AN EFFICIENCY SCHEME FOR MEC OFFLOADING PROBLEM BASED ON THE PSO ALGORITHM

Hoang Trong Minh, Nguyen Thanh Tra
Telecommunication Faculty No1, Posts and Telecommunications Institute of Technology

***Abstract*:** Mobile Edge Computing (MEC) is crucial to the aim of integrating the Internet of Things with 5G and forthcoming core technologies. MEC not only acts as an extension of cloud computing for the sake of data distribution but also provides local computing, ensures privacy, enhances system security, and improves system reliability. Particularly, the problem of system offloading is vital for data processing, computation, and security paradigms. Due to its multi-objective and multi-constraint characteristics, this problem falls inside the NP-Hard domain. Specifically, offloading tasks must concurrently achieve two objectives: energy saving and latency restriction. Hence, the heuristics approach has been a beneficial approach for both research and deployment objectives. The study will present a particle swarm optimization (PSO) algorithm-based offloading scheme to address the identified problems. The numerical simulations presented in this paper will indicate that our proposal is more efficient than that of the previous proposal.

Keywords: Mobile Edge Computing, Offloading problems, Heuristics Algorithms, Particle Swarm Optimization.

I. INTRODUCTION

5G and beyond networks are being designed to support the future digital society with main service categories as enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low latency communications (URLLC) to meet the diverse commercial and industrial demands [1] [2]. In these scenarios, the Internet of Things (IoT) plays a vital role in enabling emerging applications by connecting the physical environment to the cyberspace of communication systems [3]. ToT is taking center stage as connected devices are expected to form a significant portion of this 5G network paradigm. Besides IoT-enabled applications that will bring convenience to human life, it is a highly daunting task for 5G to support these applications, such as data rate, latency, coverage, localization, and so on. The emergence of cloud computing and its extension to the edge paradigm with the proliferation of devices is expected to lead to further innovation in IoT [4]. Hence, MEC Technology is a major driving force behind IoT integration into 5G networks to overcome the above challenges [5].

In a typical MEC architecture, MEC servers are located close to mobile users to make intelligent decisions aware of local execution or cloud-based processes. Users' tasks consist of computing demands, processing data, and security data that should be processed on the device or associated MEC servers [6]. Such situations are considered in the context of the offloading problem. In addition to the advantages MEC offers, optimizing the offloading approach has significant challenges due to the diversity of requirements, nonlinear environmental constraints, and the need for a multi-objective objective in practical applications. Hence, this sector has recently attracted a lot of research [7] [8].

The offloading problem has been recognized as an NP-Hard problem because it is a multi-objective and multi-constraint optimization problem containing nonlinear conditions from the operating environment. To deal with this, main optimization methods have to adapt real-time services in mobile edge computing, such as Lyapunov optimization, convex optimization, heuristic techniques, game theory, machine learning, and others [9] [10]. Among these approaches, in some experimental conditions, the intense fluctuation of input parameters and the requirement of simultaneous optimization of delay time and energy have led to many metaheuristics-based solutions that have increased in recent years [11]. In this paper, we propose a novel scheme for the MEC Offloading problem based on the particle swarm optimization algorithm to archive both energy and latency objects of tasks. Our contributions are two folds:

- Formulate the offloading problem in the multi-users, multi-servers scenario in conjunction with the latency and energy optimization problem on 5G's common communications systems.
- Compare and evaluate the proposed offloading method to a previous genetic algorithm-based offloading method.

This paper is organized as follows: Section II presents related work; section III briefs the background assumptions of the proposed scheme; Our proposed scheme and validation results are illustrated in IV; Last but not least, our conclusions and future works are presented in the last section.

II. RELATED WORK

Contact author: Hoàng Trọng Minh,
Email: hoangtrongminh@ptit.edu.vn
Manuscript received: 04/6/2022, revised: 24/7/2022, accepted: 22/8/2022.

