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

Xiaoxu Ren ,Xiaofei Wang ,Chao Qiu,Zhu Han,Ke Xu ,Haipeng Yao and Song Guo

DOI: 10.1109/TCC.2022.3201544
Published in: IEEE Transactions on Cloud Computing (Volume: 11, Issue: 3, 01 July-Sept. 2023)

Presented To

Name : Prof. Dr. Md. Abdur Razzaque
Designation : Chairman and Professor
Department : CSE

Presented by

Name: Shamim Ahmed & Rashed Hassan Siam
Roll: SH 2344 & FH 2343
MSc. CSE DU

Department of Computer Science and Engineering, University of Dhaka

Agenda

- AI-Bazaar
- Limitations of Related Works
- Key Contributions and Framework
- Problem Formulation
- Profit Maximization
- PB-MARL Algorithm
- Stackelberg Equilibrium
- Simulations and Results
- Limitations of this Work

Introduction

- **Growth of AI and IoE:**
  Rapid growth of AI applications demands high computational power, low latency, and high bandwidth.
- **Edge and Cloud Computing:**
  Edge and cloud computing technologies are key to meeting the computational needs of AI services.
- **Challenges in Current Frameworks:**
  Issues include underutilization of resources, inefficient allocation, and unbalanced profit-sharing mechanisms.
- **Need for a New Trading Framework:**
  A fair, efficient, and profit-balanced computing power trading framework is crucial for AI applications.

Limitations of Related Works

- **Underutilization of Resources:** Computational resources, particularly in cloud-edge infrastructures, are often either underutilized or incapable of meeting the rising demands of intensive AI tasks.
- **Lack of a Profit-Balanced Trading Mechanism**: Current computing frameworks often prioritize the self-interest of computing power providers (CPPs), neglecting fair profit-sharing mechanisms.
- **Inefficiency in AI Service Management:** Traditional frameworks struggle to provide accurate, personalized, and high-quality AI services under resource-constrained environments.

How AI-Bazaar Addresses These Problems

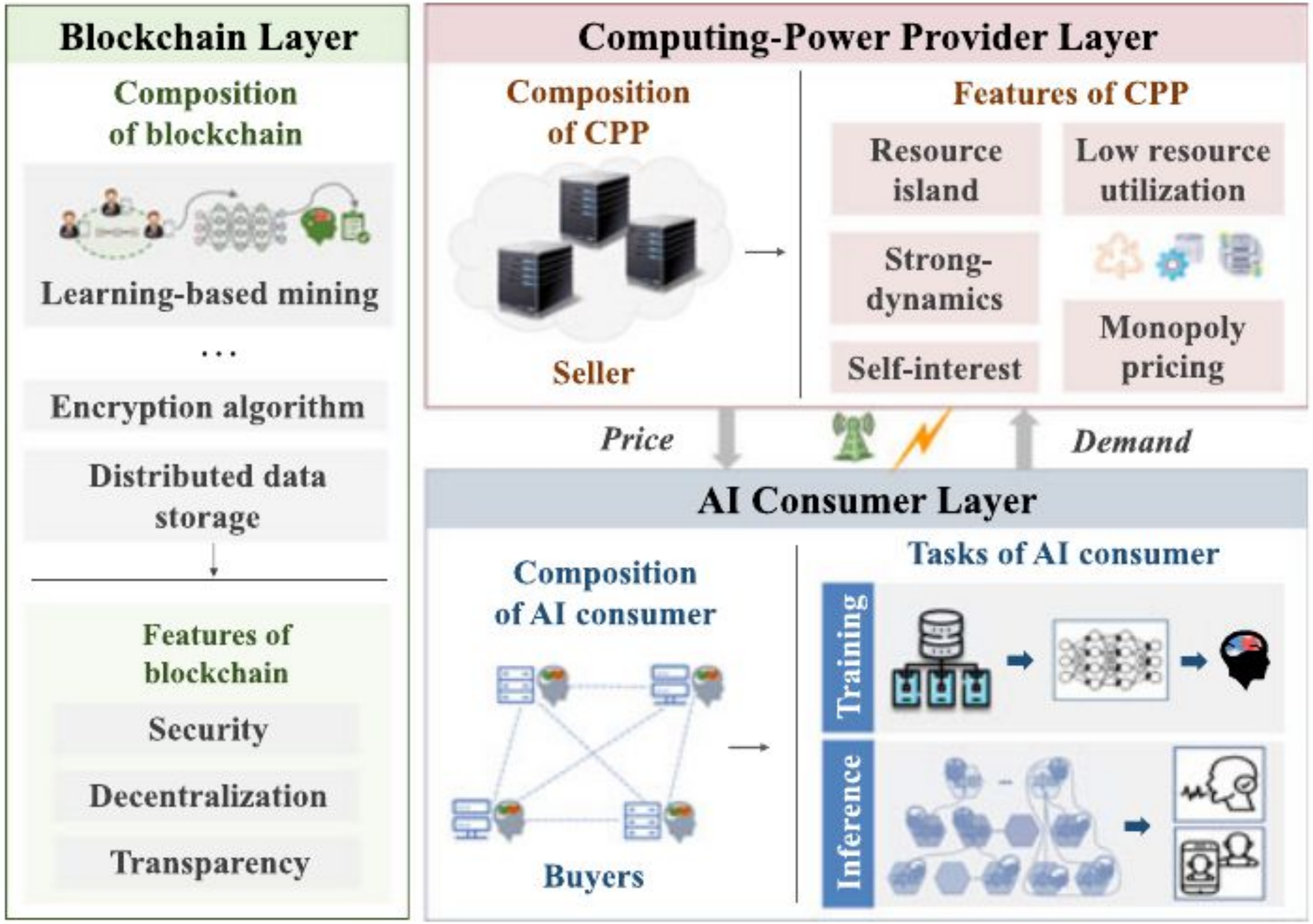
1. **Efficient Resource Utilization**: AI-Bazaar connects scattered computational resources using blockchain, optimizing their allocation across tasks.
2. **Profit-Balanced Framework**: The Stackelberg game model ensures fairness by balancing the interests of CPPs and AI consumers.
3. **Multi-Role Flexibility**: AI consumers can switch between roles (blockchain miner, AI service provider, or both), maximizing their benefits.
4. **Sustainable Blockchain Mechanism**: By employing **Proof of Learning (PoL)**, computational power is used for meaningful tasks (like training neural networks), eliminating wastage seen in PoW.

