexity of the agent is $O(S^2 \times \mathcal{A})$. Specifically, S denotes the element number in the state space set, while $\mathcal{A}$ is the number in the action space set. Additionally, we can obtain the submitted price policy of the AI consumers with the PB-MARL algorithm similar to Algorithm 1, and we will not elaborate here due to the page limit.

6 Simulation Results and Discussions

Some extensive experiments are conducted to evaluate the performance of the computing-power trading framework in

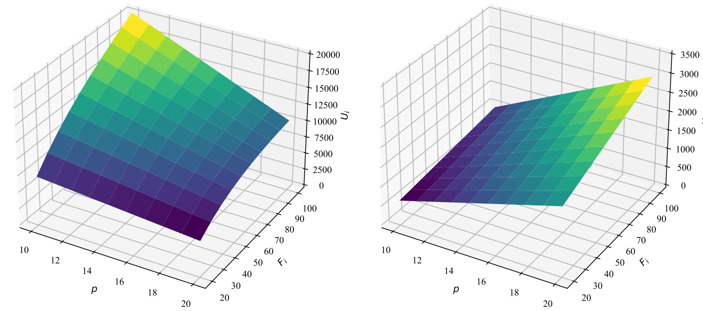


Fig. 7. The profits model of the AI consumer and CPP.

this section. Set the action space of AI consumers and CPP be $F_i \in [10, 100]$, and $P \in [10, 20]$, respectively. The given learning rate $\alpha = 0.8$, the discount factor $\gamma = 0.1$, whereas the learning parameters $\delta_{cpp}^{win} = 0.0025$ and $\delta_{cpp}^{lose} = 0.01$. Similar with [24], the other default parameters are set as Table 1, where $\mathcal{N}\backslash\{i\}$ is the other AI consumers set after removing the consumer i. The proposed framework was then implemented and tested in Python 3.7.

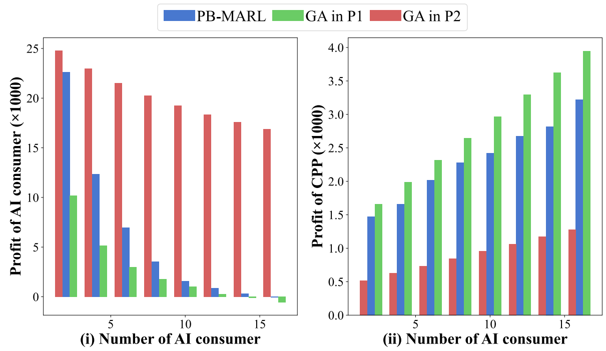
Firstly, we apply the PB-MARL algorithm for searching the AI-Bazaar equilibrium in the non-stationary environment constructed by multiple members. Fig. 6 demonstrates the convergence performance of three algorithms: PGA-APP that only needs to observe the local reward of the agent when selecting a specific action [36], MiniMax Q-learning [37] that leverages a "minimax" operator evaluated by solving a linear program to replace the "max" operator in the update step of a standard Q-learning algorithm, and PB-MARL algorithm. In Fig. 6, the red line represents the theoretically optimal solution. From Fig. 6, it can be observed that the PGA-APP algorithm cannot converge and the performance is poor. whereas, the MiniMax Q-learning and PB-MARL algorithms can converge to the AI-Bazaar equilibrium point approximately. Compared with the MiniMax Q-learning, the PB-MARL converges faster obviously due to the WoLF principle.

To get a more detailed view of the AI-Bazaar, we plot the profit models of the computing-power members shown in Fig.7. Regardless of the strategies of other computing-power members, the member in each unilateral model has its own optimal behavior. Correspondingly, problem P2 only considers the optimization of consumer's profit, and its profit model is shown in the left part of Fig.7. Meanwhile, problem P1 is to maximize the profit of the CPP, and its profit model is shown in the right part of Fig.7.

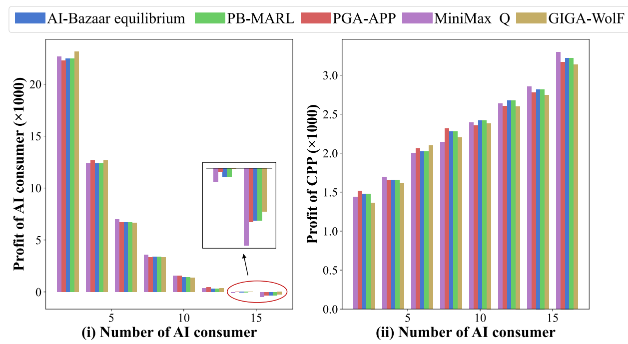
Here, we examine the profit-balance performance in AI-Bazaar. Specifically, we leverage the red bar chart to denote the AI consumer's profit calculated by addressing problem P2 through the Genetic Algorithm (GA), while the green bar chart represents the solutions by solving problem P1 through GA. Furthermore, the blue chart represents the profit obtained by addressing the Stackelberg game through the PB-MARL algorithm. From Fig. 8(a), it can be observed that the PB-MARL algorithm could balance the profit of the consumer and CPP while benefiting both of them.

