

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

ACO-based Scheme in Edge Learning NOMA Networks for Task-Oriented Communications

CARLA E. GARCIA¹, MARIO R. CAMANA¹, AND INSOO KOO²

¹Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg 1209, Luxembourg

²Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 680-749, South Korea

Corresponding author: Insoo Koo (iskoo@ulsan.ac.kr).

This work was supported in part by the ICT R&D innovation voucher conducted in 2022 with the support of the Institute of Information and Communications Technology Planning and Evaluation(IITP), funded by the government's Ministry of Science and ICT.(No. 2022-0-00805, Development of wired/wireless communication technology for installation safety communication network for workers in shaded areas of building ships). In addition, this result was supported by the "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-003).

ABSTRACT Conventional communications systems centered on data prioritize maximizing network throughput using Shannon's theory, which is primarily concerned with securely transmitting the data despite limited radio resources. However, in the realm of edge learning, these methods frequently fall short because they depend on traditional source coding and channel coding principles, ultimately failing to improve learning performance. Consequently, it is crucial to transition from a data-centric viewpoint to a task-oriented communications approach in wireless system design. Therefore, in this paper, we propose efficient communications under a task-oriented principle by optimizing power allocation and edge learning-error prediction in an edge-aided non-orthogonal multiple access (NOMA) network. Furthermore, we propose a novel approach based on the ant colony optimization (ACO) algorithm to jointly minimize the learning error and optimize the power allocation variables. Moreover, we investigate four additional benchmark schemes (particle swarm optimization, quantum particle swarm optimization, cuckoo search, and butterfly optimization algorithms). Satisfactorily, simulation results validate the superiority of the ACO algorithm over the baseline schemes, achieving the best performance with less computation time. In addition, the integration of NOMA in the proposed task-oriented edge learning system obtains higher sum rate values than those achieved by conventional schemes.

INDEX TERMS task-oriented communication, edge learning, non-orthogonal multiple access (NOMA), learning error, ant colony optimization (ACO).

I. INTRODUCTION

DEPLOYMENTS and applications in the Internet of Things (IoT) involve a vast network of interconnected users that generate substantial amounts of data. However, transmitting large quantities of data from diverse IoT devices to a distant cloud server creates significant communications challenges and increases latency in transmissions [1]. Consequently, the concept of edge computing has emerged as an alternative to traditional cloud computing to tackle these issues. Edge computing harnesses the storage, communication, and computational capabilities available at edge servers to efficiently collect and manage this immense volume of data. Additionally, edge servers facilitate quick access to the extensive data distributed across end-user devices, enabling rapid model learning and the delivery of intelligent services

and applications to IoT users [2]. Aligned with cutting-edge smart IoT sensors in 5G networks and the anticipated 6G networks, edge computing is evolving into edge intelligence, ushering in a new era of more sophisticated and intelligent IoT applications and services [3].

In the realm of edge learning, the primary goal is to swiftly acquire intelligence from the abundant, yet widely dispersed data generated by subscribed IoT users. This hinges critically on the processing of data at edge servers, and on establishing efficient communication between these servers and IoT users. However, as opposed to the ever-increasing processing capabilities of edge servers, communications encounter obstacles in the form of wireless channel issues, making it the bottleneck in achieving ultra-fast edge learning [4]. Additionally, the diverse nature of ubiquitous IoT users

and the complexity of transmission environments introduce interference that significantly undermines the reliability and communication speed of the IoT network, particularly when transmitting vast amounts of data to an edge server [5]. To tackle these challenges, traditional data-centric communication systems aim to maximize network throughput based on Shannon's theory, which focuses on securely transmitting data despite constrained radio resources [6]. Nonetheless, such approaches often prove ineffective in the context of edge learning, because they rely on classic source coding and channel coding theories, failing to enhance learning performance. Therefore, a shift in wireless system design is imperative, moving from a data-centric perspective to one that prioritizes task-oriented communication [1], [7].

In this regard, task-oriented communication aims to extend its scope beyond transmitting data at the micro level, where performance is assessed based on factors like bit or packet error rates, and instead emphasizes communication experiences that consider macro-level performance metrics, such as learning rate and inference accuracy [8]. Task-oriented communication in particular can lessen communication load by supplying only task-relevant information, such as feature extraction for edge inference, as opposed to sending all the data and ignoring information structures [9]. For instance, in [10], the authors introduced a learning-driven communication strategy designed to optimize local feature extraction, and distributed feature encoding for task-oriented purposes. This approach aims to eliminate redundant data and transmit only crucial information needed for downstream inference tasks, instead of reconstructing data samples at the edge server. In [1], the authors achieved effective communication within the task-oriented framework by optimizing power allocation and edge learning error prediction. Moreover, they implemented multi-user scheduling to mitigate interference issues in densely populated networks. Similarly, in [11], the authors focused on enhancing learning performance rather than communication throughput in the edge learning network. Therefore, they proposed an approach called learning-centric power allocation (LCPA), which is an analytically based solution for allocating radio resources in scenarios driven by learning. Simulation results showed that the LCPA scheme overcame conventional power allocation methods in terms of classification error.

None of the previous papers discussed above considered non-orthogonal multiple access (NOMA) to address spectrum scarcity issues caused by the massive connectivity for future wireless networks in task-oriented communication systems. Indeed, the IoT's rapid advances for 5G and beyond wireless networks must accommodate the massive connectivity demands imposed by the rapid growth in IoT devices. However, this reality introduces a spectrum scarcity issue, which can be dealt with through the adoption of a NOMA transmission strategy that operates in the power domain and employs techniques like superposition coding and successive interference cancellation [12]. Thus, motivated by the benefits provided by the NOMA technique and next-generation communication

systems envisioned to be task-oriented, in this paper, we investigate a low-complexity design to optimize the learning error and power allocation for task-oriented communications in an edge learning NOMA network. In the pursuit of finding optimal solutions, traditional optimization methods can introduce significant computational complexity. Additionally, these methods lack flexibility, requiring reformulation whenever network alterations occur [13]. As a remedy to these limitations, the realm of metaheuristic algorithms within artificial intelligence (AI) has emerged, offering a powerful approach to addressing intricate computation problems. Metaheuristic algorithms systematically generate potential solutions for optimization challenges, subsequently selecting the most promising option, while maintaining a balance between computational efficiency and solution accuracy.