Framework Overview

AI-Bazaar is a **blockchain-based computing-power trading framework** designed to address inefficiencies in traditional cloud-edge systems. It emphasizes **resource sharing**, **profit balance**, and **multi-role functionality** for AI consumers. The framework consists of three layers:

1. **AI Consumers Layer**: Represents resource-constrained devices (e.g., edge nodes) that demand computing power for tasks like AI training and inference.
2. **Computing-Power Provider (CPP) Layer**: Integrates distributed computational resources and rents them to AI consumers.
3. **Blockchain Layer**: Ensures secure, decentralized, and transparent management of transactions and computational tasks.

Framework



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

Role-playing Ratio

$$a_i = \frac{b_i F_i}{\sum_{j=1}^{N} b_j F_j}$$

- a_i: Fraction of consumer i's computing power allocated for mining.
- b_i: Role-playing ratio for mining.
- F_i: Computing power rented by i.

Analysis:

- If b_i increases, more of F_i is allocated to mining.
- a_i depends on F_i relative to the total rented resources $\sum_{j=1}^{N} b_j F_j$.

Mining Success Probability

$$r_i = a_i e^{-\epsilon T_p^i}$$

- r_i: Probability that consumer i successfully mines a block.
- $T_p^i = td_i B$: Propagation time for block i, dependent on block size (B) and evaluation metrics (d_i).

Analysis:

- Larger a_i improves r_i, but higher T_p^i (e.g., large blocks or poor d_i) reduces it.
- ϵ adjusts for training time, incentivizing efficient blocks.

Consumer Profit from Mining

$$U_i^m = (R + \eta B)a_i e^{-\epsilon t d_i B}$$

- R: Block reward for successful mining.
- ηB: Performance reward based on block size.
- $t d_i B$: Penalizes delays in block propagation.

Insights:

- Increasing R or η incentivizes mining.
- Poor block performance (d_i) decreases profit exponentially.

CPP Profit

$$U_{CPP} = (p - C)\sum_{i=1}^{N} F_i$$

- p: Price per unit computing power.
- C: Cost per unit.

ptimization:

- Profit increases with higher p and $\sum F_i$, but excessive p reduces demand (F_i).

Consumer Profit Maximization

$$\max_{F_i} \left[F_i((1-b_i)C_b - u_i p) + m_i U_i^m\right]$$

- C_b: Business value per unit computing power.
- u_i: Weight of cost impact on consumer utility.
- m_i: Monetary value of mining rewards.

Trade-offs:

- Consumers balance C_b (business utility) against costs (p) and mining risks.

Stackelberg Equilibrium

Equilibrium ensures:

- CPP's pricing strategy maximizes its profit.
- Consumers' purchase strategies maximize their utilities.

