

Whale Optimization Algorithm with Applications to Resource Allocation in Wireless Networks

Quoc-Viet Pham, Seyedali Mirjalili, Neeraj Kumar, Mamoun Alazab, and Won-Joo Hwang

Abstract—Resource allocation plays a pivotal role in improving the performance of wireless and communication networks. However, the optimization of resource allocation is typically formulated as a mixed-integer non-linear programming (MINLP) problem, which is non-convex and NP-hard by nature. Usually, solving such a problem is challenging and requires specific methods due to the major shortcomings of the traditional approaches, such as exponential computation complexity of global optimization, no performance optimality guarantee of heuristic schemes, and large training time and generating a standard dataset of machine learning based approaches. Whale optimization algorithm (WOA) has recently gained the attention of the research community as an efficient method to solve a variety of optimization problems. As an alternative to the existing methods, our main goal in this article is to study the applicability of WOA to solve resource allocation problems in wireless networks. First, we present the fundamental backgrounds and the binary version of the WOA as well as introducing a penalty method to handle optimization constraints. Then, we demonstrate three examples of WOA to resource allocation in wireless networks, including power allocation for energy-and-spectral efficiency tradeoff in wireless interference networks, power allocation for secure throughput maximization, and mobile edge computation offloading. Lastly, we present the adoption of WOA to solve a variety of potential resource allocation problems in 5G wireless networks and beyond.

Index Terms—Wireless and Communication Networks, Non-Orthogonal Multiple Access, Meta-heuristic Optimization, Edge Computing, Resource Allocation, Whale Optimization Algorithm.

I. INTRODUCTION

A. Motivation and Related Works

The optimization of resources in wireless networks, such as, power allocation for spectral and energy efficiency, sub-carrier assignment in orthogonal frequency-division multiple access (OFDMA) and multi-carrier non-orthogonal multiple access (NOMA), time allocation in time-division multiple

access (TDMA), computation offloading in multi-access edge computing (MEC), remote radio head selection in cloud radio access network (C-RAN), and cluster-head selection in wireless sensor networks, are typically formulated as mixed-integer non-linear programming (MINLP) problems [1], which are difficult to solve and can be NP-hard in general. Various optimization and learning techniques have been proposed to get the solutions to the MINLP problems of which some of them are as follows:

1) *Global Optimization*: some algorithms, e.g., branch-and-bound (BnB) and dynamic programming, can be used to obtain the globally optimal solution to any MINLP problem [2]. However, the exponential worst-case complexity limits their implementations and applications in wireless networks as the channels are usually highly dynamic and massive Internet of Things (IoT) devices with cellular connections will be available in the years to come.

2) *Heuristic Methods*: To reduce the exponential computational complexity of global optimization approaches, heuristic algorithms are widely used in the literature. An example of the heuristic algorithm is to relax integer variables to be continuous and to approximate the binary constraints by the quadratic ones. For instance, the subcarrier assignment to a device-to-device (D2D) pair d can be denoted by x_d^c , whose value is $x_d^c = 1$ if this D2D pair is allowed to reuse the subcarrier of the cellular user (CU) c and $x_d^c = 0$ otherwise [3]. By relaxing, x_d^c can be represented as $\hat{x}_d^c = [0, 1]$, which may be understood as the fraction of time the D2D pair d reuses the subcarrier from the CU c . Nevertheless, a major drawback of low-complexity heuristic algorithms is that the convergence to the optimal solution and thus the performance gap compared to the optimal schemes are not guaranteed.

3) *Game-Theoretic Approaches*: game theory provides a mathematical tool to study the interactions among independent rational players. More recently, game theory has played as an important tool for solving many problems in wireless and communications networking. For example, joint matching theory and coalition game was used to solve the resource allocation problem heterogeneous C-RAN with D2D communication [4], and transport theory for unmanned aerial vehicle (UAV) communications [5]. Despite its helpfulness and wide applications, game theory is based on rational responses and relies on the mathematical model that may influence the game players' strategy and outcome, which is therefore not applicable in some scenarios.

4) *Machine Learning (ML)-based Approaches*: ML has been recently integrated in wireless and communication networks as a disruptive technology that can balance between the

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Quoc-Viet Pham is with Research Institute of Computer, Information and Communication, and Won-Joo Hwang is with Department of Biomedical Convergence Engineering, Pusan National University, 46241, Republic of Korea. Email: {vietpq90@gmail.com, wjhwang@pusan.ac.kr}

Seyedali Mirjalili is with Center for Artificial Intelligence Research and Optimization, Torrens University Australia, 90 Bowen Terrace, Fortitude Valley, Brisbane, QLD 4006, Australia. Email: seyedali.mirjalili@griffithuni.edu.au

N. Kumar is with the Department of Computer Science and Engineering, Thapar University, Patiala 147004, India. Email: neeraj.kumar@thapar.edu

Mamoun Alazab is with College of Engineering, IT and Environment, Charles Darwin University, Australia. Email: alazab.m@ieee.org

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.NRF-2019R1C1C1006143). Quoc-Viet Pham is the corresponding author.

performance and computational complexity [6], [7]. However, there are some tasks that may defeat the use of ML in wireless networks. For example, generating the standard training data set for a problem is a non-trivial task, which is sometimes not available. Moreover, training a neural network with a very large data set and a large number of parameters is a time-consuming process.

To provide a viable alternative for addressing complicated resource allocation problems in wireless communications, our goal is to make use of metaheuristic algorithms¹. We are inspired by studies on metaheuristics and their applications to many large-scale and real-world engineering optimization problems, e.g., electrical engineering [8], civil engineering [9], mechanical engineering [10], and industrial engineering [11]. Recently, Mirjalili *et al.* proposed a new metaheuristic, namely *whale optimization algorithm* (WOA), which imitates the hunting behavior of humpback whales [12]. Humpback whales are generally considered as predators and their favorite diet mostly includes krill and small fish. One of the most popular feeding methods of humpback whales is the bubble-net technique, which is the unique behavior observed only in humpback whales. In bubble-net feeding, humpback whales swim in a shrinking circle and blow bubbles below the prey to force them towards the surface of the ocean [12]. It was shown in [12] that the WOA outperforms state-of-the-art metaheuristics, e.g., particle swarm optimization (PSO), gravitational search algorithm (GSA), genetic algorithm (GA), and ant colony optimization (ACO), in almost all the test scenarios. In the context of wireless networks, a mobile user can be considered as the humpback whale and the optimal solution of the optimization variables can be viewed as the prey. By imitating the hunting behavior of humpback whales, a variety of optimization problems in wireless networks can be solved by the WOA with highly competitive performance. More recently, many improved versions of the WOA were proposed, for example, binary version of the WOA [13]–[15], tradeoff between exploration and exploitation [16], chaotic WOA [17], and WOA for optimizing neural networks [18].

The WOA algorithm has considerable advantages to be considered as an efficient optimizer. First, the WOA does not require to compute gradients, which is opposed to the gradient-based algorithms, where the gradient and step size are calculated and updated in each iteration during the optimization process [19]. With the existence of massive connectivities, heterogeneous mobile devices (e.g., wearable computing devices, smartphones, IoT sensor devices, and network elements), the convergence of the heterogeneous wireless technologies, and the emergence of many new applications (e.g., virtual reality, video surveillance and analytics, and autonomous vehicles), an optimization problem may become extraordinarily complicated. Moreover, it is difficult to compute gradients in general due to the dependence on a large number of optimization variables and the requirement for huge computational capability and memory. The WOA can relax such computations, i.e., the WOA is a gradient-free method. As a population-

based algorithm; however, the computational cost is higher than gradient-based algorithms due to the need to evaluate multiple solutions using the objective function in each step of optimization. Second, the WOA is insensitive to the initial feasible solution(s), which may greatly affect the convergence and performance of the traditional methods. Next, the WOA algorithm has been equipped with adaptive mechanisms that appropriately balance the explorative and exploitative behaviors of this algorithm. This increases the chance of avoiding locally optimal solutions, which have been demonstrated in the original paper of WOA. Finally, since the WOA is easy to implement and flexible, it is applicable to a wide range of optimization problems rather than a particular problem. This advantage would make the WOA highly efficient in wireless and communication networks because of the fact that lots of objective functions, performance metrics, and constraints are considered in the optimization of wireless systems.

