Near Optimal Locality-Aware Task Allocation Towards Stable Blockchain-Based MEC System: A Potential Game Approach

Lianbo Ma, *Senior Member, IEEE,* Yuee Zhou, Xingwei Wang, *Member, IEEE,*
Carla Fabiana Chiasserini, *Fellow, IEEE,* Guangjie Han, *Fellow, IEEE*

Abstract—We consider the efficient resource allocation task in the blockchain-based mobile edge computing (MEC) system that requires decentralized transaction management to validate transactions between edge servers (ESs) and mobile devices (MDs). In such task allocation process (where MDs' resources are limited and privacy-sensitive), it is a significant challenge to guarantee individual rationality with satisfactory system stability while enabling flexible task offloading under various locality constraints (e.g., communication distance, bandwidth and delay). In this paper, we formulate the target problem as a blockchain-assisted task-resource matching model, and then propose a near optimal locality-aware resource allocation mechanism over smart contract to enable automatic and efficient transactions in MEC system. More specifically, for the service agents selection, we design the preference-based selection strategy to get highest estimated profit. For the flexible task offloading, we develop the minimum delay task graph partitioning algorithm to determine the optimal task offloading solution for MD under different resource bundles. For the task-resource matching, we propose a multi-stage task-resource matching game (based on potential game) with the second lowest cost strategy to determine the matching of task-resource and decide the price of resource bundle. For the transaction verification and block allocation, we propose a social welfare-driven consensus mechanism to enable verified transaction and fair block allocation in a reward-free way. Strict theoretical analysis and extensive simulations demonstrate that our mechanism guarantees individual rationality, Nash Equilibrium, and stable near optimal solution.

Index Terms—Mobile edge computing, Task allocation, Nash Equilibrium, Social welfare, Individual rationality.

1 Introduction

The explosive growth in advanced mobile and IoT devices has resulted in a significant amount of data being generated at the edge network [1], which enables a wide variety of promising mobile applications [2], [3], [4], [5], e.g., augmented reality, data streaming, and mobile gaming. It is estimated that by 2025, there will be around 80 billion mobile devices globally, producing 180 trillion gigabytes of data [4]. Processing such a plethora of data through cloud computing is inefficient due to the considerable quantities of bandwidth consumption [6], [7]. A natural way to solve this issue is to deploy cloud services at the edge of the network where data is generated [8]. This is achieved by mobile edge computing (MEC), allowing for the offloading of computation-intensive and latency-sensitive tasks from mobile devices (MDs) to adjacent edge servers (ESs) [9], [10], [11], [12]. Typically, there is a central trading platform for allocating resources in MEC, where MDs need to choose from services provided by adjacent SPs with sufficient computing and communication resources to meet their own demands for the execution of target tasks [13].

However, the efficient resource allocation is a challenging issue in establishing a successful MEC system [14], [15], [16]. Most existing resource allocation approaches in centralized trading platform depend on a central authority to enable the matching between tasks of MDs and resources of ESs across the network devices. In fact, the central authority may not be trusted and is fragile to the single point of failure [17], [18], [19], [20]. In this sense, the dynamic communication between MDs and ESs, and the migration of data across the system may lead to security vulnerability [21], [22]. As a result, due to the fact that the distributed participants in MEC (including MDs and ESs) are profit-driven, such fragile allocation mechanisms will result in loss for all associated MDs and ESs.

To avoid these limitations, the blockchain is then introduced into the MEC system (i.e., blockchain-based MEC, BMEC), where the resource allocation tasks are performed in a distributed way [24]. Using efficient community verification among edge nodes through consensus mechanisms such as Proof of Work (PoW) and Proof of Stake (PoS) [25], BMEC has merits of high-level security and trust without the need for any central authority [26], [28]. One line of research on BMEC is to motivate MDs and ESs to cooperate in task execution, aiming to enable the network operator to allocate edge services to a set of requesting MDs in a utility-maximization way [28]. In this paper, we thus investigate the problem how to maximize the social welfare of the

- *Lianbo Ma and Yuee Zhou are with College of Software, Northeastern University, Shenyang 110819, China. (E-mail: malb@swc.neu.edu.cn and zhouyuee@stumail.neu.edu.cn)*
- *Xingwei Wang is with School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China. (E-mail: wangxw@mail.neu.edu.cn)*
- *Carla Fabiana Chiasserini is with Politecnico di Torino, 10129 Torino, Italy, also with CNIT, 43124 Parma, Italy, and also with IEIIT-CNR, 10129 Torino, Italy (E-mail: carla.chiasserini@polito.it).*
- *Guangjie Han is with the Department of Information Science and Engineering, Hohai University, Changzhou, 213022, China. (E-mail: hanguangjie@gmail.com)*

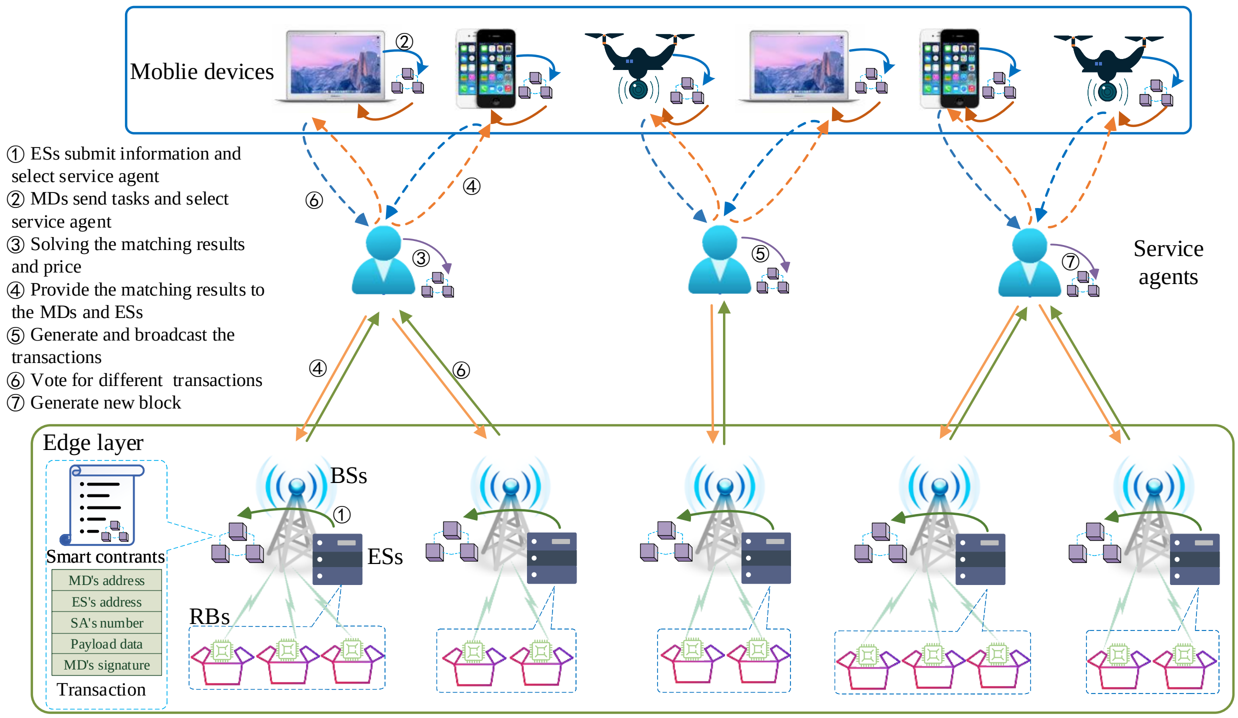


Fig. 1. An example of a BMEC model, where ESs offer resource bundles to MDs (with various resource requests for task completion).

BMEC system such that all participants can maximize their profits. A key challenge of our design is how to guarantee individual rationality with satisfactory system stability. Moreover, it is essential to guarantee that the social welfare obtained from Nash Equilibrium (NE) solution is a stable near-optimal solution, theoretically close to the optimal one.

In practice, when striving to guarantee individual rationality with good stability, the following issues should be taken into account:

(1) *Flexible task offloading is required when MDs have complex application tasks*. If MDs roughly offload all tasks to ESs, the processing time and computation cost could be increased since MD's own computing resource is wasted. This may lead to fewer MDs being satisfied and even reduce the social welfare. Thus, it is desired to allow each MD to express multiple task offloading solutions, and choose the optimal task offloading solution according to the task matching result. In this way, the MD satisfaction can be obtained and higher social welfare is achieved further.

(2) *Many locality constraints exist in real BMEC system*, including the communication distance and wireless channel conflict. Due to limited wireless recognition range, only MDs within coverage area can connect to the corresponding ESs. Besides, the wireless communication between edge devices may be slow and unstable due to congested or conflicted wireless channel sharing. Due to these locality constraints, not all ESs and MDs can be matched, which would influence the stability of the BMEC system.

(3) *Efficient consensus mechanism is required in the context of the BMEC system* due to the limitation of resource and transmission. However, existing consensus mechanisms, e.g., PoW and PoS, are not suitable to our target scenario: PoW is extremely computation-consuming and thereby being not applicable to the resource-constrained edge devices, while PoS is beneficial for wealthy players and tends to enlarge the wealth inequality among the players in BMEC.

In this paper, we formulate the target problem as a blockchain-assisted task-resource matching model, and then propose a **N**ear **O**ptimal **L**ocality-**A**ware **T**ask **A**llocation (NOLATA) mechanism over smart contract to enable automatic and efficient transactions in the BMEC system, where the social welfare maximization in joint computing and communication resource allocation problem is formulated with a set of significant locality constraints.

Our main contributions are as follows:

- To the best of our knowledge, our proposed mechanism is the first individual rational game mechanism for BMEC, which can guarantee system stability for task allocation while considering different locality constraints including communication distance, wireless channel conflict and task latency constraints.
- We propose NOLATA (derived from the potential game) for task allocation, where the smart contract preference selection strategy is devised for MDs to select smart contracts, the second lowest cost pricing strategy is utilized to decide the resource price of implemented tasks, and the social welfare-driven consensus mechanism is used to enable verified transaction and fair block allocation in a reward-free way.
- We conduct rigorous theoretical analysis and extensive simulation experiments to verify the promising performance of social welfare maximization, and and desired properties of the proposed NOLATA, i.e., individual rationality, Nash Equilibrium, and stable approximation ratio.

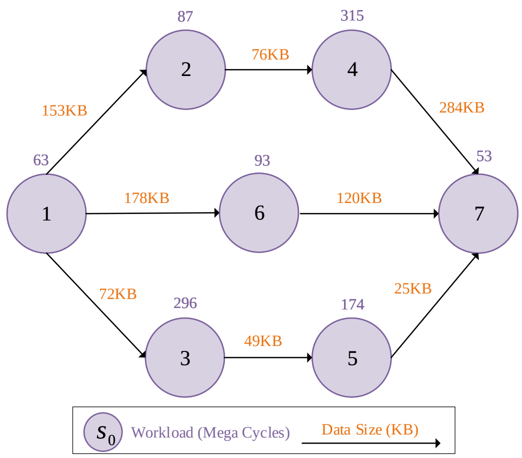


Fig. 2. An example of a directed task graph.

2 System Model and Problem Formulation

In this section, we first present the system model with a set of locality constraints. Next, we formulate the problem of maximizing social welfare in the joint allocation of computing and communication resources. The key notations are summarized in Table 1.