Additionally, we compare the profits of computing-power members between four algorithms, i.e., PGA-APP,

This article has been accepted for publication in IEEE Transactions on Cloud Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCC.2022.3201544



(a) Profits comparison with different optimization problems.



(b) Profits comparison with different algorithms.

Fig. 8. Profits performance comparison with different baselines.

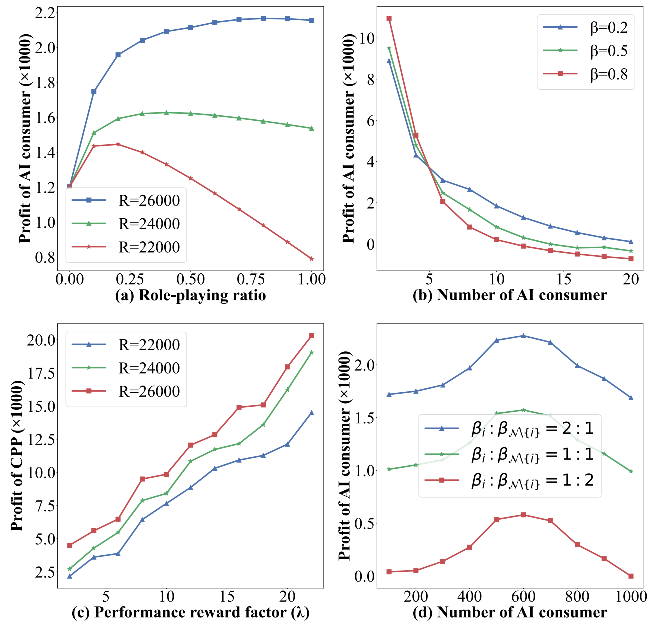


Fig. 9. Comparison of both sides profits with various blockchain factors.

MiniMax Q-learning, GIGA-WOLF [38], PB-MARL. From Fig. 8(b), we can see that the PB-MARL algorithm is closest to the AI-bazaar equilibrium compared with other baselines. Further, the AI consumer's utility may give rise to poor performance as the number of AI consumers increases. That is because as more AI consumers participate in AI-Bazaar to compete for computing resources, they are less likely to successfully solve PoI puzzles and receive rewards, further yielding poor performance. However, the CPP would gain more utility because more AI consumers will purchase computing power to support their tasks.

In our proposal, the AI consumer can play multiple roles simultaneously. In fact, it makes sense to consider the impact of the role-playing ratio on the AI consumer's profit. As shown in Fig. 9 , we compare the profit of one consumer in different block rewards. From Fig. 9 (a), for the small role-playing ratio, as blockchain mining may be more rewarding, the AI consumer's profit shows an upward tendency. However, with an increase of β_i, the AI consumer's profit decreases. That is due to devoting most of the computing power to mining may cause a high risk of revenue loss. Furthermore, the lower the block reward, the lower the profit AI consumers receive, and the more likely they fall into the mining risk group when the AI consumer chooses a large role-playing ratio.

Fig. 9 (b) compares the AI consumer's profit in different roles with the growing number of AI consumers. As shown in Fig. 9 (b), for the small-scale AI consumers, consumer i could gain the mining reward readily when it rather wants to act the miner role. In this scenario, the larger β is, the greater reward from blockchain consumer will receive. As more AI consumers enter AI-Bazaar to contend for computing power, they are less likely to successfully obtain the PoL puzzle solution, while taking more mining risks. Hence, with a large number of AI consumers, the larger β_i gets, the more the consumer's return from the blockchain slowly diminishes, and then there is a crossover point, after which it becomes more unprofitable compared to the other consumer who tends to mine with less computing power. To cope with the large-scale user situation, we can set a higher reward for stimulating more users to participate in blockchain mining.