Because of the problems of stochastic and constantly varying environments of IoT, the offloading decision-making optimization is sophisticated because of diverse influence factors and constraints. The offloading optimization problem based on metaheuristic algorithms has been proposed to overcome the complexity and real-time required in practical IoT systems. Specifically, the main objective of task offloading is to minimize the task execution time of applications running at user devices (UDs) and lower UEs' energy consumption. This sort of problem is the multi-constrained multi-purposes optimization. Study on mobile edge computing for ultra-dense cellular networks, the authors in [10] [12] proposed the distributed offloading strategy based on a binary-coded genetic algorithm designed to get an adaptive offloading decision. In a multi-user-to-multi-servers (MUMS) edge computing scenario, the proposed scheme can significantly reduce mobile devices' average latency and energy consumption in ultra-dense cellular networks. Besides focusing only on the ultra-dense configuration, the performance comparisons on other metaheuristics algorithms were not considered in this study.

Focusing on the emerging 5G applications, the authors in [13] proposed an offloading strategy based on particle swarm optimization (PSO) to achieve a relative balance between energy consumption and time delay. The simulation results show the network performance of the all-local executed offload algorithm and the random executed offload algorithm. The author in [14] proposed an intelligent particle swarm optimization (PSO) based offloading strategy with a cache mechanism to optimize mobile users' energy and delay. The PSO algorithm finds an appropriate offloading ratio to implement partial offloading. However, the trade-off ratio of the energy and delay factors is not figured out, and the algorithm performance has not been shown in these studies.

III. BACKGROUND AND ASSUMPTIONS

A. Background of the common metaheuristic algorithms

The genetic algorithm (GA) is modeled after the biological concept of a natural evolution of genomes [15]. The flowchart of the genetic algorithm is given in figure 1. The GA encodes the parameters of the objective function into a chromosome, which corresponds to a single candidate solution. Multiple chromosomes make up the genome or population. The algorithm simulates a "survival of the fittest" type scenario, where each generation of the algorithm attempts to improve upon the preceding generation. For each generation of the GA, three steps are performed: selection, crossover, and mutation. The GA encodes its chromosomes with binary strings of 0 or 1, performing well for many discrete problems, as in the full offloading problems.

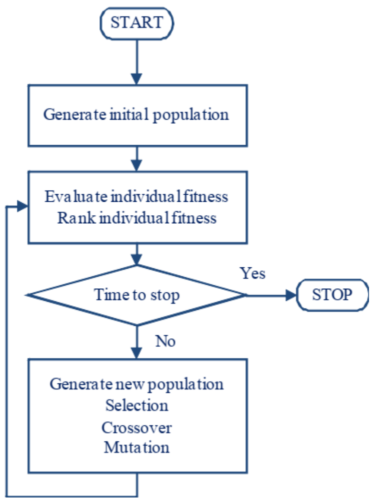

Figure 1. Flowchart for genetic algorithm

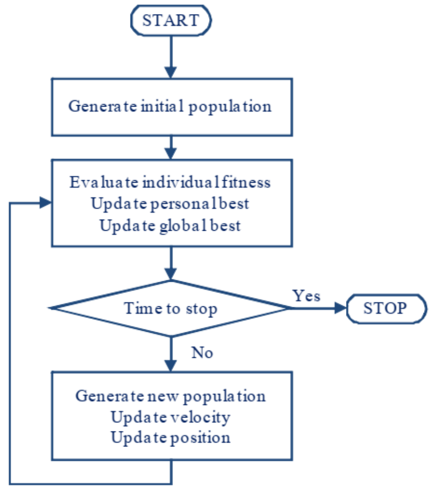

Figure 2. Flowchart for PSO

Particle Swarm Optimization (PSO) models its behavior after animals' swarming or flocking patterns [16]. It is very appealing because the simple conceptual framework and the analogy of birds flocking facilitated conceptual visualization of the search process. The basic PSO algorithm is shown in Figure 2. Instead of chromosomes, PSO has particles that make up its population, called a swarm. Unlike the GA, there is no "survival of the fittest" selection process for determining the particles that survive to the next generation, but rather just mutation. Each particle is moved from one location to another. This mutation is performed directedly, in which each particle is moved from its previous location to a new, better location.