In the domains of science and engineering, several metaheuristic algorithms have garnered substantial popularity, including the genetic algorithm [14], cuckoo search (CS) [15], ant colony optimization (ACO), and particle swarm optimization (PSO) [16]. Notably, Mohiz et al. [15] delved into a comprehensive exploration of diverse metaheuristic algorithms, discerning their effectiveness in optimizing task placement within network-on-chip cores. Their study concluded that CS outperformed baseline schemes by exhibiting minimal computational overhead. In [13], quantum particle swarm optimization (QPSO), an extension of the PSO algorithm, was applied to optimization problems for wireless communication networks. The simulation results showed that QPSO overcame the standard PSO and several metaheuristic methods to maximize the secrecy energy efficiency in a cooperative NOMA system. Motivated by the inherent benefits that metaheuristic algorithms bring to bear in tackling complex optimization dilemmas, this paper embarks on evolutionary computing algorithms for the resolution of power allocation optimization problems in an edge-learning NOMA system. Among these strategies, we propose an ACO-based scheme as a promising candidate that manages to strike a balance between accuracy and computational complexity. To comprehensively assess the performance of our proposed network, we formulated two distinct optimization problems: a single-task case (SC) and a multiple-task case (MC).

The main contributions of this paper can be summarized as follows.

- A task-oriented power allocation scheme is proposed for SC and MC optimization problems where the NOMA transmission strategy is considered in an edge learning system to improve network performance. To validate the advantage from using NOMA, we evaluate its performance against the maximum ratio combining (MRC) technique in terms of achievable data rate.
- Furthermore, we propose a novel power allocation scheme based on the ACO algorithm. This scheme is aimed at minimizing learning errors while optimizing power variables within a task-centric edge learning NOMA system. In addition, for comparison purposes, we solve the conventional sum-rate maximization prob-

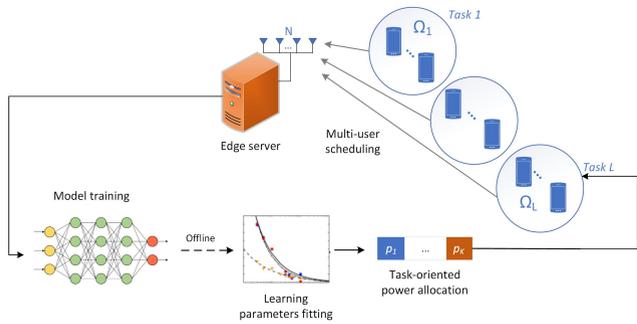


FIGURE 1: The system model of NOMA for task-oriented edge learning.

lem. To assess the practical applicability of our algorithms, we examine their performance in the context of three perception tasks related to autonomous driving systems [17].

- Simulation results show that the proposed ACO-based algorithm efficiently resolves task-specific power allocation issues with significantly reduced computation time compared to conventional baseline schemes. In particular, we investigate four additional algorithms: QPSO [13], PSO, CS, and butterfly optimization [18]. In addition, our simulations reveal that the implementation of NOMA in a task-oriented edge learning system achieved higher data rates in comparison to the standard MRC technique.

The rest of the paper is structured as follows. The system model is described in Section II. In Section III, we present the problem formulation. In Section IV, we described the proposed ACO-based optimization scheme. In section V, we provide the simulation results, and the computational complexity analysis. In section VI, we outline unsolved problems, and provide future research directions. Finally, conclusions are described in Section VII.

II. SYSTEM MODEL

We consider the task-oriented edge learning NOMA system shown in Figure 1, in which the edge server is equipped with N antennas. We consider L user groups, with L different learning tasks, $\{\tau_1, \dots, \tau_L\}$. Let us define the group of users as $\Omega = \{\Omega_1, \dots, \Omega_L\}$, where Ω_l represents the users executing the l -th task, with the number of users equal to $|\Omega_l|$. We assume each user is exclusively associated with a single task, and the total number of users is denoted as K , i.e., $K = \sum_{l=1}^L |\Omega_l|$. Furthermore, power allocation for each k^{th} user is denoted p_k .

Regarding the L distinct learning tasks depicted in Figure 1, each of these tasks encompasses a specific dataset, a learning model, the process of fine-tuning model parameters, and a task-oriented power assignment. Moreover, $\mathbf{h}_k \in \mathbb{C}^{N \times 1}$ denotes the complex-valued channel fast-fading vector from the k^{th} user to the edge server. It is assumed that users in a higher indexed group have better channel conditions than

users in a lower indexed group [19]. Without loss of generality, we consider $\|\mathbf{h}_1\|^2 \leq \dots \leq \|\mathbf{h}_K\|^2$. Following NOMA principles for uplinks that utilize the successive interference cancellation (SIC) technique, the decoding process prioritizes channels in descending order of power. This means the signal with the highest power and associated with the k^{th} user is decoded first at the edge server. Subsequently, this initial decoding contains interference from all users that have comparatively weaker channel conditions. Thus, the achievable rate of decoding a message from the k^{th} user can be expressed as follows:

$$A_k^{NOMA} = \begin{cases} \log_2 \left(1 + \frac{p_k g_{k,k}}{\sum_{j=1, j \neq k}^{k-1} p_j g_{k,j} + \sigma^2} \right), & 2 \leq k \leq K \quad (1a) \\ \log_2 \left(1 + \frac{p_k g_{k,k}}{\sigma^2} \right), & k = 1 \quad (1b) \end{cases}$$

where $g_{k,j}$ denotes the composite channel gain from the j -th user at the edge server when decoding data of the k^{th} user, computed as $g_{k,k} = \rho_k \|\mathbf{h}_k\|_2^2$ if $j = k$, and $g_{k,j} = \rho_j |\mathbf{h}_k^H \mathbf{h}_j|^2 / \|\mathbf{h}_k\|_2^2$ if $j \neq k$. ρ_k represents the path loss of the k^{th} user, and σ^2 is the variance of additive white Gaussian noise. Note that if we consider the MRC technique, interference from other users is considered noise. Subsequently, the achievable data rate of the k^{th} user can be expressed as follows:

$$A_k^{MRC} = \log_2 \left(1 + \frac{p_k g_{k,k}}{\sum_{j=1, j \neq k}^K p_j g_{k,j} + \sigma^2} \right). \quad (2)$$

At the edge server, the number of samples transmitted by a user to learn task τ_l can be calculated as follows [1]:

$$V_l = \sum_{k \in \Omega_l} \left[\frac{WTA_k^{NOMA}}{D_l} \right] + R_l \approx \sum_{k \in \Omega_l} \frac{WTA_k^{NOMA}}{D_l} + R_l, \quad (3)$$

where W denotes the total bandwidth in hertz, and T represents transmission time in seconds. For each data transmission, D_l represents the number of bits. Meanwhile, for the l^{th} pre-trained task, R_l represents the initial amount of historical data. Furthermore, in order to link wireless resource allocation with the performance of machine learning, a non-linear exponential function, $\Theta_l(V_l | a_l, b_l) \triangleq a_l V_l^{-b_l}$, is formulated to represent the characteristics of the learning error function, where the tuning parameters a_l and b_l represent non-independent and identically distributed parallel datasets, respectively. In practice, values for a_l and b_l are determined through a process of fitting the learning error function from the historical dataset. This tuning function closely aligns with the empirical data of the machine learning model, showcasing a good degree of alignment [1].