Conditions for Equilibrium:

- Consumer utilities (Ui) must be concave.
- CPP profit (Ucpp) must have a unique maximum with respect to p.

Model Analysis(1/3)

1. AI-Bazaar Equilibrium Definition:

- The equilibrium consists of strategies Fi and P satisfying conditions that no player has an incentive to deviate from their chosen strategy, considering the opponent's decisions.

2. Game Setup:

- AI consumers form a noncooperative game based on self-interest, where they choose purchase strategies from a convex set, with the strategy space being non-empty and compact.
- The utility of each consumer is continuous in their strategy space.

3. Existence of Nash Equilibrium (NE):

- **Lemma 1** confirms that the strategy space is convex, non-empty, and compact, and the utility function is continuous.
- **Theorem 1** demonstrates that the Nash equilibrium exists due to the continuity and concavity of the utility function.

Model Analysis(2/3)

4. Fixed Point and Uniqueness of NE:

- **Theorem 2** establishes that the Nash equilibrium is a fixed point of the consumers' profit function.
- **Theorem 3** shows the uniqueness of the Nash equilibrium when consumers share the same role-playing ratio bi and a specific condition is met.
- **Theorem 4** specifies that, under these conditions, the unique equilibrium for consumers can be derived.

5. Profit Maximization:

- The conditions leading to the maximization of profits in the consumer game are analyzed. The strategy functions and the specific conditions for maximizing profit under the game model are presented.
- **Lemma 2** further asserts that under the condition Rci, the consumers' strategies must satisfy a fixed inequality, which is crucial for determining optimal strategies.

Model Analysis(3/3)

1. **Initialization** of the Q-values and the policy for each state-action pair.
2. **Action Selection** using a probability distribution derived from the current policy.
3. **Observation** of the reward and the new state after taking the action.
4. **Q-value Update** using the Bellman equation, adjusting the Q-values based on the received reward and the expected future rewards.
5. **Policy Update** based on the frequency of action selections and the changes in the Q-values.
6. **Strategy Update** to adapt the policy to new experiences.
7.

Algorithm 1. The PB-MARL Algorithm for the CPP

Input: $\alpha_{cpp}, \gamma_{cpp}, \delta_{cpp}^{win}, \delta_{cpp}^{lose}$.

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 $\bar{\pi}_{cpp}(s_{cpp}^t, p)$:

$$C(s_{cpp}^t) = C(s_{cpp}^t) + 1$$

$$\bar{\pi}_{cpp}(s_{cpp}^t, p) \leftarrow \bar{\pi}_{cpp}(s_{cpp}^t, p) + \frac{1}{C(s_{cpp}^t)}(\pi_{cpp}(s_{cpp}^t, p) - \bar{\pi}_{cpp}(s_{cpp}^t, p)), \forall p \in A_{cpp}.$$

6: Update current strategy $\pi_{cpp}(s_{cpp}^t, p)$:

$$\pi_{cpp}(s_{cpp}^t, p) \leftarrow \pi_{cpp}(s_{cpp}^t, p) + \Gamma_{s_{cpp}^t, p}, \forall p \in A_{cpp}.$$

end for

until

Convergence of Algorithms

Observation: The paper compares the proposed **Profit-Balanced Multi-Agent Reinforcement Learning (PB-MARL)** algorithm with other baseline algorithms:

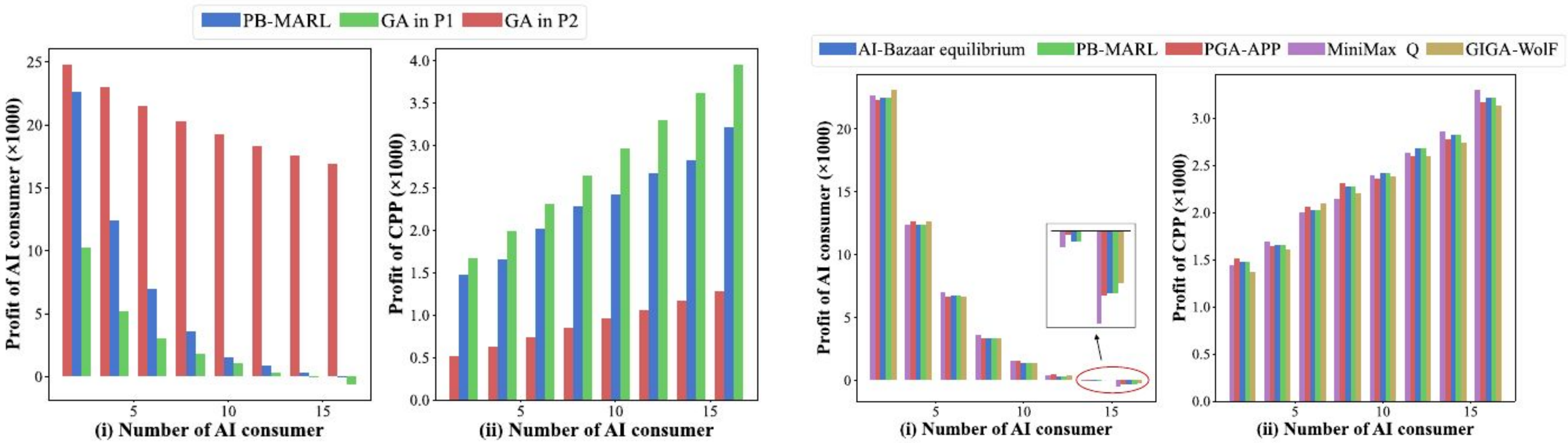
- **PGA-APP**: Only uses local agent rewards for decision-making.
- **MiniMax Q-Learning**: Employs a “minimax” operator to evaluate strategies.
- **PB-MARL**: Incorporates the WoLF (Win or Learn Fast) principle for better convergence.