Fig. 9 (c) demonstrates the impact of the performance reward factor λ. From Fig. 9 (c), the profit of the CPP would rise with the growth of λ. This is due to the fact that the higher performance reward factor will stimulate more AI consumers to purchase computing power for self-interest. Similarly, it is found that an increase in block reward brings about additional profit for the CPP as well. This is because the fixed block reward motivates the purchase demands of the AI consumers, and further boosts the gain of the CPP.

Furthermore, we assume AI-Bazaar has manifold roles and examine the impact of the transaction block size on the profit of the follower in this scenario in Fig. 9 (d). These three lines signify the profit of the AI consumer i in distinct ratio between β_i and $\beta_{\mathcal{N}\setminus\{i\}}$. From Fig. 9 (d), it can be observed that as the size of the block increases, the profit of the consumer first ascents, then decreases. This is due to the fact that when the block records fewer transactions, the increment of the performance reward is greater than the decrement of the block reward. After reaching the maximum, the degree of difficulty that consumer i obtains block reward from blockchain creeps up, and the decrease in mining incentives will be even more pronounced. Therefore, the AI consumer's profit appears to be a downtrend trend. Additionally, the greater computing power that AI consumer devotes to mining, the better revenue it will get.

7 Conclusions and Future Work

With the assistance of blockchain, this paper proposes a computing-power trading framework for ubiquitous AI services, also known as AI-Bazaar. In AI-Bazaar, AI consumers have the flexibility to switch roles between AI services and miners, while feeling free to contribute the computing power rented from the CPP for personalized demands. Accordingly, a two-stage Stackelberg game between CPP and AI consumers is formulated. Further, we develop the PB-MARL algorithm based on WoLF to find the AI-Bazaar equilibrium for better learning-based services in the resource-constrained computing environment. The experimental results further emphasize that AI consumers and CPP can make satisfying decisions and gain benefits. In the future, we will continue to improve the real framework implementation, while designing smart contracts to ensure the security and transparency of the transaction process.

8 Acknowledgement

This work was supported in part by the National Key R & D Program of China under Grant 2019YFB2101901, in part by the National Natural Science Foundation of China under Grant 62072332, Grant 62002260 and Grant 61872310, in part by the China Postdoctoral Science Foundation under Grant 2020M670654, in part by the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) under Grant GML-KF-22-03, in part by Key-Area Research and Development Program of Guangdong Province under Grant No. 2021B0101400003, in part by Hong Kong RGC Research Impact Fund under Grant No. R5060-19, General Research Fund under Grants No. 152221/19E, 152203/20E, and 152244/21E, in part by Shenzhen Science and Technology Innovation Commission under Grant JCYJ20200109142008673, in part by NSF CNS-2107216 and CNS-2128368.

References

[1] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, "6G wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 28–41, 2019.

[2] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: on-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 1, pp. 447–457, 2020.

[3] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. Chen, and L. Hanzo, "Thirty years of machine learning: The road to pareto-optimal next-generation wireless networks," *CoRR*, vol. abs/1902.01946, 2019.

[4] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, 2019.

[5] Y. Shih, W. Chung, A. Pang, T. Chiu, and H. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE Netw.*, vol. 31, no. 1, pp. 52–58, 2017.

[6] Y. Gong, H. Yao, J. Wang, M. Li, and S. Guo, "Edge intelligence-driven joint offloading and resource allocation for future 6G industrial internet of things," *IEEE Trans. Netw. Sci. Eng. (Early Access)*, 2022.

[7] G. Shao, X. Chen, X. Zeng, and L. Wang, "Deep learning hierarchical representation from heterogeneous flow-level communication data," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1525–1540, 2020.

[8] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[9] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[10] F. Wang, H. Yao, Q. Zhang, J. Wang, R. Gao, D. Guo, and M. Guizani, "Dynamic distributed multi-path aided load balancing for optical data center networks," *IEEE Transactions on Network and Service Management (Early Access)*, 2021.

[11] T. Mai, H. Yao, N. Zhang, L. Xu, M. Guizani, and S. Guo, "Cloud mining pool aided blockchain-enabled internet of things: An evolutionary game approach," *IEEE Trans. Cloud Comput. (Early Access)*, 2021.

[12] F. Li, H. Yao, J. Du, C. Jiang, Z. Han, and Y. Liu, "Auction design for edge computation offloading in SDN-based ultra dense networks," *IEEE Trans. Mob. Comput.*, vol. 21, no. 5, pp. 1580–1595, 2022.

[13] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, 2019.

[14] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 33–39, 2018.

[15] J. Li, T. Liu, D. Niyato, P. Wang, J. Li, and Z. Han, "Contract-theoretic pricing for security deposits in sharded blockchain with internet of things (iot)," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 10 052–10 070, 2021.

