Optimal Computational Power Allocation in Multi-Access Mobile Edge Computing for Blockchain

Yuan Wu [1,2,*] , **Xiangxu Chen** [1], **Jiajun Shi** [1], **Kejie Ni** [1], **Liping Qian** [1] , **Liang Huang** [1] **and Kuan Zhang** [3]

[1] College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China; 2111703318@zjut.edu.cn (X.C.); 2111703412@zjut.edu.cn (J.S.); 2111703062@zjut.edu.cn (K.N.); lpqian@zjut.edu.cn (L.Q.); lianghuang@zjut.edu.cn (L.H.)
[2] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
[3] Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, Omaha, NE 68182-0572, USA; kzhang22@unl.edu
* Correspondence: iewuy@zjut.edu.cn

check for updates

Abstract: Blockchain has emerged as a decentralized and trustable ledger for recording and storing digital transactions. The mining process of Blockchain, however, incurs a heavy computational workload for miners to solve the proof-of-work puzzle (i.e., a series of the hashing computation), which is prohibitive from the perspective of the mobile terminals (MTs). The advanced multi-access mobile edge computing (MEC), which enables the MTs to offload part of the computational workloads (for solving the proof-of-work) to the nearby edge-servers (ESs), provides a promising approach to address this issue. By offloading the computational workloads via multi-access MEC, the MTs can effectively increase their successful probabilities when participating in the mining game and gain the consequent reward (i.e., winning the bitcoin). However, as a compensation to the ESs which provide the computational resources to the MTs, the MTs need to pay the ESs for the corresponding resource-acquisition costs. Thus, to investigate the trade-off between obtaining the computational resources from the ESs (for solving the proof-of-work) and paying for the consequent cost, we formulate an optimization problem in which the MTs determine their acquired computational resources from different ESs, with the objective of maximizing the MTs' social net-reward in the mining process while keeping the fairness among the MTs. In spite of the non-convexity of the formulated problem, we exploit its layered structure and propose efficient distributed algorithms for the MTs to individually determine their optimal computational resources acquired from different ESs. Numerical results are provided to validate the effectiveness of our proposed algorithms and the performance of our proposed multi-access MEC for Blockchain.

Keywords: multi-access; mobile edge computing; computational power allocation; optimization; Blockchain

1. Introduction

Blockchain, a distributed and trustable architecture for recording and storing digital transactions, has been considered as one the promising mechanisms for enabling the secure cyber-physical systems [1]. In the framework of Blockchain, the miners participate in a mining game [2], and all miners compete with each other to be the first winner to solve the proof-of-work puzzle (which corresponds to executing a series of hashing computation). After solving the proof-of-work puzzle and broadcasting the mined block to other miners to reach the consensus, the winner can claim the consequent reward.

Nowadays, there exist several different mining pools (e.g., Slush Pool and AntPool), in which the miners can select to join and participate in the mining game/process [3]. Thanks to the nature of distributed management and independence from the central authorities, Blockchain has been expected to play a crucial role as a distributed and trustable ledger for recording and storing a variety of transactions and resource exchange, e.g., the crypto-currency and peer-to-peer electronic payment system [4], energy trading in smart grid [5,6], computation offloading in vehicular networks [7], radio resource exchange in wireless networks [8], and Internet of Things (IoTs) [9–12].

However, solving the proof-of-work puzzle requires consuming a significant amount of computational resources, which are prohibitive from the perspective of the mobile terminals (MTs). To address this difficulty, the MTs can exploit the recent advanced mobile edge computing (MEC) to enhance their computational capability [13,14]. In particular, the MEC enables the MTs to offload their computational workloads to the edge servers (ESs) equipped with sufficient computational resources and deployed at the edge of the radio access networks (e.g., the cellular base stations, BSs), which thus reduces the computation time and improves the resource utilization efficiency. As a result, MEC has been considered as one of the enabling technologies for realizing the future IoTs and the fifth generation (5G) visions which aim at providing massive connectivity, high access speed, and low latency [15]. The recent multi-access MEC, which can be enabled by the advanced Non-orthogonal multiple access (NOMA) [16–18], further allows the MTs to acquire computational resources from several ESs simultaneously, and thus yields a flexible exploitation of the computational resources from multiple ESs [19–21]. Therefore, by integrating MEC into the framework of Blockchain, the computational capability of the MTs can be effectively enhanced, which thus facilitates solving the proof-of-work puzzle and increases the successful probability in winning the mining game [22,23].

Thanks to its advantage, MEC has been exploited in many different paradigms, e.g., green IoT and vehicular networks [24–26]. In the following, we mainly review the studies about the resource management for MEC. Focusing on the single user's binary task-offloading, Melendez et al. proposed the offloading decisions for reducing the completion time [27]. Further taking into account the radio resource utilization in MEC, many studies have been devoted for the joint optimization of task offloading and radio resource allocation [28–30]. Optimal offloading strategies for partial computation offloading (in which the MTs are allowed to offload partial of their computational workloads) have been investigated in [31,32]. The scenario of multi-user MEC is close to our study here. Focusing on minimizing the delay in completing the computation tasks, there have been many studies investigating the proper multi-user computational resource allocation [33,34]. In [35], a multi-user computational resource allocation (i.e., the CPU cycle) scheme has been proposed to save the users' energy consumption. In [36], Jin et al. proposed an auction-based scheme for sharing the computational resource at the mobile edge. In [37], a revenue-maximization oriented computational resource allocation scheme has been proposed. In particular, some recent work investigated the exploitation of MEC for Blockchain. In [22], Xiong et al. exploited the MEC for Blockchain and investigated the resource management and pricing strategy. In particular, the paper formulated a two-stage Stackelberg game to maximize the profit of the edge service provider and the utilities of the miners. In [23], Liu et al. studied the computation offloading and content caching in wireless Blockchain networks with MEC. In particular, the economics based mechanisms have been envisioned as the promising approaches for enabling efficient yet distributed resource allocations in wired or wireless networks [38–40], in which the pricing mechanism and game theoretic modelling have been widely exploited. For instance, in [41], Ha et al. proposed time dependent pricing scheme for mobile data management. In [42], Tsiropoulou et al. proposed a novel joint customized price and power control for optimizing the energy-efficiency in multi-service wireless networks. In recent work [43], by taking into account the competitive wireless Internet service market, Vamvakas et al. proposed an efficient pricing strategy for the joint dynamic provider selection and the associated power allocation. In this work, we exploit the mechanism of the dual price (which is also referred as the shadow price)

to achieve an efficient allocation of the edge servers' computational resources to different mobile users/miners.

In this work, we investigate the optimal allocation of the computational resource/power in multi-access MEC enabled Blockchain. As described before, the multi-access MEC enables each MT to obtain the computational power from multiple ESs simultaneously (in the following, we use the term of computational power and the term of computational resource interchangeably). As a result, the MT can increase its successful probability when participating in the mining game and gain the consequent reward (i.e., winning the bitcoin). However, as a compensation to the ESs, the MT needs to pay for the consequent cost for acquiring the computational resources. Our contributions in this work can be summarized as follows:

- To investigate the trade-off between obtaining the computational resources from the ESs (for solving the proof-of-work) and paying for the consequent cost, we focus on a scenario in which a group of the MTs acquire the computational powers from a set of nearby ESs and pay for the consequent costs for acquiring the computational powers. Mathematically, we formulate an optimization problem in the MTs to determine their acquired computational powers from different ESs, with the objective of maximizing the MTs' total net-reward while keeping the fairness among the MTs.
- Despite the non-convexity of the formulated optimization problem, we exploit its layered structure and design two algorithms (one for the one-ES scenario and the other for the multi-ES scenario) to find the optimal solution efficiently. We also provide extensive numerical results to validate the effectiveness of our proposed algorithms and show the performance of our proposed multi-access MEC for Blockchain.

The reminder of this paper is organized as follows. We present the system model and problem formulation in Section 2. We first focus on the one-ES scenario and propose a distributed algorithm to compute the optimal solution in Section 3. We then consider the multi-ES scenario and propose a corresponding distributed algorithm to achieve the optimal solution in Section 4. We conclude this work in Section 5 and discuss the future directions.