The PSO algorithm has several advantages that make it an attractive optimization algorithm [17]:

- PSO is easy to set up and code.
- PSO is controlled by only three parameters (inertia weight, cognitive ratio, and social ratio). A slight change in any of these three controlling parameters produces a difference in performance.
- PSO is adaptable and can be combined with other optimization algorithms.

B. System model

As shown in Figure 3, the set of user devices is marked as $U=\{u_1,...,u_i,...,u_n\}$ which are distributed in the area with random distribution. MEC servers are denoted as $S=\{s_1,...,s_j,...,s_k\}$. The zones (Zone 1, Zone 2,.., Zone Z) are defined according to base station embedded MEC server coverage or clustering strategies. Current technologies assume that the 5G radio link parameters with the most common conditions.

We consider that each UD generates its tasks with the arrival rate λ_i according to the Poisson process, and one task needs to be offloaded at one time in a time slot. Denote a set of computation tasks on UD_i as $\Gamma = \{\tau_1, \tau_2, ..., \tau_M\}$ $m \in M$. A required task is triple parameters $\tau_i(m) \square < l_i, e_i^{\max}, t_i^{\max} >$.

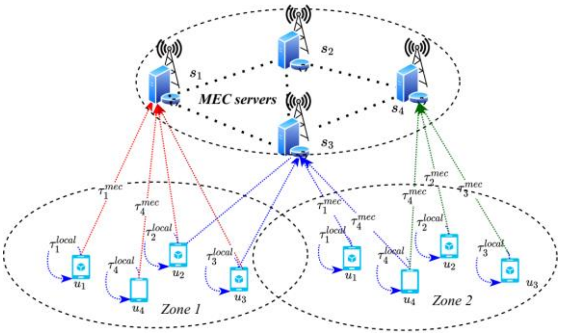

Figure 3. A general MEC model

In which l_i is the length of task (bits), the maximum requires energy $e_i^{\max}$ and deadline time $t_i^{\max}$. A task will be offloaded in the formulations as

$$\tau_i = \tau_i^{local} + \tau_i^{mec} \tag{1}$$

$$l_i = l_i^{local} + l_i^{mec} \tag{2}$$

IV. OFFLOADING PROBLEM FORMULATION

A. Communication model

Assuming that the 5G coverage areas as cluster zones and UD accesses the MEC server according to the Orthogonal Frequency Division Multiple Access (OFDMA) mechanism, the transmission rate of the user device u_i transfer task to the server s_j is calculated as follows

$$r_{i,j}^u = \frac{W_b}{N}\log_2\left(1 + \frac{p_u.h_u, b_j}{\omega_0 + \sum_{k=1,k\neq i}^{N} p_u.h_u, b_k}\right) \tag{3}$$

Where W_{b_j} is the bandwidth of the server s_j located to user devices, p_u is the transmission power of the user u_i, h_u is channel gain, and ω_0 denotes background noise. Equation (3) is based on the basic Shanon formula.

B. Local processing model

If a task $\tau_i(m)$ is processed at the user device u_i, the energy consumption and delay are calculated through the CPU cycle as below.

$$e_{i,u}^{local} = l_i^{local}.C_u.k.f_u^2 \tag{4}$$

$$t_{i,u}^{local} = \frac{l_i^{local}.C_u}{f_u} \tag{5}$$

Where $e_{i,u}^{local}$ is the energy consumption u_i for the l_i^{local} bits, f_u^2 is the CPU frequency, k is the capacitance constant, and C_u is the CPU cycle required to process the task at the microprocessor in a user's device. The symbols and definitions are illustrated in Table 1.

