Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Research on Price-Based Autonomous Group Robot Resource Allocation Strategy in Emergency Scenarios

Sun Yi
Xi'an University of Science and Technology
Zhang Xiao (zx@xust.edu.cn)
Xi'an University of Science and Technology

Research Article

Keywords: Game Theory, Autonomous Group Robots, Dynamic Pricing Strategy, Resource Allocation, Differential Pricing Model

Posted Date: September 11th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3302574/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Research on Price-Based Autonomous Group Robot Resource Allocation Strategy in Emergency Scenarios

Sun Yi[1,2] and Zhang Xiao[2,3*]

[1]College of Communication and information engineering, Xi'an University of Science and Technology, Xi'an, Shaanxi, 710054, China.
[2*]College of Communication and information engineering, Xi'an University of Science and Technology, Xi'an, Shaanxi, 710054, China.

*Corresponding author(s). E-mail(s): zx@xust.edu.cn;
Contributing authors: sunyi@xust.edu.cn;

Abstract： In unknown and dynamic emergency scenarios, achieving the collaboration of autonomous group robots for search and rescue operations can be regarded as resource allocation among robots at the micro-level. The resource allocation problem for autonomous group robots can be abstracted as a non-cooperative game, and in a dynamically changing environment, pricing becomes a critical factor for effective resource allocation.This paper starts from the perspectives of uniform pricing strategy and differential pricing strategy, respectively. It establishes master-slave game models for these two pricing strategies to describe resource allocation between resource providers and resource consumers. Furthermore, the paper utilizes game theory to model the competition for computational resources between resource-providing robots and resource-consuming robots, and solves for their Nash equilibrium solution, demonstrating its existence and uniqueness.Additionally, performance analysis and numerical analysis are conducted on both the uniform pricing model and the differential pricing model, thereby highlighting the advantages and disadvantages of different pricing models for dynamic adjustment of pricing strategies. Meanwhile, the differential pricing model introduces a fairness factor to enhance collaboration between robots and prevent resource accumulation. Simulation results indicate that under the same CPU cycle, the average processing time is shorter in the uniform pricing model, while in the differential pricing model, the resource-providing robots yield higher profits. Hence, the suitable pricing strategy can be chosen based on specific requirements.Simultaneously, as the CPU cycle increases, the cost for resource-consuming robots decreases, average processing time reduces, and the payment enthusiasm of resource-consuming robots increases. Therefore, the CPU cycle is related to the overall well-being of the autonomous group robot system.

Keywords: Game Theory, Autonomous Group Robots, Dynamic Pricing Strategy, Resource Allocation, Differential Pricing Model

1、Introduction

The concept of autonomous swarm robotics draws inspiration from the self-organized behaviors of social animals[1]. Its design aims to employ a multitude of simple-structured robots to accomplish complex tasks that a single robot cannot complete, all while striving for minimal cost, robust stability, and high efficiency[2-3]. As technology advances, autonomous group robots are garnering increased attention due to their potential for diverse applications in various environments. However, at the current research stage, scholars are predominantly focused on aspects such as robot design and implementation, with insufficient emphasis on the collaboration and cooperation among robots.

The issue of collaboration and cooperation among autonomous group robots is essentially a decision problem within a distributed multi-agent system[4]. Game theory, coincidentally, serves as a mathematical tool for studying and analyzing the interactions among decision-makers in distributed multi-agent systems[5]. It aids in capturing strategic interactions, incentives, and conflicts that arise in such systems. Autonomous group robots, being typical distributed systems, are characterized by moments of games between service providers and users, as well as between different users.In competitive markets, the quantity of resources produced by service providers depends on factors such as demand, costs, market conditions, and the objectives of the providers themselves[6]. Furthermore, the pricing of computational resources by service providers is also crucial in achieving their ultimate goals. Given these dynamics, game theory becomes a valuable framework for understanding the complex decision-making processes and interactions inherent in the collaboration and competition among autonomous group robots.

In this realm, numerous efforts have been directed towards optimal resource allocation for computational offloading among swarm robot devices[7]-[20]. Within these studies, optimization problems have been formulated and solved to achieve optimal resource allocation, aiming to maximize energy efficiency or minimize latency. The resources in question can encompass computational capacity, transmission power, and bandwidth, among other factors. However, most of these endeavors primarily focus on enhancing the overall system performance, which has led to various fairness issues within autonomous group robots.In response, many studies have emerged to address these fairness concerns[21]-[30]. Nonetheless, the approaches aimed at rectifying fairness problems may not accurately meet the preferences of certain robots. Some robots might strongly prefer to use a central robot, even if it comes at a higher cost (referred to as non-pricing edge computing systems). As a result, considering the system-level performance and varying expectations of different robots, pricing mechanisms can be a viable option for differentiating resource allocation strategies.

In the context of an autonomous group robot system, resource-providing robots possess a certain amount of computational resources, while resource-consuming robots require the use of computational resource blocks provided by the former. Taking into consideration the equilibrium benefits of both resource-providing robots and resource-consuming robots in an incomplete information scenario, this paper initially abstracts the competition process among resource-consuming robots for the resources provided by resource-providing robots into a leader-follower game model. In this model, the resource-providing robots act as leaders, and the resource-consuming robots act as followers. Due to issues such as the substantial computational task volume and computational costs, resource-consuming robots choose between locally offloading a portion of their computational tasks or sending them to resource-providing robots for offloading.In this leader-follower game, resource-providing robots set prices considering factors like computational task volume, CPU cycles, and task types. They aim to maximize their own revenue by selling resources to resource-consuming robots. On the other hand, resource-consuming robots optimize their offloading strategy to minimize their costs. Each resource-consuming robot acts independently without interference from others, while resource-providing robots share resource demands among themselves, achieving a form of local coordination.Unlike traditional models that separate resource allocation from pricing, this paper combines both concepts and proposes a pricing-based group robot resource allocation model. Specifically, the contributions of this work are as follows:

1) This paper represents the resource allocation problem of group robots as an economic resource purchase problem. It explicitly introduces the unified pricing model and the differential pricing model. It rigorously analyzes these two pricing models using

game theory methods.

2) The unified pricing model and the differential pricing model are defined as leader-follower games with a single leader (resource-providing robots) and multiple followers (resource-consuming robots). The paper calculates the Nash equilibrium solutions for both models and proves their existence and uniqueness.

3) In the differential pricing model, to ensure that the price difference does not exceed a certain level, the paper introduces a fairness factor. Specifically, in the differential pricing model, a more thorough analysis is conducted to determine how much fairness factor the resource provider should set to achieve higher total revenue compared to the unified pricing model. This analysis benefits the resource provider by allowing them to predict the revenue difference compared to the unified pricing model as long as the purchasers do not complain.

4) Starting from performance analysis and numerical analysis, the paper compares the advantages and disadvantages of the two pricing schemes. Through this analysis, resource-providing robots can choose an appropriate pricing strategy based on different environmental conditions.