B. Our Contributions

As aforementioned, the WOA has found important applications in a multitude of disciplines; however, we are not aware of any work providing the application and applicability of the WOA in wireless and communication networks. In this paper, our main motivations are to study the WOA algorithm and provide a compendious tutorial on the WOA algorithm, together with the discussion on its potential applications to various resource allocation optimization problems. The main contributions can be summarized as follows:

- We first provide a brief overview of WOA, including the mathematical model and optimization algorithm. Since the original WOA is only appropriate for continuous and unconstrained optimization problems, we present the binary version of the WOA (BWOA) and introduce the penalty method to deal with optimization constraints. Combination of the original WOA with BWOA and penalty method allows us to solve a wide range of optimization problems and obtain a high solution quality.
- To illustrate the applicability of WOA, we investigate three resource allocation problems in wireless networks: secure throughput maximization, energy and spectral efficiency tradeoff, and mobile edge computation offloading, which are then solved by the WOA algorithm. Simulation results are conducted to show that the WOA algorithm can converge very fast and achieve almost the same performance as in the existing algorithms.
- We outline some possible applications of WOA to unmanned aerial vehicle (UAV) trajectory optimization, interference management in ultra-dense networks (UDNs), user association and scheduling, mode selection, and computation offloading in multi-carrier NOMA enabled MEC systems. The results verify that the WOA is a highly promising algorithm to optimize resource allocation problems in wireless networks.

The rest of this work is organized as follows. Section II introduces the fundamentals of WOA. Section III illustrates three examples as the applications of WOA in wireless net-

¹The prefix *meta* means that a metaheuristic algorithm typically achieves a better performance than the corresponding heuristic scheme as the tradeoff of exploration and exploitation is considered in any metaheuristic algorithm.

works. Then, potential applications of WOA are discussed in Section IV. Finally, Section V concludes the paper.

II. WOA: FUNDAMENTALS AND BINARY VERSION

In this section, the fundamentals of the WOA algorithm are discussed including encircling prey, bubble-net feeding method, and search for prey. We then present the binary version of the WOA algorithm and finally introduce the penalty method as the constraint-handling technique.

A. Encircling prey

Humpback whales can recognize the location of prey (e.g., krill) and cover them completely. It is assumed in the WOA algorithm that the current best search agent is the target prey and humpback whales update their position towards the best search agent over the course of iterations. The following equations are used to mathematically formulate this behavior

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right|, \quad (1)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D}, \quad (2)$$

where \vec{A} and \vec{C} are coefficient vectors, t is the current iteration, $\vec{X}^*(t)$ is the position of the best search agent, $|\cdot|$ is the absolute value, and \cdot denotes the element-wise multiplication. The coefficient vectors \vec{A} and \vec{C} are calculated as follows [12]:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}, \quad (3)$$

$$\vec{C} = 2 \cdot \vec{r}, \quad (4)$$

where \vec{a} is linearly decreased from 2 to 0 over the course of iterations and in both exploration and exploitation phases, and \vec{r} is a random vector in $[0, 1]$. Denote by t and I_{\max} the iteration index and maximum number of iterations, respectively, the control parameter \vec{a} can be updated as $\vec{a} = 2(1 - t/I_{\max})$.

The main purpose of (3) and (4) is to balance exploration and exploitation. The perimeter r is random in both equations, which provides a stochastic behavior for position updating of the population. In Eq. (3), the range of random numbers decreases from 2 to 0. The exploration is done when $A \geq 1$, and the WOA performs exploitation when $A < 1$. To reduce the probability of permanently being trapped in local solutions when the WOA performs exploitation, the parameter C can be a random number in $[0, 2]$. This leads to boosting exploration of exploitation at any stages of optimization.

B. Bubble-Net Attacking Method

The *shrinking encircling* and *spiral updating position* mechanisms are simultaneously used to model the bubble-net attacking method of humpback whales. The shrinking encircling mechanism is achieved by setting the coefficient vector \vec{A} in $[-1, 1]$ while linearly reducing the value of \vec{a} over the course of iterations. In doing so, the new position will be located between the current position of the agent and the position of the best search agent.

To mimic the helix-shaped movement of humpback whales, the spiral equation between the location of the prey and the whale can be used as follows:

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|, \quad (5)$$

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), \quad (6)$$

where b is a constant that is used to define the logarithmic spiral shape, and l is a random number in $[-1, 1]$.

Because of the fact that humpback whales swim around the prey within a shrinking circle and move along the spiral-shaped path at the same time, the shrinking encircling method and the spiral approach are used simultaneously. To model this behavior, it is assumed that each mechanism is performed with the probability of 50% as follows:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D}, & \text{if } p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), & \text{if } p \geq 0.5 \end{cases} \quad (7)$$

where p is a random number in $[0, 1]$.

C. Search for Prey

The same approach for the shrinking encircling mechanism can be used for the prey search. However, the coefficient vector \vec{A} with $|\vec{A}| > 1$ is utilized and the position $\vec{X}^*(t)$ of the best search agent is now replaced by the position \vec{X}_{rand} of a whale selected randomly from the current population. In other words, humpback whales are forced to move away from a random whale, thus the WOA algorithm is able to extend the search space and perform the global search. The mathematical model for the prey search is given as

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X}(t) \right|, \quad (8)$$

$$\vec{X}(t+1) = \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{D}, \quad (9)$$

It is worth noting that the bubble-net attacking method and search for prey are exactly two phases in any meta-heuristic algorithm: *exploitation* and *exploration*, respectively. The bubble-net attacking method focuses on the search in a local region by exploiting the current best solution, while the search for prey is to increase the diversity of the solutions in order to achieve a global solution. As the number of iterations increases, exploitation is more desired whereas exploration is preferred at very initial iterations. Over the last few years, many efforts have been devoted to improving the WOA, which mainly focus on the exploitation and exploration capabilities, and their balance. For example, [16] proposed using the arcsine function to control the tradeoff between exploration and exploitation, and the Lévy flight trajectory was used to improve the exploration capability of the WOA in [15]. Adopting such improved versions of the WOA to optimize resource optimization problems in wireless networks in an interesting direction for the future. To conclude, the WOA algorithm can be considered as an efficient global optimizer thanks to its well balance between exploitation and exploration.

In summary, the pseudocode of the WOA algorithm is illustrated in Alg. 1. The computational complexity of computing fitness function is $\mathcal{O}(ND)$, where N is the whale

²For a minimization (maximization) problem, the search agent that has the smallest (largest) fitness value is considered as the best search agent.

Alg. 1 Pseudocode of the WOA Algorithm.

```

1: Initialize the whale population  $X_i, i = \{1, \dots, N\}$ , iteration  $t = 1$ , maximum number of iterations  $I_{\max}$ .
2: Calculate the fitness of the search agents and identify the best search agent  $\vec{X}^*(t)$ .
3: repeat
4:   for  $k \leftarrow 1$  to  $M$  (the number of whales) do
5:     Update  $a, A, C, l$  and  $p$ .
6:     if  $p < 0.5$  then
7:       if  $|A| < 1$  then
8:         Update  $\vec{D}$  by (1) and  $\vec{X}$  by (2).
9:       else
10:        Select a random  $\vec{X}_{\text{rand}}$  and update  $\vec{D}$  by (8).
11:        Update the position  $\vec{X}$  by (9).
12:      end if
13:    else
14:      Update  $\vec{D}$  by (5) and  $\vec{X}$  by (6).
15:    end if
16:  end for
17:  Calculate the fitness of each search agent.
18:  Update  $X^*(t)$  of the best search agent.
19:  Increase the iteration index  $t = t + 1$ .
20: until  $t > I_{\max}$  or the stopping criterion is satisfied.
21: Output: The best fitness value and best whale position.

```

population and D is the dimension of search agents [13]. Similarly, updating the position vector of all the search agents at each iteration requires a complexity level of $\mathcal{O}(ND)$. Therefore, the complexity of Alg. 1 can be shown to be of $\mathcal{O}(NDT)$, where T denotes the number of maximum iterations/generations.

D. Binary Whale Optimization Algorithm

The WOA algorithm in its original form is for continuous optimization; however, many problems are formulated as mixed-integer programming (MIP) problems, where each variable value can be discrete or binary. To deal with the combinational optimization, binary versions of the WOA algorithm have been proposed [13], [14]. The pseudocode of the BWOA algorithm is illustrated in Alg. 2. It is worth mentioning that the BWOA follows the same steps as in the continuous version except for the update of position vector, thus the BWOA also has a computational complexity of $\mathcal{O}(NDT)$.