Table 1. Symbols and Definitions

Symbol	**Description**
α	The balancing factor of energy and delay
λ_i	The arrival rate of a user device
C_u, C_s	The CPU cycle at a user device and a MEC server
f_u, f_s	The CPU frequency at a user device and a MEC server
l_i	The length of the task τ_i
l_i^{local}	The bit length processed by a user device
l_i^{mec}	The bit length processed by the servers
$e_{i,u}^{local}$	Energy consumption of τ_i at u_i
$e_{i,j}^{mec}$	Energy transmits the consumption of the task τ_i
$e_{i,j}^{exe}$	Energy consumption executed task τ_i at the server s_j
p_u	The power transmission of the user device u_i
t_i^u	The latency time to process the task τ_i
t_i^{trans}	The transmission time
t_i^{exe}	The executed time at a MEC server
t_i^{local}	The executed time at a local user device
p_i^{off}	The offloading proportion of the task τ_i
Γ	A set of user device tasks
S	A set of mobile edge computing servers
U	A set of user devices

C. Offload processing model

In case of partial offloading problems, to ensure generality, we denote p_i^{off} the offloading proportion of the task τ_i. Otherwise, each MEC server has a probability of being the server that UD offloads the task to, as p_s^{prob}. The probability that a user device selects MEC servers is depended on the communication model and clustering strategy.

Considering the capability of the MEC server enough to provide computation service to multiple user devices, we can get the executed time for offloaded tasks as

$$e_{i,s}^{mec} = l_i^{mec}.C_s.k.f_s^2 = (p_i^{off} \times l_i).C_s.k.f_s^2 \tag{6}$$

$$t_{i,s}^{exe} = \frac{l_i^{mec}.C_s}{f_s} = \frac{(p_i^{off} \times l_i).C_s}{f_s} \tag{7}$$

Where f_s^2 is the CPU frequency at the MEC server, and C_s is a CPU cycle required to process the task at the server.

The transmission time of a task τ_i is calculated as follows,

$$t_{i,j}^{trans} = \frac{l_i^{mec}}{r_{i,j}^u} = \frac{p_i^{off} \times l_i}{r_{i,j}^u}. \tag{8}$$

The energy consumption of transmitting the data u_i to the MEC server s_j is shown below.

$$e_{i,j}^{mec} = p_s^{prob} \times p_i^{off} \times t_{i,j}^{trans} \tag{9}$$

Based on eq (4) and eq (9), the energy consumption of the user device, including locally computational energy and transmission energy, is performed in the form of:

$$e_i^u = e_{i,u}^{local} + e_{i,j}^{mec} \tag{10}$$

The total energy to process a task τ_i for one session is

$$e_i^s = e_{i,u}^{local} + e_{i,j}^{mec} + e_{i,s}^{exe} \tag{11}$$

The task of the user's device is executed in parallel local and remote at the MEC server, and the execution latency of τ_i is

$$t_i^u = \max\{t_{i,j}^{trans} + t_{i,j}^{exe}, t_{i,u}^{local}\}. \tag{12}$$

The ultimate optimization goal is to determine latency and user energy consumption with minimal offloading decisions. This problem can be summarized as the following optimization objective function.

$$\min C(p_i^{off}, p_u) = \min \sum_{i=1}^{n} \alpha.E_i + (1-\alpha).T_i \tag{13}$$

Or

$$\min C(p_i^{off}, p_u) = \min \sum_{i=1}^{n} \alpha\left(\frac{e_i^{\max} - e_i^u}{e_i^{\max}}\right) + (1-\alpha)\left(\frac{t_i^{\max} - t_i^u}{t_i^{\max}}\right) \tag{14}$$

subject to

c1. $l_i = l_i^{local} + l_i^{mec}$

c2. $0 < p_i^{off} \le 1$

c3. $0 < p_u \le p_u^{\max}$

c4. $t_i^u \le t_i^{\max}$

c5. $e_i^u \le e_i^{\max}$

c6. $0 < \alpha < 1$

We use metaheuristics algorithms such as GA and PSO to search for the optimal offloading ratio and transmission power to reach the minimum cost. The main results of the performance of these algorithms are presented in the next session.