The remaining structure of the paper is as follows.Section 2 discusses related research and work in the field to provide context for the current study.Section 3 presents the system model that forms the foundation for the subsequent analysis. It defines the framework within which the autonomous group robot resource allocation and pricing strategies will be examined. Section 4 establishes the game models used to analyze the pricing strategies. It defines the utility functions for both resource-providing robots and resource-consuming robots, incorporating factors that influence their decisions.Section 5 calculates the optimal pricing strategies for both the unified pricing model and the differential pricing model. It outlines the process of determining prices that maximize the objectives of resource-providing and resource-consuming robots in each scenario.Finally, Section 6 draws the conclusion.

2、Related work

In existing research on resource allocation strategies, the main optimization objectives have centered around minimizing task processing latency and system energy consumption.In terms of optimizing task processing latency, there have been notable contributions. For example, in [31], a study focused on vehicular edge computing networks and designed a vertically and horizontally coordinated network architecture. It analyzed the relationships between communication, caching, and computational resources, proposing a joint optimization model. This model was solved using asynchronous distributed reinforcement learning to determine task offloading and resource management strategies. The results demonstrated a significant reduction in overall latency.Similarly, [32] addressed the problem of heterogeneous task offloading in a distributed edge computing environment as a multi-player game. Participants made offloading decisions based on incomplete information. Building upon this, Abbas et al. devised a minority game-based strategy for heterogeneous edge computing task offloading, forming subtask groups that compete for resources. This approach substantially reduced task processing latency.These studies highlight the efforts to optimize task processing latency through resource allocation strategies, incorporating elements such as edge computing, reinforcement learning, and game theory to achieve more efficient and effective outcomes.

In the realm of optimizing system energy consumption, various research efforts have been undertaken:In [33], a convex optimization problem was introduced to minimize the weighted sum of mobile device energy consumption. A priority function, based on offloading priority, was utilized to solve the problem. This priority function was determined by individual users' channel gains and local computational energy consumption. Mustafa et al. also proposed a low-complexity suboptimal strategy using average subchannel gains to address the corresponding mixed-integer optimization problem, leading to significant improvements in energy consumption optimization.In [34], a joint optimization was conducted considering two distinct resource allocation strategies. The first strategy focused on inter-user fairness, while the second prioritized system energy efficiency. Additionally, [34] framed energy optimization as a non-convex fractional programming problem in cooperative strategies. A low-complexity optimization approach was employed to achieve optimal allocation, enhancing the energy efficiency of wireless mobile edge computing systems.In [35], a combination of local edge cloud and remote edge cloud computational resources were offloaded jointly using a genetic algorithm to search for optimal offloading strategies. This approach effectively reduced total system energy consumption while moderately decreasing task latency.[36] introduced the use of sparse code multiple access (SCMA) in designing algorithms for strategy development. Compared to conventional multicast strategies, this approach exhibited superior system performance in terms of energy efficiency.These studies collectively contribute to the exploration of resource allocation strategies that aim to minimize system energy consumption while optimizing other relevant parameters like task latency and efficiency. They utilize a range of techniques such as convex optimization, cooperative strategies, and genetic algorithms to achieve their objectives.

Indeed, the mentioned research works assume a scenario where price-based service providers unconditionally fulfill any computational tasks from service recipients without considering their own profit. This assumption implies that all service providers accept complete scheduling, which is an idealized condition that might not be feasible in certain real-world applications.In practical scenarios where there are different service recipients, each of them seeks to maximize their own benefits. Moreover, service providers have limited resources to offer. Consequently, competition among service providers arises due to these factors. In such situations, it's crucial to consider the interplay of pricing strategies, resource limitations, and the diverse objectives of both service providers and recipients to achieve a more realistic and effective resource allocation approach. This approach takes into account the complex dynamics of resource availability, competition, and individual utility maximization.

In recent years, there have been several studies that consider optimizing resource pricing strategies:In [37], a unified pricing approach was considered in an edge computing system where buyers take into account energy efficiency and costs. The authors established a Stackelberg game model between users and edge sellers, determining an optimal unit price to maximize the seller's revenue.In [38], an algorithm was proposed to find Nash equilibrium among users by considering energy efficiency, cost, and latency under a given fixed pricing policy.[39] introduced a Stackelberg game model between users, considering latency and seller costs. They presented two algorithms to maximize the seller's revenue using both unified and differential pricing strategies.Additionally, [40] presented a differential pricing scheme that maximized the overall welfare of both users and sellers.In [41], differential pricing was considered in a joint system model involving device users and edge servers.Finally, [42] introduced a bidding-based auction-style pricing mechanism by considering joint communication and computational resource allocation.These studies demonstrate the growing interest in optimizing resource pricing strategies to enhance overall system performance and stakeholder satisfaction. The diverse methods and models proposed reflect the complex nature of balancing the objectives of both buyers and sellers in resource allocation scenarios.

Indeed, certain studies have considered edge computing pricing strategies in specific scenarios:In [43]-[47], authors simulated a case where mobile users employ edge computing to run blockchain applications for mining blocks. Both unified pricing and differential pricing were taken into account.In [43], a dual auction system for edge computing was discussed. Two different systems were proposed: a profit-loss balanced Double Auction and a dynamic pricing-based Double Auction. The efficiency of these systems was analyzed.In [44], resource management was formulated as a double auction game and subjected to analysis.[45] considered a system involving mobile users, edge clouds, and remote clouds.In [46], the concept of general equilibrium was introduced in a system comprising edge nodes with capacity limits and service providers with budget constraints.Moreover, [47] focused on a specific case where sellers offer edge-cloud-based caching services to users. A Stackelberg equilibrium between sellers and users was considered when providers offer cache space at a fixed price.These studies delve into nuanced scenarios, showcasing how edge computing pricing strategies can be adapted to various specific contexts. The different models and analyses highlight the versatility and applicability of pricing mechanisms in edge computing environments.

Absolutely, the mentioned studies have predominantly focused on optimal pricing mechanisms within specific scenarios or for specific metrics (such as energy efficiency, latency, fairness, etc.). However, there is a lack of comprehensive comparative analysis among different pricing strategies. Furthermore, the majority of edge computing research has primarily considered a single pricing scheme.In the context of autonomous group robots, where information is incomplete and dynamic, it is indeed essential to delve into pricing strategies under incomplete information scenarios. This enables a more nuanced understanding of how varying levels of information availability can impact the effectiveness of pricing mechanisms. Such analysis is crucial for tailoring pricing strategies that can accommodate the real-world dynamics and uncertainties inherent in autonomous group robot applications.

3、Business model and Game Model

Absolutely, in scenarios involving incomplete information, the absence or uncertainty of certain data points, variables, or factors can significantly impact the decision-making process. Robots may lack complete understanding of their environment, the capabilities of other robots, or the exact resource requirements. The lack of complete information can make it challenging to determine the optimal pricing strategy or resource allocation plan. Consequently, researching effective pricing strategies that address such incomplete information is crucial for ensuring efficient operations in dynamic and uncertain environments. The ability to make informed decisions despite incomplete information is essential for the successful functioning of autonomous systems like group robots.

This paper initiates by establishing a network model for autonomous group robots that offers services such as communication and computation. Subsequently, it incorporates the popularity of services to construct a business model. Lastly, utilizing the network and business models as foundational elements, a game-theoretic model is formulated. This approach appears to integrate various components to create a comprehensive framework that considers both the technical aspects of the network and the demand for services through the business model. The game-theoretic model built upon this foundation likely aims to optimize resource allocation, pricing, or other strategic decisions in the context of autonomous group robots. If you need further elaboration or specific details about any of these models, please feel free to ask.