According to [13], the main differences between WOA and BWOA lies in the position updating procedure and the transfer function. While in WOA, the position updating is based on the position of the best search agent and can be any continuous value within the feasible set, the position updating in BWOA is based on the toggling between the values 1 and 0. The change in the current bit is decided by a probability that is calculated according to the helix-shaped movement of humpback whales. Moreover, some concepts are considered in a transfer function [20]: *i*) the value should lie in $[0, 1]$ as the transfer function represents the probability of changing a position from 0 to 1 and vice versa, and *ii*) the transfer function should be proportional to the distance between the position of

Alg. 2 Pseudocode of the binary WOA Algorithm.

```

1: Initialize the whale population  $X_i, i = \{1, \dots, N\}$ , iteration  $t = 1$ , maximum number of iterations  $I_{\max}$ , and set the stopping tolerance  $\epsilon$ .
2: Calculate the fitness of the search agents and identify the best search agent  $\vec{X}^*(t)$ .
3: repeat
4:   for  $k \leftarrow 1$  to  $M$  (the number of whales) do
5:     Update  $a, A, C$  and generate  $p_{\text{BWOA}}$ .
6:     if  $p < 0.5$  then
7:       if  $|A| < 1$  then
8:         Update  $\vec{D}$  by (1) and  $\sigma_{\text{sem}}$  by (10).
9:         Update the position  $\vec{X}(t)$  based on (11).
10:      else
11:        Select a random agent  $\vec{X}_{\text{rand}}$  and update  $\vec{D}$ .
12:        Update  $\sigma_{\text{sp}}$  via (14) and  $\vec{X}(t)$  by (15).
13:      end if
14:    else
15:      Update  $\vec{D}$  by (5) and  $\sigma_{\text{sup}}$  by (13).
16:      Update the position  $\vec{X}(t)$  by (12).
17:    end if
18:  end for
19:  Calculate the fitness of each search agent.
20:  Update  $X^*(t)$  of the best search agent.
21:  Increase the iteration index  $t = t + 1$ .
22: until  $t > I_{\max}$  or  $\frac{|X^*(t) - X^*(t-1)|}{|X^*(t-1)|} \leq \epsilon$ .
23: Output: The best fitness value and best whale position.

```

the humpback whale and the prey, i.e., the search agent that is far away from the best search agent should have a higher probability. The modification details are given as follows.

Shrinking encircling mechanism: The step size is computed according to the following transfer function

$$\sigma_{\text{sem}} = \frac{1}{1 + \exp\left(-10\left(\vec{A} \cdot \vec{D} - 0.5\right)\right)}, \quad (10)$$

where \vec{D} and \vec{A} are computed based on (1) and (3), respectively. Actually, σ_{sem} can be treated as a probability that is used to determine whether or not the bit values should be toggled. The position of the search agents is modified as

$$\vec{X}(t+1) = \begin{cases} \mathbb{C}(\vec{X}(t)), & \text{if } p_{\text{BWOA}} < \sigma_{\text{sem}} \\ \vec{X}(t), & \text{if } p_{\text{BWOA}} \geq \sigma_{\text{sem}}, \end{cases} \quad (11)$$

where p_{BWOA} is a uniform random number in $[0, 1]$ and $\mathbb{C}(\cdot)$ denotes the ones' complement operation. More specifically, to get the ones' complement of a binary number, we can simply invert all the bits in the binary number, i.e., swapping 0s for 1s and vice versa. For example, with $\sigma_{\text{sem}} = 0.45$ and $\vec{X}(t) = [0 \ 1 \ 1 \ 0 \ 1]$, the next position will be $\vec{X}(t+1) = 1C(\vec{X}(t)) = [1 \ 0 \ 0 \ 1 \ 0]$ if the random number $p_{\text{BWOA}} < \sigma_{\text{sem}}$, e.g., $p_{\text{BWOA}} = 0.34$, and $\vec{X}(t+1) = \vec{X}(t) = [0 \ 1 \ 1 \ 0 \ 1]$ if the random number $p_{\text{BWOA}} \geq \sigma_{\text{sem}}$, e.g., $p_{\text{BWOA}} = 0.68$.

Spiral updating position: The position in the spiral updating position mechanism is computed as

$$\vec{X}(t+1) = \begin{cases} \mathcal{C}(\vec{X}(t)), & \text{if } p_{\text{BWOA}} < \sigma_{\text{sup}} \\ \vec{X}(t), & \text{if } p_{\text{BWOA}} \geq \sigma_{\text{sup}}, \end{cases} \quad (12)$$

where σ_{sup} is the step size, which can be calculated using the following transfer function

$$\sigma_{\text{sup}} = \frac{1}{1 + \exp(-10(\vec{A} \cdot \vec{D} - 0.5))}, \quad (13)$$

Here, \vec{A} and \vec{D} are calculated using (3) and (5), respectively.

Search for Prey: The step size is computed as

$$\sigma_{\text{sp}} = \frac{1}{1 + \exp(-10(\vec{A} \cdot \vec{D} - 0.5))}, \quad (14)$$

where \vec{A} and \vec{D} are calculated using (3) and (8), respectively. Hence, the position of the search agents is updated as follows:

$$\vec{X}(t+1) = \begin{cases} \mathcal{C}(\vec{X}(t)), & \text{if } p_{\text{BWOA}} < \sigma_{\text{sp}} \\ \vec{X}(t), & \text{if } p_{\text{BWOA}} \geq \sigma_{\text{sp}}. \end{cases} \quad (15)$$

It should be noted that different transfer functions can be used to map continuous search space to discrete actions. Mirjalili *et al.* in [21] proposed six transfer functions, which are divided them into two families: *s-shaped* and *v-shaped*, and evaluated the performance of the binary particle swarm optimization (PSO) with these transfer functions. Taking the *s-shaped* family as an example. Actually, a sigmoid function with different lope is used to create *s-shaped* transfer functions, which can be denoted as $T_\alpha(\cdot)$ with α being the lope of the *s-shaped* transfer function. For instance, $T_{(2)}(x) = 1/(1 + \exp(-2x))$, $T_{(1)}(x) = 1/(1 + \exp(-x))$, and $T_{(1/2)}(x) = 1/(1 + \exp(-x/2))$. We denote the distance between the position of the humpback whale and the prey by x . For a given distance x , the probability of changing the bit values becomes higher as the lope α increases. Accordingly, $T_{(2)}$ returns the higher probability than $T_{(1)}$ for the same value of the distance x , while $T_{(1/2)}$ returns the lowest probability. This fact can be pictorially observed from Fig. 1. The use of these transfer functions in an effective manner can further improve the performance of the BWOA algorithm.

E. Constraint-Dealing Techniques

Since the original WOA algorithm is for unconstrained optimization, we need to employ efficient constraint-handling techniques so as to solve constrained problems. Yang *et al.* in [22] divides constraint-handling techniques into two major categories: 1) *classic* methods, which are still widely used in its standalone form, and 2) *recent* methods, which are based on the hybrid of evolutionary ideas with the classic methods. Some of the well-known constraint-handling methods are penalty method, equality with tolerance, feasibility rules, separation of objectives and constraints, stochastic ranking, ϵ -constrained method, and multi-objective approach [22]. The penalty method is one of the simple and widely-used methods, which tries to convert constrained problems into unconstrained

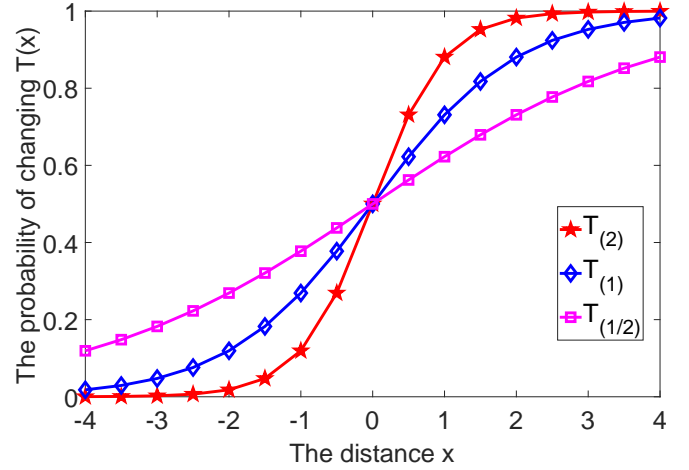


Fig. 1: Examples of s-shaped transfer functions.

problems by combining the objective and constraints together. In the following, we present the basics of the penalty method. For other constraint-handling techniques in metaheuristic optimization, the readers are invited to refer to [22, Chapter 13], the comprehensive survey paper [23], and references therein.