V. EXPERIMENTAL RESULTS

We simulate and evaluate the performance of the GA and PSO algorithms in this section. Figure 4 depicts our simulation scenario, which is similar to the scenarios used by previous authors [10] [12]. In detail, typical input parameters such as simulation area, coverage, and location of MEC host BTSs are assumed by previous authors to ensure objectivity in algorithm performance comparison.

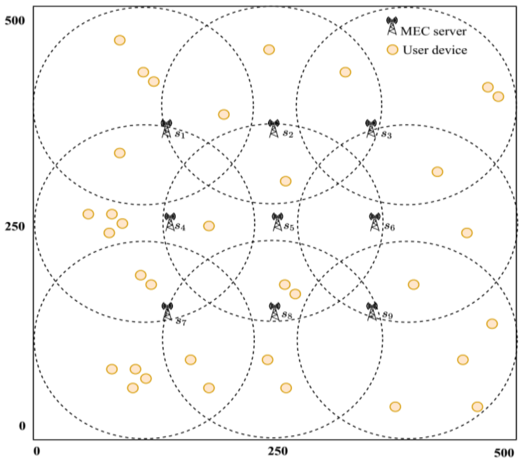

Figure 4. Simulation scenario

Table 2. Simulation parameters

Parameters	Value
The maximum iteration	50
Population size	25 50
Number of users	25 50
Maximum servers in range for a UD	3
Data volume of task (TaskSize)	[40Kb; 600Kb; 1000Kb]
BS signal coverage	250 m
Edge server CPU frequency	10GHz
BS bandwidth	10MHz
User device transmission power	20 dBm
Background noise	100 dBm

The basic parameters of the simulation are summarized in Table 2.

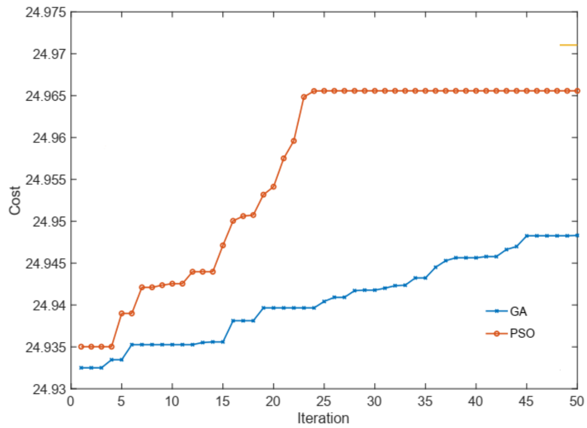

Figure 5. Cost vice versus iteration numbers

The algorithm's convergence is one of the most critical aspects of metaheuristics algorithms. Figure 5 depicts the convergence of the GA and PSO algorithms over several interactive loops. As can be seen, the PSO algorithm's convergence reached the saturated cost metric around 25 iterations. It brings a stable state more than that of the GA algorithm method.

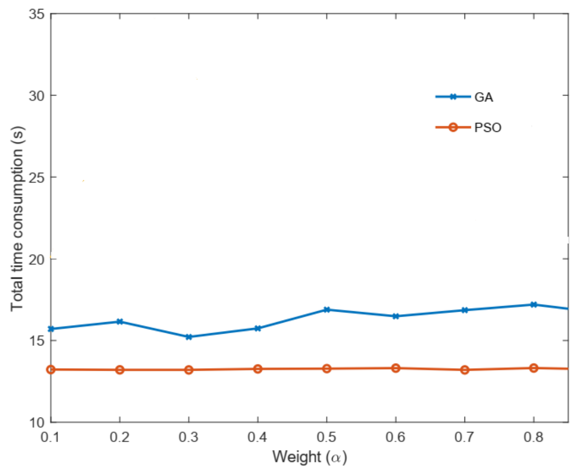

Figure 6. Total time consumption vice versus weight parameter (α)

Figure 6 demonstrates the total time consumption versus the weight parameter at the maximum iteration of 50. The weight parameter represents balancing the energy target and the delay. The results in Figure 6 show that the alpha parameter influences the time constraint through the amount of time consumed. The PSO algorithm outperforms the GA algorithm due to its proclivity to prioritize the tight deadlines of the input tasks. PSO algorithm spends the least amount of time compared to the GA algorithm. It can be seen that the total time consumption of metaheuristic algorithms, GA and PSO, does not depend much on the weight parameter.