III. PROBLEM FORMULATION

In this paper, we minimize the learning error function by jointly optimizing the power allocation variables in the proposed NOMA edge learning network for multi-users with task-oriented communication. Moreover, we consider an SC and an MC. Therefore, the optimization problem for the SC is formulated as follows:

$$\min_{\{p_k\}} a_{SC} V_{SC}^{-b_{SC}} \quad (4a)$$

subject to

$$\sum_{k=1}^K p_k = P, \quad (4b)$$

$$p_k \geq 0, \forall k \in \Omega, \quad (4c)$$

where constraint (4b) indicates that the power allocation of all users does not exceed the total available power, P . For the MC, the optimization problem is formulated as follows:

$$\min_{\{p_k\}} \sum_{l=1}^L \phi_l \times a_l V_l^{-b_l} \quad (5a)$$

subject to

$$(4b), \text{ and } (4c),$$

where $\phi_l \triangleq R_l D_l / \left(\sum_{i=1}^L R_i D_i \right)$ is the weight of diverse datasets. Note that the objective function in (5) can adapt to various learning tasks by dynamically adjusting weight factors $\phi_l, \forall l$. Furthermore, for comparison purposes, we formulate the optimization problem for the traditional sum-rate maximization as follows:

$$\max_{\{p_k\}} \sum_{k=1}^K A_k^{NOMA} \quad (6a)$$

subject to

$$(4b), \text{ and } (4c).$$

IV. ACO-BASED OPTIMIZATION METHOD

In this research article, we investigate the ACO algorithm to address the optimization problems formulated in (4a), (5a), and (6a). Initially, the ACO algorithm was introduced by Socha and Dorigo for discrete spaces, drawing inspiration from the foraging behavior of real ants [20]. In the ACO algorithm, a group collaborates to discover the optimal solution by utilizing an artificial pheromone trail to navigate through potential paths or solutions. This sharing of information through pheromone deposition enables “the ants” to construct effective paths using a discrete probabilistic approach. Nevertheless, the inherent pheromone deposition mechanism of ACO is tailored for discrete domains, necessitating adaptation for continuous spaces. Thus, the ACO algorithm designed for continuous domains achieves this by replacing discrete probability distributions with continuous probability distributions, which are represented as probability density functions

| | | | | | |
|---------|---------|---------|---------|----------|---------|
| C_1 | c_1^1 | c_1^2 | c_1^3 | $f(C_1)$ | we_1 |
| C_2 | c_2^1 | c_2^2 | c_2^3 | $f(C_2)$ | we_2 |
| \cdot | \cdot | \cdot | \cdot | \cdot | \cdot |
| \cdot | \cdot | \cdot | \cdot | \cdot | \cdot |
| C_j | c_j^1 | c_j^2 | c_j^3 | $f(C_j)$ | we_j |
| \cdot | \cdot | \cdot | \cdot | \cdot | \cdot |
| \cdot | \cdot | \cdot | \cdot | \cdot | \cdot |
| C_r | c_r^1 | c_r^2 | c_r^3 | $f(C_r)$ | we_r |

FIGURE 2: The solution archive (three users).

(PDFs) [21]. Within each iteration of the ACO algorithm, the system learns these PDFs by accumulating a historical record of candidate solutions, which is stored in a dedicated solution archive. To be more specific, ACO retains the most promising solutions within a designated file, denoted as a solution archive as depicted in Figure 2. This practice forms a Gaussian probabilistic model that implicitly simulates the concept of the continuous pheromone. It is worth noting that a single Gaussian function is limited to searching in one dimension because it possesses only one maximum. To overcome this limitation, a Gaussian kernel PDF is employed, which is essentially a weighted sum of several one-dimensional Gaussian functions.

Furthermore, to cater to multi-dimensional search spaces, a distinct Gaussian kernel PDF is constructed for each dimension. Each of these Gaussian kernel PDFs utilizes three essential vectors: the weights, the vector of means, and the vector of standard deviations. This comprehensive approach allows ACO to adapt and excel at optimization of real-valued parameters in continuous domains, making it a powerful tool for solving complex optimization problems. In this paper, we consider ACO to optimize the power allocation variables, $\{p_k\}$, which minimizes the cost function given by the learning error in both the SC and the MC. The set of power variables, p_k , is denoted by the vector \mathbf{C} . Let us change the notation of p_k to c^k where $k = 1, \dots, K$ indicates the index of the dimension vector. In this paper, $K = 6$ since we consider six users; each of them is assigned one power allocation variable, c^k . Accordingly, the group of variables is $\mathbf{C} = \{c^1, c^2, c^3, c^4, c^5, c^6\}$. Then, to build the solution archive [21], random solutions are generated in the range $[0, P]$ for each c^k . Every solution C_j of the archive is composed of a vector of the variables to be optimized, $\{c_j^1, c_j^2, \dots, c_j^k, \dots, c_j^K\}$, the associated objective function values, $\{f(C_j)\}$, and associated weight we_j where $j = 1, 2, \dots, r$. The solution archive is built in ascending order according to the performance of the objective function values, $\{f(C_1), f(C_2), \dots, f(C_r)\}$.

Moreover, the weight of the j -th element in the solution archive can be expressed as follows:

$$w e_j = \frac{1}{\sqrt{2\pi}\zeta r} \exp \left[-\frac{1}{2} \left(\frac{j-1}{\zeta r} \right)^2 \right] \quad (7a)$$

subject to

$$\sum_{j=1}^r w e_j = 1, \quad (7b)$$

where ζ is the intensification factor and is a positive real number that manages the degree of selection pressure in the process. For high values of ζ , a large number of possible solutions in the archive may be chosen. Therefore, as the value of ζ rises, convergence may be slower.

In addition, f_r denotes the sample size of the new candidate solutions. The generation of f_r is based on a guide solution. For this purpose, the Roulette wheel selection algorithm is utilized to choose a guide solution from the solution archive in accordance with equation (8) such that the higher the fitness, the higher the probability of selection as a parent solution:

$$P(C_j) = \frac{w e_j}{\sum_{a=1}^r w e_a}. \quad (8)$$

After the guide solution is chosen, the f -th new candidate solution is generated based on the Gaussian PDF. Thus, for every decision variable, a Gaussian model is built:

$$G_f^k(c_f^k) = \sum_{j=1}^r w e_j N(c_f^k; \mu_{guide,f}^k, \sigma_{guide,f}^k), \quad (9)$$

$$N(c, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma^k} \exp \left[-\frac{1}{2} \left(\frac{c_f^k - \mu_{guide,f}^k}{\sigma_{guide,f}^k} \right)^2 \right], \quad (10)$$

where $c_{guide,f}^k$ indicates the k^{th} element of the vector \mathbf{C}_{guide} in the guide solution. Moreover, the values of mean and variance for the Gaussian kernels in the guide solution can be expressed as follows:

$$\mu_{guide,f}^k = c_{guide,f}^k, \quad (11)$$

$$\sigma_{guide,f}^k = \frac{\lambda}{r-1} \sum_{q=1}^r |c_q^k - c_{guide,f}^k|, k = 1, \dots, K, \quad (12)$$

where $\lambda > 0$ represents the exploration and exploitation balance. High values of λ indicate high exploration. On the other hand, small values of λ represent high exploitation. It is worth noting that the f -th new candidate solution, $\mathbf{C}_f = \{c_f^1, \dots, c_f^K\}$ with $f = r+1, \dots, r+f_r$, is generated, dimension by dimension, based on the guide solution.