3.1Network model

In the context of an unknown scenario, this paper proposes a network model capable of providing communication and computation services. The topology of this model is depicted in Figure 1. From the perspective of resource allocation, the group robots are divided into resource-providing robots and resource-consuming robots.Among these, the resource-providing robots possess more computational resources, storage capacity, and higher computational capabilities. They can sell computational resources to neighboring robots that require them. On the other hand, the resource-consuming robots have relatively fewer computational resources and weaker computational capabilities. In collaborative scenarios, they need to purchase computational resources from robots that have more resources available.This model seems to address the resource distribution dynamics within the context of group robots in an unknown environment. By categorizing the robots into providers and consumers and considering their varying resource capacities, it likely aims to create an effective mechanism for resource exchange and allocation. If you need more detailed information or have specific questions about this model, please feel free to ask.

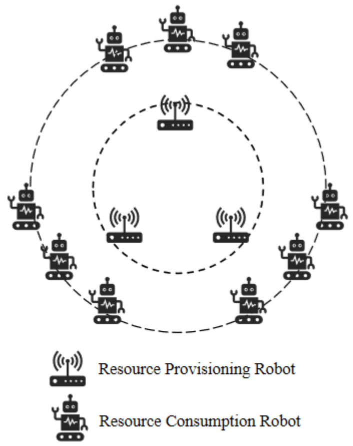

Fig1 Network Model Diagram

The group of robots comprises one type of resource-providing robot and M, where $M \in \{1,2,\cdots,M\}$,types of resource-consuming robots. The resource-providing robot has a limited amount of computational resources and aims to maximize its revenue. On the other hand, the resource-consuming robots seek to minimize their costs. These two objectives are interdependent, and they ultimately strive to achieve equilibrium in their interactions. The dynamics between the resource-providing and resource-consuming robots likely form the core of the research, exploring strategies that achieve a balance between revenue maximization and cost minimization in this complex system. If you require further elaboration or have specific inquiries, please don't hesitate to ask.

3.2Business model

Assuming that resource-consuming robot m requires C_m CPU cycles to compute 1 bit of input data and that it needs to process a total of R_m bits of input data. Among these, $0 \le \ell_m \le R_m$ bits are computed by the resource-providing robot, while the remaining $(R_m - \ell_m)$ bits are locally processed by the resource-consuming robot's CPU at a frequency denoted as F_m, measured in CPU cycles per second. This translates to a local computation time of $t_{loc,m} = (R_m - \ell_m)\ C_m/F_m$ for the resource-consuming robot.When resource-consuming robot m sends a request to the resource-providing robot, the time spent, denoted as $t_{off,m}$, encompasses various phases: $t_{u,m}$ is the time taken by the resource-consuming robot to send the request, $t_{c,m}$ represents the time taken by the resource-providing robot to prepare the resource, and $t_{d,m}$ accounts for the time taken to transmit the resource from the resource-providing robot to the resource-consuming robot.

Therefore, the time cost, denoted as $t_{off,m}$, incurred by the resource-consuming robot to purchase computational resources from the resource-providing robot is as follows:

$$t_{off,m} = t_{u,m} + t_{c,m} + t_{d,m} \qquad (1)$$

Due to the concurrent execution of local computation and the process of purchasing resources from the resource-providing robot and performing computation, the resource-consuming robot m can execute all R_m bits of data within a time period represented as $t_m = \max\{t_{loc,m}, t_{off,m}\}$.

The size of the computation result feedback to resource-consuming robot m is denoted as $\alpha_m \ell_m$. Among these parameters, $\alpha_m(\alpha_m > 0)$represents the ratio of the output bits purchased on the resource-providing robot to the input bits. This ratio is determined by the application of the resource-consuming robot. Additionally, there are parameters $t_{u,m} = \ell_m/r_m$ and $t_{d,m} = \alpha_m \ell_m / r_{B,m}$, with $r_m = \frac{B}{K}\log_2(1 + \frac{p_m h_m}{B/MN_0})$ and $r_{B,m} = \frac{B}{K}\log_2(1 + \frac{P_{B,m} h_m}{B/MN_0})$ respectively representing the uplink and downlink transmission rates of the resource-providing robot.

Among these, N_0 represents the noise power spectral density, while h_m signifies the channel gain between the resource-providing robot and resource-consuming robot m. $P_{B,m}$ and p_m respectively denote the downlink power and uplink power of resource-consuming robot m. Let $f_{c,m}$

represent the computation speed allocated to resource-consuming robot m by the resource-providing robot. Consequently, $t_{c,m} = \ell_m C_m / f_{c,m}$. For simplicity, we assume an equal allocation $f_{c,m}$, denoted as $f_{c,m} = f_C / M$ in this case, where f_C represents the total computation speed of the resource-providing robot."

Taking into account the limited computational capability of the resource-providing robot, which restricts the robot to compute received data within each computational cycle and imposes a CPU cycle limit of $\bar{F}$, this constraint can be represented as $\sum_{m=1}^{M} \ell_m C_m \leq \bar{F}$.

Where $\bar{F}$ and f_C respectively represent the computational workload and the speed of CPU cycles for the resource-providing robot.

3.3Service Popularity

In the system model, considering real-world scenarios, the computational resources of the resource-providing robot often have varying purchase frequencies, meaning their popularity differs. Therefore, the probability of resource-consuming robots purchasing computational resources, denoted as Q_c, follows a Zipf distribution.

$$Q_c = \frac{c^{-\alpha}}{\sum_{k=1}^{c} k^{-\alpha}} \tag{2}$$

In the equation, α represents the popularity level of computational resources. The larger α is, the more popular the computational resources are. Conversely, the smaller α is, the less popular the computational resources are.

Considering real-world situations, different resource-consuming robots have varying demands for purchasing computational resources, and their preferences also differ. Hence, the probability of resource-consuming robots requesting to buy computational resources is diverse. The interest preferences of resource-consuming robots can be expressed as follows:

$$Q_u = e_1 \frac{G_i^{avg}}{G_i} + e_2 \frac{N_i}{N} \tag{3}$$

In the equation: G_i^{avg} represents the average evaluation of computational resource i by the resource-consuming robots in the model region. G_i stands for the highest possible evaluation of computational resource i. N_i signifies the number of times resource-consuming robot purchases computational resource i. N represents the total number of purchases of computational resources by resource-consuming robots. e_1 and e_2 respectively denote coefficients for evaluations and purchase frequencies.

Construct the purchasing probability of computational resource by resource-consuming robot m in the model by weighting the resource popularity and the popularity based on the preferences of resource-consuming robots as follows:

$$Q = k_1 Q_c + k_2 Q_u \tag{4}$$

Where k_1 and k_2 are coefficients for resource popularity and the resource popularity based on the preferences of resource-consuming robots, respectively. If $k_1 > k_2$ holds true, it indicates that the resource-providing robot is more interested in popular computational resources. Otherwise, it suggests that resource-consuming robots purchase computational resources in pursuit of their own interests. Consequently, Q can represent the latent purchasing interest of resource-consuming robots in popular computational resources within the model, enhancing the model's alignment with real-world logic.