2. System Model and Problem Formulation

System Model

We consider a system model as shown in Figure 1. Specifically, a group of the ESs denoted by $\mathcal{K} = \{1, 2, ..., K\}$ provide the computing services to a group of the MTs which are denoted by $\mathcal{I} = \{1, 2, ..., I\}$. Enabled by the multi-access MEC, each MT $i \in \mathcal{I}$ can acquire the computational power from the ESs simultaneously. Specifically, we use x_i^k to denote MT i's computational power from ES k and use x_i^{loc} to denote MT i's local computational power. Thus, each MT i's total computational power θ_i can be expressed as:

$$\theta_i = x_i^{\text{loc}} + \sum_{k \in \mathcal{K}} x_i^k, \forall i \in \mathcal{I}. \tag{1}$$

In this work, θ_i, x_i^{loc}, and x_i^k are all measured in the unit of GHash/sec. Furthermore, we introduce α_i to denote MT i's total computational power with respect to the overall computational power of all MTs, i.e.,

$$\alpha_i = \frac{\theta_i}{\sum_{i \in \mathcal{I}} \theta_i} = \frac{x_i^{\text{loc}} + \sum_{k \in \mathcal{K}} x_i^k}{\sum_{i \in \mathcal{I}} x_i^{\text{loc}} + \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} x_i^k}. \tag{2}$$
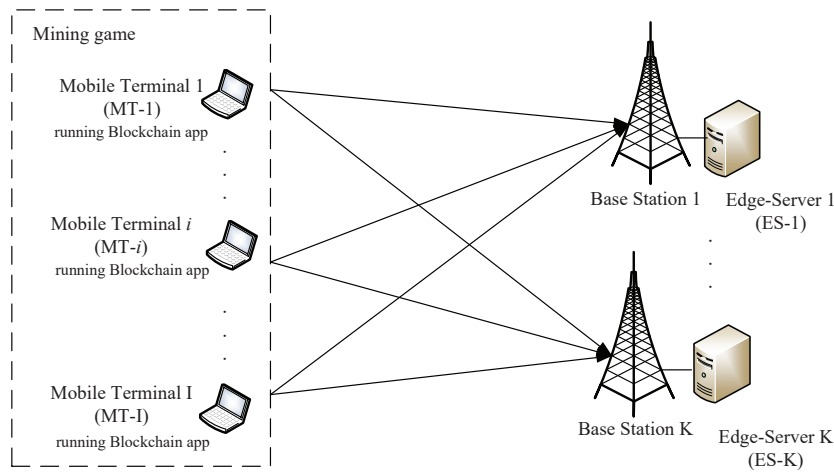
Figure 1. System model: a group of mobile terminals offload computational workloads to the edge servers.

In the mining game, the MTs compete against each other to be the first one to solve the proof-of-work puzzle and receive the reward accordingly. Similar to [22], we model the successful probability that MT *i* wins the mining game (including that MT *i* successfully mines the block and its solution reaches the consensus) as a random variable as follows:

$$P_i\left(\alpha_i, t_i\right) = \alpha_i\left(1 - \mathcal{P}_{orphan}\left(t_i\right)\right), \tag{3}$$

where t_i denotes MT *i*'s block-size, and function $\mathcal{P}_{orphan}\left(t_i\right)$ denotes the orphaning probability [22] as follows:

$$\mathcal{P}_{orphan}\left(t_i\right) = 1 - e^{-\lambda t_i}. \tag{4}$$

The use of the above $\mathcal{P}_{orphan}\left(t_i\right)$ can be explained as follows. After solving the proof-of-work puzzle, MT *i* needs to broadcast its result to other MTs for reaching the consensus. Since the broadcasting of the computation-result among the MTs suffers from a certain delay, it is possible that MT *i* fails to be the first one whose computation-result reaches the consensus among all MTs (even though that MT *i* is the first one who solves the proof-of-work puzzle). The orphaning probability $\mathcal{P}_{orphan}\left(t_i\right)$ in (4) quantifies such a probability. In particular, the same as [22,44], we express $P_{(t_i)}$ as the Poisson distribution with parameter λ, in which parameter λ denotes the inter-arrival rate of the Poisson distribution. Based on (3), we can express MT *i*'s net-reward function in winning the mining game as follows:

$$m_i = \left(R + rt_i\right) P_i\left(\alpha_i, t_i\right) - \sum_{k \in \mathcal{K}} p^k x_i^k, \tag{5}$$

where R denotes the fixed reward, and rt_i denotes the variable reward which linearly grows with MT *i*'s block-size t_i (parameter r is a fixed constant). Parameter p^k denotes the marginal price of ES k for providing the computational power to MT *i*.

In this work, we consider that the MTs acquire the computational power from the group of the ESs with the objective of maximizing the total net-reward, while keeping the fairness among them. To this end, we formulate the following total net-reward optimization (TRO) problem:

$$\text{(TRO):} \quad \max \sum_{i \in \mathcal{I}} \ln(m_i),$$

$$\text{subject to:} \quad \text{constraints (1), (2), (3), and (5),}$$

$$0 \le \sum_{i \in \mathcal{I}} x_i^k \le C^{k,\text{tot}}, \forall k \in \mathcal{K}, \tag{6}$$

$$\text{variables:} \quad x_i^k \ge 0, \forall i \in \mathcal{I}, k \in \mathcal{K}.$$

Constraint (6) ensures that all MTs' total computational power acquired from ES k cannot exceed ES k's maximum computational power $C^{k,\text{tot}}$. Notice that since both θ_i and α_i depend on $\{x_i^k\}_{k \in \mathcal{K}}$, we just treat $\{x_i^k\}_{i \in \mathcal{I}, k \in \mathcal{K}}$ as the decision variables in Problem (TRO). However, Problem (TRO) is a complicated non-convex optimization problem, and there exists no general algorithm that can solve it efficiently [45]. We will propose a distributed algorithm to compute the optimal solution in the next two sections. Specifically, in our proposed algorithm, each MT individually determines the acquired computational powers from different ESs. Then, viewing the aggregate demands from all MTs, the ESs further update their respective computational powers allocated to all MTs for maximizing the total net-reward. Thus, our algorithm does not require a central entity to collect the global information in the considered network. Nevertheless, the downside of our proposed algorithm is that it requires the message exchange between the MTs and the ESs, meaning that the MTs and ESs need to take the additional burdens on sending and receiving the required messages for reaching the optimal solution.

3. One-ES Scenario and Proposed Distributed Algorithm

3.1. Problem Formulation and Its Decomposition

We first consider one-ES scenario and aim at finding the optimal allocation of the computational resource for the MTs. For the sake of easy presentation, we use ES $k = 1$ as an example in the following. In particular, with one ES, Problem (TRO) turns into :

$$\text{(TRO-ES):} \quad \max \sum_{i \in \mathcal{I}} \ln \left((R + rt_i) \frac{x_i^{\text{loc}} + x_i^1}{\sum_{i \in \mathcal{I}} x_i^{\text{loc}} + \sum_{i \in \mathcal{I}} x_i^1} e^{-\lambda t_i} - p^1 x_i^1 \right),$$

$$\text{subject to:} \quad 0 \le \sum_{i \in \mathcal{I}} x_i^1 \le C^{1,\text{tot}},$$

$$\text{variables:} \quad x_i^1 \ge 0, \forall i \in \mathcal{I}.$$

However, Problem (TRO-ES) is still a non-convex optimization problem, which is difficult to solve in general. To efficiently solve Problem (TRO-ES), we adopt a vertical decomposition as follows. We firstly introduce an auxiliary variable μ to denote all MTs' total computational power obtained from ES 1, i.e.,

$$\sum_{i \in \mathcal{I}} x_i^1 = \mu, \tag{7}$$

with $0 \le \mu \le C^{1,\text{tot}}$.

Suppose that the value of μ is given in advance. We thus aim at solving the following subproblem:

$$\text{(TRO-ES-Sub):} \quad V_\mu^{\text{sub}} = \max \sum_{i \in \mathcal{I}} \ln(A_i x_i^1 + B_i),$$

$$\text{subject to:} \quad \sum_{i \in \mathcal{I}} x_i^1 = \mu, \tag{8}$$

$$A_i x_i^1 + B_i \geq 0, \forall i \in \mathcal{I}, \tag{9}$$

$$\text{variables:} \quad x_i^1 \geq 0, \forall i \in \mathcal{I},$$

with parameters A_i and B_i given by:

$$A_i = \frac{(R + rt_i)e^{-\lambda t_i}}{\sum_{i \in \mathcal{I}} x_i^{\text{loc}} + \mu} - p^1, \tag{10}$$

$$B_i = \frac{(R + rt_i)e^{-\lambda t_i}}{\sum_{i \in \mathcal{I}} x_i^{\text{loc}} + \mu} x_i^{\text{loc}}. \tag{11}$$

Notice that, in Subproblem (TRO-ES-Sub), we use V_μ^{sub} to denote the optimal value of Subproblem (TRO-ES-Sub), which depends on the given value of μ (i.e., the total computational power obtained from ES 1).

After solving Problem (TRO-ES-Sub) and obtaining V_μ^{sub} (for each given μ), we continue to find the optimal value of μ (denoted by μ^*) for maximizing V_μ^{sub}, i.e., solving the following top-problem:

$$\text{(TRO-ES-Top):} \quad \max V_\mu^{\text{sub}},$$

$$\text{variable:} \quad 0 \leq \mu \leq C^{1,\text{tot}}.$$

The reason for us to adopt the above proposed vertical decomposition is as follows. Given the value of μ, Subproblem (TRO-ES-Sub) is a strictly convex optimization (i.e., Proposition 1 provided below), which enables us to solve it efficiently. In the next subsection, we propose a distributed algorithm to solve Subproblem (TRO-ES-Sub) and Top-problem (TRO-ES-Top).

3.2. Proposed Algorithm to Solve Subproblem (TRO-ES-Sub)

To efficiently solve Subproblem (TRO-ES-Sub), we firstly identify the following property.

Proposition 1. *Given μ, Subproblem (TRO-ES-Sub) is a strictly convex optimization.*

Proof. Given the value of μ, the values of $\{A_i, B_i\}_{i \in \mathcal{I}}$ are all fixed. Thus, according to the convex optimization theory [45], Problem (TRO-ES-Sub) is a strictly convex optimization problem. □

The convexity of Subproblem (TRO-ES-Sub) enables us to use the Karush–Kuhn–Tucker (KKT) conditions [45] to compute the optimal solution. In particular, to solve Subproblem (TRO-ES-Sub), we identify the following three possible cases.