Consider a problem that minimizes $f_0(\mathbf{x})$ among all feasible \mathbf{x} and satisfies m inequalities and p equality constraints:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(\mathbf{x}) \leq 0, \forall i = 1, \dots, m, \\ & h_i(\mathbf{x}) = 0, \forall i = 1, \dots, p. \end{aligned} \quad (16)$$

The penalty function can be defined as $\phi(\mathbf{x}) = f_0(\mathbf{x}) + P(\mathbf{x})$, where $P(\mathbf{x})$ is the penalty term and can be defined as follows:

$$P(\mathbf{x}) = \sum_{i=1}^m \mu_i F_i(f_i(\mathbf{x})) f_i^2(\mathbf{x}) + \sum_{j=1}^p \nu_j H_j(h_j(\mathbf{x})) h_j^2(\mathbf{x}). \quad (17)$$

Here, $\mu_i \geq 0$ and $\nu_j \gg 1$ are penalty factors, which are typically identical for ease of implementation for all inequalities and equality constraints, i.e., $\mu_i = \mu \forall i$ and $\nu_j = \nu \forall j$. The index function $F_i(f_i(\mathbf{x})) = 0$ if $f_i(\mathbf{x}) \leq 0$ and $F_i(f_i(\mathbf{x})) = 1$ if $f_i(\mathbf{x}) > 0$. Similarly, the index function $H_j(h_j(\mathbf{x})) = 1$ if $h_j(\mathbf{x}) \neq 0$ and $H_j(h_j(\mathbf{x})) = 0$ if $h_j(\mathbf{x}) = 0$. The aim of the expanded objective function $\phi(\mathbf{x})$ is to decrease the fitness of infeasible solutions, but at the same time $\phi(\mathbf{x})$ is to favor the feasible solutions. As can be noted from (17), the penalty value added to the fitness of a solution is mainly controlled by the penalty factors. It is worth mentioning that the penalty function is well applicable for some problems; however, selecting the right values of penalty factors turns out to be problem-specific. According to [24], if the penalty factors are too small, an infeasible may not get enough penalty. Thus, an infeasible solution may be evolved in the evolutionary optimization process. If too large penalty factors are used, a feasible solution can be of low quality. In addition, the exploration of infeasible solutions is encouraged by the (WOA) algorithm, especially when the feasible regions are disjoint. In general, the penalty factors μ and ν are from 10^{13} to 10^{15} . In the following section, μ and ν are both set to

10^{14} for all the constraints for the sake of simplicity. An interesting direction is to integrate the WOA with other constraint-handling strategies to solve more constrained problems with potential performance improvement.

In the case of constrained problems, the computational complexity of the WOA is higher and largely depends on the numbers of equality and inequality constraints. The computation of index functions corresponding to p equality constraints and m inequality constraints requires $\mathcal{O}(Np)$ and $\mathcal{O}(Nm)$ time, respectively. Summing up these complexities with that in the original WOA, the computational complexity of the WOA to solve constrained optimization problems becomes $\mathcal{O}(N(m+p+D))$ per iteration. Since the WOA iterates at most T times, the computational complexity of the WOA to solve constrained optimization problems is $\mathcal{O}(TN(m+p+D))$.

III. WOA IN WIRELESS NETWORKS: EXAMPLES

This section aims at providing a compendious tutorial on the application of WOA to solve three fundamental optimization problems in wireless networks: max-min secrecy rate maximization, EE-SE tradeoff optimization, and MEC computation offloading. The first example is to present the application of WOA to an unconstrained continuous optimization problem, the second example is for a constrained continuous problem, and the last example is for an MINLP problem. While the first problem can be solved directly by the WOA, the constraint-handling technique needs to be used in the second example and the third problem is solved by the BWOA with the help of both the decomposition and constraint-handling techniques.

A. Power Allocation for Secrecy Rate Maximization

1) *System Model and Problem Formulation:* Consider an interference-limited wireless network (IWN) with M users, each user can be regarded as a communication link, which is composed of a single-antenna transmitter and a single-antenna receiver. Denote by p_i the transmit power of the user i and by g_{ij} the channel gain from the user j to the user i . The data rate of the user i is given as $R_i(\mathbf{p}) = \log_2(1 + p_i g_{ii}/(n_0 + I_i))$, where $I_i = \sum_{j=1}^M p_j g_{ij}$ and \mathbf{p} is the transmit power vector and n_0 is the noise power. Suppose that there is a single-antenna eavesdropper (EV), which is considered as part of the legitimate network. The wiretapped rate of the user i at the EV is given as $\Gamma_i(\mathbf{p}) = \log_2(1 + p_i g_{ei}/(n_0 + I_i^e))$, where $I_i^e = \sum_{j=1}^M p_j g_{ej}$ and g_{ej} denotes the channel gain from the user j to the EV. The secrecy rate of the user i is defined as $\Phi_i(\mathbf{p}) = \max\{R_i(\mathbf{p}) - \Gamma_i(\mathbf{p}), 0\}$. The max-min secrecy rate (MMSR) problem can be formulated as follows [25]:

$$\begin{aligned} \max_{\mathbf{p}} \quad & \Phi(\mathbf{p}) = \min_{i=1, \dots, M} [R_i(\mathbf{p}) - \Gamma_i(\mathbf{p})] \\ \text{s.t.} \quad & 0 \leq p_i \leq p_i^{\max}, \forall i = 1, \dots, M, \end{aligned} \quad (18)$$

where p_i^{\max} denotes the peak transmit power of the user i .

2) *Existing Algorithm:* Instead of using the d.c. (difference of two concave functions) representation of the secrecy rate $\Phi_i(\mathbf{p})$, Sheng *et al.* in [25] proposed a path-following procedure to solve the MMSR problem. At every iteration t , the data rate $R_i(\mathbf{p})$ and the wiretapped rate $\Gamma_i(\mathbf{p})$ are, respectively,

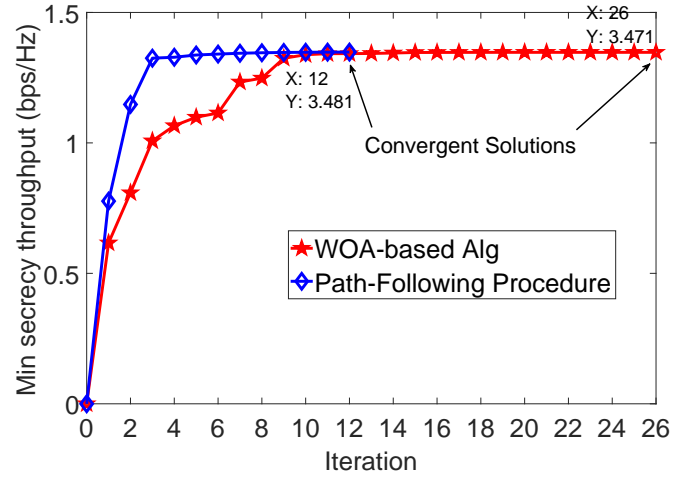


Fig. 2: Convergence evolution of the and the path-following procedure in [25] (with $M = 4$ users).

approximated by a lower bound $R_i^{(t)}(\mathbf{p})$ and an upper bound $\Gamma_i^{(t)}(\mathbf{p})$. Initialized from a feasible solution $\mathbf{p}^{(0)}$, the following convex optimization problem is solved at the t -th iteration

$$\begin{aligned} \max_{\mathbf{p}} \quad & \Phi^{(t)}(\mathbf{p}) = \min_{i=1, \dots, M} \Phi_i^{(t)}(\mathbf{p}) \\ \text{s.t.} \quad & 0 \leq p_i \leq p_i^{\max}, \forall i = 1, \dots, M. \end{aligned} \quad (19)$$

The path-following procedure is performed iteratively until the stopping criterion ϵ is met, i.e., $|\Phi^{(t)} - \Phi^{(t-1)}|/\Phi^{(t)} \leq \epsilon$.

3) *WOA-based Algorithm:* The MMSR problem is an unconstrained continuous optimization problem, thus the original WOA algorithm can be applied directly to get the solution.

