IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 24, 2020, accepted October 12, 2020, date of publication October 16, 2020, date of current version October 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031614

Joint Task Offloading and Resource Management in NOMA-Based MEC Systems: A Swarm Intelligence Approach

HUONG-GIANG T. PHAM[1], QUOC-VIET PHAM[2], (Member, IEEE), ANH T. PHAM[3], (Senior Member, IEEE), AND CHUYEN T. NGUYEN[1]

[1]School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi 100000, Vietnam
[2]Research Institute of Computer, Information and Communication, Pusan National University, Busan 46241, South Korea
[3]Computer Communications Laboratory, The University of Aizu, Aizuwakamatsu 965-8580, Japan

Corresponding author: Chuyen T. Nguyen (chuyen.nguyenthanh@hust.edu.vn)

This research was funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.04-2019.300. This work is also supported in part by the JSPS Kakenhi Project under Grant #18K11269.

ABSTRACT In this paper, we study the issue of computation offloading in non-orthogonal multiple access (NOMA)-based multi-access edge computing (MEC) systems. A joint optimization problem of offloading decision, subchannel assignment, transmit power, and computing resource allocation is investigated to improve system performance in terms of both completion time and energy consumption. The formulated problem is a mixed-integer non-linear programming one, it is therefore hard to solve. To make the problem tractable, we first decompose the problem into subproblems of computing resource allocation (CRA), transmit power control (TPC), and subchannel assignment (SA). Then, we address the CRA subproblem by a convex optimization technique. For the remaining two subproblems TPC and SA, we propose to use a gradient-free swarm intelligence approach, namely whale optimization algorithm, to provide a very general but efficient solution. Computer simulations are performed to show the convergence of the proposed algorithm, also its better performance in comparison with conventional schemes.

INDEX TERMS Computation offloading, swarm intelligence, multi-access edge computing, non-orthogonal multiple access, resource management, whale optimization algorithm.

I. INTRODUCTION

To cope with various performance requirements of emerging applications in fifth-generation (5G) and beyond networks, different advanced solutions have been taken into account, simultaneously. Among them, developed by the European Telecommunications Standards Institute (ETSI), multi-access edge computing (MEC) is a highly useful technology [1]. The principle of MEC is to move the computation capacity from the cloud to the network edge in the proximity of user equipments (UEs). Thanks to this mechanism, UEs can offload their computation tasks to MEC servers while significantly reducing the whole network transmission latency. On the other hand, non-orthogonal multiple access (NOMA) has been well known as an essential technique beyond 5G to cope with the massive Internet of Things (IoT) connectivity [2], [3]. The primary idea of NOMA is to carry over the same transmission resources (e.g., frequency and power) a superimposed signal of multiple UEs at the transmitter side. Then, successive interference cancellation (SIC) is employed at the receiver side to decode the signal of each UE, thus, increasing the total number of served UEs [3]. Owing to huge potentials of MEC and NOMA, their integration i.e., NOMA-based MEC, in 5G networks has received much attention recently [4].

One of the most important research topics in NOMA-based MEC systems is solving the issue of offloading decision and resource allocation/control. In other words, based on the shared information between UEs and MEC servers such as UEs' task loads, computing resources at UEs and the servers, transmission resources (e.g., frequency, time, and power), each UE can decide to offload its task (with which transmission resources) or to perform the task locally. The main purpose of the decision is to optimize the objective function such as the system delay, energy consumption, and computation efficiency [4]. A number of research works have

The associate editor coordinating the review of this manuscript and approving it for publication was Matti Hämäläinen.

been studied in this literature. For instance, the optimization problem of computation offloading is considered in multi-user settings in [5]–[7]. They aim at minimizing the overhead in terms of the task completion time and energy consumption of remote computation in comparison with the local one. Nevertheless, those works rely on the assumption that resource blocks are orthogonally assigned to offloading users, that is, the advantage of NOMA to support massive connectivity is not considered in these works. One of the first contributions that considers the impact of NOMA on offloading decision is investigated in [8]. This work aims at minimizing energy consumption by jointly optimizing subchannel assignment (SA), computing resources allocation (CRA), and transmit power control (TPC). To improve the objective in [8], both the completion time and energy consumption of NOMA-based MEC systems are considered in [9]; however, transmit power of NOMA users is predefined. A weighted sum overhead of time and energy consumption of NOMA-based MEC systems is minimized in [10] where SA, CRA, and TPC are jointly considered. Nevertheless, the optimization problem in [10] requires a standard form of the objective function so that both TPC and CRA problems can be solved by convex optimization approaches. This motivates us to study a general optimization approach for the above issue.

The optimization schemes of offloading decision and resources allocation in NOMA-based MEC systems can be solved by different approaches such as network optimization [7], [11], [12], game theory [6], and machine/deep learning [13]. As an attractive alternative, swarm intelligence has showed its potential in optimizing and analyzing the network performance for years. The term swarm intelligence, firstly defined in 1993, refers to the collectively intelligent behavior of a group of non-intelligent robots [14]. This behavior can unpredictably produce specific (patterns) results. Gradually, not only groups of robots, swarm intelligence describes the algorithms that mimic the collective behaviour of swarms, flocks, herds of animals, for examples, a swarm of bees, a flock of birds or a herd of wolves. According to [15], main advantages of swarm intelligence are 1) there are no assumptions about the objective function and the problem to be optimized, 2) a swarm intelligence technique can achieve the tradeoff between exploration and exploitation so as to obtain the global solution, 3) gradient information about the problem to be optimized is not required, and finally 4) it is quite simple to implement a swarm intelligence technique, as compared with other ones in game theory, convex optimization, and deep learning approaches. For instance, an efficient routing protocol based on a variant of the artificial bee colony for wireless sensor networks is proposed in [16]. The Harris Hawks optimizer is employed in [17] to solve the problem of unmanned aerial vehicle (UAV) placement, power allocation, and in NOMA-enabled visible light communications. The particle swarm optimization approach is used in [18] to generate training samples for a deep neural network in hybrid UAV-MEC systems. As one of the most recent swarm intelligence algorithms, the whale optimization algorithm (WOA) has become popular and widely used in various engineering problems since the proposal in [19]. WOA imitates the collectively hunting behavior that only is seen in the humpback whales. Having the characteristics of SI algorithms, WOA can 1) be simply implemented, 2) flexibly applied for various problems without the requirement in calculating gradient, and 3) achieve the efficiency in obtaining the global solution by balancing between exploration and exploitation phases. WOA is very competitive with other SI algorithms by test results on benchmark functions. As stated in the No Free Lunch theorem, no optimization algorithm is suited for all optimization problems [20]. In addition, it is shown in [21] that WOA can be served as a promising optimization tool for various optimization problems in wireless networks, including energy efficiency, spectral-energy efficiency tradeoff, and MEC computation offloading.

Inspired by the work in [21], we will employ WOA to solve the problem of computation offloading in NOMA-based MEC systems, and show that the WOA-based algorithm can achieve a competitive performance when it is compared with baseline and existing schemes. Our main contributions can be summarized as follows.

1) The issue of offloading decision in NOMA-based MEC systems is considered both in terms of completion time and energy consumption, taking SA, TPC, and CRA into account. The considered problem is shown to be a mixed integer non-linear programming (MINLP) problem, which is hard to solve. To make the problem tractable, we decompose the original problem into three subproblems of CRA, TPC, and SA, which are further solved in an iterative manner.
2) During each iteration, while the CRA subproblem can be solved by a convex optimization technique, we use a swarm intelligence approach, namely WOA, to solve the TPC and SA subproblems. Two additional conditions to reduce the runtime of WOA are also proposed. Thanks to the simplicity, efficiency, and flexibility of WOA, the approach is applicable for different forms of the objective function of the two subproblems.
3) We carry out computer simulations to validate the convergence of the proposed algorithm and show its better performance in comparison with other algorithms.

The rest of this article is organized as follows. Our system model and the considered optimization problem are introduced in Sections II and III, respectively. The proposed algorithm for the problem is described in details in Section IV, while simulation results are presented in Section V. Finally, Section VI concludes our works.