3.4Game Model

In this paper, resource-consuming robots utilize resources provided by resource-providing robots to perform computational tasks, while resource-providing robots must ensure that their total computation workload stays within the available CPU cycles. Therefore, to balance the supply and demand of computational resources, it's considered that resource-providing robots set a price, denoted as $\ell_m C_m$, for the CPU cycles they provide to each resource-consuming robot, denoted as K. As a result, a leader-follower game model can be employed to depict the interaction between resource-providing robots and resource-consuming robots, where resource-providing robots act as leaders, and resource-consuming robots act as followers.The resource-providing robots (leaders) start by pricing the

CPU cycles for resource-consuming robots. Subsequently, resource-consuming robots (followers) engage in local computations and purchase computational resources for computation based on the prices announced by the resource-providing robots.

The price of CPU cycles for resource-consuming robots is denoted as $\mu = (\mu_1, \mu_2, \cdots, \mu_M)$. The objective of resource-providing robots is to sell their limited computational resources to resource-consuming robots in order to maximize their revenue.

In conclusion, the utility function of resource-providing robots can be optimized as follows:

$$\mathbf{P1:}\max\ U_B(\mu) = \sum_{m=1}^{M} \mu_m \ell_m C_m,$$
$$\text{s.t.}\ \mu \geq 0.$$

Note: The data A of computational resources purchased by resource-consuming robot K is actually a function of B, as the amount of data each resource-consuming robot is willing to purchase depends on the price allocation.

On the resource-consuming robot side, the cost for each resource-consuming robot is defined as its latency plus the fee charged by the resource-consuming robot, which can be expressed as:

$$U_k(\ell_m, \mu_m) = Q \cdot \gamma \cdot (t_m + \mu_m \ell_m C_m), \quad (5)$$

This is equivalent to:

$$U_k(\ell_m, \mu_m) = \begin{cases} Q \cdot \gamma \cdot \{(\mu_m - \frac{1}{F_m})\ell_m C_m + \frac{R_m C_m}{F_m}\}, 0 \leq \ell_m \leq m_m \\ Q \cdot \gamma \cdot (\beta_m \ell_m + \mu_m \ell_m C_m), m_m < \ell_m \leq R_m \end{cases} \quad (6)$$

Where $\beta_m = \frac{1}{r_m} + \frac{C_m}{f_{c,m}} + \frac{\alpha_m}{r_{B,m}}$ and $0 < m_m < R_m$ are defined as $m_m = \frac{C_m R_m}{\beta_m F_m + C_m}$, and γ is a coefficient related to the revenue.

The goal of each resource-consuming robot is to minimize its cost by selecting the optimal data size ℓ_m for purchase, given the price μ_m set by the resource-providing robot. Therefore, the cost function for a resource-consuming robot can be expressed as:

$$\mathbf{P2:}\min U_k(\ell_m, \mu_m),$$
$$\text{s.t.}\ 0 \leq \ell_m \leq R_m.$$

From the perspective of net utility, the payment conditions in problems P1 and P2 can offset each other. The coupling between problems P1 and P2 in the leader-follower game is quite intricate. In other words, the pricing strategy of the resource-providing robots will affect the data purchase quantity of the resource-consuming robots, which in turn impacts the revenue of the resource-providing robots.

4Pricing Model

In order to analyze the considered leader-follower game, each resource-consuming robot independently solves problem P2 by given price μ to determine its purchase strategy ℓ_m^*. Resource-providing robots are aware of the purchase decisions $\ell_m^*(\mu_m)$ of each resource-consuming robot, and they determine their optimal price μ^* by solving problem P1. This process is known as backward induction. This study considers two optimal pricing strategies: uniform pricing and differential pricing. Next, we will investigate these two pricing schemes separately.

4.1Uniform Pricing

For the uniform pricing scheme, resource-providing robots set and broadcast a uniform price $\mu_1 = \mu_2 = \cdots \mu_M$ for all resource-consuming robots. With the given uniform price μ, the objective function U_m is a piecewise linear function of ℓ_m, with linearity in each interval starting from (6). With the structure of U_m, we can obtain the optimal solution for problem P2 in the following proposition:

Proposition 1: The optimal offloading strategy for each user in problem P2 follows a threshold-based strategy, where

$$\ell_m^*(\mu) = m_m x_m, \quad \forall k, \quad (7)$$

where the binary variable x_m is defined as

$$x_i = \begin{cases} 1, & \mu \leq \frac{1}{F_m} \\ 0, & otherwise \end{cases} \quad (8)$$

Proof: See Appendix A.

Based on Proposition 1, it can be inferred that there exists a threshold $1/\mu$ for purchasing computational resources. Specifically, if the CPU frequency F_m of resource-consuming robot m is less than or equal to this threshold, it tends to procure m_m bits of resources from the resource-providing robot; otherwise, it retains all bits for local computation. In other words, when resource-consuming robot m possesses a relatively lower computational speed F_k and has the potential for local computation, procuring computational resources proves beneficial.

Next, shifting our focus to Proposition 1, by substituting equation (7) into problem P1, we can reformulate the optimization problem for resource-providing robots under the unified pricing scheme as follows:

$$\mathrm{P3{:}max}\ U_B(\mu) = \mu \sum_{m=1}^{M} m_m x_m C_m \tag{9}$$

$$\mathrm{s.t.} \sum_{m=1}^{M} m_m x_m C_m \le \bar{F} \tag{10}$$

Proposition 2: Without loss of generality, the optimal unified price α^* must belong to the set $\{1/F_1 < \cdots < 1/F_{M-1} < 1/F_M\}$, after sorting $1/F_1 < \cdots < 1/F_{M-1} < 1/F_M$ in ascending order.

Proof: See Appendix B.

According to Proposition 2, the problem of maximizing revenue in P3 is simplified to a one-dimensional search problem on the n values in $\{1/F_M\}_{m=1}^{M}$. Algorithm 1 summarizes the entire approach. Specifically, resource-providing robots announce prices in decreasing order of $\{1/F_M\}_{m=1}^{M}$ to negotiate with resource-consuming robots. As a result, the total required CPU cycles sum $\sum_{m=1}^{M} \ell_m C_m$ decreases as the price μ decreases. Negotiations with other price candidates are unnecessary as long as the computational constraint (10) holds valid. This concludes the price negotiation process.

Clearly, the total complexity of Algorithm 1's search for μ^* is $\mathcal{O}(\log M)$. For the uniform pricing scheme, resource-providing robots require limited network information, namely F_m and C_m, which were collected before the algorithm's execution. In each iteration, every resource-consuming robot makes an independent offloading decision ℓ_m upon knowing the broadcasted price α from resource-providing robots and reports it back to update the price. Hence, the cloud broadcasts the price μ, each resource-consuming robot reports its purchase decision ℓ_m, and this is the information exchanged between resource-providing and resource-consuming robots in each iteration. Therefore, Algorithm 1 is a fully distributed algorithm.