2.1 System Model

We consider a standard BMEC framework comprising M ESs, N MDs, and K service agents (SAs). Each ES is equipped with a base station (BS), as shown in Fig.1. Let $[X]$ represent the set of integers $\{1, \cdots, x, \cdots, X\}$. $MD_{[N]}$, $ES_{[M]}$, $BS_{[M]}$, and $SA_{[K]}$ denote the set of MDs, ESs, BSs, and SAs, respectively. Each ES selects an SA to help them sell resources. Each MD selects an SA to purchase computing and communication resources to execute and transmit their application tasks. The SAs, acting as agents, can coordinate the transactions between MDs and ESs. A set of locality constraints are considered, including the *communication distance*, *wireless channel conflict* and *task latency* constraints (see Section 2.2 for details). On account of real BMEC scenarios, each BS employs dynamic bandwidth allocation [29] for wireless transmission. In this context, workloads are transmitted over orthogonal channels at a certain data transmission rate, and the decision to offload will be made based on latency and communication overhead.

For each MD_n, the application task is denoted by $A_n = \{D_n(V_n, E_n), f_n, t_n, (u_n, v_n)\}$, where $D_n(V_n, E_n)$ is a directed task graph used to describe the application task, f_n is the local processing capability of MD_n, t_n is the maximum latency of A_n, and (u_n, v_n) is MD_n's geographical location. In $D_n(V_n, E_n)$, V_n is the set of subtasks in A_n, and E_n denotes the data dependencies among these subtasks. Fig. 2 illustrates an example of a directed task graph, where node s_2 has 87 Mega Cycles instructions to be performed and 76KB data is involved to be transferred from s_2 to s_4. Note that, in our model, the subtasks of an MD_n's application can be offloaded to only one of its neighboring ESs.

Each ES_m is equipped with a maximum of H resource bundles (RBs), each of which can be only used by a single MD, and its geographical location is denoted by (x_m, y_m). The provision profile of each resource bundle h provided by ES_m is denoted by $RB_{\{m,h\}} = \{f_{(m,h)}\}$, where $f_{(m,h)}$ is the processing capability of $RB_{\{m,h\}}$.

We consider that each BS_m maintains a set of orthogonal channels, denoted as $\mathbb{C}_m = \{c_1, c_2, \cdots\}$, all with a bandwidth of b_m. Besides, γ_m is the wireless coverage range of BS_m. Both MDs and ESs have the option to register either as transaction clients or validators within the blockchain network. Smart contracts are utilized to automatically trigger these transactions, aiming to maximize the social welfare in the BMEC system. A transaction on the blockchain [10] is composed of:

- MD's address;
- ES's address;
- SA's number;
- Payload data: payment value and the auxiliary information (e.g., information of the RB);
- MD's signature: the publicly verifiable digital signature of the MD.

After each transaction, the details of MDs, ESs and SAs will be recorded on the blockchain in form of the payload data associated with that transaction. Specifically, all the ESs and MDs can select an SA to help them sell/purchase resources,according to their own preference. Then, the SA generates the corresponding transaction results according to its received agent requests.

System Workflow. The interactions between various participants in the BMEC system are modeled as follows.

(1) Each ES_m checks the information of SAs by calling smart contracts, and then submit its selected SA and the information of resource bundles to the smart contract.

(2) Each MD_n checks the information of SAs by calling smart contracts, and then submit its selected SA and application task $A_n = \{D_n(V_n, E_n), f_n, t_n, (u_n, v_n)\}$ to the smart contract.

(3) Each SA_k matches resources and tasks by calling smart contracts, and then determines prices for each task.

(4) SAs provide the matching results to the MDs and ESs, which decide whether to accept the results.

(5) SAs generate the transactions based on feedback of MDs and ESs, and then broadcasts it.

(6) Stakeholders vote for different SAs and their transactions.

(7) The SA that receives the highest ballots publishes the new block.

2.2 Locality Constraints

Communication Distance Constraint. In real MEC system, due to the limited wireless transmission range, ES_m is qualified to serve MD_n through BS_m when the communication distance between ES_m and MD_n is no more than the distance threshold γ_{k_m}, satisfying

$$(x_m - u_n)^2 + (y_m - v_n)^2 \leq \gamma_m, \tag{1}$$

which means that the matching of MD and ES happens only when both of them are located within the coverage range of adjacent BS.

Wireless Channel Conflict Constraint. It is imperative for MD to apply for enough wireless channels from its

TABLE 1
Notations in the BMEC model.

Notation	Description
MD, SA, BS, ES	set of mobile device MD, service agent SA, base station BS, edge server ES.
A_n	application task request of MD_n.
$RB_{\{m,h\}}$	provision profile of the h-th resource bundle of ES_m.
$D_n(V_n, E_n)$	directed task graph, where V_n denotes subtasks, and E_n denotes task dependencies.
f_n, $f_{(m,h)}$	processing capability of MD_n and $RB_{\{m,h\}}$, respectively.
t_n	latency constraint of A_n.
(u_n, v_n), (x_m, y_m)	location coordinates of MD_n and ES_m (or BS_m), respectively.
γ_m	wireless coverage range of BS_m.
b_m	bandwidth of a single channel of the associated BS_m.
$\varphi_{\{(m,h),n\}}$	task offloading solution for task A_n to resource bundle $RB_{\{m,h\}}$.
$T^{min}_{\{(m,h),n\}}$	minimum completion latency of offloading A_n to ES_m with $RB_{\{m,h\}}$.
$V^{MD}_{\{(m,h),n\}}$, $V^{RB}_{\{(m,h),n\}}$	nodes executed locally and on ES for offloading A_n to $RB_{\{m,h\}}$.
$p_{\{m,h\}}$, $\theta_{\{m,h\}}$	price of resource bundle $RB_{\{m,h\}}$ and profit of selling $RB_{\{m,h\}}$, respectively.
$d^{tran}_{(m,h),n}$	wireless transmission data volume from MD_n to ES_m with $RB_{\{m,h\}}$ (or from ES_m to MD_n).
t^{tran}_n, t^{comp}_n	transmission latency and computation latency of A_n, respectively.
pow_n	transmit power of A_n.
$W_{m,n}$	channel gain of wireless link between MD_n and BS_m.
σ^2	variance of a zero-mean additive white Gaussian noise power.
$d^{MD}_{(m,h),n}$, $d^{ES}_{(m,h),n}$	local calculation data volume and the ES calculation data volume, respectively.
$v_{m,n}$	data transmission rate between MD_n and ES_m, and $v_{m,n} = \log_2(1 + \frac{pow_n \cdot W_{m,n}}{\sigma^2})$.
$\lambda^{comp}_{m,h}$, λ^{tran}_m	cost of processing a unit data of A_n and utilizing a single channel of per unit time of BS_m.
$C^{comp}_{(m,h),n}$, $C^{tran}_{(m,h),n}$	computation cost and transmission cost of processing A_n by $RB_{\{m,h\}}$, respectively.
μ_n, ρ_n	maximum budget and profit of MD_n, respectively.
$\Delta_{M,N}$	social welfare of BMEC system with M ESs and N MDs.

adjacent BS for transmit task data. Corresponding channels are allocated to MD_n if the number of channels available on BS_m is greater than the number of channels $\mathbb{C}_{(m,n)}$ required by MD_n. In fact, when multiple MDs compete for channel allocation from a common BS, the likelihood of channel conflict significantly increases, leading to substantial communication breakdowns.

Let $\zeta_{\{n,i\}}(m) = 1$ denote that the channel c_i of BS_m is allocated to MD_n, and $|\mathbb{C}_m|$ denote the number of elements in set $\mathbb{C}_m$. To avoid channel conflict, the total requested channel of involved MDs in BS_m needs to satisfy

$$\sum_{n,i} \zeta_{\{n,i\}}(m) \leq |\mathbb{C}_m|, \tag{2}$$

which requires that BSs have enough free bandwith sources, making each MD allocated with a non-overlapping channel.

Task Latency Constraint. The task latency of an application task A_n is composed of the computation time t^{comp}_n and communication time t^{tran}_n between MD_n and ES. To be specific, the wireless transmission of task data with size of $d^{tran}_{(m,h),n}$ between MD_n and $RB_{\{m,h\}}$ leads to t^{tran}_n, which can be computed up according to the Shannon's formula [30]:

$$t^{tran}_n = \frac{d^{tran}_{(m,h),n}}{\sum_n \zeta_{\{n,i\}}(m) \cdot b_m \cdot \log_2(1 + \frac{pow_n \cdot W_{m,n}}{\sigma^2})}, \tag{3}$$

where $\sum_n \zeta_{\{n,i\}}(m)$ is the number of channels (to MD_n) allocated by BS_m , pow_n and $W_{m,n}$ are the transmit power of A_n and the channel gain of wireless channel between MD_n and BS_m, respectively, and σ^2 is the variance of a zero-mean additive white Gaussian noise power. Note that, during the execution of task A_n, other network parameters, e.g., pow_n, $W_{m,n}$, and σ^2, remain constant [23].

Suppose $d^{MD}_{(m,h),n}$ and $d^{RB}_{(m,h),n}$ represent the local calculation data volume and the ES calculation data volume, respectively. The computation latency t^{comp}_n, generated by implementing the task A_n on $RB_{\{m,h\}}$, is calculated by

$$t^{comp}_n = \max\{\frac{d^{MD}_{(m,h),n}}{f_n}, \frac{d^{RB}_{(m,h),n}}{f_{(m,h)}}\}. \tag{4}$$

Then, with consideration of the maximum latency constraint t_n of A_n, the task latency of A_n needs to satisfy

$$t^{tran}_n + t^{comp}_n < t_n. \tag{5}$$

2.3 Computation and Communication Models

The resource allocation between ESs and MDs serves for implementation of task A_n, which inevitably leads to the implementation costs. Typically, the costs of implementing task A_n consist of the usage cost of both computing and communication resources. It is assumed that the cost of processing the traffic of A_n using the resource bundle $RB_{\{m,h\}}$ is proportional to the amount of traffic to be processed. Let $\lambda_{m,h}^{comp}$ be the cost of processing a unit data of A_n. The computation cost $C_{(m,h),n}^{comp}$ of using the resource bundle $RB_{\{m,h\}}$ to process task A_n can be formulated as

$$C_{(m,h),n}^{comp} = \lambda_{m,h}^{comp} \cdot d_{(m,h),n}^{RB}. \tag{6}$$

Suppose λ_m^{tran} denotes the cost of utilizing a single channel of per unit time of BS_m, the transmission cost $C_{(m,h),n}^{tran}$ of using resource bundle $RB_{\{m,h\}}$ to process task A_n can be expressed as

$$\begin{aligned} C_{(m,h),n}^{tran} &= \lambda_m^{tran} \cdot t_n^{tran} \cdot \sum_n \zeta_{\{n,i\}}(m) \\ &= \frac{\lambda_m^{tran} \cdot d_{(m,h),n}^{tran}}{b_m \cdot v_{m,n}}, \end{aligned} \tag{7}$$

where $v_{m,n} = \log_2(1 + \frac{pow_n \cdot W_{m,n}}{\sigma^2})$. Overall, the total cost $C_{(m,h),n}$ of utilizing resource bundle $RB_{m,h}$ to process A_n can be formulated as

$$C_{(m,h),n} = C_{(m,h),n}^{comp} + C_{(m,h),n}^{tran}. \tag{8}$$

2.4 Problem Formulation

To support flexible task offloading, we determine a set of feasible resource profiles and the corresponding task offloading solutions for each MD, which is called the strategy space of MDs. Each MD is allowed to match the resource bundle with the minimal cost of task offloading. If a successful match is not found, the system continues to search for the resource bundle with the second-lowest cost. In this way, MDs can flexibly offload their tasks according to the resource profiles.