II. SYSTEM MODEL

A. NETWORK MODEL

We consider a NOMA-based MEC system, which consists of a set of N UEs defined as $\mathcal{N} = \{1, 2, \ldots, N\}$, and an MEC server that might be located at a cellular base station (BS). Each UE i has a task $I_i = \{\beta_i, \alpha_i\}$, where β_i (in

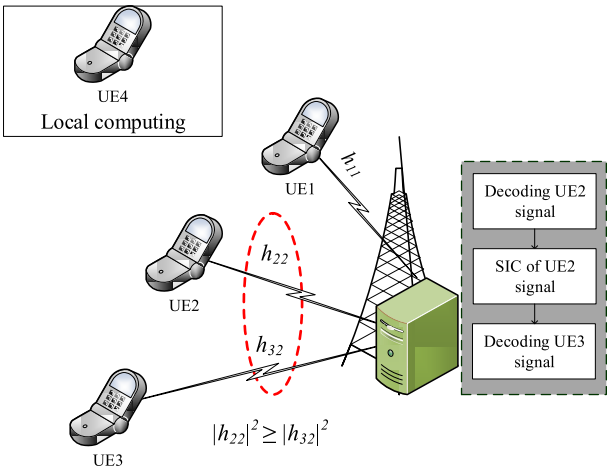

FIGURE 1. Illustration of a NOMA - based MEC system.

cycles) specifies the amount of CPU computation to complete the task and α_i (in bits) is the input data size including information of system settings, program codes, and other parameters being transmitted in case of remote computation [5]–[7]. UEs decide to execute their tasks locally, or offload them to the MEC server over one of subchannels from the set $\mathcal{S} = \{1, 2, \ldots, S\}$ to save time and energy. On the other hand, to serve multiple UEs over the same radio resource, NOMA is also adopted in this work. In particular, more than one UE can use one subchannel to transmit their signals/tasks. According to the NOMA principle [22], the message of the strongest UE (UE that has the highest channel gain) is decoded first treating weaker signals as interference. Weaker signals are successively decoded in the order of decreasing channel gains. For further convenience, we denote by p_{ij} the transmit power of UE i with the chosen subchannel j.

Fig. 1 illustrates a simple example of our systems with four UEs, where three users i.e., UE1, UE2, UE3 offload their tasks to the MEC server, while UE4 executes its task locally. Here, UE1 uses subchannel 1, whereas UE2, UE3 use the same subchannel 2 with the corresponding channel gains $|h_{22}|^2$, $|h_{32}|^2$, and $|h_{22}|^2 \geq |h_{32}|^2$. At the MEC server, UE2's signal is first decoded under the interference of UE3's signal. After using SIC to eliminate the UE2's decoded signal, UE3's signal can be also decoded. It is noted here that, for a successful decoding of UE2's signal, the condition $\frac{p_{22}|h_{22}|^2}{p_{32}|h_{32}|^2} \geq \gamma_{\text{tol}}$ need to be satisfied, where γ_{tol} is a predefined threshold required for the SIC decoder at the BS. More generally, if we denote by $\mathcal{A}_j$ the set of offloading UEs sharing the same subchannel j, the following condition should be satisfied [10]

$$\frac{p_{ij}|h_{ij}|^2}{\sum\limits_{\substack{k\in\mathcal{A}_j \\ |h_{kj}|^2\leq|h_{ij}|^2}} p_{kj}|h_{kj}|^2} \geq \gamma_{\text{tol}}, \quad \forall i \in \mathcal{A}_j. \tag{1}$$

B. LOCAL AND REMOTE COMPUTATIONS

For local computation, the time (in seconds) denoted by T_i^l that UE i takes to complete its task is

$$T_i^l = \beta_i / f_i^l, \tag{2}$$

where f_i^l is the computing capability (in cycles/second) of UE i. Also, the energy E_i^l (in Joules) consumed for this task can be found as

$$E_i^l = \kappa_i \beta_i (f_i^l)^2, \tag{3}$$

where κ_i is an energy coefficient depending on UE i's chip architecture.

On the other hand, when offloading the task to the server, UE i suffers an additional cost in term of time and energy: *i*) the time and energy for transmitting the task to server, *ii*) the time for executing the task at the server, and *iii*) the time and energy to receive the output back from server. Since the output data size is much smaller than the input one, while the data rate of downlink is relatively higher than that of uplink, we ignore the delay time and energy incurring in *iii*) as in [5]–[7]. We now show how to determine time and energy taken in steps *i*) and *ii*). In particular, we define a binary SA variable x_{ij}, $i \in \mathcal{N}$, $j \in \mathcal{S}$, where $x_{ij} = 1$ indicates that UE i offloads its task to the server through subchannel j, and $x_{ij} = 0$, otherwise. Since each UE is assumed to use at most one subchannel for offloading, the constraint below should be satisfied

$$\sum_{j\in\mathcal{S}} x_{ij} \leq 1, \quad \forall i \in \mathcal{N}. \tag{4}$$

Here, from the result of SA, offloading decision of UE i can be determined by $\sum_{j\in\mathcal{S}} x_{ij}$, in which $\sum_{j\in\mathcal{S}} x_{ij} = 0$ means UE i locally executes its task. In case of offloading, $\sum_{j\in\mathcal{S}} x_{ij} = 1$, the data rate of UE i through subchannel j can be given as

$$R_{ij} = W \log_2 \left(1 + \frac{p_{ij}|h_{ij}|^2}{n_0 + \sum\limits_{\substack{k\in\mathcal{A}_j \\ |h_{kj}|^2\leq|h_{ij}|^2}} p_{kj}|h_{kj}|^2} \right), \quad i \in \mathcal{A}_j \tag{5}$$

where W (in Hertz) is the bandwidth of a subchannel that is set to be identical for all the subchannels, n_0 is the noise power, the aforementioned set $\mathcal{A}_j$ can be clearly written as $\mathcal{A}_j = \{i \in \mathcal{N} \mid x_{ij} = 1\}$. It is noted here that the event UE i does not use subchannel j to offload refers to $p_{ij} = 0$, and thus, $R_{ij} = 0$. In this case, the transmission time taken by UE i for offloading (denoted by T_i^{off}) is only calculated based on assigned subchannel j satisfying $x_{ij} = 1$, as follows

$$T_i^{\text{off}} = \sum_{j\in\mathcal{S}} \frac{x_{ij}\alpha_i}{R_{ij}} = \alpha_i \sum_{j\in\mathcal{S}} \frac{x_{ij}}{R_{ij}}. \tag{6}$$

On the other hand, if we denote by p_i the transmit power of UE i, we have

$$0 < p_i = \sum_{j\in\mathcal{S}} x_{ij} p_{ij} \leq p_i^0, \tag{7}$$

where p_i^0 is the maximum power budget. Then, the energy cost that UE i suffers from the offloading (denoted by E_i^{off}) is

$$E_i^{\text{off}} = \frac{p_i}{\xi_i} T_i^{\text{off}} = \frac{p_i}{\xi_i} \alpha_i \sum_{j \in \mathcal{S}} \frac{x_{ij}}{R_{ij}}, \tag{8}$$

where ξ_i is power amplifier efficiency. At the server, the time for executing the task (denoted by T_i^{exe}) is calculated as

$$T_i^{\text{exe}} = \frac{\beta_i}{f_i}, \tag{9}$$

where f_i (in cycles/second) is the server's computation resource allocated to execute the task from UE i. It also implies that when $f_i = 0$, UE i executes locally. Moreover,

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{S}} x_{ij} f_i \leq f_0, \tag{10}$$

where f_0 is the total computing resource of the server. Here, the total time and energy (denoted by T_i^r and E_i^r, respectively) that UE i consumes for the offloading can be summarized as follows

$$T_i^r = T_i^{\text{off}} + T_i^{\text{exe}} = \alpha_i \sum_{j \in \mathcal{S}} \frac{x_{ij}}{R_{ij}} + \frac{\beta_i}{f_i}, \tag{11}$$

and

$$E_i^r = E_i^{\text{off}} = \frac{p_i}{\xi_i} \alpha_i \sum_{j \in \mathcal{S}} \frac{x_{ij}}{R_{ij}}. \tag{12}$$

Finally, the total time and energy taken by UE i for its task I_i, which are denoted by T_i and E_i, respectively, can be written as

$$T_i = (1 - \sum_{j \in \mathcal{S}} x_{ij}) T_i^l + \sum_{j \in \mathcal{S}} x_{ij} T_i^r, \tag{13}$$

$$E_i = (1 - \sum_{j \in \mathcal{S}} x_{ij}) E_i^l + \sum_{j \in \mathcal{S}} x_{ij} E_i^r. \tag{14}$$