4) *Simulation Results and Discussion:* We adopt the same parameters from simulation in [25]. The initial feasible point $\mathbf{p}^{(0)}$ for the path-following procedure is initialized randomly, i.e., $p_i^{(0)} = \rho(p_i^{\max} - p_i^{\min}) + p_i^{\min}$, where $p_i^{\min} = 0$ denotes the minimum transmit power of the user i and ρ is a random number in $[0, 1]$. Fig. 2 plots the convergence evolution of the path-following procedure in [25] and the proposed WOA-based algorithm versus the iteration index. As shown in Fig. 2, two algorithms can converge very quickly, the path-following procedure requires 12 iterations and that of the WOA-based algorithm is 26 iterations. However, it must be mentioned that at each iteration the path-following procedure invokes the solution of a convex problem. Since the approximate convex problem (19) evolves M optimization variables (i.e., $p_i, i \in \{1, \dots, M\}$) and M linear constraints (i.e., $0 \leq p_i \leq p_i^{\max}, i \in \{1, \dots, M\}$), the path-following procedure has a computational complexity of $\mathcal{O}(T(M^2 M^{2.5} + M^{3.5}))$ with $T = 12$ for the network scenario considered in this paper. Different from [25], at each iteration in the WOA-based algorithm, users (i.e., humpback whales in WOA) follow the best search agent to update their transmit powers, together with the transmit power of the best search agent obtained so far. As presented in Section II-C, the computational complexity of the WOA for unconstrained problem is $\mathcal{O}(NMT)$ with $T = 26$ and $N = 30$ (N is the number of search agents). Therefore, the WOA-based algorithm has the advantage of low computational complexity, which is of considerable importance for practical

applications with massive connectivity. Moreover, almost the same result can be achieved by both algorithms: 1.3481 bps/Hz for the path-following and 1.3471 bps/Hz for the WOA-based algorithm. At this point, it is concluded that the WOA-based algorithm has a lower computational complexity, but provides a very high-performance optimality, when it is compared with the existing algorithm proposed in [25].

B. Power Allocation for Energy-Spectral Efficiency Tradeoff

1) *System Model and Problem Formulation:* Consider an IWN and use the same set of notations as in Subsection III-A. The total power consumption P_{tot} includes two main parts: *transmit* power consumption p_i and *circuit* power consumption p_i^C . Thus, $P_{\text{tot}}(\mathbf{p}) = \sum_{i=1}^M (\xi_i p_i + p_i^C)$, where ξ_i is a constant power-amplifier inefficiency factor of the user i . The problem of energy efficiency (EE) and spectral efficiency (SE) tradeoff can be formulated by maximizing the global EE (GEE) subject to constraints on minimum rate requirements and transmit power budgets. Accordingly, the following optimization problem is considered for the EE-SE tradeoff [26]:

$$\begin{aligned} \max_{\mathbf{p}} \quad & q = \sum_{i=1}^M R_i(\mathbf{p}) / P_{\text{tot}}(\mathbf{p}) \\ \text{s.t.} \quad & \text{C1: } R_i(\mathbf{p}) \geq R_i^{\text{req}}, \forall i = 1, \dots, M, \\ & \text{C2: } 0 \leq p_i \leq p_i^{\text{max}}, \forall i = 1, \dots, M, \end{aligned} \quad (20)$$

where R_i^{req} denotes the minimum required rate of the user i and the unit of q is bits/Joule/Hz. The above optimization problem is non-convex and NP-Hard, it is therefore difficult to get the solution in polynomial time [26], [27].

2) *Existing Algorithm:* The problem in (20) is solved by successively applying two methods: *bisection* and *successive convex approximation* (SCA). For a given value of q , the problem in (20), can be cast as follows:

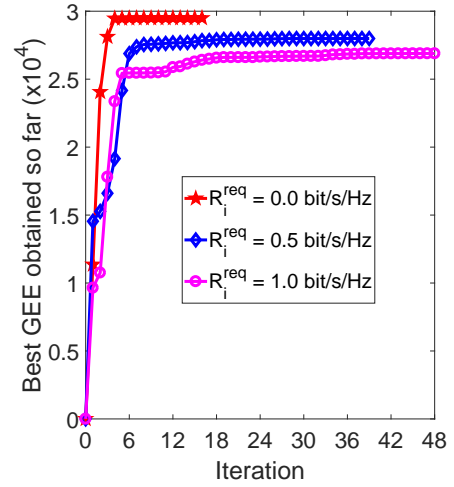
$$\max_{\mathbf{p}} \sum_{i=1}^M R_i(\mathbf{p}) - q P_{\text{tot}}(\mathbf{p}) \quad \text{s.t. C1 and C2.} \quad (21)$$

Using the d.c. representation of the new objective function and further rearranging the constraint C1, the problem in (21) can be solved by successively solving a standard convex optimization problem at each iteration. Similar to the path-following procedure above, the criterion $|R_i^{(t)} - q^{(t)} P_{\text{tot}}^{(t)}| \leq \epsilon$ is used to stop the algorithm.

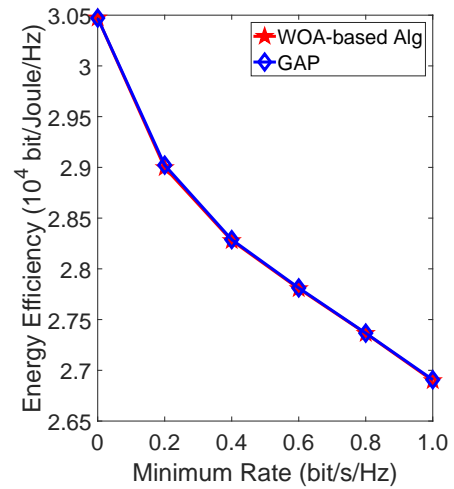
3) *WOA-based Algorithm:* The penalty method is used to deal with the minimum rate requirement constraints C1 in the problem (20). In this case, users can be regarded as a search agent (a humpback whale) and the transmit power \mathbf{p} is considered as the position of the search agents \mathbf{X} . At the iteration t , the transmit power $\mathbf{p}(t)$ (corresponding to $\mathbf{X}(t)$) can be updated by following either the shrinking encircling mechanism, spiral updating position, or search for prey. The fitness function used to evaluate the best search agent can be defined as follows:

$$\text{Fitness}(\mathbf{p}) = -\frac{\sum_{i=1}^M R_i(\mathbf{p})}{P_{\text{tot}}(\mathbf{p})} + \mu \sum_{i=1}^M F_i(f_i(\mathbf{p})) f_i^2(\mathbf{p}), \quad (22)$$

where $f_i(\mathbf{p}) = R_i^{\text{req}} - R_i(\mathbf{p})$ and $\mu = 10^{14}$. Note that the placement of a minus sign in front of the objective function is



(a) Convergence evolution.



(b) Energy efficiency vs. minimum required rates.

Fig. 3: Performance evaluation of the WOA-based algorithm.

a trick to convert a maximization problem to a minimization problem. The inequality function $f_i(\mathbf{p})$ is redefined as $f_i(\mathbf{p}) = R_i(\mathbf{p}) - R_i^{\text{req}}$ and the index function $F_i(f_i(\mathbf{x})) = 0$ if $f_i(\mathbf{x}) \geq 0$ and $F_i(f_i(\mathbf{x})) = 1$ if $f_i(\mathbf{x}) < 0$.

4) *Simulation Results and Discussion:* We use the same simulation settings as in [26]. Once the stopping criterion $|q(t) - q(t-1)| \leq 10^{-6}$ is satisfied for three consecutive times, the proposed WOA-based algorithm is terminated. From Fig. 3a, the convergence of the proposed algorithm can be obtained in a few tens of iterations, nearly three times higher than the number of main loops of the GAP in [26]. However, we must stress that the proposed WOA-based algorithm requires a single-loop operation, but the GAP is a two-loop scheme, where the outer loop is to update the energy efficiency q and the inner loop is to optimize the transmit power for a given q . Moreover, each inner loop needs to find the solution to a sequence of approximate convex problems, which is similar to the path-following procedure in [25]. More specifically, as analyzed in Section II-E, the computational complexity of the WOA is $\mathcal{O}(TN(m+M))$ with $N = 30$, $m = M$, ($p = 0$

since there is no inequality constraint), and $T = 48$ (in the case $R_i^{\text{req}} = 1.0$ bit/s/Hz). In contrast, the GAP has a computational complexity of $\mathcal{O}(T_1 T_2 (M^2 (2M)^{2.5} + (2M)^{3.5}))$, where $T_1 = 14$ and $T_2 = 12$ are the numbers of iterations needed for the outer and inner loops, respectively [26, Figs. 2 and 3]. Comparing with the GAP, the WOA has a very low computational complexity, which is crucial for practical applications. By varying the minimum rate requirement R_i^{req} , we plot the EE-SE tradeoff curve in Fig. 3b and observe that the energy efficiency decreases as the higher minimum rates are required. Moreover, the proposed WOA-based algorithm achieves almost the same performance as the existing scheme. To conclude, the WOA-based algorithm achieves competitive performance with very low computational complexity.