Let $\varphi_{\{(m,h),n\}} = \{T_{\{(m,h),n\}}^{min}, V_{\{(m,h),n\}}^{MD}, V_{\{(m,h),n\}}^{RB}\}$ be a offloading solution for offloading MD_n's task A_n to ES_m with resource bundle $RB_{\{m,h\}}$, where $T_{\{(m,h),n\}}^{min}$, $V_{\{(m,h),n\}}^{MD}$, and $V_{\{(m,h),n\}}^{RB}$ are the minimum completion latency, the nodes executed locally, and the nodes to be executed remotely, respectively. Thus, for each MD_n, the flexible task offloading scheme is formulated as follows:

$$\min\ C_{(m,h),n} = C_{(m,h),n}^{comp} + C_{(m,h),n}^{tran}, \tag{9}$$

s.t.

$$\varphi_{\{(m,h),n\}} \in \Phi_n, \tag{9.1}$$

$$(x_m - u_n)^2 + (y_m - v_n)^2 \le \gamma_m, \tag{9.2}$$

$$\sum_{n,i} \zeta_{\{n,i\}}(m) \le \text{sum}(\mathbb{C}_m), \tag{9.3}$$

$$T_{(m,h),n} < t_n, \tag{9.4}$$

where Φ_n is the strategy space of MD_n.

We next give several basic definitions, and then formulate the social welfare maximization problem with the joint allocation of computing and communication resources.

Definition 1 (ES's Profit). ES_m sells its resource bundle $RB_{\{m,h\}}$ to MD_n at a certain price, which is denoted by $p_{\{m,h\}}$. Then the profit of ES_m is calculated by

$$\sum_h \theta_{\{m,h\}} = \sum_h (p_{\{m,h\}} - C_{(m,h),n}). \tag{10}$$

Definition 2 (MD's Profit). Following [17], we consider that each MD_n has a fixed maximum budget denoted as μ_n for its task A_n. If the price $p_{\{m,h\}}$ it pays for $RB_{\{m,h\}}$ is less than μ_n, MD_n will collect a profit of

$$\rho_n = \mu_n - p_{\{m,h\}}. \tag{11}$$

Note that only when $\mu_n > p_{\{m,h\}}$, MD_n wants to buy the resource bundle $RB_{\{m,h\}}$ from ES_m.

Definition 3 (Social Welfare). The primary economic goal of task allocation is to maximize the profits of all ESs/MDs in the system under locality constraints. Let $\Delta_{M,N}$ be the social welfare in BMEC with M ESs and N MDs, the $\Delta_{M,N}$ is formulated as

$$\begin{aligned} \Delta_{M,N} &= \sum_{m,h} \theta_{\{m,h\}} + \sum_n \rho_n \\ &= \sum_{m,h,n} (\mu_n - C_{(m,h),n}) \cdot l_{\{(m,h),n\}}. \end{aligned} \tag{12}$$

If MD_n matchs $RB_{(m,h)}$, the value of $l_{\{(m,h),n\}}$ is 1, otherwise $l_{\{(m,h),n\}} = 0$.

Definition 4 (Social Welfare Maximization in JCCA). Based on the above definitions, the social welfare maximization in JCCA is formulated as

$$\max\ \Delta_{M,N} = \sum_{m,h,n} (\mu_n - C_{(m,h),n}) \cdot l_{\{(m,h),n\}}, \tag{13}$$

s.t.

$$\sum_{n=1}^{N} l_{\{(m,h),n\}} \le 1, \quad l_{\{(m,h),n\}} \in \{0,1\}, \tag{13.1}$$

$$\sum_{n=1}^{N} \sum_{m=1}^{M} l_{\{(m,h),n\}} \cdot \mathbb{C}_{(m,n)} \le |\mathbb{C}_m|. \tag{13.2}$$

The formula (13.1) guarantees that for each MD_n, its task can only be offloaded to at most one resource bundle in its adjacent ESs. The formula (13.2) ensures that for each BS_m, the total number of its sub-channels for MDs cannot exceed the number of its own available sub-channels. Our goal is to maximize the total social welfare $\Delta_{M,N}$ under a set of locality constraints (Eq. 1, Eq. 2, and Eq. 5).

Theorem 1 (NP-hard): The social welfare maximization in JCCA is NP-hard.

Proof. In fact, the social welfare maximization in JCCA with constraints (13.1) and (13.2) is a two-dimensional knapsack problem [32], [33] which is proven to be NP-hard. It is obvious that the two-dimensional knapsack problem is a special case of the social welfare maximization in JCCA. Therefore, the social welfare maximization in JCCA is also NP-hard.

3 Design of NOLATA over Smart Contracts

The proposed NOLATA is designed as a multi-stage task matching game, which integrates smart contract based *selection of service agents*, *minimum delay task graph offloading algorithm*, *task matching and pricing strategy*, and *social welfare-driven PoS consensus mechanism*, detailed as follows.

3.1 The Selection of Service Agents

At the stage of SA selection, each ES/MD needs to select an SA to help them sell/purchase resources according to its preference, aiming to get highest utility, as shown in Algorithm 1.

For ESs, such preferences can be estimated by the combination of SA's reputation and the number of past MDs. The rationality of this approach lies in the fact that 1) the higher the reputation value of SA, the greater the likelihood of successful transactions; 2) the more MDs it has, the more tasks it will assign to ES. Therefore, the preference $\varrho_{m,k}$ of ES_m over SA_k is formulated as

$$\varrho_{m,k} = \eta_m \cdot \varpi_k^I + (1 - \eta_m) \cdot \psi_k, \tag{14}$$

where ϖ_k^i is the total number of MDs who chose SA_k in the first I round, $\psi_k \in [0,1]$ is the reputation of SA_k, which is initialized to 1, and $\eta_m \in [0,1]$ is the trade-off coefficient of ES_m.

For MDs, the preference $\tau_{n,k}$ of MD_n over SA_k can be estimated by the combination of SA's reputation and resource availability:

$$\tau_{n,k} = \eta_n \cdot \chi_{n,k} + (1 - \eta_n) \cdot \psi_k, \tag{15}$$

where $\chi_{n,k} = \frac{ava_{n,k}}{all_k} \in [0,1]$ is resource availability of SC_k (all_k is the number of RBs covered by SC_k, and $ava_{n,k}$ is the number of RBs available to MD_n), and η_n is the trade-off coefficient of MD_n.

The reputation value of SA_k after i rounds is decreased by

$$\psi_k^{i+1} = \psi_k^i \cdot (1 - \varepsilon)^{e_k^i}, \tag{16}$$

where $e_k^i = \hat{\Delta}_k^i - \Delta_k^i$, $\hat{\Delta}_k^i$ is the social welfare if all MDs choosing SA_k at round i have completed matching, Δ_k^i is the real social welfare obtained by SA_k, and $\varepsilon \in [0,1]$ is the trade-off coefficient.

Algorithm 1 Service Agent Preference Selection Strategy

Input: the set SA, ES, and MD, the trade-off coefficient η_n and η_m;
Output: the maximum preference set $res = \{\varrho_{1,k}, \cdots, \varrho_{M,k}, \tau_{1,k}, \cdots, \tau_{N,k}\}$;

$res = \phi$;
for $ES_m \in ES_{[M]}$ **do**
  $\varrho = 0$;
  for $SA_k \in SA_{[K]}$ **do**
    Calculate ψ_k according to Eq. 16;
    Calculate ϖ_k^I;
    Calculate $\varrho_{m,k}$ according to Eq. 14;
    if $\varrho < \varrho_{m,k}$ **then**
      $\varrho = \varrho_{m,k}$;
    end if
  end for
  $res = res \cup \varrho$;
end for
for $MD_n \in MD_{[N]}$ **do**
  $\tau = 0$;
  for $SA_k \in SA$ **do**
    Calculate ψ_k according to Eq. 16;
    $\chi_{n,k} = \frac{ava_{n,k}}{all_k}$;
    Calculate $\tau_{n,k}$ according to Eq. 15;
    if $\tau < \tau_{n,k}$ **then**
      $\tau = \tau_{n,k}$;
    end if
  end for
  $res = res \cup \tau$;
end for
return res;

3.2 Task Offloading Scheme in Smart Contracts

In our design, we utilize the smart contracts to perform two functions (accessible to users through message invocation) for facilitating flexible task offloading between MDs and ESs:

1) "Upload": At first, MDs and SAs are enabled to upload messages to the smart contract.
2) "Offload": MDs are enabled to offload a set of subtasks to ESs managed by their selected SAs.

After "Upload", we propose and perform a smart contract based *minimum delay task graph offloading algorithm* to solve the flexible task offloading problem in Eq. 9, as shown in Algorithm 2. First, for each node $s \in V_n$, we compute the local execution time and edge execution time, i.e., t_s^{MD} and t_s^{ES}, respectively (Lines 5-7). For each edge $(s_g, s_h) \in E_n$, we compute the transmission latency, i.e., t_{gh}^{tran} (Lines 8-10). Next, we initialize the minimum completion time $T_{(m,h),n)}^{min}$ and A to be $+\infty$ and $\{s_a, s_d\}$ (where s_a and s_d are the input node and output node of $D_n(V_n, E_n)$, respectivly) (Line 12). We repeatedly choose the subtask with the smallest gain and append it to A, where $\Omega(s) = t_s^{MD} - t_s^{ES} - \sum_{v \in A} t_{vs}^{tran}$ denotes the gain obtained by subtask s when it is executed at ES (Lines 13-21). Then we determine whether executing node s_l (with the maximum gain) on ES can reduce the overall execution time. If yes, subtask s_l will be executed at ES (Lines 23-27). Then, the task offloading solution outputs, involving the minimum completion time $T_{(m,h),n}^{min}$, the corresponding node sets $V_{(m,h),n}^{MD}$ for local execution and $V_{(m,h),n}^{ES}$ for ES execution (Line 33).

3.3 Task Matching and Pricing Strategy in Smart Contracts

Based on the aforementioned strategies, we now present the smart contract based *task matching and pricing strategy* for the social welfare maximization problem. First, we utilize the smart contract to perform two two main functions in this phase as follows:

1) "Upload": MDs and ESs upload messages to the smart contract.
2) "Matching": MDs purchase ES's computing and communication resources from SAs for these ESs.