III. PROBLEM FORMULATION AND DECOMPOSITION

A. PROBLEM FORMULATION

We first denote by U_i the utility function of each UE i as follows

$$\begin{aligned} U_i &= \lambda_i^t \frac{T_i^l - T_i}{T_i^l} + \lambda_i^e \frac{E_i^l - E_i}{E_i^l} \\ &= \left[(\lambda_i^t + \lambda_i^e) - \left(\frac{\lambda_i^t T_i^r}{T_i^l} + \frac{\lambda_i^e E_i^r}{E_i^l} \right) \right] \sum_{j \in \mathcal{S}} x_{ij}, \end{aligned} \tag{15}$$

where $\frac{T_i^l - T_i}{T_i^l}$ and $\frac{E_i^l - E_i}{E_i^l}$ measures performance improvements in terms of time and energy, respectively, when offloading compared with local execution. It is noted here that $U_i = 0$ if UE i executes its task locally, while it might take negative values if remote completion time is much longer than the local one [5]. Moreover, λ_i^t and λ_i^e are EU i's preferences in consumed time and energy, respectively, where $\lambda_i^t, \lambda_i^e \in [0, 1]$ and $\lambda_i^t + \lambda_i^e = 1$. These parameters are based on the remaining battery life and requirements of the task's completion time [5]–[7]. Our objective is to maximize the system utility of all UEs when offloading i.e., $U = \sum_{i \in \mathcal{N}} U_i$. To do that, in the following, a joint optimization scheme of offloading decision, subchannel, transmit power, and computing resource allocation (ODSTCA) is studied. In particular, we first denote by $\boldsymbol{X} = \{x_{ij}, i \in \mathcal{N}, j \in \mathcal{S}\}$, $\boldsymbol{P} = \{\boldsymbol{P_{ui}}, i \in \mathcal{N}\} = \{\boldsymbol{P_{sj}}, j \in \mathcal{S}\}$, and $\boldsymbol{F} = \{f_i, i \in \mathcal{N}\}$ the SA, the power, and computation allocation vectors, respectively. Here, $\boldsymbol{P_{ui}} = \{p_{ij}, j \in \mathcal{S}\}$ and $\boldsymbol{P_{sj}} = \{p_{ij}, i \in \mathcal{N}\}$, represent the power of each UE and each subchannel, respectively. Then, the optimization scheme can be expressed as follows

$$\max_{\boldsymbol{X}, \boldsymbol{P}, \boldsymbol{F}} U(\boldsymbol{X}, \boldsymbol{P}, \boldsymbol{F}) \tag{16}$$

$$\text{s.t. C1 : } x_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{N}, \ \forall j \in \mathcal{S}, \tag{17}$$

$$\text{C2: } \sum_{j \in \mathcal{S}} x_{ij} \leq 1, \quad \forall i \in \mathcal{N}, \tag{18}$$

$$\text{C3: } p_{ij} \geq 0, \quad \forall i \in \mathcal{N}, \ \forall j \in \mathcal{S}, \tag{19}$$

$$\text{C4: } \sum_{j \in \mathcal{S}} x_{ij} p_{ij} \leq p_i^0, \quad \forall i \in \mathcal{N}, \tag{20}$$

$$\text{C5: } \frac{p_{ij} |h_{ij}|^2}{\sum\limits_{k \in \mathcal{A}_j, \ |h_{kj}|^2 \leq |h_{ij}|^2} p_{kj} |h_{kj}|^2} \geq \gamma_{\text{tol}}, \quad \forall j \in \mathcal{S}, \ \forall i \in \mathcal{A}_j, \tag{21}$$

$$\text{C6: } f_i \geq 0, \quad \forall i \in \mathcal{N}, \tag{22}$$

$$\text{C7: } \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{S}} x_{ij} f_i \leq f_0. \tag{23}$$

In ODSTCA problem, C1 and C2 imply that each UE can only use at most one subchannel to offload the task to the server. C3 and C4 refer to requirements of the allocated transmit power, while C5 specifies the condition to implement efficient SIC at the server. In addition, C6 and C7 show the constraints of the computing capability of the server.

B. PROBLEM DECOMPOSITION

It is observed that (16) is an MINLP problem, which might take exponential time to obtain the optimal solution. To make the problem more tractable, the structure of objective function and constraints are separated into subproblems as the approach in [5]–[7]. In particular, by denoting $\mathcal{A} = \cup_{j \in \mathcal{S}} \mathcal{A}_j$ as the set of all offloading UEs, the objective function $Z(\boldsymbol{X}, \boldsymbol{P}, \boldsymbol{F})$ is rewritten as follows

$$\begin{aligned} U(\boldsymbol{X}, \boldsymbol{P}, \boldsymbol{F}) &= \sum_{i \in \mathcal{A}} \left[(\lambda_i^t + \lambda_i^e) - \left(\frac{\lambda_i^t T_i^r}{T_i^l} + \frac{\lambda_i^e E_i^r}{E_i^l} \right) \right] \\ &= \sum_{i \in \mathcal{A}} (\lambda_i^t + \lambda_i^e) - W(\boldsymbol{X}, \boldsymbol{P}, \boldsymbol{F}), \end{aligned} \tag{24}$$

where $W(\boldsymbol{X}, \boldsymbol{P}, \boldsymbol{F})$ refers to as the overhead in time and energy at remote computation compared with the local one. The problem can be now understood as maximizing the number of offloading UEs, while minimizing the extra overhead. From (11), (12), (24), $W(\boldsymbol{X}, \boldsymbol{P}, \boldsymbol{F})$ can be further decomposed

as

$$W(\boldsymbol{X},\boldsymbol{P},\boldsymbol{F}) = \sum_{i\in\mathcal{A}} \frac{\lambda_i^t \beta_i}{f_i} + \sum_{i\in\mathcal{A}} (\eta_i + \gamma_i p_i) \sum_{j\in\mathcal{S}} \frac{x_{ij}}{R_{ij}} = G(\boldsymbol{X},\boldsymbol{F}) + F(\boldsymbol{X},\boldsymbol{P}), \quad (25)$$

where $\eta_i = \frac{\lambda_i^t \alpha_i}{T_i^l}$, $\gamma_i = \frac{\lambda_i^e \alpha_i}{\xi_i E_i^l}$, $G(\boldsymbol{X},\boldsymbol{F}) = \sum_{i\in\mathcal{A}} \frac{\lambda_i^t \beta_i}{f_i}$, and $F(\boldsymbol{X},\boldsymbol{P}) = \sum_{i\in\mathcal{A}} (\eta_i + \gamma_i p_i) \sum_{j\in\mathcal{S}} \frac{x_{ij}}{R_{ij}}$.

The original problem (16) can be now considered as three subproblems. In particular, for a given offloading decision $\boldsymbol{X}$, minimizing $W(\boldsymbol{X},\boldsymbol{P},\boldsymbol{F})$ are equivalent to two independent subproblems, namely CRA and TPC as follows

$$\min_{\boldsymbol{F}} \; G(\boldsymbol{X},\boldsymbol{F}) \quad \text{s.t. C6, C7.} \quad (26)$$

$$\min_{\boldsymbol{P}} \; F(\boldsymbol{X},\boldsymbol{P}) \quad \text{s.t. C3, C4, C5,} \quad (27)$$

where rewritten $F(\boldsymbol{X},\boldsymbol{P}) = \sum_{j\in\mathcal{S}} \sum_{i\in\mathcal{A}_j} \frac{(\eta_i + \gamma_i p_{ij})}{R_{ij}}$. On the other hand, for a given optimal solution of $\boldsymbol{F}^*$, $\boldsymbol{P}^*$ obtained from (26), (27), respectively, (16) is equivalent to the SA subproblem as

$$\max_{\boldsymbol{X}} \; \sum_{i\in\mathcal{A}} U(\boldsymbol{X}) \quad \text{s.t. C1, C2,} \quad (28)$$

where $U(\boldsymbol{X}) = (\lambda_i^t + \lambda_i^e) - G(\boldsymbol{X},\boldsymbol{F}^*) - F(\boldsymbol{X},\boldsymbol{P}^*)$. In the next section, we in turn present our solution for each subproblem in order to obtain the final optimal of (16).

IV. PROPOSED METHOD

After decomposing the original problem, the CRA and TPC subproblems can be solved with fixed subchannel allocation and the SA can be addressed corresponding to the results of the CRA and TPC. As will be presented in this section, the solution for the CRA is solved by a convex optimization technique, as presented in the previous works, while the solutions for the TPC and SA subproblems are obtained by WOA.