Key Findings:

- PB-MARL converges faster than both PGA-APP and MiniMax Q-Learning.
- PGA-APP fails to reach equilibrium, while MiniMax Q-Learning converges slower than PB-MARL.

Reason: PB-MARL’s ability to adapt its learning rate based on success or failure accelerates convergence.

Results on Different Baselines



(a) Profits comparison with different optimization problems.

(b) Profits comparison with different algorithms.

Profit-Balance in AI-Bazaar

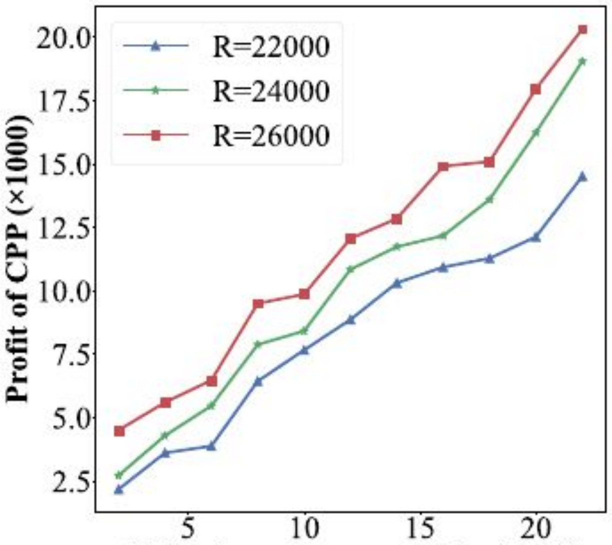
The **profit-balance mechanism** ensures that both CPPs and AI consumers benefit fairly:

- **Blue Bars**: Profits under the PB-MARL algorithm (Stackelberg equilibrium).
- **Red/Green Bars**: Profits under unilateral optimization (GA-based solutions).

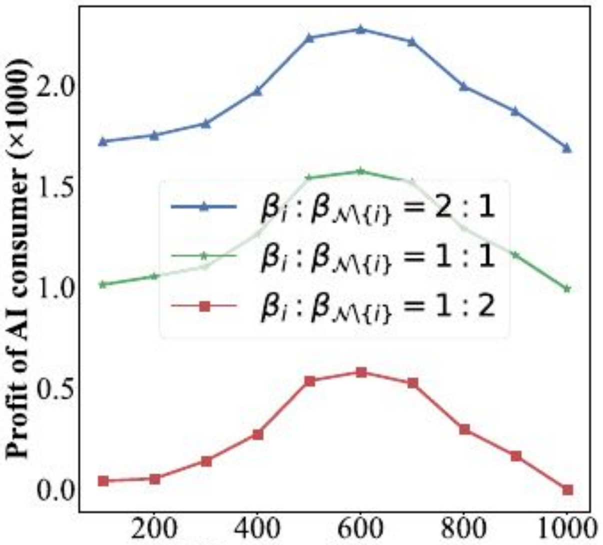
Result:

- PB-MARL achieves a balanced profit, ensuring both CPP and consumers gain reasonable benefits.
- Unilateral optimization often favors one party, making it less sustainable.