Case I in which we have $A_i < 0, \forall i \in \mathcal{I}$, and $\sum_{i \in \mathcal{I}} \left(-\frac{B_i}{A_i}\right) \geq \mu$.

In Case I, we define $C_i = -A_i, \forall i \in \mathcal{I}$. With $\{B_i, C_i\}_{i \in \mathcal{I}}$, Problem (TRO-ES-Sub) can be equivalently changed into Problem (TRO-ES-Sub-I):

$$\text{(TRO-ES-Sub-I):} \quad V_\mu^{\text{sub}} = \max \sum_{i \in \mathcal{I}} \ln(B_i - C_i x_i^1),$$

$$\text{subject to:} \quad \sum_{i \in \mathcal{I}} x_i^1 \geq \mu, \tag{12}$$

$$\text{variables:} \quad 0 \leq x_i^1 \leq \frac{B_i}{C_i}, \forall i \in \mathcal{I}.$$

Problem (TRO-ES-Sub-I) is again a strictly convex optimization problem. Moreover, it can be observed that constraint (12) is strictly binding at the optimum. Thus, we introduce the dual variable λ to relax (12) and obtain the Lagrangian function as (where the subscript "I" stands for Case I):

$$L_{\mathrm{I}}(\{x_i^1\}_{i\in\mathcal{I}}, \lambda) = \sum_{i\in\mathcal{I}} \ln(B_i - C_i x_i^1) + \lambda\left(\sum_{i\in\mathcal{I}} x_i^1 - \mu\right). \tag{13}$$

With the KKT condition and (13), we can derive the optimal solution for Problem (TRO-ES-Sub-I) as follows:

$$x_i^{1*} = \max\left\{\frac{B_i}{C_i} - \frac{1}{\lambda^*}, 0\right\}, \forall i \in \mathcal{I}, \tag{14}$$

where λ^* is determined according to the following condition:

$$\sum_{i\in\mathcal{I}} x_i^{1*} = \mu. \tag{15}$$

Based on (14) and (15), we can propose the following distributed algorithm (i.e., Algorithm 1) to solve Problem (TRO-ES-Sub-I). Notice that, in Algorithm 1, exploiting the monotonic property of (14), we use the bisection-search (i.e., from Step 3 to Step 11) to find λ^* until condition (15) is satisfied.

Algorithm 1: To solve Problem (TRO-ES-Sub-I) and determine $\{x_i^1\}^*$ and V_μ^{sub}

    Input: Each MT i's A_i and B_i.
    Initialization: ES 1 sets $\overline{\lambda}$ as a sufficiently large number and $\underline{\lambda} = 0$. Set the tolerance for the
    computational error ε as a small number.
    while $|\overline{\lambda} - \underline{\lambda}| > \varepsilon$ **do**
        ES 1 sets $\lambda^{\mathrm{cur}} = \frac{\overline{\lambda} + \underline{\lambda}}{2}$ and broadcasts λ^{cur} to all MTs.
        Each MT $i \in \mathcal{I}$ sets $x_i^1 = \max\left\{\frac{B_i}{C_i} - \frac{1}{\lambda^{\mathrm{cur}}}, 0\right\}$ and reports x_i^1 to ES 1.
        if $\sum_{i\in\mathcal{I}} x_i^1 > \mu$ **then**
            ES 1 updates $\overline{\lambda} = \lambda^{\mathrm{cur}}$.
        else
            ES 1 updates $\underline{\lambda} = \lambda^{\mathrm{cur}}$.
        end if
    end while
    ES 1 sets $\lambda^* = \lambda^{\mathrm{cur}}$ and broadcasts λ^* to all MTs in $\mathcal{I}$.
    Each MT $i \in \mathcal{I}$ sets $x_i^{1*} = \max\left\{\frac{B_i}{C_i} - \frac{1}{\lambda^*}, 0\right\}$ and reports the value of $\ln(A_i x_i^{1*} + B_i)$ to ES 1.
    Output: ES 1 outputs $V_\mu^{\mathrm{sub}} = \sum_{i\in\mathcal{I}} \ln(B_i - C_i x_i^{1*})$ based on all MTs' reports.

Case II in which there exists a subset of MT $i \in \mathcal{I}$ with $A_i > 0$. In particular, we denote this subset as $\mathcal{I}_{\mathrm{sub}} = \{i \in \mathcal{I}|A_i > 0\}$.

In Case II, we can derive the optimal solution of Problem (TRO-ES-Sub) as follows. For each MT i with $A_i < 0$, we set $x_i^* = 0$ directly. For the MTs in $\mathcal{I}_{\mathrm{sub}}$, we can express Problem (TRO-ES-Sub) as:

$$\text{(TRO-ES-Sub-II):} \qquad V_\mu^{\mathrm{sub}} = \max \sum_{i\in\mathcal{I}_{\mathrm{sub}}} \ln(A_i x_i^1 + B_i),$$

$$\text{subject to:} \qquad \sum_{i\in\mathcal{I}_{\mathrm{sub}}} x_i^1 \leq \mu, \tag{16}$$

$$\text{variables:} \qquad x_i^1 \geq 0, \forall i \in \mathcal{I}_{\mathrm{sub}}.$$

It can be observed that Subproblem (TRO-ES-Sub-II) is a strictly convex optimization problem, and the optimal solution occurs when constraint (16) is binding. We thus introduce the dual variable λ to relax (16) and obtain the following Lagrangian function:

$$L_{\mathrm{II}}(\{x_i^1\}_{i\in\mathcal{I}},\lambda) = \sum_{i\in\mathcal{I}_{\mathrm{sub}}} \ln(A_i x_i^1 + B_i) + \lambda(\mu - \sum_{i\in\mathcal{I}_{\mathrm{sub}}} x_i^1). \tag{17}$$

With the KKT condition and (17), we can derive the optimal solution for Subproblem (TRO-ES-Sub-II) as follows:

$$x_i^{1*} = \max\left\{\frac{1}{\lambda^*} - \frac{B_i}{A_i}, 0\right\}, \forall i \in \mathcal{I}_{\mathrm{sub}}, \tag{18}$$

with λ^* determined according to the following condition

$$\sum_{i\in\mathcal{I}_{\mathrm{sub}}} x_i^{1*} = \mu. \tag{19}$$

Based on (18) and (19), we can propose the following distributed algorithm (i.e., Algorithm 2) to solve Problem (TRO-ES-SubII). Notice that, in Algorithm 2, exploiting the monotonic property of (18), we use the bisection-search (i.e., from Step 4 to Step 12) to find λ^* until condition (19) is satisfied.

Algorithm 2: To solve Problem (TRO-ES-Sub-II) and determine $\{x_i^{1*}\}$ and V_μ^{sub}

Input: Each MT i's A_i and B_i.

Initialization: ES 1 sets $\overline{\lambda}$ as a sufficiently large number and $\underline{\lambda} = 0$. Set the tolerance for the

computational error ε as a small number.

MT $i \in \mathcal{I}$ sets $x_i^{1*} = 0$ if its $A_i \leq 0$ and reports to ES 1.

while $|\overline{\lambda} - \underline{\lambda}| > \varepsilon$ **do**

  ES 1 sets $\lambda^{\mathrm{cur}} = \frac{\overline{\lambda}+\underline{\lambda}}{2}$ and broadcasts λ^{cur} to the MTs in $\mathcal{I}_{\mathrm{sub}}$.

  Each MT $i \in \mathcal{I}_{\mathrm{sub}}$ sets $x_i^1 = \max\left\{\frac{1}{\lambda^{\mathrm{cur}}} - \frac{B_i}{A_i}, 0\right\}$ and reports x_i^1 to ES 1.

  if $\sum_{i\in\mathcal{I}} x_i^1 > \mu$ **then**

    ES 1 updates $\underline{\lambda} = \lambda^{\mathrm{cur}}$.

  else

    ES 1 updates $\overline{\lambda} = \lambda^{\mathrm{cur}}$.

  end if

end while

ES i sets $\lambda^* = \lambda^{\mathrm{cur}}$ and sends λ^* to all MTs in $\mathcal{I}_{\mathrm{sub}}$.

Each MT $i \in \mathcal{I}_{\mathrm{sub}}$ sets $x_i^{1*} = \max\left\{\frac{1}{\lambda^{\mathrm{cur}}} - \frac{B_i}{A_i}, 0\right\}$ and reports the value of $\ln(A_i x_i^{1*} + B_i)$ to ES 1.

Output: ES 1 outputs $V_\mu^{\mathrm{sub}} = \sum_{i\in\mathcal{I}_{\mathrm{sub}}} \ln(A_i x_i^{1*} + B_i) + \sum_{i\in\mathcal{I}, i\notin\mathcal{I}_{\mathrm{sub}}} \ln(B_i)$ based on all MTs' reports.

<u>*Case III*</u> in which we have $A_i < 0, \forall i \in \mathcal{I}$ and moreover, $\sum_{i\in\mathcal{I}}\left(-\frac{B_i}{A_i}\right) < \mu$. Case III is a trivial case in which Problem (TRO-ES-Sub) is infeasible.