[16] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Blockchain and machine learning for communications and networking systems," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 1392–1431, 2020.

[17] C. Qiu, H. Yao, X. Wang, N. Zhang, F. R. Yu, and D. Niyato, "AI-Chain: Blockchain energized edge intelligence for beyond 5g networks," *IEEE Netw.*, vol. 34, no. 6, pp. 62–69, 2020.

[18] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Trans. Ind. Informatics*, vol. 16, no. 6, pp. 4177–4186, 2020.

[19] C. Chenli, B. Li, Y. Shi, and T. Jung, "Energy-recycling blockchain with proof-of-deep-learning," in *IEEE International Conference on Blockchain and Cryptocurrency*, May 2019, pp. 19–23.

[20] X. Wang, X. Ren, C. Qiu, Y. Cao, T. Taleb, and V. C. M. Leung, "Net-in-ai: A computing-power networking framework with adaptability, flexibility, and profitability for ubiquitous AI," *IEEE Netw.*, vol. 35, no. 1, pp. 280–288, 2021.

[21] M. Król, S. Mastorakis, D. Oran, and D. Kutscher, "Compute first networking: Distributed computing meets ICN," in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, Sep. 2019, pp. 67–77.

[22] X. Tang, C. Cao, Y. Wang, S. Zhang, Y. Liu, M. Li, and T. He, "Computing power network: The architecture of convergence of computing and networking towards 6G requirement," *China Communications*, vol. 18, no. 2, pp. 175–185, 2021.

[23] J. Lee, A. Mtibaa, and S. Mastorakis, "A case for compute reuse in future edge systems: An empirical study," in *2019 IEEE Globecom Workshops*, Dec. 2019, pp. 1–6.

[24] Z. Chang, W. Guo, X. Guo, Z. Zhou, and T. Ristaniemi, "Incentive mechanism for edge-computing-based blockchain," *IEEE Trans. Ind. Informatics*, vol. 16, no. 11, pp. 7105–7114, 2020.

[25] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4585–4600, 2019.

[26] A. Asheralieva and D. Niyato, "Distributed dynamic resource management and pricing in the iot systems with blockchain-as-a-service and uav-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1974–1993, 2020.

[27] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial internet of things," *IEEE Trans. Ind. Informatics*, vol. 15, no. 6, pp. 3602–3609, 2019.

[28] B. Xiao, X. Fan, S. Gao, and W. Cai, "Edgetoll: A blockchain-based toll collection system for public sharing of heterogeneous edges," *CoRR*, vol. abs/1912.12681, 2019. [Online]. Available: http://arxiv.org/abs/1912.12681

[29] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Trans. Parallel Distributed Syst.*, vol. 30, no. 9, pp. 1975–1989, 2019.

[30] Z. Xiong, S. Feng, D. Niyato, P. Wang, and Z. Han, “Optimal pricing-based edge computing resource management in mobile blockchain,” in *2018 IEEE International Conference on Communications*, May 2018, pp. 1–6.

[31] T. Bahreini, H. Badri, and D. Grosu, “An envy-free auction mechanism for resource allocation in edge computing systems,” in *2018 IEEE/ACM Symposium on Edge Computing, Oct. 2018*, 2018, pp. 313–322.

[32] Z. Han, D. Niyato, W. Saad, T. Baar, and A. Hjrungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*, 1st ed. USA: Cambridge University Press, 2012.

[33] D. Gale, “The game of hex and the brouwer fixed-point theorem,” *The American Mathematical Monthly*, vol. 86, pp. 818–827, 1979.

[34] K. Etessami and M. Yannakakis, “On the complexity of nash equilibria and other fixed points,” *SIAM J. Comput.*, vol. 39, no. 6, pp. 2531–2597, 2010.

[35] L. Jouffe, “Fuzzy inference system learning by reinforcement methods,” *IEEE Trans. Syst. Man Cybern. Part C*, vol. 28, no. 3, pp. 338–355, 1998.

[36] C. Zhang and V. R. Lesser, “Multi-agent learning with policy prediction,” in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, Jul. 2010, pp. 927–934.

[37] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in *Machine Learning, Proceedings of the Eleventh International Conference*, Jul. 1994, pp. 157–163.

[38] M. H. Bowling, “Convergence and no-regret in multiagent learning,” in *Advances in Neural Information Processing Systems, Dec. 2004*, 2004, pp. 209–216.