Figure 7 shows the total energy consumption versus the weight parameter at the maximum iteration of 50. PSO has the best performance on the total energy consumption and keeps unchanged at the small value with the weight parameter. The GA algorithm's total energy consumption slightly fluctuates at a very high value. The gap in total energy consumption between PSO and GA reaches 50 J. In the fact that figure 7 shows the results from an energy perspective of a function of two energy and delay variables. In the general case, when the deadline is larger than the requirement task delays, the efficiency function becomes an optimal function of one energy variable. Hen, a good algorithm will give stable results in terms of adaptive energy with different weights. The results in the figure show that the response of PSO is better than GA in energy saving.

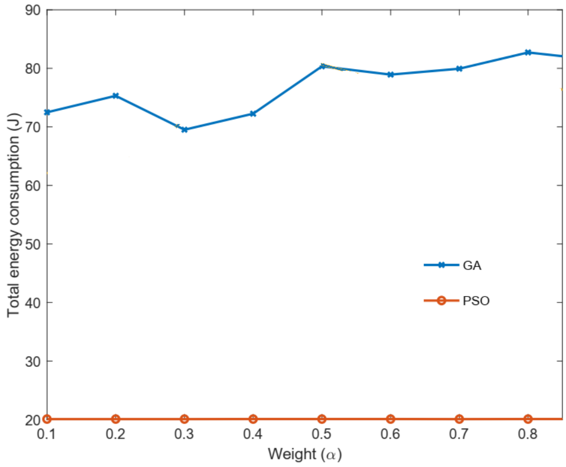

Figure 7. Total energy consumption vs. weight (α)

Thus, the impact of the weight parameter on the cost, total time consumption, and total energy consumption of three metaheuristic algorithms, including the GA and PSO

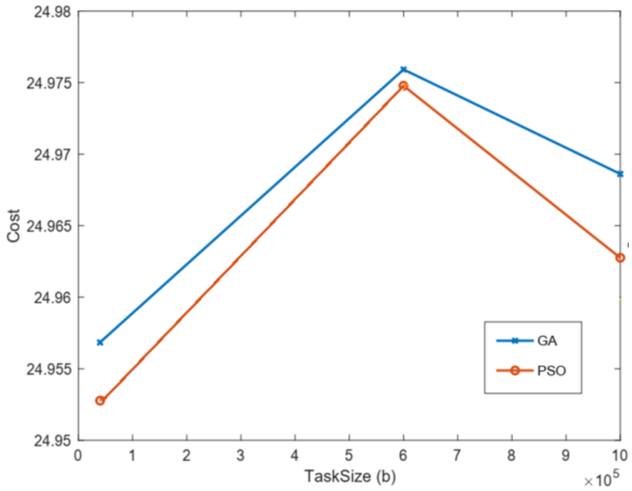

Figure 8. Cost vs. TaskSize

algorithm, is validated, showing that PSO exposes the efficiency in the total time consumption and total energy consumption but has a very high cost, while GA is reversed.

The cost depends on the data volume of the task of two metaheuristic algorithms with the maximum iteration of 50, which is illustrated in Figure 8. The cost performance of GA is the worst. All two algorithms show that the optimal efficiency value varies with the input packet size and tends to be similar. It shows that the size of different tasks directly affects network performance. This conclusion is beneficial for IoT application deployment models according to the data traffic provided by the sensors.