Table1 Optimal Uniform Pricing Policy for Problem P3

Algorithm1 Optimal Uniform Pricing Policy for Problem P3

1:Resource provision robot initializaties $\tau = K$ and $\mu^\tau = 1/F_\tau$.
2:**Repeat**
3:Each resource consumption robot determines the optimal purchase data size $\ell_m^*(\mu^\tau)$ based on (7)
4:Resource provision robots calculate their revenue $U_B(\mu^\tau)$ based on (9).
5:**if** $\sum_{m=1}^{M} \ell_m^*(\mu^\tau) C_m \le \bar{F}$ **then**
6: Update the price $\mu^{\tau-1} = 1/F_{\tau-1}$ and $\tau \leftarrow \tau - 1$;
7:**else**
8: Set $U_B(\mu^\tau) = 0$;**break**;
9:**end if**
10:**until** $\tau \le 0$.
11:Output $\mu^* \leftarrow argmax_{\mu^\tau} U_B(\mu^\tau)$.

4.2Differentiated pricing

In this context, we consider the general case where resource-providing robots charge different types of resource-consuming robots at different prices. Similar to the uniform pricing case, the optimal solution to Problem P2 is still given by (7), with the substitution of μ in x_m being replaced by μ_m. Problem P1 can be formulated as:

$$\mathrm{P4{:}max}\ U_B(\mu_m) = \sum_{m=1}^{M} \mu_m m_m x_m C_m$$

$$\mathrm{s.t.}\ \mu \ge 0 \tag{11}$$

In particular, for resource-consuming machine m, the price μ_m is actually a function of x_m. More specifically, the optimal price $x_m = 1$, denoted as $\mu_m \le 1/f_i$ for resource-consuming machine m, is determined by $\mu_m^* = 1/f_m$ since the objective function of Problem P4 is a monotonically increasing function of μ_m. When $x_m = 0$ occurs, resource-providing robots set the price for resource-consuming machine m to $\mu_m^* = \infty$, without earning any profit. Based on the above analysis, Problem P4 can be equivalently represented as:

$$P4^{'}: max\ U_B(x_m) = \sum_{m=1}^{M} \frac{m_m x_m C_m}{F_m},$$

$$\mathrm{s.t.}\ 0 \le x_m \le 1$$

(12)

Problem $P4^{'}$ is actually a 0/1 knapsack problem with weights given by $m_m c_m$ and values corresponding to resource-consuming machine m denoted as $m_m C_m / F_m$. Since this problem is NP-complete, there is no efficient algorithm to solve it optimally. However, we can apply dynamic programming to solve the aforementioned 0/1 knapsack problem in pseudo-polynomial time.

To solve Problem $P4^{'}$, each resource-consuming machine needs to report m_m, C_m, and F_m to the resource-providing machine. There is no need for iteration between resource-consuming machines and resource-providing machines. Upon obtaining the optimal price μ_m^*, resource-consuming machine m decides its optimal strategy based on Equation (7). Therefore, the differential pricing scheme is also a distributed algorithm, but it requires more information and has higher complexity compared to the uniform pricing scheme.

5Simulation and discussion

In this section, we provide several numerical results for both the uniform pricing algorithm and the differential pricing algorithm to analyze their performance. All simulations in this section are based on Python 3.5.

Consider a set of resource-consuming robots as {1, 2, 3, 4, 5}, and contemplate the following scenario: Two resource-consuming robots (Robot 1 and Robot 2) are engaged in executing terrain exploration algorithms, while another two robots are concurrently running speech recognition programs. Additionally, one particular robot (Robot 5) is currently performing image detection algorithms.Suppose the sizes of data associated with their respective tasks are represented by the collection (a1, a2, a3, a4, a5) = (1, 2, 8, 10, 15) MB. Furthermore, let's assume that each resource-consuming robot's constant μi is defined by (μ1, μ2, μ3, μ4, μ5) = (1, 1.5, 2, 3, 5). The total allocation of CPU cycles for computing resources is set at 8 GHz. Notably, in a reference [26], the processing densities for speech recognition algorithms and 400-frame videos are measured at 31680 and 2640 cycles per bit, respectively. Consequently, it is reasonable to attribute a processing density of 31680, 31680, and 264 cycles per bit to resource-consuming robots 3, 4, and 5, respectively. Similarly, we assume a processing density of 20000 cycles per bit for the speech recognition program.

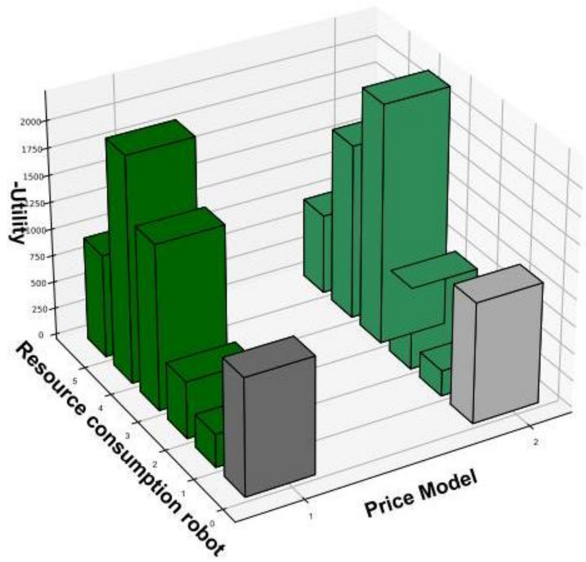

Fig 2 Absolute Value of Utility Function

In Figure 2, the absolute values of the utility functions are visualized using a 3D graph, with the average values represented by gray bars. The results indicate that the absolute values of the utility functions exhibit an increasing trend in the order of the uniform pricing algorithm and the differentiated pricing algorithm. This implies that the actual values decrease in the same order.

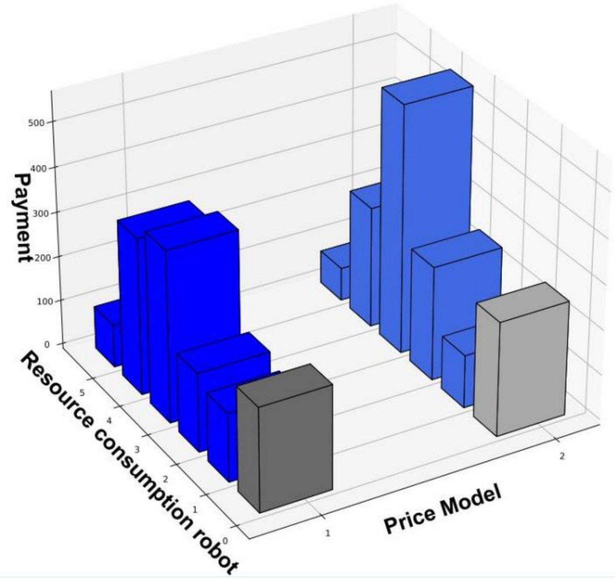

Fig3 Payment situation of consumer robots

Furthermore, Figure 3 depicts the payment scenarios for users in both models, illustrating the trend of increasing payments in the order of the uniform pricing algorithm and the differentiated pricing algorithm.