Algorithm 2 Minimum Delay Task Graph Offloading Algorithm

Input: $A_n = \{D_n(V_n, E_n), f_n, t_n, (u_n, v_n)\}$, ES_m's geographical location (x_m, y_m), $RB_{m,h} = \{f_{(m,h)}\}$;
Output: $\varphi_{\{(m,h),n\}} = \{T^{min}_{\{(m,h),n\}}, V^{MD}_{\{(m,h),n\}}, V^{RB}_{\{(m,h),n\}}\}$

if $(x_m - u_n)^2 + (y_m - v_n)^2 > \gamma_m$ **then**
  return False;
end if
$\hat{V} = V_n$;
for $s \in V_n$ **do**
  $t^{MD}_s = d_s/f_n,\ t^{ES}_s = d_s/f_{(m,h)}$;
end for
for $(s_g, s_h) \in E_n$ **do**
  Calculate t^{tran}_{gh} according to Eq. 3;
end for
while $|V_n| > 1$ **do**
  $T^{min}_{\{(m,h),n\}} = +\infty,\ A = \{s_a, s_d\}$;
  while $A \neq V_n$ **do**
    $min = +\infty,\ n_{min} = null$;
    for $s \in V_n/A$ **do**
      $\Omega(s) = t^{MD}_s - t^{ES}_s - \sum_{v \in A} t^{tran}_{vs}$;
      if $\Omega(s) < min$ **then**
        $min = \Omega(s),\ s_{min} = s$;
      end if
    end for
    $A = A \cup s_{min}$;
  end while
  Select the last subtask added to A as s_l;
  $T_{\{A/\{s_l\}, s_l\}} = t^{MD}_n - [t^{MD}_s - t^{ES}_s - \sum_{v \in A} t^{tran}_{vs}]$;
  if $T_{\{A/\{s_l\}, s_l\}} < T^{min}_{\{(m,h),n\}}$ **then**
    $T^{min}_{\{(m,h),n\}} = T_{\{A/\{s_l\}, s_l\}}$, $V^{MD}_{\{(m,h),n\}} = A/\{s_l\}$, $V^{RB}_{\{(m,h),n\}} = \hat{V} - V^{MD}_n$;
  end if
  $V_n = V_n/\{s_l\}$;
end while
if $T^{min}_{\{(m,h),n\}} > t_n$ **or** $\sum_{n,i} \zeta_{\{n,i\}}(m) \leq \mathrm{num}(\mathbb{C}_m)$ **then**
  return False;
end if
return $\varphi_{\{(m,h),n\}} = \{T^{min}_{\{(m,h),n\}}, V^{MD}_{\{(m,h),n\}}, V^{RB}_{\{(m,h),n\}}\}$

Task Matching. To achieve maximum profit, each SA_k needs to consider a subset of tasks that may lead to maximum profit given limited resource bundles. Therefore, we transform the problem of qualified resource bundle selection by each MD into the problem of pursuing a minimum weight perfect matching in a bipartite graph.

In bipartite graph $G_k = (V_k, E_k)$ of SA_k, the node set V_k contains two disjoint subsets $\mathcal{A}$ and $\mathcal{RB}$, representing the set of task A_n and resource bundle $RB_{\{m,h\}}$, respectively. The edge $(A_n, RB_{\{m,h\}})$ denotes the match between A_n and $RB_{\{m,h\}}$ under locality constraints. Recall that the profit received by ES_m due to serving task A_n is

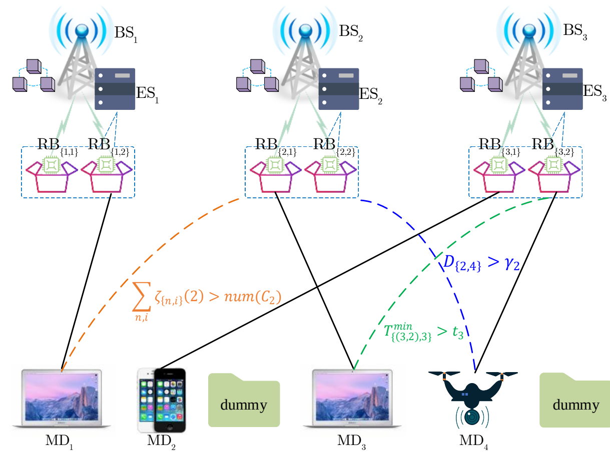


Fig. 3. An example of the constructed bipartite graph G_k.

$\sum_h \theta_{\{m,h\}} = \sum_h (p_{\{m,h\}} - C_{(m,h),n})$. Since $p_{\{m,h\}}$ is not determined, maximizing the profits of all ESs managed by SA_k means minimizing their cost of performing tasks. We thus consider the cost of matching A_n and $RB_{\{m,h\}}$ as the weight of edge $(A_n, RB_{\{m,h\}})$, i.e., $\omega(A_n, RB_{\{m,h\}}) = C_{(m,h),n}$.

As shown in Fig. 3, the resource bundle $RB_{\{m,h\}}$ can serve task A_n of MD_n *iff* all locality constraints are satisfied. For example, the distance $D_{2,4}$ between ES_2 (BS_2) and A_4 does not satisfy the distance constraint, thus A_4 cannot match resource bundles $RB_{\{2,1\}}$ and $RB_{\{2,2\}}$. Additionally, the minimum latency of completing task A_3 over $RB_{\{3,2\}}$ exceeds the latency constraint of t_3, thus A_3 cannot match resource bundle $RB_{\{3,2\}}$. Moreover, $RB_{\{2,1\}}$ is not able to allocate enough communication resources to task A_1, thus A_1 cannot match resource bundle $RB_{\{2,1\}}$. We set the weight of edge $(A_n, RB_{\{m,h\}})$ as ∞ if either locality constraint is not satisfied, which indicates the illegal matching between A_n and $RB_{\{m,h\}}$. Besides, it is highly likely that $|\mathcal{A}|$ is not equal to $|\mathcal{RB}|$. In this situation, we will add enough *dummy* nodes for the party with fewer nodes to ensure that $|\mathcal{A}| = |\mathcal{RB}|$. The weight of the edge associated with the *dummy* node is set to infinity.

With constructed G_k, we can find a minimum weighted perfect matching $\mathbb{M}_k$, using the HK matching algorithm [27]. The $\mathbb{M}_k$ will minimize the cost of all ESs managed by SA_k, i.e., each edge $(A_n, RB_{\{m,h\}})$ in $\mathbb{M}_k$ enables MD_n to obtain resource bundle $RB_{\{m,h\}}$ so that task A_n can be completed at minimum cost.

Opportunity Cost Principle of Conflict Resolution. For multiple MDs, all of which request the same channel of BS_m/ES_m that has requested resource bundles $(RB_{\{m,1\}}, \cdots, RB_{\{m,h\}})$, if the free channel of BS_m can only satisfy one of them, then the wireless channel conflict happens. In addition, if multiple MDs can achieve the lowest task implementation cost on the same resource bundle $RB_{\{m,h\}}$ of ES_m, this will inevitably lead to resource bundle conflict between MDs, similar to the wireless channel conflict. To resolve these conflict, we design an opportunity cost strategy to determine which one of multiple MDs has right to access the requested channel (or resource bundle).

Definition 5 (Opportunity Cost). Given $MD_O = \{MD_1, ..., MD_N\}$ and their matched ES_m, if only one MD can be selected as winner, the opportunity cost of ES_m is the highest-valued foregone alternative of $MD_O \backslash MD_n$ when a choice is made (i.e., $MD_n \in MD_O$ is selected), defined as

$$OC_m^n = \max\{\theta_{\{(m,h),n\}}(MD_z) | \forall\, z \neq n, 1 \leq z \leq N\}, \tag{17}$$

where $\theta_{\{(m,h),n\}}$ is the profit of the resource bundle $RB_{m,h}$ requested by MD_n of ES_m, which is calculated by Eq. 10.

This opportunity cost strategy can optimize the allocation of limited resources. When multiple MDs request the same wireless channel (or resource bundle) of BS_m (or ES_m), the opportunity cost of Eq. 17 is used as the measure to determine the winner. That is, the MD with lowest opportunity cost will win the competition of getting the wireless channel (or resource bundle).

Task Pricing. Each MD/ES is selfish and it only chooses the most beneficial scheme. Aiming to incentive each task A_n to select it, each ES_m follows the second price principle [34] to set the price of its resource bundle severing for each task. The price set by ES_m for the task should not be higher than the second lowest cost set by current ESs for it. This is the highest price that ES_m would like to get away with charging A_n; charging any more would give some ESs an incentive to compete for this task, which ensures that both seller ES_m and buyer MD_n receive the maximum benefit simultaneously.

3.4 Social Welfare-Driven PoS Consensus Mechanism

Due to the limited computing and communication resources in the target BMEC system, conventional consensus mechanisms such as PoW and PoS often suffer from inefficiency and wealth inequality [35], [36]. To handle the above issue, we utilize the stake of each stakeholder by the profits they can earn from the respective transaction, and then introduce a social welfare-driven PoS mechanism for BMEC, where the stakeholder with the highest ballot proportion is granted the right to generate a new block.

In our design, a single block is published in each time slot, and each epoch is made up by several consecutive time slots. At the starting of each epoch, a genesis block logs all stakeholders holding stakes. Any MD, ES, or SA can participate in the blockchain as a stakeholder based on their stakes (i.e., profit or reputation). However, MD/ES stakeholders cannot publish blocks, to purchase/sell resources, they must vote for their selected SA stakeholder to help them obtain the right to publish blocks. SA stakeholders compete for the right to publish blocks to fulfill the tasks of MD stakeholders, sell the resources of ES stakeholders, and earn rewards from blockchain. Note that only the SA whose normalized reputation is greater than the threshold can register as a stakeholder who can publish blocks. The proposed mechanism includes three stages within each epoch:

Stage 1 (Ballot allocation): At the starting of each epoch, MD and ES stakeholders are allocated to different ballot proportions based on their stakes.

In the conventional PoS mechanism, the stake relies on the coinage. However, the BMEC system emphasizes the maximization of social welfare. In this context, we define the stakes of stakeholders as the profits they can earn from the respective transaction. The ballot proportion of MD stakeholder i and ES stakeholder j are defined by Eq. 18 and Eq. 19, respectively.

$$\kappa_i = \frac{\rho_i}{\sum_n \rho_n + \sum_m \theta_m}, \tag{18}$$

$$\kappa_j = \frac{\theta_j}{\sum_n \rho_n + \sum_m \theta_m}, \tag{19}$$

where ρ_i and θ_j are the profits of MD stakeholder i and ES stakeholder j, respectively.

Stage 2 (Voting of stakeholders): MD stakeholders and ES stakeholders vote for its selected SA stakeholder. The SA stakeholder that receives the highest ballots publishes the new block.

Stage 3 (Legitimacy verification): At each validation stage, the SA stakeholder that receives the highest ballots generates and then broadcasts the information about its transaction. Due to the transparency and verifiability of the number of profits and ballots, all the stakeholders can verify the legitimacy of the block publisher. If the validation fails, they will continue to validate the SP stakeholder that can obtain the second highest ballots. Then, the first SP stakeholder that is successfully validated will publish block.

3.5 NOLATA Algorithm

To sum up, Algorithm 3 shows the main procedures of the proposed NOLATA, which is a multi-stage task matching game via integrating the aforementioned strategies and mechanisms, where SAs and MDs are strategic players. At **Stage 1**, MDs and ESs select an SA to purchase resources or help them sell resources (Line 4). At **Stage 2**, the SA obtains a feasible and flexible policy set for task offloading for each MD by invoking *the minimum delay task graph offloading algorithm* through smart contract (Lines 5-8). At **Stage 3**, the SA decides which resource bundle completes MD's task and sets the corresponding price through *the task matching and pricing strategy* in smart contract (Lines 9-15). And at **Stage 4**, blockchain nodes validate the transaction that achieves the highest ballot proportion (Lines 16-19).

4 Near Optimal and Stability Analysis for NOLATA

In this section, we first give basic definitions, and then analyze the properties of the proposed NOLATA.

4.1 Definitions and Lemmas

Definition 6 (Potential Game). Let K be the number of SPs, A be the set of action profiles over the action sets L_i of each ES and θ_i be the payoff function. Given a game $\mathbb{G} = (L_1, L_2, \cdots, L_K, \theta_1, \theta_2, \cdots, \theta_K)$, we call the function $\Delta : L_1 \times L_2 \times \cdots \times L_K \to \mathbb{R}$ a potential function for $\mathbb{G}$ if $\forall i \in \{1, 2, \cdots, K\}$, $\forall l_{-i} \in L_{-i}$, $\forall l_i, \tilde{l}_i \in L_i$, and

$$\theta_i(l_i, l_{-i}) - \theta_i(\tilde{l}_i, l_{-i}) = \Delta(l_i, l_{-i}) - \Delta(\tilde{l}_i, l_{-i}). \tag{20}$$

Then we call a game that has a potential function as a **potential game**, where there exists a Nash Equilibrium.