A. COMPUTATION RESOURCE ALLOCATION

The CRA problem in (26) has been showed as a convex problem, and fully addressed in [5]. We adopt this contribution in our work to obtain the optimal solution and objective value, respectively, as follows

$$\boldsymbol{F}^* = \{f_i^*, i \in \mathcal{N}\}, \; f_i^* = \frac{\sqrt{\lambda_i^t f_i^l}}{\sum_{i\in\mathcal{A}} \sqrt{\lambda_i^t f_i^l}} f_0,$$

$$G(\boldsymbol{X},\boldsymbol{F}^*) = \frac{\left(\sum_{i\in\mathcal{A}} \sqrt{\lambda_i^t f_i^l}\right)^2}{f_0}. \quad (29)$$

B. WOA FOR TRANSMIT POWER CONTROL

To solve the TPC subproblem, in this section, WOA is employed thanks to its simplicity and efficiency. WOA is a new nature-inspired meta-heuristic searching algorithm proposed in [19] that mimics the unique bubble-net hunting behavior of humpback whales. It is shown in [21] that WOA can achieve a competitive performance with a low computation complexity compared to algorithms in the areas of convex optimization and game theory. The work in [21] also discusses the use of WOA in a number of resource management problems in 5G wireless networks and beyond.

In bubble-net hunting, the whales simultaneously move along an upward spiral and create a net of bubbles in order to disorient and encircle the prey, then feed them on the surface of water [23]. Correspondingly, WOA models this behavior by three maneuvers namely, *search for prey (SFP)*, *shrinking encircling mechanism (SEM)* and *spiral updating position (SUP)*. The positions of the whales (or search agents) during finding the optimal solution (the position of the prey) are continuously updated, which depends on the particular maneuver. In particular, if it is *SFP*, a whale randomly picks another whale in the population, then moves far away this whale to explore its surroundings, discover potential areas that might contain the prey. If it is *SEM* or *SUP*, the whale will shrink the corral or swim in helix-shape towards the prey's position, respectively. From the swarm intelligence perspective, SFP belongs to exploration phase, while SEM and SUP are in exploitation phase. This process of searching and position updating is terminated when the position of the prey, be assumed as the best agent, is found. It is noted here that the determination of which maneuver is selected during the search process depends on the WOA's parameters, which controls the balance between exploration and exploitation phases.

On the other hand, the former version of WOA is designated for unconstrained optimization problems only. To solve TPC subproblem in (27), we therefore adopt a penalty method applied in [21] to transform the subproblem into an unconstrained one. In particular, constraints C4, C5 can be rewritten as

$$h_4(\boldsymbol{P}_{\boldsymbol{u}i}) = \sum_{j\in\mathcal{S}} x_{ij} p_{ij} - p_i^0, \quad i \in \mathcal{N},$$

$$h_5(\boldsymbol{P}_{\boldsymbol{s}j}) = P_{\text{tol}} \times \sum_{\substack{k\in\mathcal{A}_j \\ |h_{kj}|^2 \le |h_{ij}|^2}} |h_{kj}|^2 p_{kj} - |h_{ij}|^2 p_{ij}, \quad j \in \mathcal{S}, i \in \mathcal{A}_j. \quad (30)$$

Note that, constraint C3 is handled by limiting the variable domain of the algorithm. Then, by denoting $\Gamma(h_i) = 0$ if $h_i \le 0$ and $\Gamma(h_i) = 1$, otherwise, a penalty term is defined as

$$P_{\text{TPC}}(\boldsymbol{X},\boldsymbol{P}) = \sum_{i\in\mathcal{N}} \mu_i \Gamma(h_4(\boldsymbol{P}_{\boldsymbol{u}i})) h_4^2(\boldsymbol{P}_{\boldsymbol{u}i})$$

$$+ \sum_{j\in\mathcal{S}} \sum_{i\in\mathcal{A}_j} \nu_{ij}\Gamma(h_5(\boldsymbol{P}_{sj}))h_5^2(\boldsymbol{P}_{sj}), \quad (31)$$

where μ_i, ν_{ij} are penalty factors, which represent the violation level of the considering solution to the corresponding constraints. These factors, in general is in range of $[10^{13}, 10^{15}]$ [21], mostly depending of the value range of objective function. To explain how the penalty method works, we give an example. It is supposed that, for a given SA $\boldsymbol{X}$, we are considering an allocated transmit power solution $\boldsymbol{P}$ that, for at least one UE, the maximum power condition is violated, i.e. $h_4(\boldsymbol{P}_{ui}) > 0$, then $\Gamma(h_4(\boldsymbol{P}_{ui})) = 1$, therefore, $P_{\text{TPC}}(\boldsymbol{X}, \boldsymbol{P}) = \mu_i h_4^2(\boldsymbol{P}_{ui}) \gg F(\boldsymbol{X}, \boldsymbol{P}^*)$, with respect to $\boldsymbol{P}^*$ is the best solution had found so far, correspondingly $F(\boldsymbol{X}, \boldsymbol{P}^*)$ is the smallest objective value. Thus, $F(\boldsymbol{X}, \boldsymbol{P}) + P_{\text{TPC}}(\boldsymbol{X}, \boldsymbol{P}) \gg F(\boldsymbol{X}, \boldsymbol{P}^*)$, i.e. $\boldsymbol{P}$ takes penalty for violating the constraints. Therefore, it is impossible for $\boldsymbol{P}$ to become the best solution. As a result, the transformed TPC problem can be expressed as

$$\min_{\boldsymbol{P}} \ [F(\boldsymbol{X}, \boldsymbol{P}) + P_{\text{TPC}}(\boldsymbol{X}, \boldsymbol{P})]. \quad (32)$$

For a given SA $\boldsymbol{X}$, (32) can be efficiently solved by WOA as in Algorithm 1. The transmit power $\boldsymbol{P}$ is represented to positions of the whales (search agents). Our algorithm aims to determine their best position, i.e., the transmit power $\boldsymbol{P}^*$ that results in the smallest objective function value in (32). In particular, the searching process, which is iterative, begins with a random whale population. In each iteration, the MEC server searches for the current best position and updates the search agents' position in two stages, namely, *Stage I-C* and *Stage II-C*, respectively. Here, *C* stands for continuous WOA. In *Stage I-C*, for the position of each search agent, the server first amends if this position excesses the power budget of UEs. TCP subproblem, then, finds the best current (whale's) position determined by the smallest calculated fitness in (32), which is referred to (lines 14-18) in the algorithm. In *Stage II-C*, the positions of all whales are updated according to the above mentioned maneuvers of the WOA, i.e., *SFP*, *SEM*, and *SUP*. The algorithm is terminated when reaching I_1 iterations or a predefined converged condition is satisfied (lines 19-23).

C. BINARY WOA FOR SUBCHANNEL ASSIGNMENT

In this section, we propose to use binary WOA (BWOA) approach for the SA subproblem in (28). Note that, SA is an optimization problem that works on binary variables. In particular, the server decides whether or not a UE should utilize a subchannel among subchannels, from that, decides the offloading decision of that UE. In order to deal with the integer programming problem SA, an exhaustive search on all possible solutions could be impractical. Particularly, for the considered NOMA-based MEC system consisting of N UEs and S subchannels, a UE can locally execute or offload through one of S subchannels, thus there are $(S+1)^N$ feasible SA solutions. For example, we consider a NOMA-based

Algorithm 1 WOA for TPC Problem

1: **Initialization:**
2: The whale population $\{\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots \boldsymbol{P}_{M_1}\}$ where M_1 is the number of search agents
3: The iteration index $t = 0$, the convergence index $t_1 = 0$, the maximum number of iterations I_1, the maximum number of iterations for convergence I_α
4: The current best fitness value $F^*(t) = \infty$
5: The convergence threshold ε
6: **while** $\{t_1 < I_\alpha$ **or** $t < I_1\}$ **do**
7: $t = t + 1$
8: *Stage I-C* :
9: **for** $k = 1 \rightarrow M_1$ **do**
10: **if** $\boldsymbol{P}_k \notin$ the power range of UEs **then**
11: Amend $\boldsymbol{P}_k$
12: **end if**
13: Calculate the fitness $F(\boldsymbol{X}, \boldsymbol{P}_k)$ according to (32)
14: **if** $F(\boldsymbol{X}, \boldsymbol{P}_k) < F^*(t)$ **then**
15: $\boldsymbol{P}^* = \boldsymbol{P}_k$
16: $F^*(t-1) = F^*(t)$
17: $F^*(t) = F(\boldsymbol{X}, \boldsymbol{P}_k)$
18: **end if**
19: **if** $\mid F^*(t) - F^*(t-1) \mid < \varepsilon$ **then**
20: $t_1 = t_1 + 1$
21: **else**
22: $t_1 = 0$
23: **end if**
24: **end for**
25: *Stage II-C* :
26: **for** $k = 1 \rightarrow M_1$ **do**
27: Update WOA's parameters
28: Update the new position $\boldsymbol{P}_k$
29: **end for**
30: **end while**
31: **Output:** The TPC solution $\boldsymbol{P}^*$ and the best fitness value $F(\boldsymbol{X}, \boldsymbol{P}^*) = F^*(t)$

MEC system with $N = 15$, $S = 5$, the number of SA solutions is $(5+1)^{15} = 4.7 \times 10^{11}$, that has the polynomial complexity to be implemented in practice. In investigating an efficient approach, we apply binary WOA (BWOA) to solve the SA problem.