Xiaoxu Ren (S’19) is currently pursuing the Ph.D. degree in College of Intelligence and Computing, Tianjin University, Tianjin, China. She received the B.S. degree in College of Science from Inner Mongolia University of Technology, China in 2016. Her current research interests include machine learning, computing power networking, edge intelligence, and blockchain.

Chao Qiu (S’15, M’19) is currently a Lecturer in the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University. She received the B.S. degree from China Agricultural University in 2013 in Communication Engineering and the Ph.D. from Beijing University of Posts and Telecommunications in 2019 in Information and Communication Engineering. From September 2017 to September 2018, she visited Carleton University, Ottawa, ON, Canada, as a Visiting Scholar. Her current research interests include edge computing, edge intelligence, and blockchain.

Xiaofei Wang (S’06, M’13, SM’18) is currently a Professor with the Tianjin Key Laboratory of Advanced Networking, School of Computer Science and Technology, Tianjin University, China. He got master and doctor degrees in Seoul National University from 2006 to 2013, and was a Post-Doctoral Fellow with The University of British Columbia from 2014 to 2016. Focusing on the research of social-aware cloud computing, cooperative cell caching, and mobile traffic offloading, he has authored over 140 technical papers in the IEEE JSAC, the IEEE TWC, the IEEE WIRELESS COMMUNICATIONS, the IEEE COMMUNICATIONS, the IEEE TMM, the IEEE INFOCOM, and the IEEE SECON. In 2017, he received the “Fred W. Ellersick Prize” from the IEEE Communication Society.

Zhu Han (S’01, M’04, SM’09, F’14) 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, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han 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, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019, and ACM distinguished Member since 2019. Dr. Han is a 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of the 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: “for contributions to game theory and distributed management of autonomous communication networks.”

Ke Xu (M’02, SM’09) received the Ph.D. degree from the Department of Computer Sci- ence and Technology, Tsinghua University, Beijing, China. He currently serves as a Full Professor for Tsinghua University. He has published more than 200 technical articles and holds 11 U.S. patents in the research areas of next-generation internet, blockchain systems, the Internet of Things, and net- work security. He is a member of ACM. He served as the Steering Committee Chair for IEEE/ACM IWQoS. He is an Editor of IEEE INTERNET OF THINGS JOURNAL. He has guest edited several special issues in IEEE and Springer journals.

Haipeng Yao (M'16, SM'20) is a Professor in Beijing University of Posts and Telecommunications. Haipeng Yao received his Ph.D. in the Department of Telecommunication Engineering at University of Beijing University of Posts and Telecommunications in 2011. His research interests include future network architecture, network artificial intelligence, networking, space-terrestrial integrated network, network resource allocation and dedicated networks. He has published more than 100 papers in prestigious peer-reviewed journals and conferences. Dr. Yao has served as an Editor of IEEE Network, IEEE Access, and a Guest Editor of IEEE Open Journal of the Computer Society and Springer Journal of Network and Systems Management. He has also served as a member of the technical program committee as well as the Symposium Chair for a number of international conferences, including IWCMC 2019 Symposium Chair, ACM TUR-C SIGSAC2020 Publication Chair.

Song Guo (M'02, SM'11, F'19) received the bachelor's degree in computer software from the Huazhong University of Science and Technology, the master's degree in computer engineering from the Beijing University of Posts and Telecommunications, and the Ph.D. degree in computer science from the University of Ottawa. He is currently a Full Professor and an Associate Head with the Department of Computing, The Hong Kong Polytechnic University. Before joining PolyU, he was a Professor with the University of Aizu, Japan. His research interests are mainly in the areas of big data, cloud computing, mobile computing, and distributed systems. Dr. Guo is an IEEE Fellow (Computer Society) and the Editor-in-Chief of the IEEE OPEN JOURNAL OF THE COMPUTER SOCIETY. He was a recipient of the 2019 IEEE TCBD Best Conference Paper Award, 2018 IEEE TCGCC Best Magazine Paper Award, 2019 and 2017 IEEE Systems Journal Annual Best Paper Award, and other six Best Paper Awards from IEEE/ACM conferences. His work was also recognized by the 2016 Annual Best of Computing: Notable Books and Articles in Computing in ACM Computing Reviews. Some of his Transactions papers were selected as Featured or Spotlight papers. He was a Distinguished Lecturer of IEEE Communications Society (ComSoc) and served in the IEEE ComSoc Board of Governors. He has been named on editorial board of a number of prestigious international journals like the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING. He has also served as chairs of organizing and technical committees of many international conferences.