Finally, f_r new candidate solutions are assessed to obtain $\{f(\mathbf{C}_f)\}$ and are added to the solution archive: $A = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{r+f_r}\}$. Then, the best r solutions are preserved for the subsequent iteration, and the remaining solutions are discarded, thereby restoring the solution archive's size to r .

Algorithm 1 ACO-based scheme to solve problems (4a) and (5a).

- 1: **inputs:** set the parameters $Ite_{ACO}, K, f_r, \zeta, \lambda, N, P$.
- 2: Compute r random solutions in the range of $[0, P]$ for the power allocation variables to be optimized, $\{p_k\}$, denoted by $\mathbf{C}_j = \{c_j^1, \dots, c_j^K\}, j = 1, 2, \dots, r$, and assess their performance by solving the optimization problems to obtain $\{f(\mathbf{C}_j)\}$.
- 3: Sort the r solutions in the archive, $A = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_r\}$
- 4: Evaluate the weights in accordance with (7a).
- 5: **while** $It \leq Ite_{ACO}$ **do**
- 6: **for** $f = r+1$ to $r+f_r$ **do**
- 7: **for** $k = 1$ to K **do**
- 8: Chose a guide solution from the solution archive.
- 9: Generate a sample, c_f^k , from a Gaussian distribution with parameters $\mu_{guide,f}^k, \sigma_{guide,f}^k$.
- 10: **end**
- 11: Save and assess the generated solution $\mathbf{C}_f = \{c_f^1, \dots, c_f^K\}$ by solving problem (4a) or (5a) to obtain $\{f(\mathbf{C}_f)\}$.
- 12: **end**
- 13: Update solution archive $A = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{r+f_r}\}$ with the best r candidate solutions and remove the remaining. Sort the r solutions in the archive.
- 14: Increase the number of iterations: $It = It + 1$.
- 15: **end while**
- 16: **Output:** Set $\mathbf{C}_1 = \{c_1^1, \dots, c_1^K\}$ as the best solution for power variables $\{p_1, \dots, p_k, \dots, p_K\}$ of problem (4a) or (5a).

Accordingly, in each iteration It , where $It = 1, 2, \dots, Ite_{ACO}$, the results in the solution archive are updated to achieve the most optimal outcome recorded to this point. Ite_{ACO} denotes the maximum number of iterations in the algorithm. Finally, the best result so far is the optimization result until meeting the termination criteria. Otherwise, the ACO algorithm recalculates the probabilistic model. Algorithm 1 summarizes the ACO algorithm to minimize the learning error in problems (4a) and (5a) while optimizing power allocation $\{p_k\}$. Note that optimization problems (4a) and (5a) are solved individually by the ACO algorithm.

V. SIMULATION RESULTS

In this section, we showcase simulation results to assess the performance of the designed schemes in comparison to benchmark methods. To generate the simulation results, we used a computer with a 4 GHz i7-6700K CPU and 16 GB of RAM. The simulation parameter settings for the proposed wireless communications were similar to those in [11]. Specifically, we set communication bandwidth $W = 180\text{kHz}$, the number of users to $K = 6$, the total power budget at $P = 20\text{mW}$, the path loss of the k^{th} user was $\rho_k = -90\text{dB}$, and channel \mathbf{h}_k was based on $\mathcal{CN}(0, \rho_k \mathbf{I})$. Since we considered multiple-user connectivity in an IoT

TABLE 1: Simulation parameters for swarm-based schemes.

| Algorithm | Parameters |
|-----------|---|
| ACO | Sample size, $f_r = 20$ Number of iterations, $Ite_{ACO} = 400$ |
| PSO | Number of iterations, $Ite_{PSO} = 500$ Number of particles, $N_{PSO} = 80$ Inertia weight, $Ine = 0.7$ Scaling factor, $c_1 = 1.496$ Scaling factor, $c_2 = 1.496$ |
| QPSO | Number of iterations, $Ite_{QPSO} = 500$ Number of particles, $N_{QPSO} = 55$ |
| CS | Number of iterations, $Ite_{CS} = 500$ Number of nests, $N_{CS} = 55$, Probability of abandoned maximum, $p_a^{max} = 0.5$, Probability of abandoned minimum, $p_a^{min} = 0.25$ |
| BOA | Number of generations, $G_{BOA} = 500$ Number of agents, $S_{BOA} = 70$ Sensory modality, $sm = 0.01$ Power exponent, $pe = 0.01$ Switch probability, $sp = 0.5$ |

network, we assumed the number of user sets is equal to the number of L different tasks [1]. Therefore, we considered three tasks, one for each set of users (i.e., three user sets). Each set was composed of two users. Thus, for task-oriented learning at the edge, we considered three tasks in autonomous driving [17]—Task 1: weather classification utilizing RGB images and a CNN, Task 2: object detection using point cloud data and sparsely embedded convolutional detection (SECOND), and Task 3: traffic design using RGB images and YOLOV5. In these experiments, datasets were generated by the open-source autonomous driving simulation platform called CarlaFLCAV. These datasets are available online at <https://github.com/SIAT-INVS/CarlaFLCAV>. In the simulation experiments, Task 2 was selected to perform the SC. Meanwhile, the three-task case representing the MC is given by Task 1, Task 2, Task 3. Each RGB image contained $D_1 = D_3 = 0.7\text{MB}$, and that of each point cloud sample $D_2 = 1.6\text{MB}$. Moreover, the number of historical data samples was $R_1 = R_2 = R_3 = 300$. The learning parameters for Task 1, Task 2, and Task 3 were $(a_1, b_1) = (10.34, 1.2)$, $(a_2, b_2) = (0.5, 0.1)$, and $(a_3, b_3) = (8.89, 0.64)$, respectively. The results were averaged over several channel realizations.

In the simulations, we considered five swarm-intelligence schemes: ACO, PSO, QPSO, CS, and BOA. The settings for the parameters of each algorithm listed in Table 1 were determined by analyzing the optimal outcomes obtained from several experiments.