Not similar to WOA where the updating position is not continuous value within the variable domain, that in BWOA is only 0 or 1. To do this, the update mechanism in BWOA with helix-shaped movement behavior of humpback whale might be determined by other transfer functions rather than that in WOA, for example, as in [21] and [24]. In particular, by rewriting the constraint C2 as $h_2(x_{ij}) = \sum_{j\in\mathcal{S}} x_{ij} - 1, i \in \mathcal{N}$, the penalty term as in WOA can be defined as

$$P_{\text{SA}}(\boldsymbol{X}) = \sum_{i\in\mathcal{N}} \theta_i \Gamma(h_2(x_{ij}))h_2^2(x_{ij}), \quad (33)$$

where θ_i is the penalty factor. We also omit C1 constraint since it is already considered as the domain of our problem.

Then, by denoting $\overline{U}(\boldsymbol{X}) \triangleq \sum_{i\in\mathcal{A}}(\lambda_i^t + \lambda_i^e) - G(\boldsymbol{X}, \boldsymbol{F}^*) - F(\boldsymbol{X}, \boldsymbol{P}^*) - P_{\text{SA}}(\boldsymbol{X})$, the SA subproblem (28) is equivalent to the following

$$\max_{\boldsymbol{X}} \ \overline{U}(\boldsymbol{X}). \tag{34}$$

To solve (34), we first find $G(\boldsymbol{X}, \boldsymbol{F}^*)$ by addressing the CRA subproblem according to (22), for a given value of $\boldsymbol{X}$. Then, the objective function in (34) can be rewritten as

$$\overline{U}(\boldsymbol{X}) = H(\boldsymbol{X}) - F(\boldsymbol{X}, \boldsymbol{P}^*), \tag{35}$$

where $H(\boldsymbol{X}) = \sum_{i\in\mathcal{A}}(\lambda_i^t + \lambda_i^e) - P_{\text{SA}}(\boldsymbol{X}) - G(\boldsymbol{X}, \boldsymbol{F}^*)$. We also define $\overline{U}^* = \overline{U}(\boldsymbol{X}^*)$, where $\boldsymbol{X}^*$ is the current best agent determined from the previous iteration.

Condition 1: If $H(\boldsymbol{X}) \le \overline{U}^*$, the MEC server does not need to optimize the transmit power for the solution $\boldsymbol{X}$.

Proof: If the Condition 1 is true, the fitness value of the current position $\overline{U}(\boldsymbol{X}) = H(\boldsymbol{X}) - F(\boldsymbol{X}, \boldsymbol{P}^*)$ is obviously smaller than $\overline{U}^*$. Thus the current position is farther from the target prey than the current best agent. Hence, it is unnecessary to optimize the transmit power. □

On the other hand, if $H(\boldsymbol{X}) > \overline{U}^*$, we are going to find the lower bound of $F(\boldsymbol{X}, \boldsymbol{P})$ denoted by $\widetilde{F}(\boldsymbol{X}, \boldsymbol{P})$, which helps to determine the upper bound of $\overline{U}(\boldsymbol{X})$, as follows

Lemma 1: $\widetilde{F}(\boldsymbol{X}, \boldsymbol{P})$ is the lower bound of $F(\boldsymbol{X}, \boldsymbol{P})$, and

$$\widetilde{F}(\boldsymbol{X}, \boldsymbol{P}) = \sum_{j\in\mathcal{S}} \sum_{i\in\mathcal{A}_j} \frac{(\eta_i + \gamma_i p_{ij})}{\widetilde{R}_{ij}}, \tag{36}$$

where $\widetilde{R}_{ij} = W \log_2(1 + n_0^{-1} p_{ij} |h_{ij}|^2)$.

Proof: We can easily observe that

$$\begin{aligned} &\frac{p_{ij}|h_{ij}|^2}{n_0 + \sum_{|h_{kj}|^2 \le |h_{ij}|^2} |h_{kj}|^2 p_{kj}} \\ &\le \frac{p_{ij}|h_{ij}|^2}{n_0}, \quad j \in \mathcal{S},\ i, k \in \mathcal{A}_j \\ &\Leftrightarrow R_{ij} \le \widetilde{R}_{ij} \Leftrightarrow F(\boldsymbol{X}, \boldsymbol{P}) \ge \widetilde{F}(\boldsymbol{X}, \boldsymbol{P}). \end{aligned}$$

The bound is tight for $\{\forall j \in \mathcal{S} \mid \exists! i \in \mathcal{A}_j\}$. □

Then, we only need to solve an equivalent lower-bound optimization problem as follows

$$\begin{aligned} &\min_{\boldsymbol{P}} \ \widetilde{F}(\boldsymbol{X}, \boldsymbol{P}) \\ &\text{s.t. C3, C4.} \end{aligned} \tag{37}$$

It is noted here that, the received signals over the same subchannel is supposedly decoded perfectly at the server. Therefore, the condition C5 has been removed. In addition, in cases an UE is assigned with one subchannel, (37) is equivalent to the original problem (27).

Lemma 2: The problem described in (37) is a quasiconvex optimization problem, and the optimal solution can be obtained after a number of steps.

Proof: For a given SA $\boldsymbol{X}$, the problem in (37) can be decomposed into individual subproblems for offloading users over the corresponding subchannel as follows

$$\begin{aligned} &\min_{p_{ij}} \ \frac{W^{-1}(\eta_i + \gamma_i p_{ij})}{\log_2(1 + n_0^{-1} p_{ij} |h_{ij}|^2)}, \quad j \in \mathcal{S},\ i \in \mathcal{A}_j \\ &\text{s.t. } 0 \le p_{ij} \le p_i^0. \end{aligned} \tag{38}$$

Then, based on the study in [5], the objective function of this problem is recognized quasiconvex. Therefore, the optimal solution of (37) denoted by $\widetilde{\boldsymbol{P}}^*$ can be found via solving individual subproblems (38) by an evolved bisection algorithm that is terminated after $\lceil \log_2(p_i^0/\varepsilon) \rceil$ iterations. □

Condition 2: If $H(\boldsymbol{X}) - \widetilde{F}(\boldsymbol{X}, \widetilde{\boldsymbol{P}}^*) \le \overline{U}^*$, the server does not need to solve the TPC subproblem.

Proof: According to Lemmas 1 and 2, $F(\boldsymbol{X}, \boldsymbol{P}^*) \ge \widetilde{F}(\boldsymbol{X}, \boldsymbol{P}^*) \ge \widetilde{F}(\boldsymbol{X}, \widetilde{\boldsymbol{P}}^*)$. Hence, $\overline{U}(\boldsymbol{X}) \le H(\boldsymbol{X}) - \widetilde{F}(\boldsymbol{X}, \widetilde{\boldsymbol{P}}^*)$. If $H(\boldsymbol{X}) - \widetilde{F}(\boldsymbol{X}, \widetilde{\boldsymbol{P}}^*) \le \overline{U}^* \Leftrightarrow \overline{U}(\boldsymbol{X}) \le \overline{U}^*$, the TPC step is unnecessary. □

By Condition 2, the MEC server can omit cases in which UEs use unfavorable subchannels to offload, which might reduce the transmission overhead and shorten the algorithm's running time.