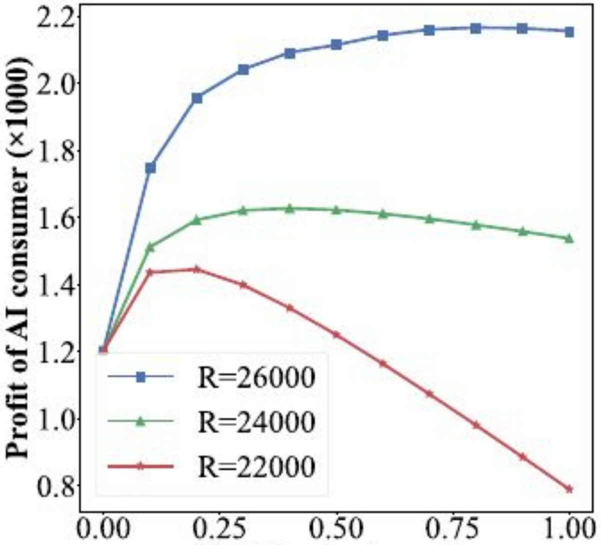
Results on Various Blockchain Factors



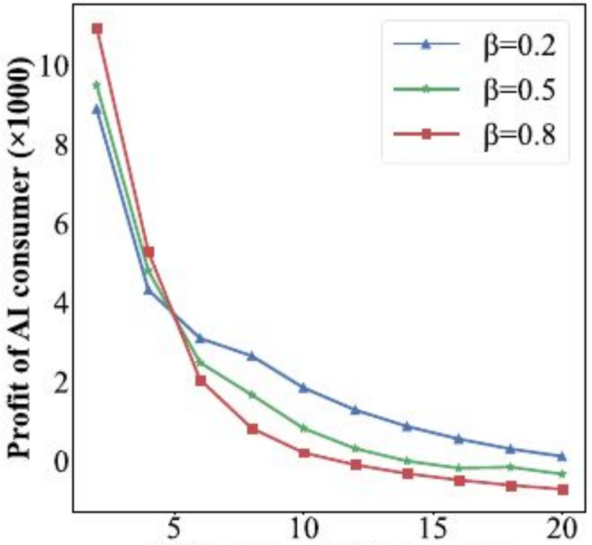
(c) Performance reward factor (λ)



(d) Number of AI consumer



(a) Role-playing ratio



(b) Number of AI consumer

Role-Playing Ratio (bi) and Consumer Profits

Impact of bi on Profits:

- When bi (mining allocation) is low, consumers gain more from AI services.
- Increasing bi boosts mining rewards but reduces AI service profits.

Critical Insight:

- For small block rewards, high bi leads to diminishing returns due to mining risks.
- For large block rewards, higher bi increases profits, but only up to a point.

Example:

- At bi=0.4, the balance between mining and AI service profits is optimal.
- At bi>0.6, mining risks outweigh benefits.

PB-MARL Algorithm

a. Block Reward:

- Higher block rewards incentivize more consumers to allocate computing power to mining.
- Increases overall profit for CPPs, as more consumers purchase computing power.

b. Performance Reward Factor (λ):

- Higher λ leads to better quality mining and encourages more purchases.
- CPP profits increase as AI consumers compete for higher rewards.

c. Block Size (B):

- Initially, increasing block size improves consumer profits (more transactions recorded).
- Beyond a certain size, mining becomes harder, reducing profits.

Advantages of the AI-Bazaar Framework

Resource Utilization:

- Efficiently allocates unused computational power for both blockchain and AI tasks.

Decentralized Control:

- Eliminates reliance on centralized intermediaries by using blockchain.

Dynamic Role Allocation:

- Consumers can switch between roles (AI training, inference, mining) dynamically, maximizing their benefits.

Fair Profit Sharing:

- Stackelberg game ensures balanced profits between CPPs and AI consumers.

Limitations

- **No real framework** is implemented.
- Assumes **uniform behavior** across AI consumers.
- The complexity of the **PB-MARL algorithm** may increase significantly with more participants, leading to **scalability concerns**.
- The use of **PoL consensus mechanism** still introduces additional computational and network overheads.
- Excludes **network bandwidth**, potentially limiting its **practical applicability**.

References

[1] Z. Zhang et al., “6G wireless networks: Vision, requirements, architecture, and key technologies,” IEEE Veh. Technol. Mag., vol. 14, no. 3, pp. 28–41, Sep. 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, Jan. 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,” IEEE Commun. Surv. Tutorials, vol. 22, no. 3, pp. 1472–1514, Jan. 2020.

[4] 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, Jan./Feb. 2017.

[5] F. Wang et al., “Dynamic distributed multi-path aided load balancing for optical data center networks,” IEEE Trans. Netw. Serv.Manage., vol. 19, no. 2, pp. 991–1005, Jun. 2022.

Any Question?

Thank You