C. Mobile Edge Computation Offloading

1) *System Model and Problem Formulation:* Consider an MEC scenario with one MEC server that is collocated with the eNB, and M users. Each user has a computation task $I_i = \{D_i, C_i\}$, where D_i is the input size (in bits) and C_i is the number of CPU cycles needed to accomplish the task. Denote by a_i the offloading decision of the user i . Here, $a_i = 1$ if the task I_i is offloaded and $a_i = 0$ otherwise.

The completion time and energy consumption of the user i can be expressed as $T_i = a_i T_i^r + (1 - a_i) T_i^l$ and $E_i = a_i E_i^r + (1 - a_i) E_i^l$, where T_i^l/T_i^r and E_i^l/E_i^r are, respectively, the task completion time and energy consumption of the user i when the task I_i is executed locally/remotely. According to [28], T_i^l , E_i^l , T_i^r , and E_i^r can be given as $T_i^l = C_i/f_i^l$, $E_i^l = \alpha (f_i^l)^{\gamma-1} C_i$, $T_i^r = C_i/f_i + D_i/R_i$, and $E_i^r = D_i p_i / \varsigma R_i$, where f_i^l is the local computing capability of the user i , $\alpha = 10^{-11}$, $\gamma = 2$, f_i is the computing resource allocated to the user i by the MEC server, ς is the power amplifier efficiency, and R_i is the data rate of the user i . With the assumptions that the allocation of subcarriers is orthogonal and predefined, the data rate R_i is given as $R_i(p_i) = W \log_2(1 + p_i h_i / n_0)$ with h_i being the channel gain from the user i to the eNB.

In order to maximize the improvement in terms of completion time and energy consumption by computation offloading, the overall utility can be defined as follows:

$$v_i(a_i, p_i, f_i) = a_i \left(\beta_i^t \frac{T_i^l - T_i^r}{T_i^l} + \beta_i^e \frac{E_i^l - E_i^r}{E_i^l} \right), \quad (23)$$

where β_i^t and β_i^e the user preferences on completion time and energy consumption, respectively. The problem of optimizing the computation offloading decision and resource allocation to maximize the overall utility can be formulated as follows [28]:

$$\begin{aligned} \max_{\mathbf{a}, \mathbf{p}, \mathbf{f}} \quad & \sum_{i=1}^M v_i(a_i, p_i, f_i) \\ \text{s.t.} \quad & \text{C1 : } a_i = \{0, 1\}, \forall i = 1, \dots, M, \\ & \text{C2 : } 0 \leq p_i \leq p_i^{\max}, \forall i = 1, \dots, M, \\ & \text{C3 : } f_i > 0, \forall i \in \mathcal{S}, \text{ C4 : } \sum_{i \in \mathcal{S}} f_i \leq f_0, \\ & \text{C5 : } \sum_{i=1}^M a_i \leq N, \end{aligned} \quad (24)$$

where $\mathcal{S} = \{i = 1, \dots, M | a_i = 1\}$ is the set of offloading users, f_0 is the maximum computing resource of the MEC

server, and N is the number of subcarriers, which indicates that N users are allowed to offload their computation tasks. The constraints C3 and C4 in the problem (24) imply that the MEC server only allocates computing resources to the offloading users and the total allocated resources is required to be less than the maximum computing capability f_0 .

2) *Existing Algorithm:* Since the problem (24) is NP-hard, the authors in [28] proposed decomposing the original problem into two subproblems. The first subproblem (JCCR) is to optimize computation and communication resources for a given offloading decision $\hat{\mathbf{a}}$ as follows:

$$\max_{\mathbf{p}, \mathbf{f}} \sum_{i \in \mathcal{S}} v_i(\hat{a}_i, p_i, f_i) \quad \text{s.t. C2, C3, C4.} \quad (25)$$

The second subproblem (COD) is to optimize the offloading decisions for a given $(\hat{\mathbf{p}}, \hat{\mathbf{f}})$ and can be written as

$$\max_{\mathbf{a}} \sum_{i \in \mathcal{S}} v_i(a_i, \hat{p}_i, \hat{f}_i) \quad \text{s.t. C1 and C5.} \quad (26)$$

For a given $\hat{\mathbf{a}}$, the transmit power \mathbf{p} and the computing resources \mathbf{f} are decoupled from each other; therefore, the JCCR subproblem can be further decomposed into two separate problems of \mathbf{p} and \mathbf{f} . Then, the solutions of \mathbf{p} and \mathbf{f} can be solved using the convex optimization and bisection method, respectively. It is shown that the objective of the COD subproblem is a submodular optimization problem, thus the offloading decision \mathbf{a} is solved through a heuristic algorithm that is based on the submodular optimization.

3) *WOA-based Algorithm:* To evaluate the effectiveness of the BWOA algorithm and fairly compare the performance of the BWOA-based algorithm with the existing one, we propose using the decomposition technique to get two subproblems of \mathbf{a} and (\mathbf{p}, \mathbf{f}) . While the JCCR subproblem is solved using the same approach as in [28], the BWOA algorithm is utilized to solve the COD subproblem so as to obtain the solution of offloading decision. It is worth noting that this subsection is devoted to solving the binary optimization problem (26), which is opposed to Subsections III-A and III-B, where the WOA algorithm is used to solve continuous optimization problems. The fact is that one can utilize the combination of WOA and BWOA to solve the original problem (24). However, such algorithms are not our focus, thus they are omitted in this paper. To learn more, the interested readers can refer to [29] and the references therein for the optimization of channel allocation and power control in D2D communications using particle swarm optimization.

To deal with the inequality C5, a straightforward approach is checking to see whether the constraint C5 is satisfied. If the constraint is satisfied, the solution is feasible and the evaluation of the fitness function can be carried out. Once the constraint is violated, the solution is discarded and one should proceed with a new solution. Obviously, this approach is slow and inefficient. To achieve a better performance, using the penalty method, the fitness function can be defined as follows:

$$\text{Fitness}(\mathbf{a}) = - \sum_{i \in \mathcal{S}} v_i(a_i, \hat{p}_i, \hat{f}_i) + \mu F(f(\mathbf{a})) f^2(\mathbf{a}), \quad (27)$$

where $f(\mathbf{a}) = \sum_{i=1}^M a_i - N$.

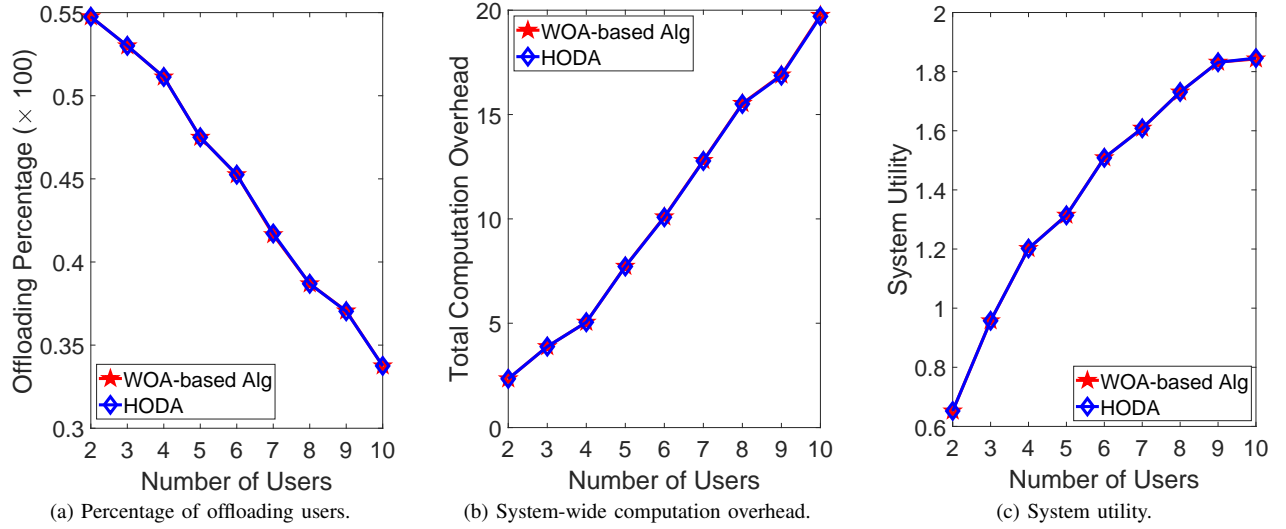


Fig. 4: Performance comparison when the number of users varies from 2 to 10. The WOA-based algorithm performs very close to the solution obtained by the centralized heuristic algorithm HODA in [28].