Definition 7 (PoA). Price of Anarchy (PoA) = (global optimal solution / the minimum value of the NE solution). In principle, the larger the PoA, the worse the NE solution.

Algorithm 3 NOLATA: Near Optimal Locality-Aware Task Allocation

Input: the sets $MD_{[N]}$, $ES_{[M]}$, and $SA_{[K]}$;
Output: final transaction results;

Each MD_n calls the "upload" function and sends task A_n, budget μ_n to smart contract;
Each ES_m calls the "upload" function and reports its basic information to smart contract (e.g., the number of RBs);
Smart contract verifies the information of MDs and ESs;
/* **Stage 1.** MDs and ESs select an SA. */
Invoke Algorithm 1;
/* **Stage 2.** The smart contract obtains the policy space PS_n for each MD_n. */
for $MD_n \in MD_{[N]}$ **do**
    Invoke Algorithm 2;
    PS_n=$PS_n \cup \varphi_{\{(m,h),n\}}$;
end for
/***Stage 3.** Task matching and pricing*/
Construct the bipartite graph $G_k = (V_k, E_k)$;
Find the minimum weight perfect matching $\mathbb{M}_k$;
if channel conflict or RB conflict **then**
    Use opportunity cost principle to select winners, according to Eq. 17;
end if
Set the price for each MD_n based on the second lowest cost;
Generate the transactions according to MDs/SPs' choices;
/***Stage 4.** Blockchain nodes validate the transaction*/
Smart contract generates and broadcasts transactions for each SA;
Distribute the ballot proportion to each stakeholder according to Eq. 18 and Eq. 19;
Each stakeholder votes for the transaction they support;
Blockchain nodes validate the transaction that achieves the highest ballot proportion;
return Final transaction results.

Definition 8 (Regret). For the social welfare maximization problem, the regret of the game is defined as the expected deviation of the social welfare of the obtained solution from the optimal solution in each round, i.e., $regret = E\left[\Delta - \Delta^*\right]$, where Δ denotes the social welfare obtained by NOLATA and Δ^* is the obtained social welfare based on the optimal strategy.

Lemma 1 (Individual Rationality): The proposed task matching game for the social welfare maximization problem in BMEC is individual rational.

Proof. For each MD_n, it has a budget μ_n for fulfilling its own task A_n. For A_n, it gathers a profit of $\mu_n - p_{\{m,h\}}$ if $RB_{\{m,h\}}$ is matched. Since an MD only buys a resource bundle $RB_{\{m,h\}}$ if $\mu_n > p_{\{m,h\}}$ at the cost of $p_{\{m,h\}}$, the mechanism is individual rational for MDs. For each ES_m, its profit depends on the prices it sets for tasks and the cost of executing them. Since the price it sets for each A_n is not higher than the second lowest implementation cost of the ES for A_n, which makes the MD to consider ES_m as its best choice, and then leads to a non-negative profit, the game thus is individual rational for ESs as well. Besides, the social welfare of the BMEC system is composed of the total profits of all MDs and ESs, and according to our matching principle, the social welfare of the BMEC system will be non-negative.

Lemma 2 (Potential Game): The proposed NOLATA for the social welfare maximization problem in the task matching is a **potential game** with potential function $\Delta_{M,N}$.

Proof. Showing that the game of NOLATA is a potential game, we need to show that if ES_m changes its strategy L, then the change in social welfare $\Delta_{M,N}$ is exactly the change in ES_m's revenue. To prove this, let's assume that ES_m chooses to quit the game. If ES_m drops out, each A_n that was served by ES_m switches to its second best choice $\tilde{ES}_m$. For task A_n, $\tilde{ES}_m$ is the second best position of ES_m, then the cost $p_{\{m,h\}}$ of $\tilde{ES}_m$ is the price of ES_m. Assuming the price of $\tilde{ES}_m$ is $\tilde{p}_{\{\tilde{m},\tilde{h}\}}$, the changing value of social welfare is the social welfare when strategy $\tilde{L}$ is adopted minus that when strategy L is adopted, i.e.,

$$\begin{aligned} & \Delta_{M,N}(\tilde{L}) - \Delta_{M,N}(L) \\ = \ & [\tilde{p}_{\{\tilde{m},\tilde{h}\}} - p_{\{m,h\}} + \mu_n - \tilde{p}_{\{\tilde{m},\tilde{h}\}}] \\ - \ & [p_{\{m,h\}} - C_{(m,h),n} + \mu_n - p_{\{m,h\}}] \\ = \ & C_{(m,h),n} - p_{\{m,h\}}, \end{aligned} \tag{21}$$

where $C_{(m,h),n}$ is the cost of ES_m, $\tilde{L}$ is the strategy of $\tilde{ES}_m$, and L is the strategy of ES_m.

In strategy $\tilde{L}$, the revenue of ES_m is $\theta_{\{m,h\}}(\tilde{L}) = 0$, and in strategy L, the revenue of ES_m is $\theta_{\{m,h\}}(L) = p_{\{m,h\}} - C_{(m,h),n}$. So the revenue change of ES_m is

$$\begin{aligned} & \theta_{\{m,h\}}(\tilde{L}) - \theta_{\{m,h\}}(L) \\ = \ & C_{(m,h),n} - p_{\{m,h\}} \\ = \ & \Delta_{M,N}(\tilde{L}) - \Delta_{M,N}(L). \end{aligned} \tag{22}$$

The proposed game thus is a potential game.

Lemma 3 (Submodularity): $\Delta(A)$ is **submodular**. As the number of elements in the strategy set increases, the function increment caused by adding a single element to the strategy set decreases monotonically, also known as the diminishing marginal utility. Mathematically, for any strategy set $P \subseteq Q \subseteq R$, $\exists \{e\} \in R - Q$, we have $\Delta_{M,N}(\{e\} \cup Q) - \Delta_{M,N}(Q) \leq \Delta_{M,N}(\{e\} \cup P) - \Delta_{M,N}(P)$.

Proof. The social welfare of the task matching game consists of the revenues of both ESs and MDs, i.e., $\Delta_{M,N} = \sum_{n=1}^{N} \rho_n + \sum_{m=1}^{M} \theta_m$, and the strategy set for ESs is the set of all resource bundles they own. Let P be their selected set of resource bundles to serve MDs so that an NE can be reached. To show that $\Delta_{M,N}$ is submodular, we need to show that the marginal benefit to $\Delta_{M,N}$ is diminishing as more resource bundles are added into set P to serve the

MDs. By rewriting the social welfare, we have

$$\begin{aligned} &\Delta_{M,N}(P) = \sum_{m=1}^{M} \theta_m + \sum_{n=1}^{N} \rho_n \\ = &\sum_{r\in R}\sum_{A_n\in A} [\mu_n - p_{\{m,h\}} + p_{\{m,h\}} - C_{(m,h),n}] \\ = &\sum_{r\in R}\sum_{A_n\in A} [\mu_n - C_{(m,h),n}]. \end{aligned} \tag{23}$$

We assume that $\{e\}$ is a better resource bundle for any task compared with those in set R, and A is the set of all tasks. Similarly, we have

$$\begin{aligned} &\Delta_{M,N}(P\cup\{e\}) \\ = &\sum_{r\in\{P\cup\{e\}\}} [\sum_{A_n\in A} (\mu_n - C_{(m,h),n})] \\ = &\ \Delta_{M,N}(P) + (\mu_n - C^{\{e\}}_{(m,h),n}) - (\mu_n - C_{(m,h),n}) \\ = &\ \Delta_{M,N}(P) + C_{(m,h),n} - C^{\{e\}}_{(m,h),n}, \end{aligned} \tag{24}$$

where A_n is the task assigned to resource bundle $\{e\}$ under strategy P whose implementation cost on resource bundle $\{e\}$ is $C^{\{e\}}_{(m,h),n}$, and

$$\begin{aligned} &\Delta_{M,N}(Q\cup\{e\}) \\ = &\sum_{r\in\{Q\cup\{e\}\}} [\sum_{A_{n'}\in A} (\mu'_n - C_{(m,h),n'})] \\ = &\ \Delta_{M,N}(Q) + (\mu'_n - C^{\{e\}}_{(m,h),n'}) \\ - &\ (\mu'_n - C_{(m,h),n'}) \\ = &\ \Delta_{N,M}(Q) + C_{(m,h),n'} - C^{\{e\}}_{(m,h),n'}, \end{aligned} \tag{25}$$

where $A_{n'}$ is the task assigned to resource bundle $\{e\}$ under strategy Q whose implementation cost on resource bundle $\{e\}$ is $C^{\{e\}}_{(m,h),n'}$.

It is clear that in the original strategy P, the saving in the implementation cost for task A_n is higher than that of task $A_{n'}$; otherwise, task $A_{n'}$ will be assigned to $\{e\}$ instead of A_n. Therefore, we have

$$C_{(m,h),n'} - C^{\{e\}}_{(m,h),n'} \le C_{(m,h),n} - C^{\{e\}}_{(m,h),n}, \tag{26}$$

which also means that

$$\begin{aligned} &\Delta_{M,N}(\{e\}\cup Q) - \Delta_{M,N}(Q) \\ \le &\ \Delta_{M,N}(\{e\}\cup P) - \Delta_{M,N}(P). \end{aligned} \tag{27}$$

Thus, the submodular of $\Delta_{M,N}(P)$ is proven.

4.2 Nash Equilibrium and Stablility

Theorem 2: (Price of Anarchy) The proposed mechanism NOLATA can obtain the NE solution, and the PoA of the NE solution obtained is at most 2.

Proof. We first prove that the solution obtained by the proposed mechanism is an NE solution. Recall that $\Delta_{M,N}$ is the social welfare we defined in the task matching game. As shown in Lemma 2, $\Delta_{M,N}$ is the potential function. Let Q be a pure strategy vector maximizing $\Delta_{M,N}(Q)$. Since any move by player B_i that results in a new strategy vector $\tilde{Q}$, $\Delta_{N,M}(\tilde{Q}) \le \Delta_{M,N}(Q)$ holds. By Lemma 2, we further have $\beta_i(\tilde{Q}) - \beta_i(Q) = \Delta_{M,N}(\tilde{Q}) - \Delta_{M,N}(Q) \le 0$, where β_i is the profit of player B_i. Obviously, B_i's profit can not increase from any move, and thus Q is stable. Note that any strategy Q with the property that $\Delta_{M,N}(Q)$ cannot be decreased by altering any one strategy in Q is an NE solution. This also means that the proposed mechanism converges to NE.

Let Q be the set of resource bundles selected at the NE, and O be the strategy space of the socially optimal solution. Obviously, $\Delta_{M,N}(Q) \le \Delta_{M,N}(Q\cup O)$ holds. Assuming that O_i is the strategy of the first i player in the optimal solution, we have (for ease of writing, we omit the subscript {N,M})

$$\begin{aligned} &\Delta(O) - \Delta(Q) \le \Delta(Q\cup O) - \Delta(Q) \\ = &\sum_{i=1}^{N+M} [\Delta(Q\cup O_i) - \Delta(Q\cup O_{i-1})]. \end{aligned} \tag{28}$$

By Lemma 3, we have

$$\begin{aligned} &\Delta(Q\cup O_i) - \Delta(Q\cup O_{i-1}) \\ = &\ \Delta(Q\cup O_i \cup \{o_i\}) - \Delta(Q\cup O_{i-1}) \\ \le &\ \Delta(Q - \{q_i\}\cup\{o_i\}) - \Delta(Q-\{q_i\}) \le \beta_i(Q), \end{aligned} \tag{29}$$

where o_i and q_i are the strategy of the player B_i in the optimal solution and the NE solution, respectively. We can further obtain

$$\Delta(O) - \Delta(Q) = \sum_{i=1}^{M+N} \beta_i(Q) = \Delta(Q) \tag{30}$$

Based on Eq. 30, we finally construct

$$\Delta_{M,N}(O) \le 2\Delta_{M,N}(Q), \tag{31}$$

which means that the PoA of the proposed mechanism is at most 2. The **Theorem 2** is proved.