As a summary of the above Case I, Case II, and Case III, we propose the following Algorithm 3 to solve Problem (TRO-ES-Sub) and determine $\{x_i^{1*}\}_{i\in\mathcal{I}}$ and V_μ^{sub}. In Algorithm 3, we use Algorithm 1 (in Step 7) and Algorithm 2 (in Step 10) as the subroutines. Until now, we have completed solving Problem (TRO-ES-Sub) for the given value of μ.

Algorithm 3: To solve Problem (TRO-ES-Sub) and determine $\{x_i^1\}^*$ and V_μ^{sub}

Input: The value of μ.

Each MT $i \in \mathcal{I}$ reports its x_i^{loc} to ES 1.

ES 1 computes $X^{\text{loc}} = \sum_{i \in \mathcal{I}} x_i^{\text{loc}}$ and sends X^{loc} to all MTs.

Each MT i uses (10) to compute $A_i = \frac{(R+rt_i)e^{-\lambda t_i}}{X^{\text{loc}}+\mu} - p^1$, and uses (11) to compute $B_i = \frac{(R+rt_i)e^{-\lambda t_i}}{X^{\text{loc}}+\mu} x_i^{\text{loc}}$.

Each MT reports its (A_i, B_i) to ES 1.

if Case I holds **then**

    Use Algorithm 1 to output $\{x_i^{1*}\}_{i \in \mathcal{I}}$ and V_μ^{sub}.

else

    if Case II holds **then**

        Use Algorithm 2 to output $\{x_i^{1*}\}_{i \in \mathcal{I}}$ and V_μ^{sub}.

    else

        Output that Problem (TRO-ES-Sub) is infeasible under the current μ.

    end if

end if

3.3. Proposed Algorithm to Solve Top-Problem (TRO-ES-Top)

We continue to solve Top-problem (TRO-ES-Top) in this subsection. Notice that, for each given μ, we can use Algorithm 3 to obtain the value of V_μ^{sub}. However, the difficulty in solving Top-problem (TRO-ES-Top) lies in that we cannot derive V_μ^{sub} analytically. As a result, Top-problem (TRO-ES-Top) is an optimization problem in which the objective function cannot be analytically given, which thus prevents us from using the conventional gradient based approach to solve it. Fortunately, the viable interval of the μ is fixed, namely, $\mu \in [0, C^{1,\text{tot}}]$. Exploiting this property, we can use a linear-search (LS) with a very small step-size to numerically find the best value of μ (which is denoted by μ^*) that can maximize V_μ^{sub}. The details are shown in the following Algorithm 4. Notice that, in Step 3, we use Algorithm 3 as the subroutine to determine the value of V_μ^{sub} (and the corresponding optimal $\{x_i^{1*}\}_{i \in \mathcal{I}}$) under the currently enumerated μ.

Algorithm 4: To solve Top-problem (TRO-ES-top) and determine μ^*

Initialize: CBV $= 0$ and CBS $= \varnothing$. Set Δ as a very small step-size.

for $\mu = \Delta : \Delta : C^{1,\text{tot}}$ **do**

    Use Algorithm 3 to determine V_μ^{sub} and $\{x_i^{1*}\}_{i \in \mathcal{I}}$.

    if $V_\mu^{\text{sub}} >$ CBV **then**

        Set CBS $= \{x_i^{1*}\}_{i \in \mathcal{I}}$ and CBV $= V_\mu^{\text{sub}}$.

    end if

end for

Output: Set the optimal solution according to CBS.

The computational complexity of our Algorithm 4 can be analyzed as follows. First, as the internal subroutines of Algorithm 4, both Algorithms 1 and 2 require executing a bisection search with the complexity of $\log_2\left(\frac{\bar{\lambda}-\underline{\lambda}}{\epsilon}\right)$. In addition, the linear search in Algorithm 4 requires consuming $\frac{C_1^{\text{tot}}}{\Delta}$ rounds of the iterations (notice that only one of the two subroutines, i.e., Algorithms 1 and 2, will be invoked in each round of the iterations). As a result, the total complexity of our proposed Algorithm 4 is given by $\frac{C_1^{\text{tot}}}{\Delta} \log_2\left(\frac{\bar{\lambda}-\underline{\lambda}}{\epsilon}\right)$.

3.4. Numerical Results for One-ES Scenario

We present the numerical results to validate the effectiveness of our proposed algorithms to solve Problem (TRO-ES). We set the parameter-settings according to the data provided in [46] (Table 1 lists the detailed settings). Specifically, we set $\lambda = \frac{1}{600}$ (i.e., the average generating-time for each block is 10 min) and each block-size $t_i = 1$ Mbit. Meanwhile, according to [46], we set $R = 7000\$$ for each block and set $r = 1000$ \$/Mbit. In addition, for MT i, the local computational power x_i^k is randomly generated according to a uniform distribution within [1,2] GHash/sec. Finally, we set $p^1 = 10$ \$/GHash according to the ES's unit cost for providing the computational power.

To illustrate the rationale of our proposed decomposition, we provide Figure 2 to show how V_μ^{sub} varies with different μ. We test a 5-MT case (in the left subplot) and a 10-MT case (in the right subplot). Figure 2 shows that V_μ^{sub} firstly increases when μ increases, and then gradually decreases when μ is beyond a certain threshold. Such a phenomenon of V_μ^{sub} matches the intuition very well, namely, neither a too small μ nor too small μ will be beneficial to the computation offloading. On the one hand, when μ is too small, the MTs can only obtain a small amount of computational power from the ES, which results in a small total net-reward. On the other hand, when μ is too large, a large cost is incurred for obtaining the computational power, which again degrades the total net-reward. This phenomenon is the motivation of our work, i.e., to find the optimal trade-off between exploiting the computational power provided by the ES and the consequent cost.

Table 1. Important parameter settings.

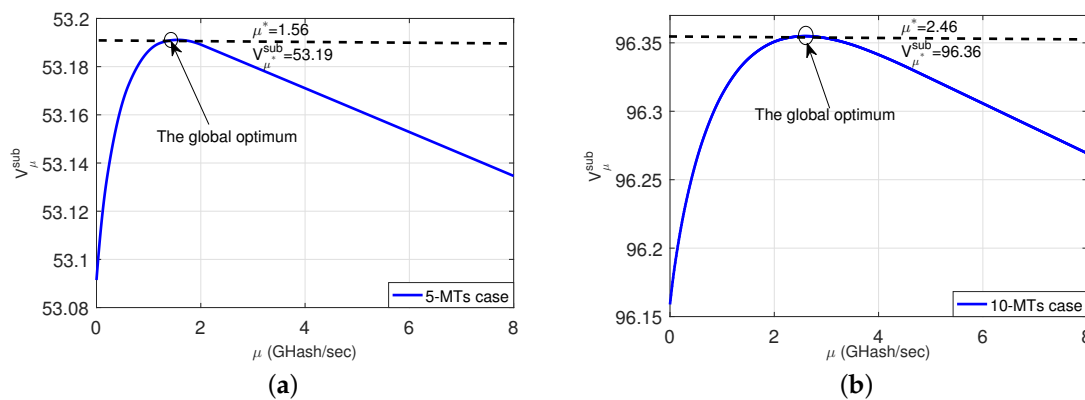| Definition and Symbol	Value
MT i's local computational power x_i^{loc}	uniform distribution between 1 and 2 GHash/s
MT i's block-size t_i	1 Mbit
Fixed reward for each block R	7000\$
Coefficient of variable reward r	1000 \$/Mbit
Average generating-rate of the block λ	$\frac{1}{600}$
Marginal price for computational power p^1	10 \$/GHash

(a)

(b)

Figure 2. V_μ^{sub} versus different μ. (**a**): a 5-MT case; (**b**): a 10-MT case.

Table 2 validates the effectiveness of our Algorithm 4 for solving Top-problem (TRO-ES-Top). For the purpose of comparison, we use a benchmark scheme that exploits the convexity of Problem (TRO-ES-Sub). Specifically, we use CVX [45] (which is a commercial solver for convex optimization) to solve Problem (TRO-ES-Sub) directly for each given μ and then execute a linear search of $\mu \in [0, C^{1,\text{tot}}]$ to solve Top-problem (TRO-ES-Top). Table 2 shows the optimal value achieved by different schemes and the corresponding computation-time. Notice that all the results are obtained on a PC with Intel Core i5-4590 CPU@3.3GHz. As shown in Table 2, our Algorithm 4 can achieve the global

optimum solution as the benchmark scheme, and, moreover, our Algorithm 4 consumes a significantly less computation-time than the benchmark scheme, which thus validates the effectiveness of our proposed algorithm.

Table 2. Performance of our proposed algorithm.