Figure 3 and Figure 4 show the convergence behavior for the SC and the MC, respectively, of the proposed ACO algorithm and the four additional swarm intelligence baseline schemes when the number of antennas is equal to $N = 6$, and the variance of additive white Gaussian noise is $\sigma = -90\text{dBm}$. The learning error is computed by (4a) and (5a) for the SC and the MC, respectively. From Figure 3 and Figure 4, we observe that as the number of iterations increased, the learning error decreased. Moreover, for the SC and the MC,

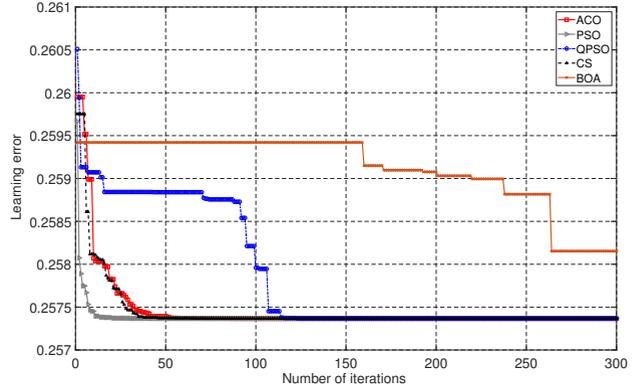


FIGURE 3: Convergence behavior of the swarm intelligence schemes for the SC.

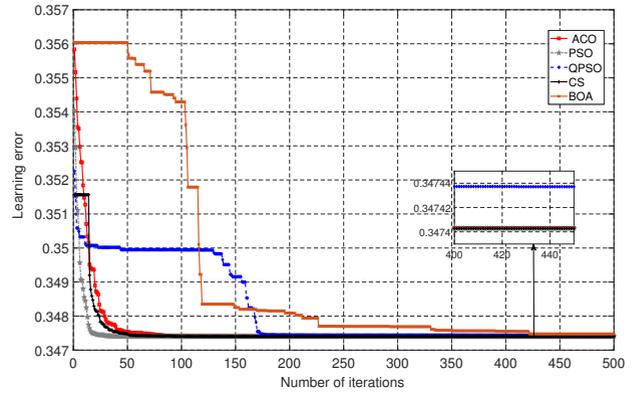


FIGURE 4: Convergence behavior of the swarm intelligence schemes for the MC.

observe that PSO achieved faster convergence than the other swarm-learning algorithms, followed by CS and the proposed ACO algorithm. On the other hand, the worst performance was given by BOA, followed by the QPSO algorithm.

To validate the superiority of the ACO algorithm over PSO and CS, we evaluated the algorithms in terms of computational complexity and computation time. In particular, the computational complexity of PSO depends of the number of particles, N_{PSO} , and the number of iterations, Ite_{PSO} . Therefore, its computational complexity is expressed as $\mathcal{O}(N_{PSO} \cdot Ite_{PSO})$. The computational complexity of CS relies on the number of nests, N_{CS} , the probability of abandonment, pa , and the number of iterations, Ite_{CS} . Therefore, the computational complexity for CS is given by $\mathcal{O}(N_{CS} + (pa \cdot N_{CS}) \cdot Ite_{CS})$. Regarding the proposed ACO algorithm, computational complexity is based on the sample size, f_r , and the number of iterations, Ite_{ACO} , which results in $\mathcal{O}(f_r \cdot Ite_{ACO})$. Accordingly, we can appreciate the ACO algorithm has the least computational complexity because it requires fewer particles and iterations than the PSO and CS algorithms. In Table 2, we evaluate the investigated schemes in terms of computational time and learning error performance. Overall, we can see that the learning error values vary slightly between the algorithms in both the SC and the MC.

TABLE 2: Computational time.

| Algorithm | SC-learning error | SC-computation time [s] | MC-learning error | MC-computation time [s] |
|-----------|-------------------|-------------------------|-------------------|-------------------------|
| ES | 0.2569 | 1250 | 0.34737 | 3587 |
| ACO | 0.2574 | 0.623 | 0.34740 | 0.803 |
| PSO | 0.2574 | 1.124 | 0.34740 | 1.288 |
| CS | 0.2574 | 2.231 | 0.34743 | 2.94 |
| QPSO | 0.2574 | 3.21 | 0.34743 | 3.613 |
| BOA | 0.2581 | 2.36 | 0.34747 | 2.94 |

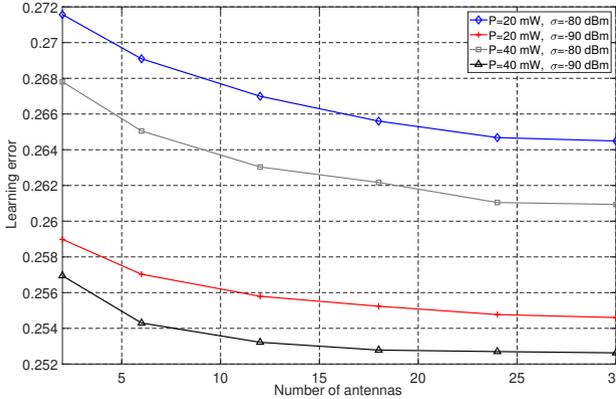


FIGURE 5: Learning error versus number of antennas in the SC.

However, the main difference among of them is computation time; the least computation time was obtained by the proposed ACO algorithm where the result is remarkably lower than that obtained by CS and PSO. This can be attributed to the fact that the ACO algorithm achieves convergence with a smaller number of particles compared to its counterparts. Moreover, in Table 2, we compare the learning performance and computational complexity between the swarm-based algorithms and the traditional exhaustive search (ES) method, typically employed to identify optimal solutions in optimization problems. The ES technique, while thorough, is burdened by its significant computational demands and slow convergence rate due to its systematic evaluation of every potential solution. Notably, the computational complexity of exhaustive search scales with the number of candidate solutions. In contrast, our proposed-based scheme called ACO, offers expedited convergence towards near-optimal solutions and requires less computational overhead compared to ES.

Figure 5 and Figure 6 show the learning error computed by (4a) and (5), respectively, considering different available transmission power levels. To gain more insight into the proposed edge learning system, Figure 5 and Figure 6 show the learning error versus the number of antennas, with two values for the variance of additive white Gaussian noise: $\sigma = -80\text{dBm}$ and $\sigma = -90\text{dBm}$. We can see from Figure 5 and Figure 6 the benefit of a multiple-antenna system, because as the number of antennas increased the learning error decreased for both the SC and the MC. This demonstrates the advantage of multiple antenna at the edge server.

Furthermore, to validate the advantage of the NOMA sys-

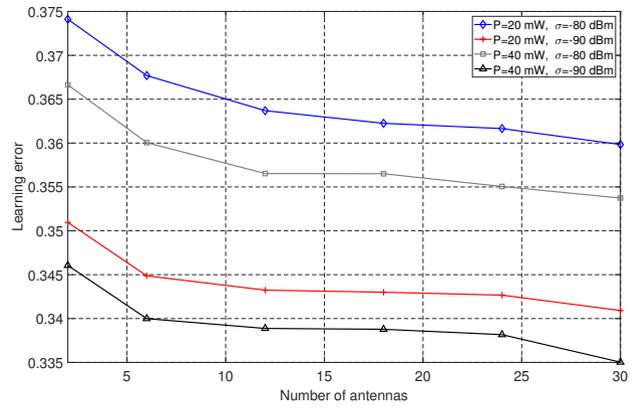


FIGURE 6: Learning error versus number of antennas in the MC.