We employ the BWOA-based algorithm to solve the SA problem, from that we obtain task offloading and resource allocation for the ODSTCA problem as described in Algorithm 2, where the searching process is also iterative and contains two stages i.e., *Stage I-B* and *Stage II-B*, in which, *B* stands for BWOA. *Stage I-B* is in charge of updating the current best position. Here, compared to WOA, there is no step for amending positions since the whale population is firstly initiated as random binary positions. The updating position is done by toggling between 0 and 1, which does not make the whales going beyond the variable domain. Conditions 1 and 2 are also checked in *Stage I-B* (lines 12 and 14, respectively). After satisfying both the conditions, the TPC subproblem is addressed by Algorithm 1. Then the fitness value of the current position $\overline{U}(\boldsymbol{X}_k)$ can be recalculated, which helps to update the the current best position (lines 17-22). On the other hand, in *Stage II-B*, positions of whales in the population are updated. The algorithm terminates if either reaching I_2 iterations or satisfying a predefined stop condition as in Algorithm 1.

Here, it is worth to remind two important notations in Algorithm 2. Firstly, the transfer functions proposed in [21] are used for mapping onto three maneuvers *SFP, SEM, SUP*. Secondly, solving the TPC problem for the total M_2 whales' position in all iterations can take time. Hence, for a given SA, before optimizing transmit power, two conditions are proposed to check. This is in order to omit cases that not making benefit for the system utility. Only potential solutions passed both conditions need to optimize the transmit power.

D. COMPLEXITY ANALYSIS

In this section, we analyse the complexity of the proposed algorithm, presented in Algorithm 2. The BWOA is applied in cooperation with the penalty method, a convex optimization technique (to solve the CRA subproblem), and WOA (to solve the TPC subproblem) for solving the constrained problem ODSTCA, which consists of two stages.

In order to calculate the fitness value for each search agent (solution) in *Stage I-B*, the CRA subproblem is firstly solved

Algorithm 2 The Proposed Algorithm for ODSTCA Problem

Initialization:
The binary whale population $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots \boldsymbol{X}_{M_2}\}$ where M_2 is the number of search agents
The iteration index $t = 0$, the convergence index $t_1 = 0$, the maximum number of iterations I_2, the maximum number of iterations for convergence I_β
The current best fitness value $\overline{U}^*(t) = -\infty$
The convergence threshold ε
while $\{t_1 < I_\beta$ **or** $t < I_2\}$ **do**
  $t = t + 1$
  Stage I-B :
  for $k = 1 \rightarrow M_2$ **do**
    Solve the CRA problem (29) and obtain $G(\boldsymbol{X}_k, \boldsymbol{F}^*)$
    Calculate $H(\boldsymbol{X}_k)$
    if $H(\boldsymbol{X}_k) > \overline{U}^*(t)$ **then**
      Solve the lower bound problem (37) and obtain $\widetilde{F}(\boldsymbol{X}_k, \widetilde{\boldsymbol{P}}^*)$
      if $H(\boldsymbol{X}_k) - \widetilde{F}(\boldsymbol{X}_k, \widetilde{\boldsymbol{P}}^*) > \overline{U}^*(t)$ **then**
        Solve the TPC problem by Algorithm 1 and obtain $F(\boldsymbol{X}_k, \boldsymbol{P}^*)$
        Calculate the fitness value $\overline{U}(\boldsymbol{X}_k)$ (35)
        if $\overline{U}(\boldsymbol{X}_k) > \overline{U}^*(t)$ **then**
          $\boldsymbol{X}^* = \boldsymbol{X}_k$
          Save $\boldsymbol{F}^*, \boldsymbol{P}^*$
          $\overline{U}^*(t-1) = \overline{U}^*(t)$
          $\overline{U}^*(t) = \overline{U}(\boldsymbol{X}_k)$
        end if
        if $\mid \overline{U}^*(t) - \overline{U}^*(t-1) \mid < \varepsilon$ **then**
          $t_1 = t_1 + 1$
        else
          $t_1 = 0$
        end if
      end if
    end if
  end for
  Stage II-B :
  for $k = 1 \rightarrow M_2$ **do**
    Update WOA's parameters
    Update the new position $\boldsymbol{X}_k$
  end for
end while
Determine the offloading decision set $\mathcal{A}$ from the SA solution $\boldsymbol{X}^*$
Output: The resource allocation solution $\boldsymbol{F}^*, \boldsymbol{P}^*, \boldsymbol{X}^*$, the offloading decision solution $\mathcal{A}$

as IV-A, which has the complexity $\mathcal{O}(N)$. Next, Condition 1 is tested. Only the solutions that passed this condition need to address the lower bound problem, which is equivalent to solving at most N individual subproblems as Lemma 2, the complexity is $\mathcal{O}(N \log_2(p_i^0/\varepsilon))$. Finally, only the solutions that passed Condition 2 is solved the TPC problem, following Algorithm 1. Similar to [21], in Algorithm 1, the WOA is applied with penalty method for N inequality constraints C4 and at most NS constraints C5. For a M_1-whale population, the dimension of whales is NS, the WOA iterates maximum I_1 iterations, hence the computational complexity of Algorithm 1 is $\mathcal{O}(I_1\ M_1(N + NS + NS))$, which is equivalent to $\mathcal{O}(I_1\ M_1\ NS)$. We consider the worst case that a solution SA $\boldsymbol{X}$ passed both Condition 1, 2, the complexity in *Stage I-B* for each solution can be summarized as $\mathcal{O}(N + N \log_2(p_i^0/\varepsilon) + I_1\ M_1\ NS)$, which is equivalent to $\mathcal{O}(N \log_2(p_i^0/\varepsilon) + I_1\ M_1\ NS)$.

In *Stage II-B*, the position of each whale is updated and index functions are checked to evaluate the penalty term, the complexity is $\mathcal{O}(NS+N)$. Therefore, if we denote by I_2 the maximum number of iterations for Algorithm 2 to converge, and M_2 the number of search agents, the complexity of the proposed algorithm is $\mathcal{O}(I_2\ M_2(N \log_2(p_i^0/\varepsilon) + I_1\ M_1\ NS + NS))$, which is eventually equivalent to

$$\mathcal{O}(I_2 M_2(N \log_2(p_i^0/\varepsilon) + I_1 M_1 NS)). \tag{39}$$

V. PERFORMANCE EVALUATION AND DISCUSSIONS

In this section, computer simulations are performed to validate the correctness of our proposed algorithm ODSTCA, as well as to show its effectiveness in comparison with the conventional ones. In particular, we consider the other following algorithms

1) *Exhaustive search* (EX): The MEC server searches for all feasible cases of subchannel allocation ($(S + 1)^N$ in total), optimizes the computing resource, transmit power allocation for each case, and then chooses the best solution.
2) *All Remote Joint Optimization Algorithm* (ARJOA): The MEC server accepts all offloading UEs, then jointly optimizes the subchannel, transmit power, and computing resource allocation [5].
3) *Independent Offloading Joint Optimization Algorithm* (IOJOA): All UEs independently make their offloading decision (whether to offload or not), then a joint optimization scheme of subchannel, transmit power, and computing resource is performed at the MEC server [5].
4) OFDMA: Orthogonal frequency division multiple access technique is used instead of NOMA, i.e., each subchannel is allocated to at most a UE [5].
5) *All Local Computing Algorithm* (ALCA): All UEs execute their tasks locally.

Other simulation settings are described as follows. A MEC server is located with a BS at the center of a small cell of radius 250 m. If not stated otherwise, the number of users $N = 18$. The number of subchannels $S = 5$, each with a bandwidth of $W = 1$ MHz. The noise power $n_0 = -114$ dBm. Similar to [25], we adopt the distance-dependent path-loss model as $140.7 + 36.7 \log_{10} d$ [dB], where d (in km) is the distance from each UE to the server. The channels between UEs and the server are supposedly independent. The simulation results are averaged by 300 realizations. Moreover,

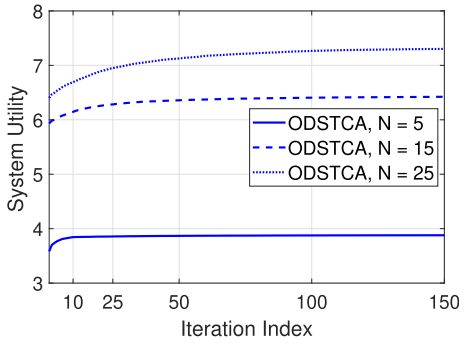

FIGURE 2. Convergence behavior of ODSTCA in terms of system utility.