5-MT Case	t_i = 0.2 Mbit	t_i = 0.4 Mbit	t_i = 0.6 Mbit	t_i = 0.8 Mbit	t_i = 1 Mbit
LS Proposed	52.4482	52.6437	52.8338	53.0190	53.1994
Computing Time	0.314 s	0.278 s	0.249 s	0.212 s	0.261 s
Benchmark	52.4482	52.6437	52.8338	53.0190	53.1994
Computing Time	148.5 s	154.4 s	140.39 s	143.2 s	146.2 s
10-MT Case	t_i = 0.2 Mbit	t_i = 0.4 Mbit	t_i = 0.6 Mbit	t_i = 0.8 Mbit	t_i = 1 Mbit
LS Proposed	94.8489	95.2408	95.6220	95.9932	96.3549
Computing Time	0.346 s	0.352 s	0.344 s	0.331 s	0.322 s
Benchmark	94.8489	95.2408	95.6220	95.9932	96.3549
Computing Time	197.4 s	197.6 s	198.4 s	199.5 s	210.2 s

Figure 3 evaluates the impact of the ES's price for providing the computational power to the MTs. When the price increases, the MTs become conservative in using the computational power from the ES, and thus the total computational power acquired from ES 1 decreases (as shown in the right subplot). Accordingly, all MTs' total net-reward gradually decreases when the price increases (as shown in the left subplot).
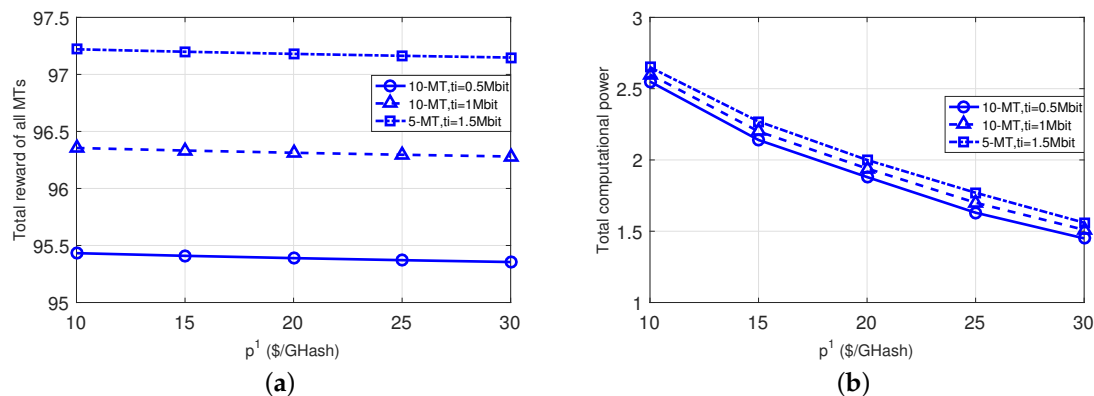

Figure 3. Impact of different ES prices. (**a**): total net-reward; (**b**): total computational power.

4. Multi-ES Scenario and Proposed Distributed Algorithm

4.1. Problem Decomposition

We next consider the multi-ES scenario and focus on solving Subproblem (TRO-Sub) and Top-problem (TRO-Top). As we have described before, Problem (TRO-ES) is a non-convex optimization problem which is difficult to solve in general. To this end, we again adopt a vertical decomposition, by introducing an auxiliary variable v_i which denotes MT i's totally acquired computational power from all ESs, namely,

$$v_i = \sum_{k \in \mathcal{K}} x_i^k, \forall i \in \mathcal{I}. \tag{20}$$

Firstly, we assume that the values of $\{v_i\}_{i \in \mathcal{I}}$ are given in advance, and we aim at solving the subproblem as follows:

$$\text{(TRO-Sub):} \quad H^{\text{sub}}_{\{v_i\}_{i \in \mathcal{I}}} = \max \sum_{i \in \mathcal{I}} \ln \left((R + rt_i) \frac{x_i^{\text{loc}} + v_i}{\sum_{i \in \mathcal{I}} \left(x_i^{\text{loc}} + v_i \right)} e^{-\lambda t_i} - \sum_{k \in \mathcal{K}} p^k x_i^k \right), \quad (21)$$

$$\text{subject to:} \quad 0 \le \sum_{i \in \mathcal{I}} x_i^k \le C^{k,\text{tot}}, \forall k \in \mathcal{K}, \quad (22)$$

$$\sum_{k \in \mathcal{K}} x_i^k = v_i, \forall i \in \mathcal{I}, \quad (23)$$

$$\text{variables:} \quad x_i^k \ge 0, \forall i \in \mathcal{I}, k \in \mathcal{K}.$$

After solving Subproblem (TRO-Sub) and obtaining $H^{\text{sub}}_{\{v_i\}_{i \in \mathcal{I}}}$ (for the given $\{v_i\}_{i \in \mathcal{I}}$), we continue to solve the top-problem as:

$$\text{(TRO-Top):} \quad \max H^{\text{sub}}_{\{v_i\}_{i \in \mathcal{I}}},$$
$$\text{variables:} \quad 0 \le v_i \le Q^{\max}, \forall i \in \mathcal{I}, \quad (24)$$

where we set $Q^{\max} = \sum_{k \in \mathcal{K}} C^{k,\text{tot}}$.

4.2. Distributed Algorithm to Solve Subproblem (TRO-Sub)

The reason for us to adopt the above decomposition is that we can propose a distributed algorithm to solve Subproblem (TRO-Sub). Specifically, given $\{v_i\}_{i \in \mathcal{I}}$, Subproblem (TRO-Sub) is a convex optimization problem. Thus, we again introduce the dual variable λ^k to relax constraint (22) with respect to ES k, and obtain the corresponding Lagrangian function as:

$$L\left(\{x_i^k\}, \{\lambda^k\} \right) = \sum_{i \in \mathcal{I}} \ln \left((R + rt_i) M - \sum_{k \in \mathcal{K}} p^k x_i^k \right) + \sum_{k \in \mathcal{K}} \lambda^k \left(C^{k,\text{tot}} - \sum_{i \in \mathcal{I}} x_i^k \right), \quad (25)$$

where the fixed parameter M (under the given $\{v_i\}_{i \in) \in \mathcal{I}}$) is:

$$M = \frac{x_i^{\text{loc}} + v_i}{\sum_{i \in \mathcal{I}} \left(x_i^{\text{loc}} + v_i \right)} e^{-\lambda t_i}. \quad (26)$$

An observation on (25) shows that it can be separated as follows:

$$L\left(\{x_i^k\}, \{\lambda^k\} \right) = \sum_{i \in \mathcal{I}} L_i(\{x_i^k\}_{\forall k \in \mathcal{K}}, \{\lambda^k\}_{\forall k \in \mathcal{K}}) + \sum_{k \in \mathcal{K}} \lambda^k C^{k,\text{tot}}, \quad (27)$$

where, for each MT i, its associated Lagrangian function is:

$$L_i(\{x_i^k\}_{\forall k \in \mathcal{K}}, \{\lambda^k\}_{\forall k \in \mathcal{K}}) = \ln \left((R + rt_i) M - \sum_{k \in \mathcal{K}} p^k x_i^k \right) - \sum_{k \in \mathcal{K}} x_i^k \lambda^k, \forall i \in \mathcal{I}. \quad (28)$$

Based on (28), we formulate each MT i's local optimization problem as follows:

$$\text{(TRO-Sub-MT}i\text{):} \quad \{\tilde{x}_i^k\} = \arg \max \ln \left((R + rt_i) M - \sum_{k \in \mathcal{K}} p^k x_i^k \right) - \sum_{k \in \mathcal{K}} x_i^k \lambda^k, \quad (29)$$

$$\text{subject to:} \quad \sum_{k \in \mathcal{K}} x_i^k = v_i, \forall i \in \mathcal{I}, \quad (30)$$

$$\text{variables:} \quad x_i^k \ge 0, \forall i \in \mathcal{I}, k \in \mathcal{K}.$$

To further determine the optimal values of $\{\lambda^k\}_{k\in\mathcal{K}}$ (i.e., the optimal solution of the dual problem), we use the following subgradient method:

$$\lambda^k = \max\left\{\lambda^k - \varepsilon\left(C^{k,\mathrm{tot}} - \sum_{i\in\mathcal{I}}\tilde{x}_i^k\right), 0\right\}, \forall k \in \mathcal{K}, \tag{31}$$

where ε is the step-size for the dual-updating. Notice that (31) means that each ES k can individually update its λ^k based on all MTs' reported $\{\tilde{x}_i^k\}_{i\in\mathcal{I}}$. Based on the above, each MT i's local optimization problem (TRO-Sub-MTi) and each MT k's dual-updating in (31), we propose the following distributed algorithm to solve Problem (TRO-Top) and Problem (TRO-Sub).

In particular, according to [45], using the diminishing step-size (in Step 7) enables us to reach the dual optimality. Thus, Algorithm 5 is guaranteed to converge to the optimal solution of Subproblem (TRO-Sub) and determine $H^{\mathrm{sub}}_{\{v_i\}_{i\in\mathcal{I}}}$.

Algorithm 5: To solve Subproblem (TRO-Sub) and determine $H^{\mathrm{sub}}_{\{v_i\}_{i\in\mathcal{I}}}$

Input: $\{v_i\}_{i\in\mathcal{I}}$.

Initialization: Set the iteration-index $l = 1$. Each ES $k \in \mathcal{K}$ initializes $\lambda^k(l)$.

while $\max_{k\in\mathcal{K}}|\lambda^k(l) - \lambda^k(l-1)| > \varepsilon$ **do**

     Each ES k broadcasts $\lambda^k(l)$ to all MTs.