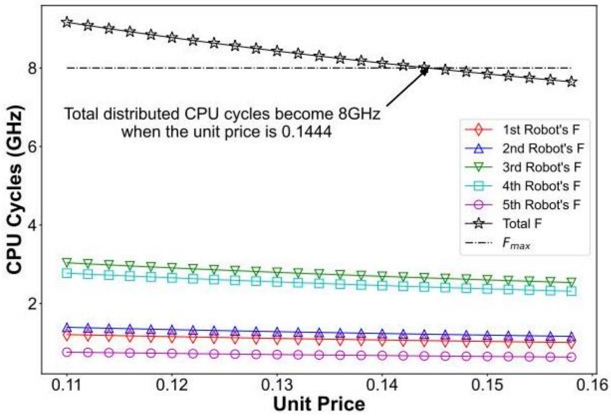

Fig 4 Distributed CPU cycles and total

As evident from Figure 4, it can be observed that when the payment for the third resource-consuming robot shifts from point $0.5\mu^*(137.5)$ to point $1.5\mu^*(412.5)$, the utility of the third resource-consuming robot reaches its maximum value of -1354 at point $\mu 3 = 275.0$, which corresponds to the computed Nash equilibrium.

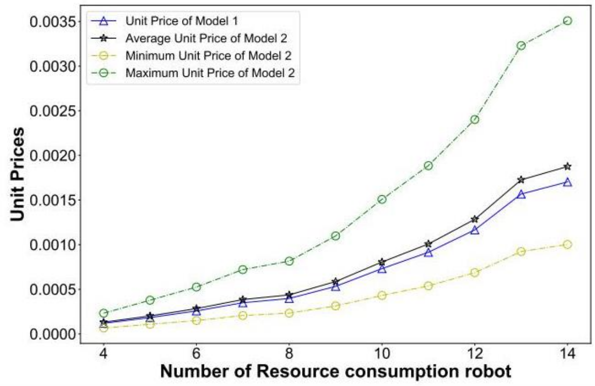

Fig5: Changes in the number of resource consuming robots in each model as the unit price increases

In addition, a simulation with randomly generated resource-consuming robots was also designed to emulate real-world scenarios. It was assumed that in emergency situations, the total CPU cycles allocated for resource allocation is 100, with each robot requiring 10 CPU cycles. The fairness coefficient was set to 3.5. Furthermore, each resource-consuming robot was generated and assigned a random value of μ (constant) and its data size within the range of 0 to 10. We differentiated the number of resource-consuming robots and plotted the unit prices for both models for each user count. Figure 5 effectively presents the results.

In the above analysis, two models were analyzed based on the same set of robots, while also increasing the total CPU cycles allocated for resource allocation. It can be intuitively predicted that as the CPU cycles increase, the utility of resource-consuming robots will increase, and the overall latency for resource-consuming robots will decrease. Additionally, according to the principle of supply and demand, it can be anticipated that pricing will decrease.

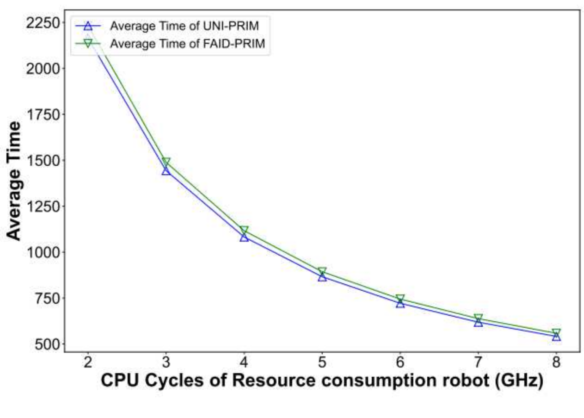

Fig 6: Changes in average processing time of resource consuming robots as CPU cycles increase

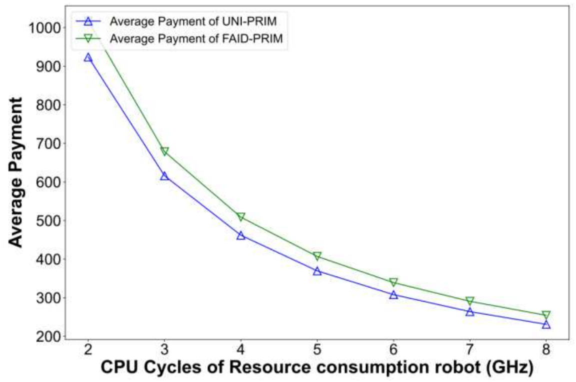

Fig 7: Changes in payment pricing for resource consumption robots as CPU cycles increase

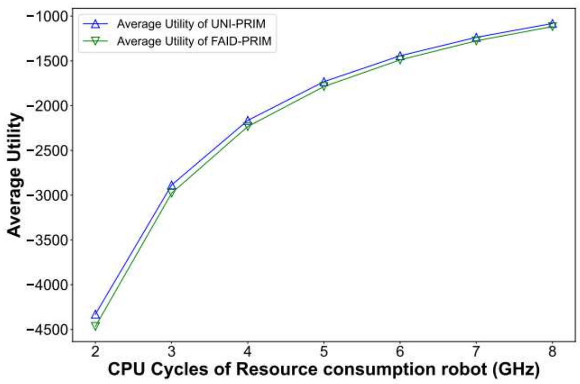

Fig 8: Changes in the number of resource consuming robots as CPU cycles increase

Increasing the allocated CPU cycles from 2GHz to 8GHz has been depicted in the changes in average processing time, pricing, and social welfare, as shown in Figures 7, 8, and 9. Figures 7 and 8 illustrate that the average processing time and pricing for resource-consuming robots tend to decrease with the increase in the total CPU cycles, following a trend that resembles an inverse relationship. Consequently, the average utility of resource-consuming robots often increases with the increase in the CPU cycles allocated by resource-providing robots, implying that enhancing the total CPU cycles allocated by resource-providing robots can lead to greater satisfaction among resource-consuming robots. These numerical results establish a direct correlation between the allocated CPU cycles for resource allocation and the overall well-being of the system.

6Conclusion

This paper investigates the problem of resource allocation for autonomous group robots in an incomplete information scenario based on pricing strategies. To achieve this, the paper models the computation resource allocation from a game-theoretical perspective, introducing both the uniform pricing model and the differential pricing model and conducting a rigorous comparative analysis between them. The introduction of a fairness factor in the differential pricing model enhances collaboration among robots. It is concluded that different pricing mechanisms result in varying degrees of collaboration and stability within autonomous group robots. Furthermore, the paper conducts performance and numerical analyses on the two pricing models, outlining their respective advantages and disadvantages. This enables autonomous group robots to adaptively select different pricing strategies in response to changing environmental conditions in an incomplete information setting.

Appendix A

For a given α, the optimal solution of problem P2 is determined by

$$\ell_m^*(\mu)\begin{cases} = m_m, & \mu < \frac{1}{F_m}, \\ \in [0, m_m], & \mu = \frac{1}{F_{m,}} \\ = 0 & otherwise. \end{cases} \tag{17}$$

For all resource-providing robots, the probability of event $\mu = 1/F_m$ occurring is 0. In this scenario, let $\ell_{\mathrm{m}} = \mathrm{m_m}$. Proof is complete.