4.3 Near Optimal Solution

Theorem 3 (Near Optimal Solution): Algorithm NOLATA has a regret of $\frac{(I+1)}{aJ}\ln K$.

Proof. Let's first assume that $\sum_i e^i_k = E_k$, and $SA^{I+1}_* = \text{argmax}_{1\le k\le K}\ \psi^{I+1}_k$.

Let $\Phi^I = \sum_{k=1}^K \psi^I_k$ be the total reputation of round I. The reputation of each SA after round I then is

$$\psi^{I+1}_k = \psi^I_k \cdot \Pi^T_{t=1}(1-\varepsilon)^{e^I_k} = (1-\varepsilon)^{E_k}. \tag{32}$$

Due to $1-\varepsilon \in [0,1]$ and the fact that $\forall\ 0 < a < b$, we have $(1-\varepsilon)^a \ge (1-\varepsilon)^b$. Then, the Φ^{I+1} satisfies

$$\Phi^{I+1} \ge \psi^{I+1}_* = (1-\varepsilon)^{E_*} \ge (1-\varepsilon)^{\Delta^*}. \tag{33}$$

The relation between the reputation and the social welfare in the game can be formulated as

$$\frac{\Phi^{i+1}}{\Phi^i} = \frac{\sum_{k=1}^K \psi^{i+1}_k}{\Phi^i} = \frac{\sum_{k=1}^K \psi^i_k \cdot (1-\varepsilon)^{e^i_k}}{\Phi^i}. \tag{34}$$

Let's first define $p^i_k = \frac{\psi^i_k}{\Phi^i}$. Then, due to the fact that $\forall a, b \ge$

$0, x > 0$, we have $(1-\varepsilon)^x < 1 - ax + bx^2$. We can obtain

$$\begin{aligned}\frac{\Phi^{i+1}}{\Phi^i} &\le \frac{\sum_{k=1}^{K}\psi_k^i[1 - a\cdot e_k^i + b\cdot (e_k^i)^2]}{\Phi^i} \\ &\le K\cdot\sum_{k=1}^{K}[(1 - a\cdot e_k^i + b\cdot (e_k^i)^2)\cdot p_k^i],\end{aligned} \tag{35}$$

Since $e_k^i = \hat{\Delta}_k^i - \Delta_k^i \ll \infty$, we can find a positive integer J that holds $e_k^i \le J\cdot\Delta_k^i$. Then Eq. 35 can be expressed as

$$\begin{aligned}\frac{\Phi^{i+1}}{K\cdot\Phi^i} &\le 1 - aJ\sum_{k=1}^{K}(\Delta_k^i p_k^i) + bJ^2\sum_{k=1}^{K}((\Delta_k^i)^2 p_k^i) \\ &= 1 - aJ\cdot E[\Delta(i)] + bJ^2\cdot E[\Delta^2(i)],\end{aligned} \tag{36}$$

where $\Delta(i)$ is the social welfare in round i and $\Delta = \sum_i \Delta(i)$. By taking the logarithm operation on Eq. 36, and due to the fact that $\forall\ x > 0$, $\ln(1+x) < x$ holds, we have

$$\begin{aligned}\ln\frac{\Phi^{i+1}}{\Phi^i} &< \ln K(1 - aJ\cdot E[\Delta(i)] + bJ^2\cdot E[\Delta^2(i)]) \\ &< \ln K - aJ\cdot E[\Delta(i)] + bJ^2\cdot E[\Delta^2(i)].\end{aligned} \tag{37}$$

Further, we can have

$$\sum_{i=1}^{I}\ln\frac{\Phi^{i+1}}{\Phi^i} < I\ln K - aJ\cdot E[\Delta] + bJ^2\cdot E[\Delta^2]. \tag{38}$$

Based on Eq. 33, we have

$$\sum_{i=1}^{I}\ln\frac{\Phi^{i+1}}{\Phi^i} = \ln\Phi^{I+1} - \ln\Phi^1 \ge \ln(1-\varepsilon)^{\Delta^*} - \ln K. \tag{39}$$

If the value of b is taken as 0, we have

$$I\ln K - aJ\cdot E[\Delta] > \Delta^*\cdot\ln(1-\varepsilon) - \ln K. \tag{40}$$

Through linear transformation of Eq. 40, we can get

$$aJ\cdot E[\Delta] < (I+1)\ln K - \Delta^*\cdot\ln(1-\varepsilon). \tag{41}$$

That is to say,

$$E[\Delta - \Delta^*] < \frac{(I+1)}{aJ}\ln K - \Delta^*\cdot(\frac{\ln(1-\varepsilon)}{aJ} + 1). \tag{42}$$

If the value of a is taken as $\frac{1}{J}\ln\frac{1}{(1-\varepsilon)} > 0$, we have

$$regret = E[\Delta - \Delta^*] < \frac{(I+1)}{aJ}\ln K. \tag{43}$$

Then **Theorem 3** is proved.

5 Performance Evaluation

5.1 Experimental Setting

We conduct extensive simulations to evaluate the performance of NOLATA. The system parameters are given in Table 2. In addition, for each MD, we use the method in [37] to generate the directed task graph for evaluation. We follow a random distribution to generate the cost matrix (including computation cost and transmission cost) of ESs, with its values varied from 20 to 40. We then set the budget of MD as the average cost of all resource bundles serving this MD. The running time of each algorithm is obtained based on a machine with a 2.90GHz Intel i5-9400F CPU and 8 GiB RAM. Unless otherwise stated, these parameters are taken as the default.

TABLE 2
A comparision of different social welfare maximization mechanisms.

Description	Value
the number of MDs	[100,1000]
the number of ESs	[10,100]
the number of SAs	10
the number of RBs owned by ES_m	[10,20]
processing capability of $RB_{\{m,h\}}$	{8,10,12,14,16}GHz
the number of sub-channels of BS_m	{4,6,8,10,12,14}
the bandwidth of sub-channels	20 MHz

Then, we compare NOLATA with two popular allocation methods including: 1) Social Welfare Improved Double Auction (SWIDA) [10], which introduces a smart contract based double auction mechanism in BMEC to build up the leasing association between the sellers and buyers, and determine the pricing of the winners. 2) Truthful Combinatorial Double Auctions for Mobile Edge Computing (TCDA) [6], which employs a truthful combinatorial double auction mechanism with a dynamic pricing strategy.

5.2 Property Verification of Individual Rationality

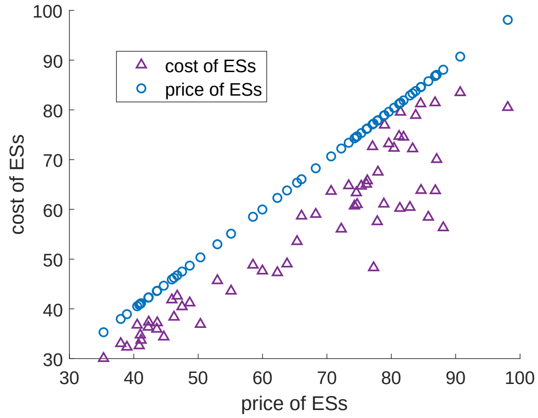
To assess the individual rationality of the NOLATA, we plot the curve of the cost of the ESs vs the price of the ESs, and the curve of the budget of the MDs vs the price of the ESs, as shown in Fig. 4. Fig. 4 (a) indicates that the cost of each ES always does not exceed its price. At the same time, Fig. 4 (b) shows that the price of each ES does not exceed the budget of the corresponding MD. These results demonstrate that all MDs and ESs can achieve non-negative profits, which encourages them to participate in our BMEC system. As a result, the NOLATA is individual rational.

5.3 Performance of NOLATA

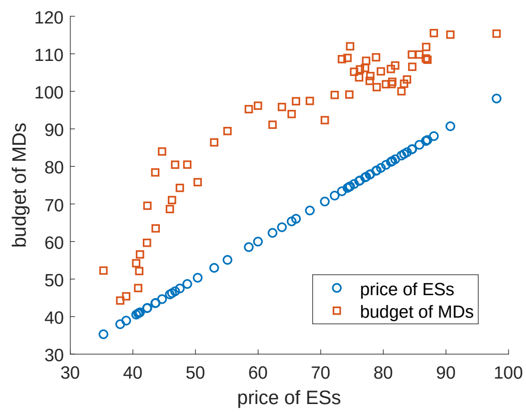
We evaluate our NOLATA using different performance indicators, including social welfare, runtime, and the profits of MDs and ESs, as shown in Figs. 5-6.

Social Welfare. From Fig. 5 (a), we can observe that the social welfare obtained by NOLATA is very close to the optimal social welfare, which support **Theorem 3**. At the same time, we can also observe that as the number of tasks increases, the effectiveness of NOLATA is very stable, which validates the near-optimality of NOLATA in the large-scale BMEC system, i.e., NOLATA can deal with a large number of tasks without sacrificing performance. Moreover, our NOLATA consistently demonstrates higher social welfare compared to TCDA and SWIDA. This is attributed to the fact that NOLATA encourages the MDs to offload tasks in a more flexible manner, providing them with increased opportunities to match resource bundles effectively.

Computational Efficiency. Fig. 5 (b) illustrates the comparison in terms of computational efficiency by recording the runtime results with an increasing number of tasks. TCDA doesn't require the consensus mechanism verification process since it operates within the MEC system, and thereby it has a shorter runtime. NOLATA operates