     Given $\{\lambda^k(l)\}_{k\in\mathcal{K}}$, each MT i solves its local Problem (TRO-Sub-MTi) and obtain its $\{\tilde{x}_i^k\}_{k\in\mathcal{K}}$.

     Each MT i reports its $\tilde{x}_i^k$ to each ES k.

     After collecting $\{\tilde{x}_i^k\}_{i\in\mathcal{I}}$ from all MTs, each ES k updates

$$\lambda^k(l) = \max\left\{\lambda^k(l-1) - \varepsilon(l)\left(C^{k,\mathrm{tot}} - \sum_{i\in\mathcal{I}}\tilde{x}_i^k\right), 0\right\}, \forall k \in \mathcal{K},$$

     where $\varepsilon(l) = \frac{a}{b+l}$. Parameters a and b are the given values.

     Set $l = l + 1$.

end while

Each MT i sets $\ln\left((R + rt_i)M - \sum_{k\in\mathcal{K}}p^k\tilde{x}_i^k\right)$ and reports to ES 1.

ES 1 sets $H^{\mathrm{sub}}_{\{v_i\}_{i\in\mathcal{I}}} = \sum_{i\in\mathcal{I}}\ln\left((R + rt_i)M - \sum_{k\in\mathcal{K}}p^k\tilde{x}_i^k\right)$.

Output: $H^{\mathrm{sub}}_{\{v_i\}_{i\in\mathcal{I}}}$.

4.3. Proposed Algorithm to Solve Top-Problem (TRO-Top)

We then continue to solve Problem (TRO-Top). The difficulty in solving Problem (TRO-Top) lies in the fact that we cannot express $H^{\mathrm{sub}}_{\{v_i\}_{i\in\mathcal{I}}}$ analytically for each MT i. To tackle this difficulty, we adopt the idea of simulated annealing (SA) [47] to determine the optimal values of $\{v_i\}_{i\in\mathcal{I}}$ (which are denoted by $\{v_i^*\}_{i\in\mathcal{I}}$). The details are shown in the following Algorithm 6. Based on the idea of SA, our Algorithm 6 executes a randomized search for finding $\{v_i^*\}_{i\in\mathcal{I}}$:

- If the newly generated $\{v_i\}_{i\in\mathcal{I}}$ (which is randomly generated within the range of the previously located $\{v_i\}_{i\in\mathcal{I}}$) can improve the current best value (CBV), we then accept the newly generated $\{v_i\}_{i\in\mathcal{I}}$.
- If the newly generated $\{v_i\}_{i\in\mathcal{I}}$ fails to improve the CBV, we then accept it according to a certain probability, such that we can avoid being trapped by a local optimum. In particular, the probability for us to accept a non-improvement $\{v_i\}_{i\in\mathcal{I}}$ gradually decreases according to an annealing process in which the annealing temperature decreases gradually. As a result, it is more likely that we will refuse to accept a non-improved $\{v_i\}_{i\in\mathcal{I}}$.

Notice that, in Step 7, given the newly generated $\{v_i'\}_{i\in\mathcal{I}}$, we use Algorithm 5 to compute the value of $H^{\text{sub}}_{\{v_i'\}_{i\in\mathcal{I}}}$. However, regarding our proposed Algorithm 6, we notice that it is very challenging to analytically derive its computational complexity. The key reason is due to the difficulty in deriving the complexity of the subroutine, i.e., Algorithm 5 which relies on the sub-gradient method to reach the convergence.

Algorithm 6: To solve Top-problem (TRO-top)

Initialization: Initialize the temperature $T_1 = 9.9$, the decaying-rate $d = 0.9$, the lowest temperature

$T_{\text{final}} = 0.001$, $N_{\text{count}} = 0$, and the time index $t = 1$.

Randomly generate $\{v_i\}_{i\in\mathcal{I}}$. Set CS $= \{v_i\}_{i\in\mathcal{I}}$.

Use Algorithm 5 to compute $H^{\text{sub}}_{\{v_i\}_{i\in\mathcal{I}}}$. Set CV $= H^{\text{sub}}_{\{v_i\}_{i\in\mathcal{I}}}$.

while $T_t > T_{\text{final}}$ **do**

  Set $t = t + 1$.

  Randomly generate $\{v_i'\}_{i\in\mathcal{I}}$ a plane whose central is $\{v_i\}_{i\in\mathcal{I}}$.

  Use CVX to compute $H^{\text{sub}}_{\{v_i'\}_{i\in\mathcal{I}}}$.

  if $H^{\text{sub}}_{\{v_i'\}_{i\in\mathcal{I}}} > $ CV **then**

    Set CV $= H^{\text{sub}}_{\{v_i'\}_{i\in\mathcal{I}}}$.

    Set CS $= \{v_i'\}_{i\in\mathcal{I}}$.

    Set $N_{\text{count}} = 0$.

  else

    Generate a number ϱ according to the uniform distribution within $[0, 1]$.

    if $\varrho < \exp\left(\dfrac{H^{\text{sub}}_{\{v_i'\}_{i\in\mathcal{I}}} - \text{CV}}{T_t}\right)$ **then**

      CV $= H^{\text{sub}}_{\{v_i'\}_{i\in\mathcal{I}}}$.

      Set CS $= \{v_i'\}_{i\in\mathcal{I}}$.

      Set $N_{\text{count}} = 0$.

    else

      Update $N_{\text{count}} = N_{\text{count}} + 1$.

    end if

  end if

  if $N_{\text{count}} \geq 30$ **then**

    Break.

  end if

  Update $T_t = T_{t-1} * d$.

end while

Output: $\{v_i^*\}$ = CS and CV as the maximum value of the objective function of Problem (TRO).

4.4. Numerical Results for Multi-ES Scenario

In this subsection, we show the performance of our proposed algorithms for the multi-ES scenario.

Figure 4 shows the convergence of Algorithm 5 (under the given $\{v_i\}_{i\in\mathcal{I}}$). We use a 5-MT and 3-ES case, with the three ESs having $\{p^1, p^2, p^3\} = \{10, 20, 30\}\$/\text{GHash}$. In Figure 4a, we set $\{v_i\} = \{1, 2, 3, 4, 5\}$ for the five MTs. The top-subplot in Figure 4a shows the convergence of the dual variables $\{\lambda^1, \lambda^2, \lambda^3\}$, and the bottom-subplot in Figure 4a shows the convergence of $H^{\text{sub}}_{\{v_i\}_{i\in\mathcal{I}}}$ to the

dual optimum (which is denoted by the red-dash line). In Figure 4b, we set $\{v_i\} = \{1, 2, 3, 4, 5\}$ for the five MTs, and the results show the same convergence property as Figure 4a.

Figure 5 shows the convergence of our Algorithm 6 for solving Top-problem (TRO-top). The left-subplot shows the case of 5-MT and 3-ES (which is used in Figure 4 before), and the right-subplot shows the case of 10-MT and 3-ES. The results show that our algorithm can quickly converge to the optimal solution (i.e., $\{v_i^*\}_{i \in \mathcal{I}}$) and reach the global optimum of the total net-reward of all MTs.
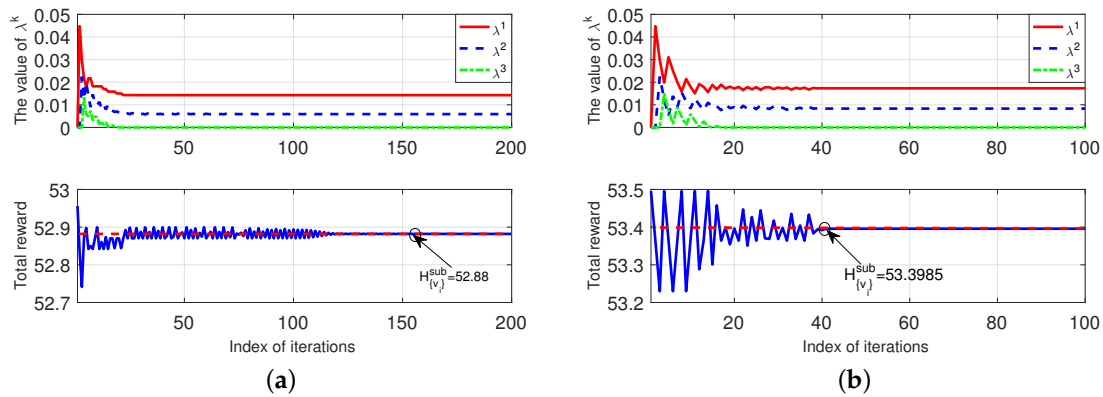
Figure 4. Convergence of our proposed Algorithm 5. (**a**): $\{v_i\} = \{1, 2, 3, 4, 5\}$. (**b**): $\{v_i\} = \{3, 3, 3, 3, 3\}$.
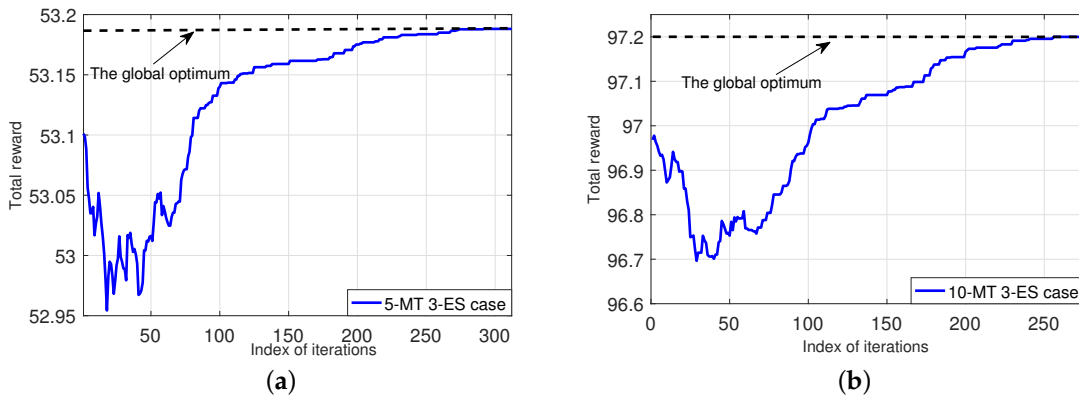
Figure 5. Convergence of our Algorithm 6. (**a**): a 5-MT and 3-ES case; (**b**): a 10-MT and 3-ES case.