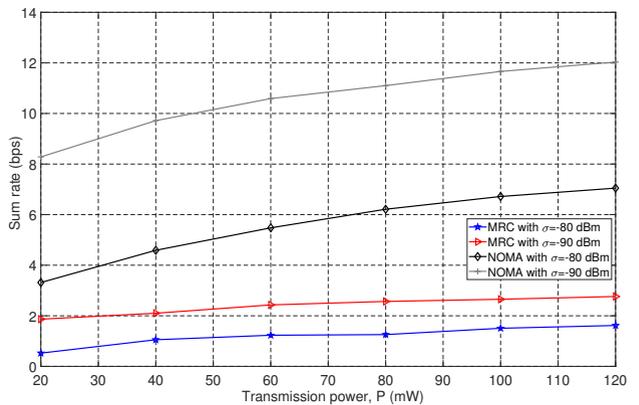


FIGURE 7: Sum rate versus transmission power in the SC between the proposed edge learning task-oriented communications with NOMA and the baseline with MRC.

tem against the conventional MRC method, Figure 7 and Figure 8 show the sum rate given by the summation of a user's rate versus the total power budget. Overall, observe that as the transmission power increased, the sum rate improved. This is because by increasing the total available power, a greater amount of power becomes available for data transmission from the users to the edge server, resulting in an increase in the achievable rate that minimizes the objective function defined by the learning error. It is worth noting that according to equations (3) and (5), an increase in the sum rate corresponds to a decrease in learning error. Moreover, from Figure 6 and Figure 7, we can see that the edge learning system with the NOMA technique outperformed the conventional edge learning with the MRC technique in terms of sum rate. This is because NOMA can remove the interference from other users by applying SIC. In this manner, the decoding process is carried out in descending order according to the channel conditions, as expressed in (1), instead of treating the interference from other users as noise, as in (2).

Figure 9 shows the relationship between learning error and transmission power in our proposed scheme in (5), which is designed to minimize the learning error. The baseline scheme,

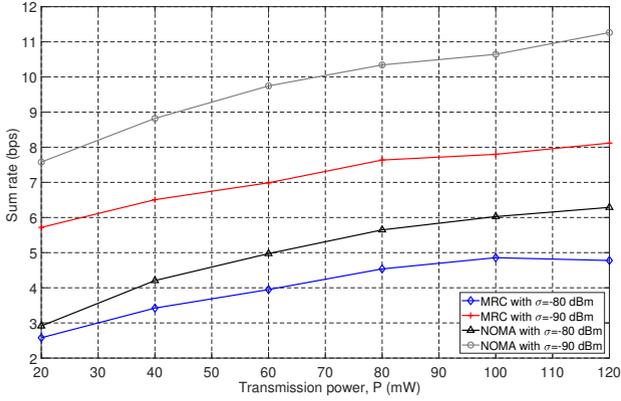


FIGURE 8: Sum rate versus transmission power in the MC between the proposed edge learning task-oriented communications with NOMA and the baseline with MRC.

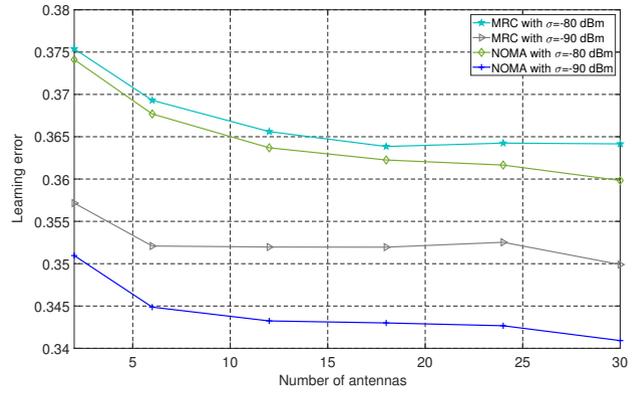


FIGURE 10: Learning error versus number of antennas between the proposed edge learning task-oriented communications with NOMA and the baseline with MRC.

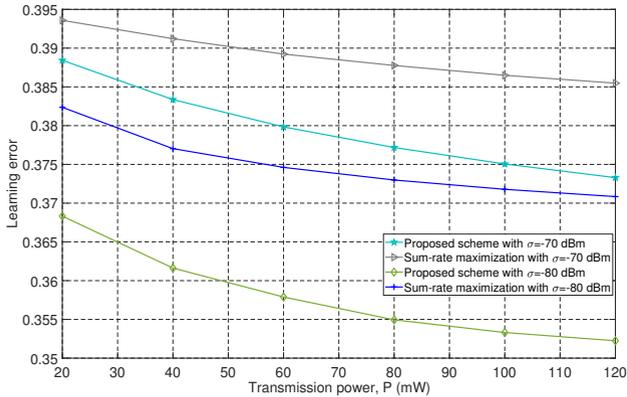


FIGURE 9: Learning error versus transmission power in the MC.

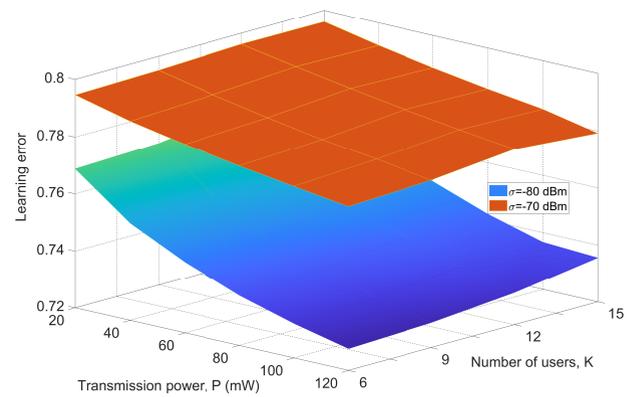


FIGURE 11: Learning error versus transmission power and number of users in the MC.

(6), is primarily focused on the traditional goal of maximizing the sum rate. The figure reveals a compelling trend as the power budget increased across all scenarios—a reduction in learning error. However, insight emerges when comparing these two schemes. Our proposed optimization framework in (5) outperformed the conventional sum-rate maximization approach in (6) by achieving a significantly lower learning error. This superiority stems from our incorporation of ML techniques in our optimization problem, a facet that the sum-rate maximization scheme in (6) neglects.

Furthermore, Figure 10 shows the edge learning error versus the number of antennas between the proposed task-oriented communication with NOMA and the baseline with MRC. From Figure 10, we can observe that NOMA is able to reduce learning error compared to the benchmark MRC scheme, underscoring the efficacy of NOMA in enhancing edge learning outcomes.