4) *Simulation Results and Discussion:* We conduct experiments to compare the performance in terms of system utility, system-wide (i.e., total) computation overhead, and percentage of offloading, between the proposed WOA-based algorithm and the heuristic-offloading decision algorithm (HODA) in [28]. These metrics are all evaluated at the convergent solution: 1) system utility is defined as the objective of the optimization problem (24), 2) system-wide computation overhead is defined as $Z_{\text{tot}} = \sum_{i=1}^M (1 - a_i)(\beta_i^t T_i^l + \beta_i^e E_i^l) + a_i(\beta_i^t(T_i^t + T_i^e) + \beta_i^e E_i^r)$, and 3) percentage of offloading is defined as the ratio between the number of offloading users and the total users, i.e., $P_{\text{off}} = \mathbf{a}^T \mathbf{1}/N$. The same set of simulation parameters for HODA [28] is utilized in our experiments. Further, each plot is achieved by averaging the results over 200 realizations and users are randomly positioned in each realization.

Fig. 4 shows that as the number of users increases, both the algorithms have a wider range of users with various channel conditions, local computing capabilities, and computation tasks, thus they can provide continuous improvements on system utility. Since each user needs to compete with more users for computation offloading and the number of subcarriers is limited, some users are not allowed to offload despite the fact that they can still benefit from remote execution. Therefore, the percentage of offloading and the system-wide computation overhead both increase as the number of users increases. The WOA-based algorithm has a low complexity level of $\mathcal{O}(N(1 + M))$ (there is no equality constraint and only one inequality constraint C5), compared with the $\mathcal{O}(M^3)$ and $\mathcal{O}(2^M)$ complexities of the HODA and optimal solution, respectively. As can be observed from Fig. 4, the WOA-based algorithm performs very close to the solution achieved by the heuristic centralized HODA [28]. With the number of users $M = 8$, the WOA-based algorithm obtains the system utility of 1.731, whereas the system utility of 1.7311 is achieved by the HODA. Moreover, [28] verified via simulations that the HODA is able to remain its performance within 5% of the

optimal solution by the exhaustive search. Hence, the WOA performs competitively with a low computational complexity.

Through the above three examples, we utterly demonstrate that the WOA is a promising tool to solve various resource allocation problems in wireless and communication networks. All of the experiments show that the WOA-based algorithm can obtain the solution with almost the same performance compared to the existing algorithms with convergence guarantee. It is reasonable since both exploration and exploitation are evolved in the optimization process, thus the WOA can be regarded as a global optimizer.

IV. WOA IN WIRELESS NETWORKS: POTENTIAL APPLICATIONS

Although the applications of WOA to wireless networks remain unexplored, three examples in the previous section really justify the effectiveness of the WOA-based algorithms. In the following subsections, a number of potential applications of WOA in wireless networks are discussed. From these discussions, we expect that the WOA will be an effective tool for optimizing upcoming wireless systems, where many new technologies, services, network architectures will be emerged, e.g., federated learning, intelligent reflecting surface, cell-free massive multi-input multi-output (MIMO), holographic and terahertz communications. For a comprehensive survey of 5G networks and speculative study on 6G, we invite the readers to read [30], [31] and references therein.

A. MEC Offloading in Multi-Carrier NOMA

With the emergence of new compute-intensive and energy-hungry mobile applications, small users (e.g., IoT devices) tend to offload their computations to the centralized cloud and/or edge servers, e.g., fog node and MEC server for reduced completion latency and energy consumption [7], [32]. Conventionally, computation offloading is performed by applying orthogonal multiple access (OMA) schemes to the transmission

between the server and users. For instance, in time-division multiple access TDMA-MEC (OFDMA-MEC), users are allocated a fraction of time (a subcarrier) in an orthogonal manner to offload their tasks to the MEC server. However, it may not be efficient in MEC systems when a very large number of users are trying to perform computation offloading over a limited number of orthogonal subcarriers. Integrating NOMA to MEC can not only avoid severe delay but also reduce energy consumption [33]. Compared with the conventional problem in OMA-MEC, joint optimization of computation offloading and resource allocation (e.g., communication and computation resources) in NOMA-MEC is further exacerbated by unique characteristics of NOMA, such as user clustering and decoding order. Since such problems are usually difficult to solve directly, most of the literature in MEC considered decomposing the original problem into smaller subproblems, which are alternately and iteratively solved until the final solution is achieved or the stopping criterion is satisfied.

The applicability of WOA to solve the joint problem of computation offloading and resource allocation in NOMA-MEC is quite straightforward. Using optimization-based approaches, one needs to analyze the properties of considered problems and network scenarios, and specific methods are required for a particular problem. In contrast, the WOA can be applied to a wide range of the computation offloading problems and the result can be served as a benchmark for measuring the performance comparison with other methods. A potential approach is to directly apply the WOA to solve the joint problem, where the original WOA deals with continuous variables, such as power allocation, computing resource assignment, and local computing capability control, and the binary WOA solves binary variables, for example, offloading decision, subcarrier assignment, and NOMA cluster selection. Another promising direction is to decompose the problem into smaller subproblems, by which several ones are solved using conventional methods such as game theory and convex optimization while the remaining ones are solved using WOA. Our third example provides a clear illustration of this direction. In particular, we proposed decomposing the joint optimization of power control, computing resource allocation, and offloading decision into three subproblems, and then utilized the BWOA algorithm for offloading decision and used convex optimization for transmit power and computing resource allocation.

B. Interference Management

In wireless networks, *interference management* (IM) has become an underlying optimization problem that helps to achieve many objectives, such as throughput maximization, energy/computation efficiency, QoS assurance, and network planning. The IM problem becomes further serious and complicated in ultra-dense networks (UDNs), where many interfering nodes (e.g., users, access points, and eNBs) may coexist in a small area [34]. Without proper IM schemes, the network performance can be reduced significantly while a significant amount of resources can be wasted. In addition, IM is closely interwoven with resource allocation in a complex way.

In general, IM is mathematically formulated as non-convex and NP-hard optimization problems and a common approach

used in the literature is to find the near-optimal solution, which is sometimes not guaranteed to finally converge to the optimal values. Clearly, such problems can be solved using the WOA algorithm with the help of constraint-dealing techniques. Given an optimization problem, the constraints can be avoided safely by integrating them into the objective function. As a global optimizer, the WOA is expected to make the performance for a particular resource allocation problem highly competitive among the proposed algorithms.

C. User Association and D2D Mode Selection

Besides IM, *user association* (UA) is of importance to enhance the network load balancing, spectrum, and energy efficiency in UDNs. UA is meant to decide which eNB a user should connect [35]. The well-known UA schemes are based on basic metrics, such as signal-to-interference-plus-noise ratio and received signal strength indicator. Different from the previous generations, 5G will not only provide communication but also computation, caching, and control; therefore, there are many more metrics that can be used to yield an efficient UA solution, for example, outage and coverage probability, spectrum/energy efficiency, QoS, fairness, computation-aware capability, and caching-aware contents [32], [36].

D2D communication is a promising approach to facilitate high data rate services in a short range to increase the performance of 5G networks [3], [29]. By means of *mode selection*, two users in a D2D pair can select between two modes: direct communication reusing subcarriers of CUs, and indirect communication as normal CUs with the help of eNB [37]. In LTE-unlicensed, mode selection can be considered as the decision on which spectrum (i.e., licensed or unlicensed spectrum) D2D pairs should use for their direct communication [29]. D2D model selection, together with consideration of subcarrier assignment, is usually formulated as MIP problems.

Among the state-of-the-art approaches, matching theory, coalition games, and graph theory are considered as the most promising techniques, which try to find the binary high-quality solution within a reasonable time and with a reduced computational complexity in comparison to the exhaustive search. WOA provides an alternative way to solve such problems. Although there is no convergence proof, numerical results can be used to confirm the effectiveness of the binary WOA algorithm. Our demonstration of the third example showed that the WOA algorithm can achieve almost the same performance compared to the heuristic centralized scheme.