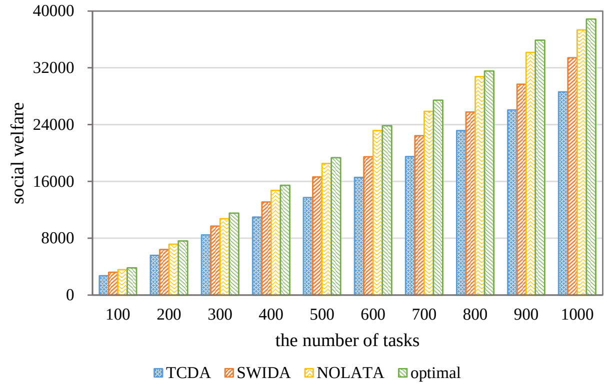


(a) ES's cost and ES's price

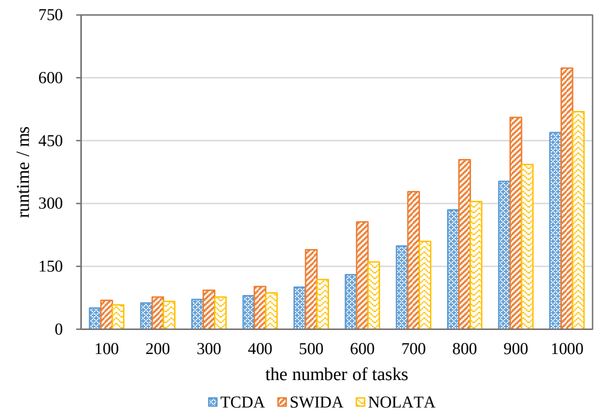


(b) MD's budget and ES's price

Fig. 4. Cost & Price & Budget



(a) Social welfare of NOLATA, SWIDA, and TCDA



(b) Runtime of NOLATA, SWIDA, and TCDA

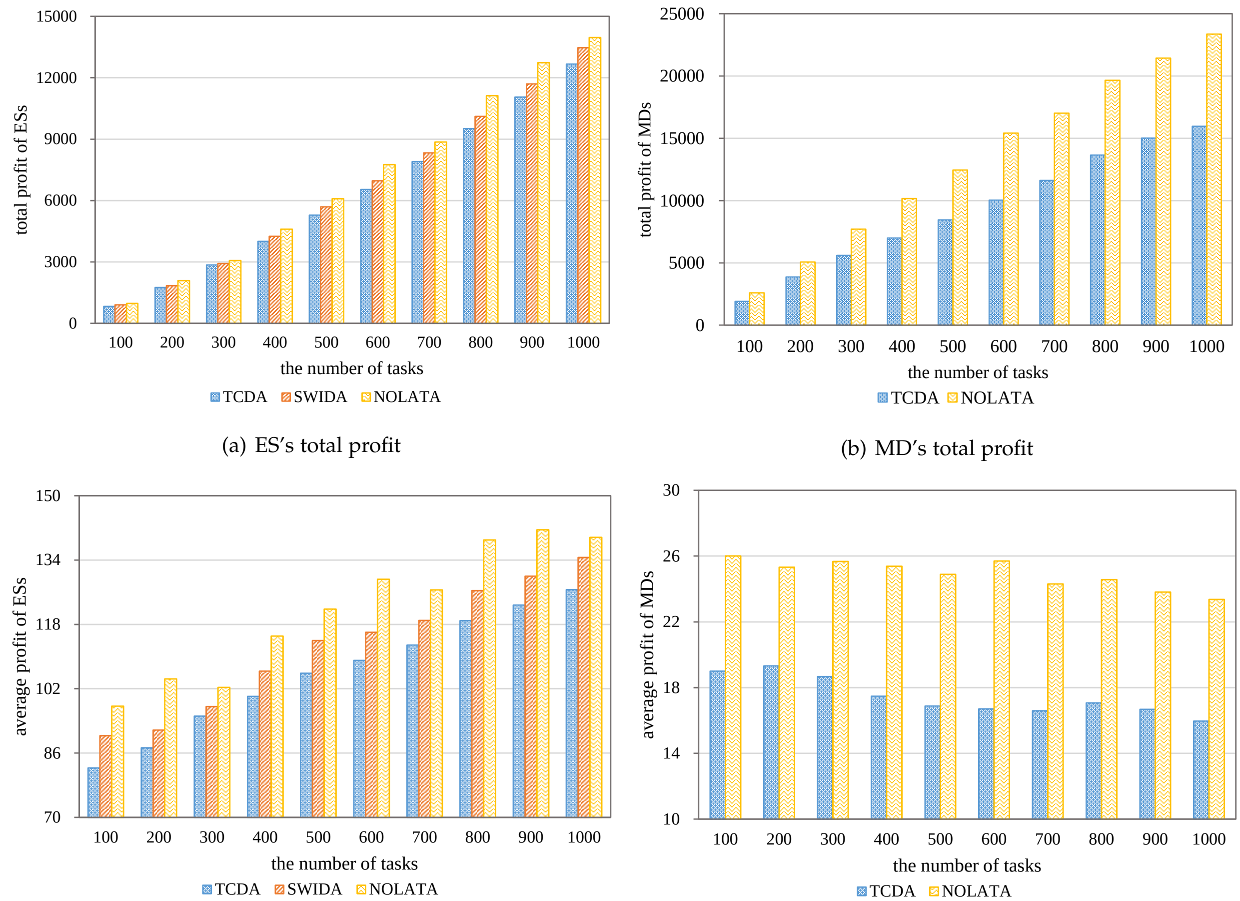
Fig. 5. Social welfare and runtime.

within BMEC and needs to perform social welfare-driven consensus mechanism, which requires computation cost. However, the runtime of NOLATA is only slightly longer than that of TCDA, which also exhibits the efficiency of the proposed consensus mechanism. Furthermore, the runtime of NOLATA is significantly lower than SWIDA, which uses consensus mechanism. This substantial difference is attributed to the efficiency of the consensus mechanism used in NOLATA, which allows all verifiers to unanimously and efficiently validate transactions that maximize the social welfare, eliminating the computation cost wasted in the voting process used by SWIDA.

ESs/MDs' Profits. Fig. 6 (a) - (b) show the total profits of ESs and MDs obtained by each method, respectively. We can see that the general trend is that the total profits of ESs and MDs grow accordingly with the increase of the number of tasks. However, unlike the average profit of ESs in Fig. 6 (c), which increases with the number of tasks, the average profit of MDs declines in a stable way, as shown in Fig. 6 (d). This is due to the fact that the fiercer competition among MDs contributes to higher pricing of ESs, which finally reduces the profit of MDs. Since SWIDA didn't consider the profit of MDs, we didn't show them in Fig. 6 (b) and Fig. 6 (d). Moreover, it is clear that the total profit and average profit of NOLATA is significantly better than that of its counterpart, because that NOLATA is able to enable flexible task offloading, resulting in more tasks being met.

5.4 Performance of Consensus Mechanisms

We consider that the reward for publishing a new block is proportional to the number of MDs successfully trading in the new block. That is to say, the SA (serving as miner) who completes more tasks will receive more rewards from the system. Therefore, the wealth level of different SAs can be represented by the number of tasks they have successfully matched, and the more tasks they have completed, the richer the SA will be. The SA whose task completion rate is below the poverty line (i.e. 70% of the average task), can be regarded as an impoverished SA. Fig. 7 compares the number of tasks completed by each SA in our proposed social welfare-driven PoS mechanism with the conventional PoS mechanism. Obviously, compared with conventional PoS, it is observed that our proposed social welfare-driven PoS mechanism can enable SAs to complete task matching more fairly. When the total number of tasks is fixed, conventional PoS mechanisms always generate a few excessively wealthy SAs and a large number of poor SAs, while our proposed mechanism has fewer poor SAs, and the number of tasks completed by different SAs is relatively uniform. Therefore, our proposed social welfare-driven PoS mechanism can reduce the wealth inequality among the SAs.



(a) ES's total profit

(b) MD's total profit

(c) ES's average profit

(d) MD's average profit

Fig. 6. Profits of ESs and MDs.

6 Related Work

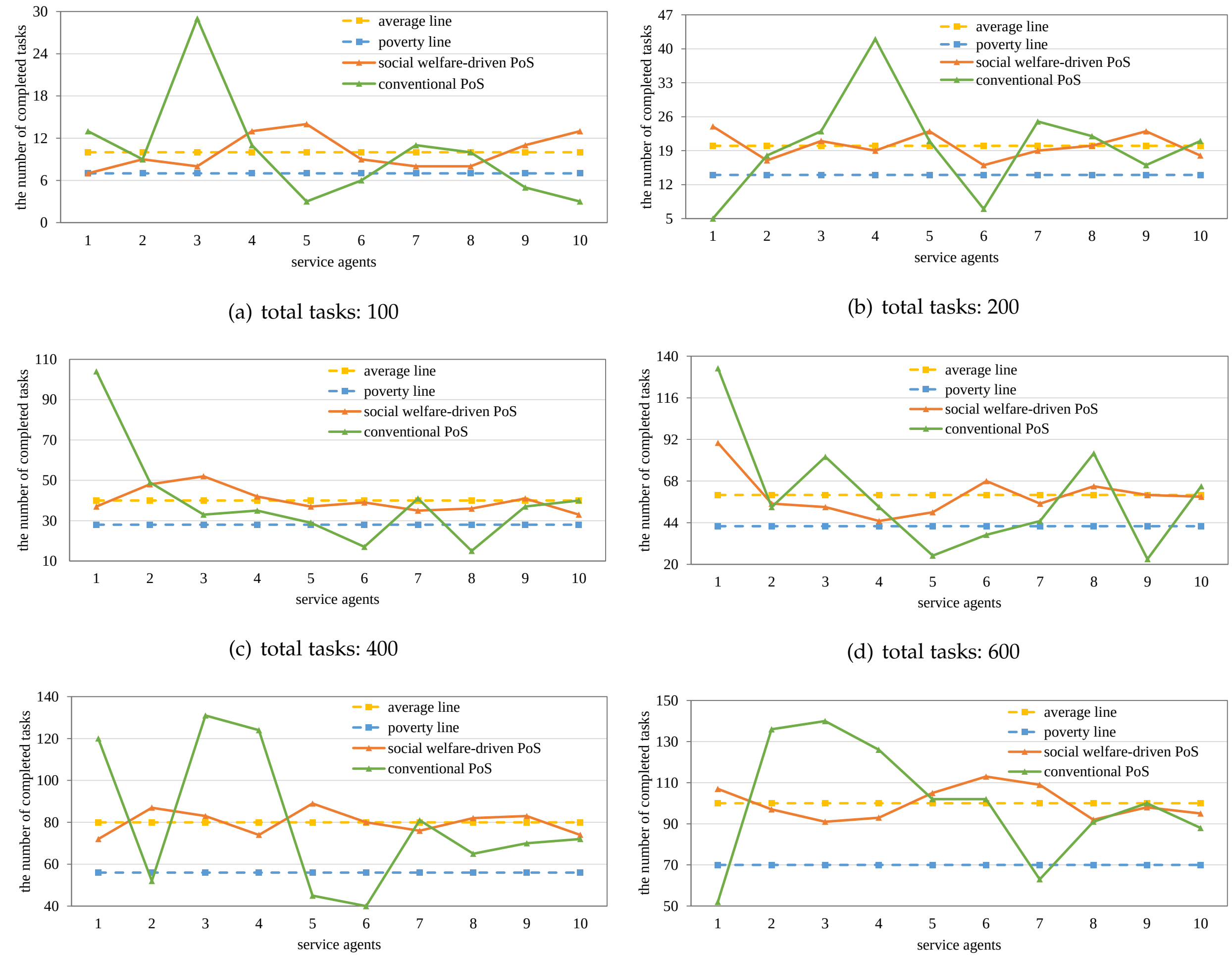
Recently, the design of resource/task allocation mechanisms for the BMEC system has received a surge of attention [10], [21], [22], [38], [39], [40], [41]. Pioneer attempts focus on the design of auction-based task allocation mechanisms [10], [38], [39], [41]. For example, [41] propose an auction-based cluster federated learning scheme. Besides, a smart contract based double auction mechanism [10] is developed and used in the BMEC system. However, this method focuses on the profit of sellers rather than those of both buyers and sellers, and thus cannot get the maximized total social welfare of the system. Aiming at the trading between computation service provider and blockchain miners, [38] proposes an auction-based model for efficient computing resource allocation. However, it concentrates on computing resource allocation, but ignores communication resources. Then, [39] proposes to explore a hierarchical auction model for both computing and communication resource allocation, which however neglects the impact of computation latency and cannot guarantee individual rationality.

Fortunately, the game theory becomes a promising way to deal with complex resource/task allocation problems in the BMEC system, where several game-based methods have been proposed and developed [21], [22], [40]. For example, a potential game mechanism for the task offloading of edge devices is designed and the existence of a Nash Equilibrium in the mechanism is analyzed [21]. [40] utilizes the mean field game to model the collective scheduling behaviours of all edge servers. [22] proposes to model the interactions between service providers (sellers) and miners (buyers) as a stochastic Stackelberg game under private information to maximize the profit of miners. However the above game theory based approaches do not consider various locality constraints in the practical BMEC system.