Appendix B

Using the method of proof by contradiction, the following is demonstrated: Suppose the optimal price μ^* exists within the interval $(1/F_i, 1/F_{i+1}), \forall i \in \{1, \cdots, M-1\}$. Next, consider scenario $\tilde{\mu} = 1/F_{i+1}$. According to equation (6), it is evident that for values $\tilde{\mu}$andμ^*, when condition $\mathrm{m} = 1, \cdots \mathrm{i}$holds true, $x_m = 0$occurs; similarly, when condition $\mathrm{m} = \mathrm{i}+1, \cdots \mathrm{M}$ is satisfied, $x_i = 1$ follows. Hence, the total CPU cycles for the summation of data $\sum_{m=1}^{M} m_m x_m c_m$are equal to the CPU cycles for data $\tilde{\mu}$ in scenario μ^*. Given that the objective function $U_B(\mu)$, provided in equation (7), is a monotonically increasing linear function of price μ, it is always possible to achieve higher profits in scenario $\tilde{\mu} = 1/F_{i+1}$ than in scenario μ^*. This contradicts the assumption of the optimal solution for problem P3 when condition $1/F_i < \mu^* < 1/F_{i+1}$ is valid. As a result, the optimal price α^* must necessarily belong to the set $\{1/F_1, \cdots 1/F_{M-1}, 1/F_M\}$.

Acknowledgements

The authors would like to thank everyone involved in the work that was presented in this paper.

Funding

Not applicable.

Author’s information

Sun Yi was born in Jingyang County, Shaanxi Province in March 1972. In May 2001, he graduated from Xi'an Jiaotong University majoring in electrical engineering and obtained a doctor's degree in engineering, professor and master's supervisor. In March 2003, he worked in the Department of Electronic Science and Technology of Xi'an University of Science and Technology.

The main research directions include intelligent information processing, embedded systems and applications, machine learning, game theory, blockchain, etc. He has presided over more than 20 provincial and enterprise entrusted projects to tackle key scientific and technological problems in Shaanxi and Xi'an, published more than 40 papers in Chinese and foreign academic journals and international conferences, owned more than 50 patents, software copyrights, and edited or translated 5 monographs.

Zhang Xiao was born in Yanan, Shaanxi in December 2002. In June 2022, he graduated from Xi'an University of Science and Technology with a bachelor's degree. In September 2022, he was admitted to the electronic information major of Xi'an University of Science and Technology and is now a graduate student of Xi'an University of Science and Technology.

The main research interests are group robots, game theory, etc. It was wrote two papers in school. One Chinese core, one SCI. He has participated in many competitions and won prizes, such as electronic design competition, mathematical modeling, etc.

Author’s Contributions

ZX is responsible for writing papers, constructing models, and implementing algorithms; SY is responsible for overall control of the paper.

Corresponding authors

Correspondence to ZhangXiao.

Ethics approval and consent to participate

Ethics approval

I have read the Nature Portfolio journal policies on author responsibilities and submit this manuscript in accordance with those policies.

Consent to participate

I confirm that I understand journal Swarm Intelligence is a transformative journal. When research is accepted for publication, there is a choice to publish using either immediate gold open access or the traditional publishing route.

Competing interest

I declare that the authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Consent for Publication

The results/data/figures in this manuscript have not been published elsewhere, nor are they under consideration (from you or one of your Contributing Authors) by another publisher.

Availability Material

All of the material is owned by the authors and/or no permissions are required.

References:

[1] Beni, G.: Swarm intelligence. Complex Social and Behavioral Systems: Game Theory and Agent-Based Models, 791–818 (2020)

[2] Debie, E., Kasmarik, K., Garratt, M.: Swarm robotics: A survey from a multitasking perspective. ACM Computing Surveys (2023)

[3] Bredeche, N., Fontbonne, N.: Social learning in swarm robotics. Philosophical Transactions of the Royal Society B 377(1843), 20200309 (2022)

[4] Kuang, Z., Ma, Z., Li, Z., Deng, X.: Cooperative computation offlfflffloading and resource allocation for delay minimization in mobile edge computing. Journal of Systems Architecture 118, 102167 (2021)

[5] Li, X.: A computing offlfflffloading resource allocation scheme using deep reinforcement learning in mobile edge computing systems. Journal of Grid Computing 19, 1–12 (2021)

[6] Sharif, Z., Jung, L.T., Razzak, I., Alazab, M.: Adaptive and priority-based resource allocation for effiffifficient resources utilization in mobile edge computing. IEEE Internet of Things Journal (2021)

[7] Kuang, Z., Ma, Z., Li, Z., Deng, X.: Cooperative computation offlfflffloading and resource allocation for delay minimization in mobile edge computing. Journal of Systems Architecture 118, 102167 (2021)

[8] Sellami, B., Hakiri, A., Yahia, S.B.: Deep reinforcement learning for energyaware task offlfflffloading in join sdn-blockchain 5g massive iot edge network. Future Generation Computer Systems 137, 363–379 (2022)

[9] Huang, X., Gong, S., Yang, J., Zhang, W., Yang, L., Yeo, C.K.:Hybrid market based resources allocation in mobile edge computing systems under stochastic information. Future Generation Computer Systems 127, 80–91 (2022)

[10] Tun, Y.K., Dang, T.N., Kim, K., Alsenwi, M., Saad, W., Hong, C.S.: Collaboration in the sky: A distributed framework for task offlfflffloading and resource allocation in multi-access edge computing. IEEE Internet of Things Journal 9(23), 24221–24235 (2022)

[11] Nath, S., Wu, J.: Deep reinforcement learning for dynamic computation offlfflffloading and resource allocation in cache-assisted mobile edge computing systems. Intelligent and Converged Networks 1(2), 181–198 (2020)

[12] Jia, M., Zhang, L., Wu, J., Guo, Q., Gu, X.: Joint computing and communication resource allocation for edge computing towards huge leo networks. China Communications 19(8), 73–84 (2022)

[13] Zaman, S.K.u., Jehangiri, A.I., Maqsood, T., Haq, N.u., Umar, A.I., Shuja, J., Ahmad, Z., Dhaou, I.B., Alsharekh, M.F.: Limpo: Lightweight mobility prediction and offlfflffloading framework using machine learning for mobile edge computing. Cluster Computing 26(1), 99–117 (2023)

[14] Xie, Z., Song, X., Cao, J., Xu, S.: Energy effiffifficiency task scheduling for battery level-aware mobile edge computing in heterogeneous networks. ETRI Journal 44(5), 746–758 (2022)

[15] Bajaj, K., Sharma, B., Singh, R.: Implementation analysis of iot-based offlfflffloading frameworks on cloud/edge computing for sensor generated big data. Complex & Intelligent Systems 8(5), 3641–3658 (2022)

[16] Li, T., He, X., Jiang, S., Liu, J.: A survey of privacy-preserving offlfflffloading methods in mobile-edge computing. Journal of Network and Computer Applications 203, 103395 (2022)

[17] Yadav, A.M., Sharma, S.: Cooperative task scheduling secured with blockchain in sustainable mobile edge computing. Sustainable Computing: Informatics and Systems 37, 100843 (2023)

[18] Santos, J., Wauters, T., De Turck, F.: Effiffifficient management in fog computing, 1–6 (2023). IEEE