D. UAV Trajectory Optimization

The optimization of the trajectory of UAVs has received significant efforts from research communities as a method to improve the performance of UAV systems [7], [38]. However, the trajectory of UAVs is greatly affected by a different number of factors, e.g., flight time, energy constraints, collision avoidance, computation efficiency, and energy harvesting causality. One common way to optimize the trajectory of UAVs is to discretizing the transmission period T into N different time slots and assume that the UAV's location is approximately unchanged within each time slot [39]. In doing so, the joint

problem of trajectory and resource allocation can be optimized by applying the successive optimization techniques, that is, the resource allocation problem is solved for a given trajectory, then the trajectory is updated for a given resource allocation vector, and these two phases are alternately and iteratively performed until the stopping criterion is satisfied.

Even though, the optimization problems to be solved within each time slot are usually non-convex and solving them mostly utilizes the convex approximation and heuristic methods. Due to great advantages, one can consider applying WOA to solve the trajectory optimization problem and it is expected that the WOA algorithm is more competent for optimizing the trajectory of UAVs than the existing mechanisms in terms of solution quality and convergence speed.

V. CONCLUSION

We have presented the fundamentals of the whale optimization algorithm (WOA) and its applications to resource allocation in wireless and communication networks. To demonstrate the applicability of WOA in wireless and communication networks, we have investigated three problems and some preliminary results have been presented. Also, potential use cases in NOMA-MEC systems, interference management in heterogeneous networks, user association in UDNs, mode selection in D2D communication, and trajectory optimization in UAV systems, have been highlighted.

Since the application of WOA to wireless and communication networks remain unexplored and the WOA-based algorithm can provide the competitive performance compared with the state-of-the-art algorithms, WOA can be used as a benchmark for performance comparison when someone proposes his own methods for any optimization problem. We hope that this paper will be serving as a starting reference for the adoption of WOA in optimizing resource allocation in wireless networks as well as other engineering branches.

REFERENCES

- [1] Q.-V. Pham, L. B. Le, S.-H. Chung, and W.-J. Hwang, "Mobile edge computing with wireless backhaul: Joint task offloading and resource allocation," *IEEE Access*, vol. 7, pp. 16 444–16 459, Jan. 2019.
- [2] Y. Pochet and L. A. Wolsey, *Production planning by mixed integer programming*. B-1348 Louvain-la-Neuve, Belgium: Springer Science & Business Media, 2006.
- [3] F. Boabang, H.-H. Nguyen, Q.-V. Pham, and W.-J. Hwang, "Network-assisted distributed fairness-aware interference coordination for device-to-device communication underlaid cellular networks," *Mobile Information Systems*, vol. 2017, pp. 1–11, Jan. 2017.
- [4] B. Zhang, X. Mao, J. Yu, and Z. Han, "Resource allocation for 5G heterogeneous cloud radio access networks with D2D communication: A matching and coalition approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5883–5894, Jul. 2018.
- [5] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Wireless communication using unmanned aerial vehicles (UAVs): Optimal transport theory for hover time optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8052–8066, Dec. 2017.
- [6] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, Apr. 2017.
- [7] 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," *arXiv preprint arXiv:1906.08452*, 2019. [Online]. Available: <http://arxiv.org/abs/1906.08452>
- [8] P. R. Sahu, P. K. Hota, and S. Panda, "Power system stability enhancement by fractional order multi input SSSC based controller employing whale optimization algorithm," *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3, pp. 326–336, Dec. 2018.
- [9] Y. Moodi, S. R. Mousavi, A. Ghavidel, M. R. Sohrabi, and M. Rashki, "Using response surface methodology and providing a modified model using whale algorithm for estimating the compressive strength of columns confined with FRP sheets," *Construction and Building Materials*, vol. 183, pp. 163–170, Sep. 2018.
- [10] M. A. Algalaway, A. Y. Abdelaziz, S. F. Mekhamer, and S. H. A. Aleem, "Considerations on optimal design of hybrid power generation systems using whale and sine cosine optimization algorithms," *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3, pp. 312–325, Dec. 2018.
- [11] S. Khalilpourazari, S. H. R. Pasandideh, and A. Ghodrattama, "Robust possibilistic programming for multi-item EOQ model with defective supply batches: Whale optimization and water cycle algorithms," *Neural Computing and Applications*, pp. 1–28, Apr. 2018.
- [12] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016.
- [13] V. Kumar and D. Kumar, "Binary whale optimization algorithm and its application to unit commitment problem," *Neural Computing and Applications*, pp. 1–29, Oct. 2018.
- [14] H. F. Eid, "Binary whale optimisation: an effective swarm algorithm for feature selection," *International Journal of Metaheuristics*, vol. 7, no. 1, pp. 67–79, May 2018.
- [15] M. Abdel-Basset, D. El-Shahat, and A. K. Sangaiah, "A modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 3, pp. 495–514, Mar. 2019.
- [16] X. Wu, S. Zhang, W. Xiao, and Y. Yin, "The exploration/exploitation tradeoff in whale optimization algorithm," *IEEE Access*, vol. 7, pp. 125 919–125 928, Sep. 2019.
- [17] G. Kaur and S. Arora, "Chaotic whale optimization algorithm," *Journal of Computational Design and Engineering*, vol. 5, no. 3, pp. 275–284, Jul. 2018.
- [18] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Computing*, vol. 22, no. 1, pp. 1–15, Jan. 2018.
- [19] Q.-V. Pham and W.-J. Hwang, "Network utility maximization-based congestion control over wireless networks: A survey and potential directives," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1173–1200, Second Quarter 2017.
- [20] E. Rashedi, H. Nezamabadi-pour, and S. Saryzadi, "BGSA: binary gravitational search algorithm," *Natural Computing*, vol. 9, no. 3, pp. 727–745, Sep. 2010.
- [21] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1–14, Oct. 2013.
- [22] X.-S. Yang, *Nature-Inspired Optimization Algorithms*. 32 Jamestown Road, London NW1 7BY: Elsevier, 2014.
- [23] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, Dec. 2011.
- [24] T. P. Runarsson and Xin Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, Sep. 2000.
- [25] Z. Sheng, H. D. Tuan, A. A. Nasir, T. Q. Duong, and H. V. Poor, "Power allocation for energy efficiency and secrecy of wireless interference networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3737–3751, Jun. 2018.
- [26] Y. Li, M. Sheng, C. Yang, and X. Wang, "Energy efficiency and spectral efficiency tradeoff in interference-limited wireless networks," *IEEE Communications Letters*, vol. 17, no. 10, pp. 1924–1927, Oct. 2013.
- [27] Q.-V. Pham and W.-J. Hwang, "Fairness-aware spectral and energy efficiency in spectrum-sharing wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 207–10 219, Nov. 2017.
- [28] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.
- [29] G. G. Girmay, Q. Pham, and W. Hwang, "Joint channel and power allocation for device-to-device communication on licensed and unlicensed band," *IEEE Access*, vol. 7, pp. 22 196–22 205, Feb. 2019.
- [30] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1617–1655, Third Quarter 2016.

- [31] P. Yang, Y. Xiao, M. Xiao, and S. Li, "6G wireless communications: Vision and potential techniques," *IEEE Network*, vol. 33, no. 4, pp. 70–75, Jul. 2019.
- [32] Q.-V. Pham, T. L. Anh, 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. 75 868–75 885, Nov. 2018.
- [33] Q.-V. Pham, T. H. Nguyen, Z. Han, and W.-J. Hwang, "Coalitional games for computation offloading in NOMA-enabled multi-access edge computing," *IEEE Transactions on Vehicular Technology*, 2019, in press.
- [34] Y. Teng, M. Liu, F. R. Yu, V. C. M. Leung, M. Song, and Y. Zhang, "Resource allocation for ultra-dense networks: A survey, some research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2134–2168, Third Quarter 2019.
- [35] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.
- [36] D. Liu, L. Wang, Y. Chen, M. ElKashlan, K. Wong, R. Schober, and L. Hanzo, "User association in 5G networks: A survey and an outlook," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1018–1044, Second quarter 2016.
- [37] G. Yu, L. Xu, D. Feng, R. Yin, G. Y. Li, and Y. Jiang, "Joint mode selection and resource allocation for device-to-device communications," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 3814–3824, Nov. 2014.
- [38] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.
- [39] F. Cheng, S. Zhang, Z. Li, Y. Chen, N. Zhao, F. R. Yu, and V. C. M. Leung, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.