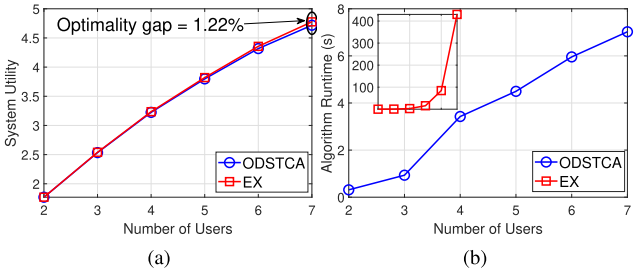

FIGURE 3. Comparison of ODSTCA and EX regarding the number of UEs: (a) system utility, (b) algorithm runtime.

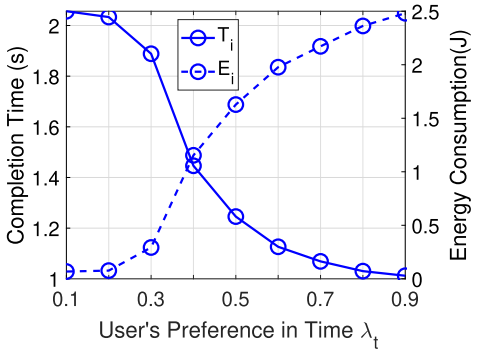

FIGURE 4. Completion time and energy consumption of each UE with respect to the UE's preference in time.

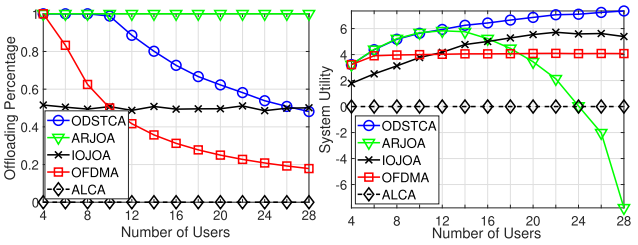

FIGURE 5. Performance comparison of ODSTCA and conventional schemes regarding the number of UEs: (a) offloading percentage, (b) system utility.

the computing capacity of the server, if not stated otherwise, is set by 10 GHz. In regard to the condition of implementing efficient SIC technique, the threshold γ_{tol} is set by 0 dB, while each UE's maximum transmit power p_i^0 is 24 dBm [22]. The energy coefficient of each UE $\kappa_i = 5 \times 10^{-27}$, and power amplifier efficiency is $\xi_i = 1$. The computing capacity of each UE is randomly selected from $\{0.5, 0.8, 1\}$ (GHz). For the computation task, we consider a face recognition application, where the input data size $\alpha_i = 420$ (kB), $\beta_i = 1000$ (Megacycles) [26]. The preference coefficients in time and energy are both set to 0.5.

A. CONVERGENCE BEHAVIOR AND SUBOPTIMALITY OF THE PROPOSED ALGORITHM

We first study in Fig. 2 the convergence behavior of the proposed ODSTCA in terms of system utility, for different numbers of UEs $N = 5$, 15 and 25. It is seen that the algorithm converges after a number of iterations depending on the number of UEs, while in this example, ODSTCA converges in all cases after 100 iterations. Moreover, in Fig. 3, we compare the performance of the proposed algorithm with that of EX in terms of the system utility and the algorithm runtime, for a given number of UEs. The results are obtained with Windows Desktop PC, quad-core 3.1 GHz CPU, 16 GB RAM. In this scenario, $S = 3$, and $N = 2-7$ are set. It is seen that, while the performance of the two methods is almost the same in this setting, the runtime of ODSTCA is much smaller than that of EX. This confirms the validity and effectiveness of the proposed algorithm.

B. IMPACT OF THE SYSTEM PARAMETERS ON THE ALGORITHMS' PERFORMANCE

In Fig. 4, we illustrate the effect of UE's preference in time λ_i^t to the completion time and energy consumption of each UE in the system. It is noted here that $\lambda_i^e = 1 - \lambda_i^t$. It can be seen from the figure that when λ_i^t increases the completion time decreases at the cost of the energy. This is because the tasks that might take a shorter time to complete have a higher probability to offload than the others with a longer one.

Fig. 5 depicts the offloading percentage and system utility of the proposed ODSTCA and the above conventional schemes i.e., ARJOA, IOJOA, OFDMA, and ALCA, for a given number of UEs. It is seen that the ODSTCA significantly outperforms other baseline schemes especially with a large number of UEs, which can be explained as follows

1) In ALCA, UEs do not take advantage from remote computation so that offloading percentage, and thus, the system utility is always 0.
2) Offloading percentages of ARJOA and IOJOA are, respectively, 1 and 0.5. In this case, when the number of UEs increases ($N \geq 14$ (ARJOA) and $N \geq 22$ (IOJOA)), the system utility decreases due to limited resources of subchannels S, the bandwidth W, the computing capacity f_0 of MEC server, etc.
3) Although both offloading percentages of OFDMA and ODSTCA decrease according to the increase of N, the number of offloading UEs with ODSTCA is higher

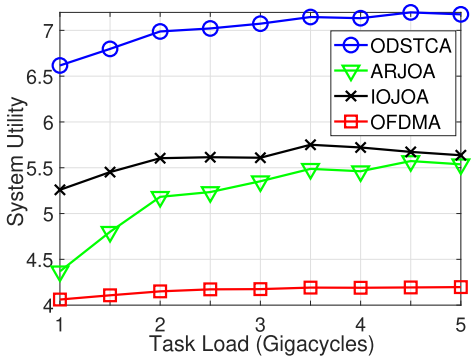

FIGURE 6. Performance comparison between schemes with respect to the task loads.

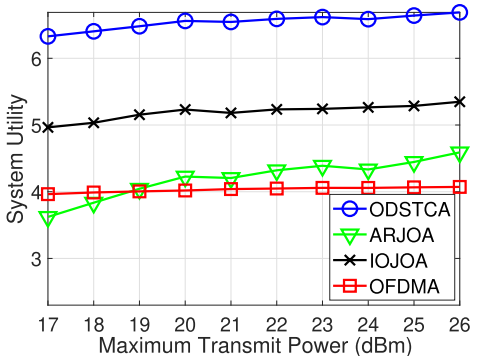

FIGURE 8. Performance comparison between schemes with respect to the UEs' maximum transmit power.

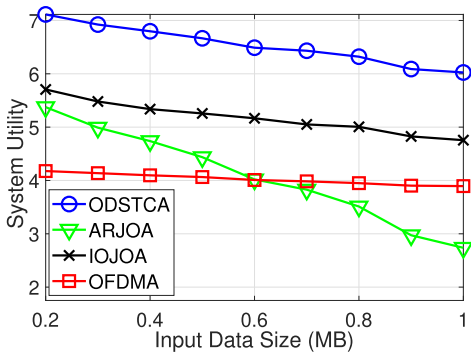

FIGURE 7. Performance comparison between schemes with respect to the task input sizes.

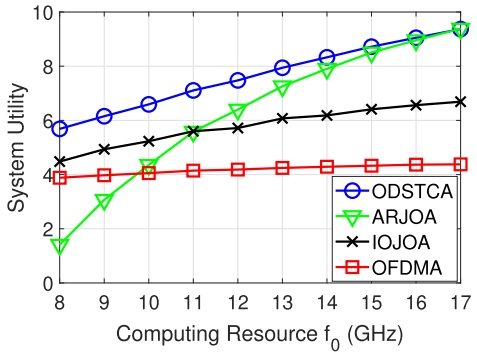

FIGURE 9. Performance comparison between schemes with respect to the MEC server's computing capacity.

than that with OFDMA. The reason is that ODSTCA, by exploiting NOMA, helps more UEs to have benefit from remote computation over the same number of subchannels. Therefore, the system utility of ODSTCA is higher than that of OFDMA.

Figs. 6 and 7 illustrate the impacts of task loads and task input data sizes, respectively, on the performance of the above-considered schemes. Here, we skip showing the ALCA's performance (from now on) without loss of generality since its offloading percentage is 0. It is logical to observe that the performance is proportional and inversely proportional to the task load and input data size, respectively. Nevertheless, in all cases, ODSTCA outperforms the other methods in terms of system utility thanks to the efficiency of the proposed optimization algorithms.

Fig. 8 illustrates the impact of UEs' maximum transmit power p_i^0 on the system performance. It can be seen that when the power increases, the performance of SIC at the MEC server is more efficient and thus, the system utility of all methods except OFDMA (without OFDMA) increases until it reaches a threshold. The reason for this performance saturation is due to the signal interference to other UEs on the same subchannel.