Table 3 evaluates the accuracy and efficiency of our proposed Algorithm 6 for solving Top-problem (TRO-Top), in comparison with the exhaustive-search method. In the exhaustive-search method, we enumerate all possible $\{v_i\}_{i \in \mathcal{I}}$. However, the exhaustive-search method consumes a significant computation complexity. We thus consider two cases, namely, a 5-MT and 2-ES case and a 3-MT and 2-ES case. We set $\{p^1, p^2\} = \{10, 20\}\$/GHash$ for the two ESs, and vary each MT's block-size t_i from 0.2 Mbit to 1 Mbit. For each tested case, we show the total net-reward (i.e., the top-value) in each cell, and the corresponding computation time (i.e., the bottom-value) in each cell. The results in Table 3 show that our Algorithm 6 achieves the optimal solution exactly the same as the exhaustive-search method, but consuming significantly less computation-time.

To show the advantage of our proposed algorithms, we further compare the performance of our proposed algorithms with that of a heuristic equal-allocation scheme in which each ES equally divides its total computational power to be shared by all MTs. Figure 6a below shows the results under the scenario of 10 MTs and one ES, and Figure 6b shows the results under the scenario of five MTs and two

ESs. Both figures validate that our proposed algorithms can outperform the equal-allocation scheme. This advantage essentially comes from that we properly allocate the computational powers at different ESs for the MTs.

Finally, in Figure 7, we evaluate the impact of the ESs' prices for providing the computational power to the MTs. We use the same parameter-settings as Table 3, but fix $p^1 = 10\$/GHash$ and vary p^2 from 6 \$/GHash to 14 \$/GHash. Both subplots show that all MTs' totally acquired computational power from ES-2 gradually decreases due to the increasing price. As a result, the MTs are encouraged to acquire more computational power from ES-1.

Table 3. Accuracy and efficiency of our Algorithm 6.

| 3-MT 2-ES Case	t_i = 0.2 Mbit	t_i = 0.4 Mbit	t_i = 0.6 Mbit	t_i = 0.8 Mbit	t_i = 1 Mbit
Exhaustive Meth.	33.6742	33.7930	33.9124	34.0287	34.1356
Computing Time	889.9 s	896.2 s	913.2 s	926.2 s	917.8 s
MultiTop-Algo.	33.6742	33.7930	33.9124	34.0287	34.1356
Computing Time	93.10 s	74.93 s	108.8 s	87.29s	59.83 s
5-MT 2-ES Case	t_i = 0.2 Mbit	t_i = 0.4 Mbit	t_i = 0.6 Mbit	t_i = 0.8 Mbit	t_i = 1 Mbit
Exhaustive Meth.	52.4315	52.6387	52.8214	53.0064	53.1970
Computing Time	1547 s	1847 s	1648 s	1788 s	1980 s
MultiTop-Algo.	52.4315	52.6387	52.8214	53.0064	53.1970
Computing Time	63.78 s	73.05 s	53.85 s	57.67 s	83.51 s
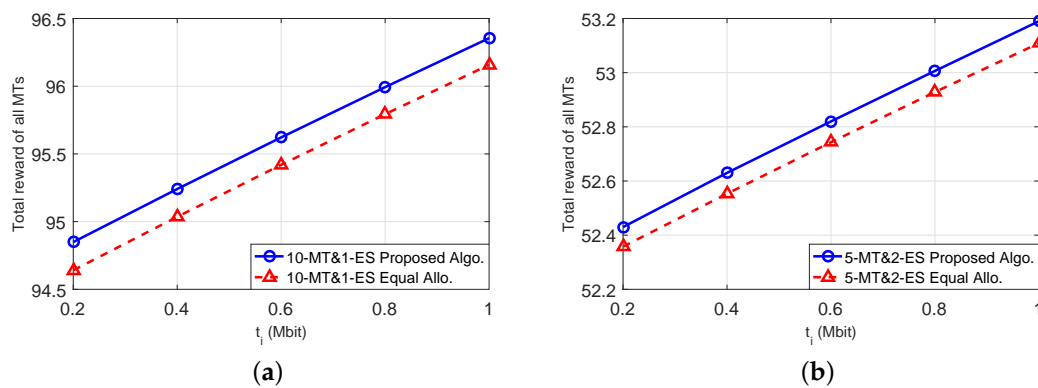
Figure 6. Advantage of our proposed algorithms in comparison with an Equal-Allocation scheme. (**a**): 10-MT and 1-ES case; (**b**): 5-MT and 2-ES case
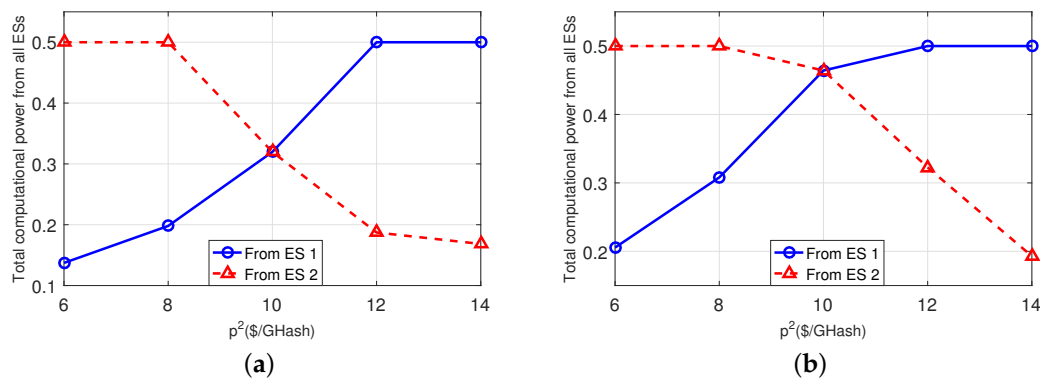

Figure 7. Impact of the ES prices. (**a**): 3-MT and 2-ES case; (**b**): 5-MT and 2-ES case.

5. Conclusions

In this work, we have investigated the optimal computational power allocation for the multi-access MEC enabled Blockchain. In particular, we focused on the scenario in which the group of the MTs acquire the computational power from the ESs, with the objective of maximizing all MTs' total net-reward in the mining process while keeping the fairness among the MTs. By exploiting the layered structure of the formulated optimization problem, we have proposed two distributed algorithms, namely, one for the single-ES scenario and another for the multi-ES scenario, to efficiently compute the MTs' optimal computational power allocations. Extensive numerical results have been provided to validate the effectiveness of our proposed algorithms. In this work, we mainly focused on the reward optimization from the MTs' perspective. For our future work, we will further consider the revenue of the ESs in providing the multi-access MEC service and investigate how different ESs adjust their prices for optimizing their revenues.

Author Contributions: Conceptualization, all co-authors; Methodology, all co-authors; Software, not applicable; Validation, all co-authors; Formal Analysis, all co-authors; Investigation, all co-authors; Resources, all co-authors; Data Curation, all co-authors; Writing-Original Draft Preparation, Y.W., X.C., J.S. and K.N.; Writing-Review and Editing, all co-authors; Visualization, all co-authors; Supervision, not applicable; Project Administration, Y.W., L.Q., L.H. and K.Z.; Funding Acquisition, Y.W., L.Q., L.H. and K.Z.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tschorsch, F.; Scheuermann, B. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2084–2123. [CrossRef]
2. Houy, N. The Bitcoin Mining Game. Available online: https://ssrn.com/abstract=2407834 (accessed on 1 August 2018).
3. Mining Pool Hub. Available online: https://miningpoolhub.com/ (accessed on 1 August 2018).
4. Hurlburt, G. Might the Blockchain Outlive Bitcoin? *IT Prof.* **2016**, *18*, 12–16. [CrossRef]
5. Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things. *IEEE Trans. Ind. Inf.* **2018**, *14*, 3690–3700. [CrossRef]
6. Kang, J.; Yu, R.; Huang, X.; Maharjan, S.; Zhang, Y.; Hossain, E. Enabling Localized Peer-to-Peer Electricity Trading Among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains. *IEEE Trans. Ind. Inf.* **2017**, *13*, 3154–3164. [CrossRef]
7. Liu, H.; Zhang, Y.; Yang, T. Blockchain Enabled Security in Electric Vehicles Cloud and Edge Computing. *IEEE Netw. Mag.* **2018**, *32*, 78–83. [CrossRef]
8. Kotobi, K.; Bilen, S. Blockchain-enabled Spectrum Access in Cognitive Radio Networks. In Proceedings of the 2017 Wireless Telecommunications Symposium (WTS'2017), Chicago, IL, USA, 26–28 April 2017; pp. 1–6.
9. Christidis, K.; Devetsikiotis, M. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access.* **2016**, *4*, 2292–2303. [CrossRef]
10. Novo, O. Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT. *IEEE Int. Things J.* **2018**, *5*, 1184–1195. [CrossRef]
11. Yang, X.; Wang, X.; Wu, Y.; Qian, L.; Lu, W.; Zhou, H. Small-Cell Assisted Secure Traffic Offloading for Narrow-Band Internet of Thing (NB-IoT) Systems. *IEEE Int. Things J.* **2018**, *5*, 1516–1526. [CrossRef]
12. Zhang, K.; Ni, J.; Yang, K.; Liang, X.; Ren, J.; Shen, X. Security and Privacy for Smart City Applications: Challenges and Solutions. *IEEE Commun. Mag.* **2017**, *55*, 122–129. [CrossRef]
13. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
14. Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile Edge Computing: A Survey. *IEEE Int. Things J.* **2018**, *5*, 450–465. [CrossRef]