[19] Hassan, S.R., Ahmad, I., Nebhen, J., Rehman, A.U., Shafifiq, M., Choi,J.- G.: Design of latency-aware iot modules in heterogeneous fog-cloud computing networks. Computers, Materials & Continua 70(3) (2022)

[20] Gasmi, K., Dilek, S., Tosun, S., Ozdemir, S.: A survey on computation offlfflffloading and service placement in fog computing-based iot. The Journal of Supercomputing 78(2), 1983–2014 (2022)

[21] Tang, Q., Xie, R., Yu, F.R., Chen, T., Zhang, R., Huang, T., Liu, Y.: Distributed task scheduling in serverless edge computing networks for the internet of things: A learning approach. IEEE Internet of Things Journal 9(20), 19634–19648 (2022)

[22] Menaka, M., Kumar, K.S.: Workflflow scheduling in cloud environment–challenges, tools, limitations & methodologies: A review. Measurement: Sensors, 100436 (2022)

[23] Malik, U.M., Javed, M.A., Zeadally, S., Islam, S.: Energy-effiffifficient fog computing for 6g-enabled massive iot: Recent trends and future opportunities. IEEE Internet of Things Journal 9(16), 14572–14594 (2021)

[24] Hu, B., Du, J., Chu, X.: Enabling low-latency applications in vehicular networks based on mixed fog/cloud computing systems, 722–727 (2022). IEEE

[25] Liu, Y., Su, Z., Wang, Y.: Energy-effiffifficient and physical-layer secure computation offlfflffloading in blockchain-empowered internet of things. IEEE Internet of Things Journal 10(8), 6598–6610 (2022)

[26] Mohajer, A., Daliri, M.S., Mirzaei, A., Ziaeddini, A., Nabipour, M., Bavaghar, M.: Heterogeneous computational resource allocation for noma: Toward green mobile edge-computing systems. IEEE Transactions on Services Computing 16(2), 1225– 1238 (2022)

[27] Tran-Dang, H., Kim, D.-S.: Disco: Distributed computation offlfflffloading framework for fog computing networks. Journal of Communications and Networks 25(1), 121–131 (2023)

[28] Li, X., Fan, R., Hu, H., Zhang, N., Chen, X., Meng, A.:Energy-effiffifficient resource allocation for mobile edge computing with multiple relays. IEEE Internet of Things Journal 9(13), 10732–10750 (2021)

[29] Liu, H., Long, X., Li, Z., Long, S., Ran, R., Wang, H.-M.: Joint optimization of request assignment and computing resource allocation in multi-access edge computing. IEEE Transactions on Services Computing 16(2), 1254–1267 (2022)

[30] Ashraf, M., Shiraz, M., Abbasi, A., Albahli, S.: Distributed application execution in fog computing: A taxonomy, challenges and future directions. Journal of King Saud University-Computer and Information Sciences 34(7), 3887–3909 (2022)

[31] Feng, C., Han, P., Zhang, X., Yang, B., Liu, Y., Guo, L.:Computation offlfflffloading in mobile edge computing networks: A survey. Journal of Network and Computer Applications 202, 103366 (2022)

[32] Mustafa, E., Shuja, J., Zaman, S.K., Jehangiri, A.I., Din, S., Rehman,F., Mustafa, S., Maqsood, T., Khan, A.N.: Joint wireless power transfer and task offlfflffloading in mobile edge computing: a survey. Cluster Computing 25(4), 2429–2448 (2022)

[33] Tang, L., Hu, H.: Computation offlfflffloading and resource allocation for the internet of things in energy-constrained mec-enabled hetnets. IEEE Access 8, 47509–47521 (2020)

[34] Wu, M., Qi, W., Park, J., Lin, P., Guo, L., Lee, I.: Residual energy maximization for wireless powered mobile edge computing systems with mixed-offlfflffloading. IEEE Transactions on Vehicular Technology 71(4), 4523–4528 (2022)

[35] Chai, R., Song, X., Chen, Q.: Joint task offlfflffloading, cnn layer scheduling, and resource allocation in cooperative computing system. IEEE Systems Journal 14(4), 5350–5361 (2020)

[36] Gao, M., Shen, R., Li, J., Yan, S., Li, Y., Shi, J., Han, Z., Zhuo, L.: Computation offlfflffloading with instantaneous load billing for mobile edge computing. IEEE Transactions on Services Computing 15(3), 1473–1485 (2020)

[37] Abbas, Z.H., Ali, Z., Abbas, G., Jiao, L., Bilal, M., Suh, D.-Y., Piran, M.J.: Computational offlfflffloading in mobile edge with comprehensive and energy effiffifficient cost function: a deep learning approach. Sensors 21(10), 3523 (2021)

[38] Rahmani, A.M., Mohammadi, M., Mohammed, A.H., Karim, S.H.T.,Majeed, M.K., Masdari, M., Hosseinzadeh, M.: Towards data and computation offlfflffloading in mobile cloud computing: taxonomy, overview, and future directions. Wireless Personal Communications 119, 147–185 (2021)

[39] Tong, Z., Deng, X., Mei, J., Dai, L., Li, K., Li, K.: Stackelberg game-based task offlfflffloading and pricing with computing capacity constraint in mobile edge computing. Journal of Systems Architecture 137, 102847 (2023)

[40] Bi, J., Yuan, H., Zhang, K., Zhou, M.: Energy-minimized partial computation offloading for delay-sensitive applications in heterogeneous edge networks. IEEE Transactions on Emerging Topics in Computing 10(4), 1941–1954 (2022)

[41] Apostolopoulos, P.A., Tsiropoulou, E.E., Papavassiliou, S.: Risk-aware data offlfflffloading in multi-server multi-access edge computing environment. IEEE/ACM Transactions on Networking 28(3), 1405–1418 (2020)

[42] Huang, X., Gong, S., Yang, J., Zhang, W., Yang, L., Yeo, C.K.: Hybrid marketbased resources allocation in mobile edge computing systems under stochastic information. Future Generation Computer Systems 127, 80–91 (2022)

[43] Yang, Y., Liu, Z., Liu, Z., Chan, K.Y., Guan, X., et al.: Joint optimization of edge computing resource pricing and wireless caching for blockchain-driven networks. IEEE Transactions on Vehicular Technology 71(6), 6661–6670 (2022)

[44] Fan, Y., Wang, L., Wu, W., Du, D.: Cloud/edge computing resource allocation and pricing for mobile blockchain: an iterative greedy and search approach. IEEE Transactions on Computational Social Systems 8(2), 451–463 (2021)

[45] Jain, V., Kumar, B.: Auction based cost-effiffifficient resource allocation by utilizing blockchain in fog computing. Transactions on Emerging Telecommunications Technologies 33(7), 4469 (2022)

[46] Baranwal, G., Kumar, D., Vidyarthi, D.P.: Bara: A blockchain-aided auctionbased resource allocation in edge computing enabled industrial internet of things. Future Generation Computer Systems 135, 333–347 (2022)

[47] Ma, L., Wang, X., Wang, X., Wang, L., Shi, Y., Huang, M.: Tcda: Truthful combinatorial double auctions for mobile edge computing in industrial internet of things. IEEE Transactions on Mobile Computing 21(11), 4125–4138 (2021)