In addition, to showcase the outstanding generalization of our proposed ACO-based approach for task-oriented communication in NOMA networks, we consider three distinct architectures for various classification: a 6-layer convolutional neural network (CNN6) deployed for clas-

sifying the MNIST dataset [22], a deep residual network consisting of 110 layers (ResNet110) applied to the CIFAR10 dataset [23], and a PointNet model utilized for processing 3D point clouds within the ModelNet40 dataset [24]. The learning parameters for these tasks are: $(a_1, b_1, R_1, D_1) = (7.3, 0.69, 300, 6276)$, $(a_2, b_2, R_2, D_2) = (8.15, 0.44, 1600, 24584)$, and $(a_3, b_3, R_3, D_3) = (0.95, 0.24, 800, 192008)$, respectively. For further insights into obtaining these learning parameters, readers are directed to Section III of [11]. Accordingly, Figure 11 shows the learning error performance by using the aforementioned three-task case, versus the transmission power, P , and different number of users, K when the number of antennas is equal to $N=2$. Similar to Figure 9, from Figure 11, we can observe that as the transmission power increases P , the learning error is improved. On the other hand, from Figure 11, we can see that as the number of user increases, the learning error slightly rises since the total transmission power allocated to each user need to satisfy a maximum value, P . Therefore, as more users are served in the system, less power will be assigned to each user, as well as, the interference between users increases as the number of users increases.

It is worth to highlight that for time-varying channel conditions, the implementation of the ACO algorithm is imperative within each coherence time subsequent to channel estimation. Approaches rooted in swarm intelligence offer a promising avenue for optimization, boasting both efficiency and low complexity. These methods yield solutions that are nearly optimal while demanding minimal computational resources and ensuring stable convergence [25]. Such characteristics make them particularly suitable for supporting delay-sensitive applications across wireless communication networks. Particularly in scenarios characterized by highly dynamic channel conditions, the necessity for rapid algorithms capable of furnishing nearly optimal solutions akin to ACO becomes apparent. Furthermore, there exists the option to adjust ACO parameters, such as sample size and maximum iteration count, to expedite convergence of the objective function. However, such alterations entail the risk of obtaining sub-optimal solutions, potentially local optima rather than global optimal.

VI. FUTURE WORKS

Future directions in research should investigate the integration of reinforcement learning (RL) techniques into the task-oriented communication framework for edge learning systems. RL is a branch of machine learning that focuses on learning optimal decision-making policies through interactions with an environment. In the context of edge learning and communication systems, RL could be utilized to dynamically adapt communication strategies based on the learning task at hand [26]. This could involve optimizing power allocation, resource allocation, and scheduling decisions in real-time to maximize the learning performance of edge devices. Here, RL algorithms could adapt communication parameters based on feedback from the learning process, such as error rates or learning progress.

Moreover, within the framework of mobile edge computing systems, mobile devices utilize servers to delegate tasks for low-latency computing, a process that can occur in either partial or binary modes. In the partial mode, computational tasks are divided into two segments: one segment is processed locally on the mobile device while the other is transferred to a nearby mobile edge computing server for execution. Conversely, in the binary mode, the entire task is either completed locally on the device or transferred entirely to a nearby mobile edge computing server via the uplink connection [12][27]. In terms of future directions, an interest approach to explore is to joint task offloading and resource optimization in NOMA-based vehicular networks [27][28] with edge computing AI technology. This can be beneficial from the resource management perspective since both, the user and the server can perform tasks. However, it also brings some complexity and security challenges to solve because of the load burden at the user and the task transmission in an open wireless environment susceptible to eavesdroppers' attacks.

Additionally, the reduction of the communication overhead plays a crucial role in wireless communications for optimizing network performance, reducing costs, minimizing

latency, and improving resource allocation. In this paper, the communication overhead for power allocation with ACO is primarily attributed to uplink channel estimation, which is generally performed using pilot symbols. In particular, let us consider an edge computing system with a coherence interval composed of Ω symbol periods. The pilot sequences assigned to users are designed to be pair-wise orthogonal and consist of ς symbols [29]. During the coherence interval, channel estimation takes ς symbols, while the data uplink transmission occurs in the remaining $\Omega - \varsigma$ symbols. It's important to note that the required number of symbols in the pilot sequence, ς , increases with the number of users. For example, we can select $\varsigma \geq K$ to avoid pilot contamination. Therefore, the channel estimation overhead increases with the number of users. However, it is worth noting that state-of-art schemes for power allocation also necessitate channel estimation procedures. Therefore, concerning communication overhead, the proposed ACO-based scheme involves similar communication overhead compared to state-of-the-art schemes. In future work, reducing the channel estimation overhead can be studied through pilot-reuse and investigating schemes to mitigate pilot contamination in those scenarios.

VII. CONCLUSION

In this paper we proposed a novel power allocation design based on the ACO algorithm to optimize the learning error in a task-oriented edge NOMA system for an SC and an MC. The proposed ACO-based scheme provides a low-complexity solution because it requires fewer particles and iterations to achieve convergence than required by the comparative swarm learning techniques. Moreover, the ACO algorithm effectively achieved the best performance with less computation time than its counterparts. Furthermore, simulation results demonstrated that the integration of NOMA in the proposed task-oriented edge learning system reaches higher achievable data rates.

REFERENCES

- [1] H. Xie, M. Xia, P. Wu, S. Wang, and H. V. Poor, "Edge Learning for Large-Scale Internet of Things With Task-Oriented Efficient Communication," *IEEE Transactions on Wireless Communications*, pp. 1—1, 2023, doi: 10.1109/twc.2023.3271665.
- [2] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward Massive Machine Type Cellular Communications," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 120—128, Feb. 2017, doi: 10.1109/mwc.2016.1500284wc.
- [3] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457—7469, Aug. 2020, doi: 10.1109/jiot.2020.2984887.
- [4] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge Artificial Intelligence for 6G: Vision, Enabling Technologies, and Applications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 5—36, Jan. 2022, doi: 10.1109/jsac.2021.3126076.
- [5] A. Badi and I. Mahgoub, "ReapIoT: Reliable, Energy-Aware Network Protocol for Large-Scale Internet-of-Things (IoT) Applications," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13582—13592, Sep. 2021, doi: 10.1109/jiot.2021.3066531.
- [6] W. Cui, K. Shen, and W. Yu, "Spatial Deep Learning for Wireless Scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1248—1261, Jun. 2019, doi: 10.1109/jsac.2019.2904352.

[7] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 447–457, Jan. 2020, doi: 10.1109/twc.2019.2946140.

[8] Y. Shi, Y. Zhou, D. Wen, Y. Wu, C. Jiang, and K. B. Letaief, "Task-Oriented Communications for 6G: Vision, Principles, and Technologies," *IEEE Wireless Communications*, vol. 30, no. 3, pp. 78–85, Jun. 2023, doi: 10.1109/mwc.002.2200468.