15. Tran, T.; Hajisami, A.; Pandey, P.; Pompili, D. Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges. *IEEE Commun. Mag*. **2017**, *55*, 54–61. [CrossRef]

16. Liu, Y.; Qin, Z.; Elkashlan, M.; Ding, Z.; Nallanathan, A.; Hanzo, L. Non-orthogonal Multiple Access for 5G and Beyond. *Proc. IEEE* **2017**, *105*, 2347–2381. [CrossRef]

17. Qian, L.; Wu, Y.; Zhou, H.; Shen, X. Dynamic Cell Association for Non-orthogonal Multiple-access V2S networks. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 2342–2356. [CrossRef]

18. Wu, Y.; Qian, L.; Mao, H.; Yang, X.; Shen, X. Optimal Power Allocation and Scheduling for Non-Orthogonal Multiple Access Relay-Assisted Networks. *IEEE Trans. Mob. Comput.* **2018**, 17, 2591–2606. [CrossRef]

19. Guo, H.; Liu, J. Collaborative Computation Offloading for Multi-access Edge Computing Over Fiber-Wireless Networks. *IEEE Trans. Veh. Technol*. **2018**, *67*, 4514–4526. [CrossRef]

20. Qian, L.; Feng, A.; Huang, Y.; Wu, Y.; Ji, B.; Shi, Z. Optimal SIC Ordering and Computation Resource Allocation in MEC-aware NOMA NB-IoT Networks. *IEEE Internet Things J.* **2018**. [CrossRef]

21. Wu, Y.; Ni, K.; Zhang, C.; Qian, L.; Tsang, D. NOMA Assisted Multi-Access Mobile Edge Computing: A Joint Optimization of Computation Offloading and Time Allocation. *IEEE Trans. Veh. Technol.* **2018**. [CrossRef]

22. Xiong, Z.; Feng, S.; Niyato, D.; Wang, P.; Han, Z. Edge Computing Resource Management and Pricing for Mobile Blockchain. Available online: https://arxiv.org/pdf/1710.01567.pdf (accessed on 1 August 2018).

23. Liu, M.; Yu, F.; Teng, Y.; Leung, V.; Song, M. Joint Computation Offloading and Content Caching for Wireless Blockchain Networks. In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018; pp. 517–522.

24. Zhang, K.; Leng, S.; He, Y.; Maharjan, S.; Zhang, Y. Mobile Edge Computing and Networking for Green and Low-latency Internet of Things. *IEEE Commun. Mag*. **2018**, *56*, 36–44. [CrossRef]

25. Wu, Y.; Qian, L.; Zheng, J.; Zhou, H.; Shen, X. Green-Oriented Traffic Offloading through Dual-Connectivity in Future Heterogeneous Small-Cell Networks. *IEEE Commun. Mag*. **2018**, *56*, 140–147. [CrossRef]

26. Guo, H.; Liu, J.; Zhang, J.; Sun, W.; Kato, N. Mobile-Edge Computation Offloading for Ultra-Dense IoT Networks, *IEEE Int. Things J*. **2018**. [CrossRef]

27. Melendez, S.; McGarry, M. Computation Offloading Decisions for Reducing Completion Time. In Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2017; pp. 160–164.

28. Chen, X.; Jiao, L.; Li, W.; Fu, X. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing. *IEEE/ACM Trans. Netw*. **2016**, *24*, 2795–2808. [CrossRef]

29. Chen, X.; Zhang, H.; Wu, C.; Mao, S.; Ji, Y.; Bennis, M. Optimized Computation Offloading Performance in Virtual Edge Computing Systems via Deep Reinforcement Learning. Available online: https://arxiv.org/abs/1805.06146v1 (accessed on 1 August 2018).

30. Liu, J.; Mao, Y.; Zhang, J.; Letaief, K. Delay-Optimal Computation Task Scheduling for Mobile-Edge Computing Systems. In Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, Spain, 10–15 July 2016; pp. 1451–1455.

31. Wang, Y.; Sheng, M.; Wang, X.; Wang, L.; Li, J. Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling. *IEEE Trans. Commun*. **2016**, *64*, 4268–4282. [CrossRef]

32. Kao, Y.; Krishnamachari, B.; Ra, M.; Bai, F. Hermes: Latency Optimal Task Assignment for Resource-Constrained Mobile Computing. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Kowloon, Hong Kong, China, 26 April–1 May 2015; pp. 1894–1902.

33. Chen, M.; Liang, B.; Ming, D. Joint Offloading and Resource Allocation for Computation and Communication in Mobile Cloud with Computing Access Point. In Proceedings of the 2017 IEEE Conference on Computer Communications (INFOCOM), Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.

34. Wu, Y.; Shi, J.; Ni, K.; Qian, L.; Zhu, W.; Shi, Z.; Meng, L. Secrecy-based Delay-aware Computation Offloading via Mobile Edge Computing for Internet of Things. *IEEE Internet Things J.* **2018**. [CrossRef]

35. Barbarossa, S. Joint Allocation of Computation and Communication Resources in Multiuser Mobile Cloud Computing. In Proceedings of the 2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Darmstadt, Germany, 16–19 June 2013; pp. 26–30.

36. Jin, A.; Song, W.; Zhuang, W. Auction-based Resource Allocation for Sharing Cloudlets in Mobile Cloud Computing. *IEEE Trans. Emerg. Top. Comput*. **2018**, *6*, 45–57. [CrossRef]

37. Hoang, D.; Niyato, D.; Wang, P. Optimal Admission Control Policy for Mobile Cloud Computing Hotspot With Cloudlet. In Proceedings of the 2012 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, 1–4 April 2012; pp. 3145–3149.

38. MacKie-Mason, J.; Murphy, L.; Murphy, J. Responsive Pricing in the Internet. In *Chapter of Internet Economics*; The MIT Press: Cambridge, UK, 1998; pp. 279–303.

39. Tsiropoulou, E.; Vamvakas, P.; Papavassiliou, S. Energy Efficient Uplink Joint Resource Allocation Non-cooperative Game with Pricing. In Proceedings of the 2012 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, 1–4 April 2012; pp. 2352–2356.

40. Loiseau, P.; Schwartz, G.; Musacchio, J.; Amin, S. Incentive Schemes for Internet Congestion Management: Raffles versus Time-of-Day Pricing. In Proceedings of the IEEE Allerton Conference, Monticello, IL, USA, 28–30 September 2011; pp. 103–110.

41. Ha, S.; Sen, S.; Wong, C.; Im, Y.; Chiang, M. TUBE: Time Dependent Pricing for Mobile Data. In Proceedings of the ACM SIGCOMM, Helsinki, Finland, 13–17 August 2012; pp. 247–258.

42. Tsiropoulou, E.; Vamvakas, P.; Papavassiliou, S. Joint Customized Price and Power Control for Energy-Efficient Multi-Service Wireless Networks via S-Modular Theory. *IEEE Trans. Green Commun. Netw.* **2017**, *1*, 17–28. [CrossRef]

43. Vamvakas, P.; Tsiropoulou, E.; Papavassiliou, S. Dynamic Provider Selection and Power Resource Management in Competitive Wireless Communication Markets. *Mob. Netw. Appl.* **2018**, *23*, 86–99. [CrossRef]

44. Peter, R. A Transaction Fee Market Exists Without a Block Size Limit. Available online: https://www.bitcoinunlimited.info/downloads/feemarket.pdf (accessed on 1 August 2018).

45. Boyd, S.; Vandenberghe, L. Convex Optimization. In *Convex Optimization*; Cambridge University Press: England, UK, 2004.

46. Blockchain. Available online: https://data.bitcoinity.org (accessed on 1 August 2018).

47. Talbi, E. *Metaheuristics: From Design to Implementation. In Metaheuristics: From Design to Implementation*; Wiley Press: Manhattan, NY, USA, 2009.