To better understand the comparison of our NOLATA with the existing works, the comparative study is summarized in Table 3. Most of existing mechanisms for the BMEC applications do not explicitly consider the locality constraints and flexible task offloading. Meanwhile, they cannot guarantee a set of desirable properties required by real applications, such as individual rationality, Nash Equilibrium, and stable approximation ratio. Therefore, it is urgently needed to devise effective task allocation mechanisms to bridge the gap between existing works and requirements of real BMEC system.

7 Conclusion

In this paper, we propose a near optimal locality-aware task allocation mechanism for the BMEC system. We first construct a social welfare maximization model with a set of locality constraints including, i.e., *communication distance*, *wireless channel conflict* and *task latency constraints*. Then, we



(a) total tasks: 100

(b) total tasks: 200

(c) total tasks: 400

(d) total tasks: 600

(e) total tasks: 800

(f) total tasks: 1000

Fig. 7. The number of tasks completed by each service agents under different consensus mechanisms.

TABLE 3
A comparision of different social welfare maximization mechanisms.

Mechanism	Properties[1]							
	FTO	JCC	CD	WCC	TL	IR	NE	NO
[10]	×	✓	×	×	×	✓	✓	×
[38]	×	×	×	×	×	✓	✓	✓
[39]	×	✓	×	×	✓	×	×	×
[21]	×	✓	×	✓	✓	×	✓	✓
[22]	×	×	×	×	×	×	✓	✓
[40]	×	✓	×	×	✓	×	×	✓
[41]	×	✓	×	×	×	✓	✓	×
NOLATA	✓	✓	✓	✓	✓	✓	✓	✓

[1] Flexible task offloading (FTO), Joint Computing and Communication Resource Allocation (JCC), Communication Distance (CD), Wireless Channel Conflict (WCC), Task Latency (TL), Individual Rationality (IR), Nash Equilibrium Solution (NE), Near-Optimality of Solutions (NO)

design a *multi-stage task matching game*, which integrates the preference-based selection of service agents, *the minimum delay task graph partitioning* for task offloading, and *social welfare-driven PoS consensus mechanism*. The significant merit of NOLATA is the theoretical guarantee of the desirable features including individual rationality, Nash Equilibrium, and stable approximation ratio, with explicit consideration of flexible task offloading in the joint computing and communication resource allocation process. Extensive simulation results verify the effectiveness and efficiency of the proposed NOLATA. In future work, we will extend the NOLATA mechanism to one-to-many MD-RB association scenario, where each MD can be served by multiple RBs at the same time.

Acknowledgment

This work is supported by the National Key R&D Program of China under Grant No. 2022YFB4500800.

References

[1] Y. Mao, C. You, J. Zhang, and K. Huang, "A survey on mobile edge computing: The communication perspective," IEEE communications surveys & tutorials, vol. 19, no. 4, pp. 2322–2358, August 2017.

[2] C. An, A, Zhou, and J. Pei, "Octopus: Exploiting the Edge Intelligence for Accessible 5G Mobile Performance Enhancement," IEEE/ACM Transactions on Networking, vol. 31, no. 6, pp. 2454 – 2469, 2023.

[3] C. Wang, Q. Hu, and D. Yu, "Online learning for failure-aware edge backup of service function chains with the minimum latency" IEEE/ACM Transactions on Networking, vol. 31, no. 6, pp. 2730 – 2744, 2023.

[4] C. Madarasingha, K. Thilakarathna, and A. Zomaya, "OpCASH: Optimized Utilization of MEC Cache for 360-Degree Video Streaming with Dynamic Tiling," IEEE International Conference on Pervasive Computing and Communications, pp. 34–43, 2022.

[5] S. Yue, J. Ren, N. Qiao, "TODG: Distributed task offloading with delay guarantees for edge computing," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 7, pp. 1650–1665, 2021.

[6] L. Ma, X. Wang, X. Wang, and L. Wang, "TCDA: Truthful Combinatorial Double Auctions for Mobile Edge Computing in Industrial Internet of Things," IEEE Transactions on Mobile Computing, vol. 21, no. 11, pp. 4125–4138, March 2021.

[7] Y. Chen, S. Zhang, Y. Jin, "LOCUS: User-perceived delay-aware service placement and user allocation in MEC environment," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 7. pp. 581–1592, 2021.

[8] H. Guo, J. Zhang, J. Liu, and H. Zhang, "Energy-aware computation offloading and transmit power allocation in ultradense IoT networks," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4317–4329, October 2018.

[9] Y.C, Wu, T.Q. Dinh, Y. Fu, and C. Lin, "A Hybrid DQN and Optimization Approach for Strategy and Resource Allocation in MEC Networks" IEEE Transactions on Wireless Communications, vol.20, no. 7, pp. 4282–4295, February 2021.

[10] Y. Du, Z. Wang, J. Li, and L. Shi, "Blockchain-Aided Edge Computing Market: Smart Contract and Consensus Mechanisms," IEEE Transactions on Mobile Computing, vol. 22, no. 6, pp. 3193–3208, January 2022.

[11] Z. Ma, S. Zhang, Z. Chen, "Towards revenue-driven multi-user online task offloading in edge computing," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 5, pp. 1185–1198, 2021.

[12] Y. Ding, K. Li, C. Liu, "A potential game theoretic approach to computation offloading strategy optimization in end-edge-cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 6, pp. 503–1519, 2021.

[13] Y. Zhang, M. A. Kishk, and M.S. Alouini, "Computation Offloading and Service Caching in Heterogeneous MEC Wireless Networks," IEEE Transactions on Mobile Computing, vol.22, no. 6, pp. 3241–3256, December 2021.

[14] H. Liu, S. Long, Z. Li Z, "Revenue Maximizing Online Service Function Chain Deployment in Multi-Tier Computing Network," IEEE Transactions on Parallel and Distributed Systems, vol. 34, no. 3, pp. 781–796, 2022.

[15] J. Lee, D. Kim, and D. Niyato, "Market Analysis of Distributed Learning Resource Management for Internet of Things: A Game-Theoretic Approach," IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8430–8439, May 2020.

[16] T. Bahreini, H. Badri, D. Grosu, "Mechanisms for resource allocation and pricing in mobile edge computing systems," IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 3, pp. 667–682, 2021.

[17] Z. Xu, H. Ren, W. Liang, Q. Xia, "Near Optimal and Dynamic Mechanisms Towards a Stable NFV Market in Multi-Tier Cloud Networks," IEEE International Conference on Pervasive Computing and Communications, pp. 1–10, 2021.

[18] S. Liu, Y. Yu, X. Lian, and Y. Feng, "Dependent Task Scheduling and Offloading for Minimizing Deadline Violation Ratio in Mobile Edge Computing Networks", IEEE Journal on Selected Areas in Communications, vol.41, no. 2, pp. 538–554, January 2023.

[19] S. He, Y. Lu, Q. Tang, "Blockchain-Based P2P Content Delivery With Monetary Incentivization and Fairness Guarantee," IEEE Transactions on Parallel and Distributed Systems, vol. 34, no. 2, pp. 746–765, 2022.

[20] L. Yuan, Q. He, S. Tan, "CoopEdge+: Enabling Decentralized, Secure and Cooperative Multi-Access Edge Computing Based on Blockchain," IEEE Transactions on Parallel and Distributed Systems, vol. 34, no. 3, pp. 894–908, 2022.

[21] D. C. Nguyen, M. Ding, P. N. Pathirana, and A. Seneviratne, "Cooperative Task Offloading and Block Mining in Blockchain-Based Edge Computing With Multi-Agent Deep Reinforcement Learning," IEEE Transactions on Mobile Computing, vol. 22, no. 4, pp. 2021–2037, October 2021.

[22] A. Asheralieva and D. Niyato, "Learning-Based Mobile Edge Computing Resource Management to Support Public Blockchain Networks," IEEE Transactions on Mobile Computing, vol. 20, no. 3, pp. 1092–1109, December 2019.

[23] S. Zhang, G. Cui, and Y. Long, " Joint computing and communication resource allocation for satellite communication networks with edge computing," China Communications, vol. 18, no. 7, pp. 236–252, 2021.

[24] J. Kang, Z. Xiong, and D. Niyato, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2906–2920, January 2019.

[25] L. Liu, J. Feng, Q. Pei, and C. Chen, "Blockchain-Enabled Secure Data Sharing Scheme in Mobile-Edge Computing: An Asynchronous Advantage Actor–Critic Learning Approach," IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2342–2353, December 2020.

[26] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "BEdgeHealth: A Decentralized Architecture for Edge-Based IoMT Networks Using Blockchain," IEEE Internet of Things Journal, vol. 8, no. 14, pp. 11743–11757, February 2021.

[27] N. Blum, "A simplified realization of the hopcroft-karp approach to maximum matching in nonbipartite graphs," Research report, Universit at Bonn, 1999.

[28] Y. Zuo, S. Jin, S. Zhang, and Y. Han, "Delay-Limited Computation Offloading for MEC-Assisted Mobile Blockchain Networks," vol. 69, no. 12, pp. 8569–8584, September 2021.

[29] Z. Lin, Z. Ni, and L. Kuang, "Dynamic beam pattern and bandwidth allocation based on multi-agent deep reinforcement learning for beam hopping satellite systems," IEEE Transactions on Vehicular Technology, vol. 71. no. 4 pp.3917–3930, April 2022.

[30] C. E. Shannon, "A mathematical theory of communication", The Bell system technical journal, vol. 27, no. 3, pp. 379–423, July 1948.

[31] A. A. Zarir, G. A. Oliva, Z. M. Jiang, and A. E. Hassan, "Developing cost-effective blockchain-powered applications: A case study of the gas usage of smart contract transactions in the ethereum blockchain platform," ACM Transactions on Software Engineering and Methodology, vol. 30, no. 3, pp. 1–38, March 2021.

[32] R. Wang and Z. Zhang, "Set Theory-Based Operator Design in Evolutionary Algorithms for Solving Knapsack Problems," IEEE Transactions on Evolutionary Computation, vol. 25, no. 6, pp. 1133–1147, May 2021.

[33] Z. Li, L. Tang, and J. Liu, "A Memetic Algorithm Based on Probability Learning for Solving the Multidimensional Knapsack Problem," IEEE Transactions on Cybernetics, vol. 52, no. 4, pp. 2284–2299, July 2020.

[34] S. Bikhchandani , "Reputation in repeated second-price auctions," Journal of Economic Theory, vol.46, no. 1, pp. 97–119, October 1988.

[35] K. Gai, J. Guo, L. Zhu, and S. Yu, "Blockchain Meets Cloud Computing: A Survey," IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 2009–2030, April 2020.

[36] R. Yang, F. R. Yu, P. Si, and Z. Yang, "Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges," IEEE Communications Surveys & Tutorials, vol. 21, no, 2, pp. 1508–1532, January 2019.

[37] C. Daniel, M. Grégory, P. Swann, and T. Denis, "Random graph generation for scheduling simulations," Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, 2010.

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

[39] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2082–2091, September 2017.

[40] W. Xiong, Y. Jiancheng, and L. S. C. John, "Decentralized Scheduling and Dynamic Pricing for Edge Computing: A Mean Field Game Approach," IIEEE/ACM Transactions on Networking, vol. 31, no. 3, pp. 965–978, September 2022.

[41] R. Lu, W. Zhang, and Y. Wang, "Auction-based cluster federated learning in mobile edge computing systems," IEEE Transactions on Parallel and Distributed Systems, vol. 34, no. 4, pp. 1145–1158, 2023.