[9] D. Gunduz et al., "Beyond Transmitting Bits: Context, Semantics, and Task-Oriented Communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, Jan. 2023, doi: 10.1109/jsac.2022.3223408.

[10] J. Shao, Y. Mao, and J. Zhang, "Task-Oriented Communication for Multidevice Cooperative Edge Inference," *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 73–87, Jan. 2023, doi: 10.1109/twc.2022.3191118.

[11] S. Wang, Y.-C. Wu, M. Xia, R. Wang, and H. V. Poor, "Machine Intelligence at the Edge With Learning Centric Power Allocation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7293–7308, Nov. 2020, doi: 10.1109/twc.2020.3010522.

[12] C. E. Garcia, M. R. Camana, and I. Koo, "Particle Swarm Optimization-Based Secure Computation Efficiency Maximization in a Power Beacon-Assisted Wireless-Powered Mobile Edge Computing NOMA System," *Energies*, vol. 13, no. 21, p. 5540, Oct. 2020, doi: 10.3390/en13215540.

[13] C. E. Garcia, M. R. Camana and I. Koo, "Ensemble Learning Aided QPSO-Based Framework for Secrecy Energy Efficiency in FD CR-NOMA Systems," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 2, pp. 649–667, June 2023, doi: 10.1109/TGCN.2022.3219111.

[14] J. H. Holland, "Outline for a logical theory of adaptive systems," *J. Assoc. Comput. Mach.*, vol. 9, no. 3, pp. 297–314, 1962, doi: 10.1145/321127.321128.

[15] M. J. Mohiz, N. K. Baloch, F. Hussain, S. Saleem, Y. B. Zikria, and H. Yu, "Application mapping using cuckoo search optimization with Lévy flight for NoC-based system," *IEEE Access*, vol. 9, pp. 141778–141789, 2021, doi: 10.1109/ACCESS.2021.3120079.

[16] D. Maldonado, J. Cepeda, S. Gamboa, J. A. Torres, and P. J. Cruz, "Optimum frequency response with decentralized drivers in power system," *IET Generation, Transmission & Distribution*, vol. 17, no. 9, pp. 1985–1998, Mar. 2023, doi: 10.1049/gtd2.12774.

[17] S. Wang et al., "Federated Deep Learning Meets Autonomous Vehicle Perception: Design and Verification," *IEEE Network*, vol. 37, no. 3, pp. 16–25, May 2023, doi: 10.1109/nmnet.104.2100403.

[18] M. O. Okwu and L. K. Tartibu, *Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications*, vol. 927. Cham, Switzerland: Springer, 2020.

[19] D. Song, W. Shin, J. Lee and H. V. Poor, "Sum-Throughput Maximization in NOMA-Based WPCN: A Cluster-Specific Beamforming Approach," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10543–10556, 1 July 2021, doi: 10.1109/JIOT.2021.3049956.

[20] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155–1173, Mar. 2008, doi: 10.1016/j.ejor.2006.06.046.

[21] S. M. Kalami Heris and H. Khaloozadeh, "Ant Colony Estimator: An intelligent particle filter based on," *Engineering Applications of Artificial Intelligence*, vol. 28, pp. 78–85, Feb. 2014, doi: 10.1016/j.engappai.2013.11.005.

[22] Li Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov. 2012, doi: 10.1109/msp.2012.2211477.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. Comp. Vis. Pat. Rec. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/cvpr.2016.90.

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. Comp. Vis. Pat. Rec. (CVPR)*, Jun. 2017, pp. 77–85, doi: 10.1109/cvpr.2017.16.

[25] P. Zhang, H. Yao, C. Fang, and Y. Liu, "Multi-objective enhanced particle swarm optimization in virtual network embedding," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, Jul. 2016, doi: 10.1186/s13638-016-0669-2.

[26] Z. Yang, Y. Liu and Y. Chen, "Distributed Reinforcement Learning for NOMA-Enabled Mobile Edge Computing," *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, Dublin, Ireland, 2020, pp. 1-6, doi:10.1109/ICCWorkshops49005.2020.9145457.

[27] Xincuo Xu, Kai Liu, Penglin Dai, Feiyu Jin, Hualing Ren, Choujun Zhan, Songtao Guo, "Joint task offloading and resource optimization in NOMA-based vehicular edge computing: A game-theoretic DRL approach," *Journal of Systems Architecture*, Volume 134, 2023.

[28] [4] Mohsan, S.A.H.; Li, Y.; Shvetsov, A.V.; Varela-Aldás, J.; Mostafa, S.M.; Elfikky, A. "A Survey of Deep Learning Based NOMA: State of the Art, Key Aspects, Open Challenges and Future Trends," *Sensors*, 2023, 23, 2946. <https://doi.org/10.3390/s23062946>.

[29] [5] N. T. Ti, L. B. Le, and Q. Le-Trung, "Computation Offloading in MIMO Based Mobile Edge Computing Systems Under Perfect and Imperfect CSI Estimation," *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 2011–2025, Nov. 2021



CARLA E. GARCÍA received the B.Eng. degree in electronics and telecommunications engineering from the Escuela Politécnica Nacional (EPN), Quito, Ecuador, in 2016. She was awarded the Brain Korean (BK) 21+ scholarship for her Master and PhD studies in South Korea. She received the M.Sc. degree and a Ph.D degree from the University of Ulsan, South Korea, in 2020 and 2023, respectively. She held the positions of Research Assistant and Postdoctoral Researcher at the Department of Electrical, Electronic and Computer Engineering, University of Ulsan, South Korea. She joined the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, where she currently holds a Research Associate position.



MARIO R. CAMANA received his B.E. degree in electronics and telecommunications engineering from the Escuela Politécnica Nacional (EPN), Quito, Ecuador, in 2016, and his Ph.D. degree in Electrical Engineering from the University of Ulsan, Ulsan, South Korea, in 2023. He held the positions of Research Assistant and Postdoctoral Researcher at the University of Ulsan, South Korea, and the position of Postdoctoral Researcher at Korea University, Seoul, South Korea. He is currently a Research Associate at the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg. His research interests include machine learning, optimizations, ISAC, wireless power transfer and MIMO communications.



INSOO KOO received the BE from Kon-Kuk University, Seoul, Korea, in 1996, and an M.Sc. and a Ph.D. from the Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 1998 and 2002, respectively. From 2002 to 2004, he was with the Ultrafast Fiber-Optic Networks Research Center, GIST, as a Research Professor. In 2003, he was a Visiting Scholar with the Royal Institute of Science and Technology, Stockholm, Sweden. In 2005, he joined the University of Ulsan, Ulsan, Korea, where he is currently a Full Professor. His current research interests include spectrum sensing issues for CRNs, channel and power allocation for cognitive radios (CRs) and military networks, SWIPT MIMO issues for CRs, MAC and routing protocol design for UW-ASNs, and relay selection issues in CCRNs.