We also describe in Fig. 9 the impact of the MEC server's computation capacity f_0 on system performance. In general, it is seen that the higher f_0, the better performance of all schemes thanks to the offloading mechanism. Nevertheless, in both the two above figures, ODSTCA obtains the highest system utility, which again, confirms the efficiency of the proposed algorithm in comparison with the conventional ones.

VI. CONCLUSION

In this paper, we studied the issue of computation offloading in NOMA-based MEC systems. Our optimization problem ODSTCA was modeled as an MINLP one, which was then decomposed into subproblems of CRA, TPC, and SA for more tractable. A swarm intelligence approach i.e., WOA, was proposed to solve the TPC and SA subproblems, while the CRA was handled by convex optimization technique. Computer simulations were performed with different system parameters to validate the correctness and efficiency of the proposed algorithm. The obtained results showed that the algorithm converged after a number of iterations. Its performance was comparable with that of the exhaustive search, while the computational complexity was much lower. Moreover, the proposed algorithm was shown to outperform conventional and baseline schemes such as ARJOA, IOJOA, OFDMA, and ALCA in terms of system utility.

As for future works, it is desirable to consider NOMA-based multi-server MEC systems [27]. A number of research issues should be further investigated, for examples, the hierarchical MEC model with small-cell and macro-cell MEC

servers, the computation migration between MEC servers, and the joint design of user association, offloading decision, and resource allocation. Although optimization problems in such scenarios would be more challenging, but they have the potential to improve the quality of service for UEs and network performance.

REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017.

[2] S. M. R. Islam, N. Avazov, O. A. Dobre, and K.-S. Kwak, "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 721–742, 2nd Quart., 2017.

[3] M. Vaezi, G. A. Aruma Baduge, Y. Liu, A. Arafa, F. Fang, and Z. Ding, "Interplay between NOMA and other emerging technologies: A survey," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 4, pp. 900–919, Dec. 2019.

[4] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.

[5] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.

[6] Q.-V. Pham, T. Leanh, N. H. Tran, B. J. Park, and C. S. Hong, "Decentralized computation offloading and resource allocation for mobile-edge computing: A matching game approach," *IEEE Access*, vol. 6, pp. 75868–75885, 2018.

[7] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[8] A. Kiani and N. Ansari, "Edge computing aware NOMA for 5G networks," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1299–1306, Apr. 2018.

[9] Q.-V. Pham, H. T. Nguyen, Z. Han, and W.-J. Hwang, "Coalitional games for computation offloading in NOMA-enabled multi-access edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1982–1993, Feb. 2020.

[10] W. Zhou, L. Lin, J. Liu, D. Zhang, and Y. Xie, "Joint offloading decision and resource allocation for multiuser NOMA-MEC systems," *IEEE Access*, vol. 7, pp. 181100–181116, 2019.

[11] Z. Yang, C. Pan, J. Hou, and M. Shikh-Bahaei, "Efficient resource allocation for mobile-edge computing networks with NOMA: Completion time and energy minimization," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7771–7784, Nov. 2019.

[12] J. Zhu, J. Wang, Y. Huang, F. Fang, K. Navaie, and Z. Ding, "Resource allocation for hybrid NOMA MEC offloading," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4964–4977, Jul. 2020.

[13] L. Qian, Y. Wu, F. Jiang, N. Yu, W. Lu, and B. Lin, "NOMA assisted multi-task multi-access mobile edge computing via deep reinforcement learning for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, early access, Jun. 10, 2020, doi: 10.1109/TII.2020.3001355.

[14] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots and Biological Systems: Towards a New Bionics?* Berlin, Germany: Springer, 1993, pp. 703–712.

[15] Q.-V. Pham, D. C. Nguyen, S. Mirjalili, D. T. Hoang, D. N. Nguyen, P. N. Pathirana, and W.-J. Hwang, "Swarm intelligence for next-generation wireless networks: Recent advances and applications," 2020, *arXiv:2007.15221*. [Online]. Available: http://arxiv.org/abs/2007.15221

[16] Z. Wang, H. Ding, B. Li, L. Bao, and Z. Yang, "An energy efficient routing protocol based on improved artificial bee colony algorithm for wireless sensor networks," *IEEE Access*, vol. 8, pp. 133577–133596, 2020.

[17] Q.-V. Pham, T. Huynh-The, M. Alazab, J. Zhao, and W.-J. Hwang, "Sum-rate maximization for UAV-assisted visible light communications using NOMA: Swarm intelligence meets machine learning," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10375–10387, Oct. 2020.

[18] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu, and K. Yang, "Deep-learning-based joint resource scheduling algorithms for hybrid MEC networks," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6252–6265, Jul. 2020.

[19] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[20] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[21] Q.-V. Pham, S. Mirjalili, N. Kumar, M. Alazab, and W.-J. Hwang, "Whale optimization algorithm with applications to resource allocation in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4285–4297, Apr. 2020.

[22] M. Shipon Ali, H. Tabassum, and E. Hossain, "Dynamic user clustering and power allocation for uplink and downlink non-orthogonal multiple access (NOMA) systems," *IEEE Access*, vol. 4, pp. 6325–6343, 2016.

[23] *Humpback Whale Behaviors*. Accessed: Jun. 12, 2020. [Online]. Available: https://www.adfg.alaska.gov/static/viewing/pdfs/whale_behaviors.pdf

[24] V. Kumar and D. Kumar, "Binary whale optimization algorithm and its application to unit commitment problem," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2095–2123, Apr. 2020.

[25] X. Chu, D. Lopez-Perez, Y. Yang, and F. Gunnarsson, *Heterogeneous Cellular Networks: Theory, Simulation and Deployment*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[26] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2012, pp. 59–66.

[27] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Edge intelligence for energy-efficient computation offloading and resource allocation in 5G beyond," *IEEE Trans. Veh. Technol.*, early access, Aug. 4, 2020, doi: 10.1109/TVT.2020.3013990.

HUONG-GIANG T. PHAM received the B.E. degree in communications engineering from the Hanoi University of Science and Technology (HUST), Vietnam, in June 2020. She is currently a Research Assistant with the Communications Theory and Applications Research Group (CTARG), School of Electronics and Telecommunications, HUST. Her research interest includes resource allocation in multiaccess edge computing (MEC) systems.

QUOC-VIET PHAM (Member, IEEE) received the B.S. degree in electronics and telecommunications engineering from the Hanoi University of Science and Technology, Vietnam, in 2013, and the Ph.D. degree in telecommunications engineering from Inje University, South Korea, in 2017. From September 2017 to December 2019, he was with Kyung Hee University, Changwon National University, and Inje University on various academic positions. He is currently a Research Professor with the Research Institute of Computer, Information and Communication, Pusan National University, South Korea. His research interests include convex optimization, game theory, machine learning to analyze and optimize edge/cloud computing, and 5G and beyond networks. He received the Best Ph.D. Dissertation Award in engineering from Inje University in 2017.

ANH T. PHAM (Senior Member, IEEE) received the B.E. and M.E. degrees in electronics engineering from the Hanoi University of Technology, Vietnam, in 1997 and 2000, respectively, and the Ph.D. degree in information and mathematical sciences from Saitama University, Japan, in 2005. From 1998 to 2002, he was with NTT Corporation, Vietnam. Since 2005, he has been a Faculty Member with The University of Aizu, where he is currently a Professor and the Head of the Computer Communications Laboratory, Division of Computer Engineering. He has authored/coauthored over 200 peer-reviewed articles on these topics. His research interests include communication theory and networking with a particular emphasis on modeling, design, and performance evaluation of wired/wireless communication systems and networks. He is a member of IEICE and OSA.

CHUYEN T. NGUYEN received the B.E. degree in electronics and telecommunications from the Hanoi University of Science and Technology (HUST), Vietnam, in 2006, the M.S. degree in communications engineering from National Tsing-Hua University, Taiwan, in 2008, and the Ph.D. degree in informatics from Kyoto University, Japan, in 2013. From September to November 2014, he was a Visiting Researcher with The University of Aizu, Japan. He is currently an Assistant Professor with the School of Electronics and Telecommunications, HUST. His current research interests include MAC protocol design and reliable transmission in wireless systems. He received the Fellow Award from the Hitachi Global Foundation in August 2016.