VI. CONCLUSION

This paper proposes a partial load offloading model with a multi-user multi-servers scenario for mobile edge computing. We propose the PSO algorithm to apply to the offloading problem not mentioned in the literature. The numerical results of our proposed method have been examined with the GA algorithm on the exact scenarios. The performance parameters of the proposed method in the analytical method have shown that algorithm convergence and stability with weight change during optimization reduce time complexity. We hope to implement practical test systems as part of our ongoing work in the future.

ACKNOWLEDGMENT

This work is supported by Project 02-HV-2022-VT1.

REFERENCES

[1] 5GPPP Architecture Working Group. "View on 5G Architecture: Version 2.0." (2017): 1-117.

[2] Series, M. "IMT Vision–Framework and overall objectives of the future development of IMT for 2020 and beyond." Recommendation ITU 2083 (2015): 21.

[3] Q. Qi, X. Chen, C. Zhong, and Z. Zhang, "Integration of energy, computation and communication in 6G cellular Internet of things," IEEE Comm. Letters, vol. 24, no. 6, pp. 1333–1337, 2020.

[4] Hu, Yun Chao, et al. "Mobile edge computing—A key technology towards 5G." ETSI white paper 11.11 (2015): 1-16.

[5] Liyanage, Madhusanka, et al. "Driving forces for multi-access edge computing (MEC) IoT integration in 5G." ICT Express 7.2 (2021): 127-137.

[6] Cao, Bin, et al. "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework." IEEE Communications Magazine 57.3 (2019): 56-62.

[7] Jehangiri, Ali Imran, et al. "Mobility-aware computational offloading in mobile edge networks: a survey." Cluster Computing 24.4 (2021): 2735-2756.

[8] Feng, Chuan, et al. "Computation offloading in mobile edge computing networks: A survey." Journal of Network and Computer Applications (2022): 103366.

[9] Sadatdiynov, Kuanishbay, et al. "A review of optimization methods for computation offloading in edge computing networks." Digital Communications and Networks, 2022.

[10] Liao, Zhuofan, et al. "Adaptive offloading in mobile edge computing for ultra-dense cellular networks based on genetic algorithm." Journal of Cloud Computing 10.1 (2021): 1-16.

[11] Kavyashree, S., and H. A. Chaya Kumari. "Survey on Computation Offloading Strategies in Cellular Networks with Mobile Edge Computing." Data Intelligence and

Cognitive Informatics. Springer, Singapore, 2022. 567-575.

[12] Wang, Jin, et al. "A probability preferred Priori offloading mechanism in mobile edge computing." IEEE Access 8 (2020): 39758-39767.

[13] Li, Shun, et al. "Computation Offloading Strategy for Improved Particle Swarm Optimization in Mobile Edge Computing." 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA). IEEE, 2021.

[14] Zhou, Wenqi, et al. "Offloading strategy with PSO for mobile edge computing based on cache mechanism." Cluster Computing (2021): 1-13.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. Int. Conf. Neural Netw. (ICNN), vol. 4, 1995, pp. 1942–1948.

[16] J. Holland, Adaptation in Natural and Artificial Systems. Ann Arbor, MI, USA: Univ. of Michigan, 1975.

[17] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh and S. Mirjalili, "Particle Swarm Optimization: A Comprehensive Survey," in IEEE Access, vol. 10, pp. 10031-10061, 2022,

MỘT LƯỢC ĐỒ HIỆU QUẢ CHO BÀI TOÁN GIẢM TẢI ĐIỆN TOÁN BIÊN DI ĐỘNG DỰA TRÊN THUẬT TOÁN PSO

Tóm tắt: Mobile Edge Computing (MEC) đóng vai trò then chốt trong mục tiêu tích hợp Internet of things với các công nghệ nền tảng 5G và beyond. Không chỉ đóng vai trò như một phần kéo dài của điện toán đám mây cho mục tiêu phân tán dữ liệu, MEC cung cấp khả năng xử lý cục bộ, đảm bảo tính riêng tư, nâng cao tính bảo mật và tăng độ tin cậy của toàn hệ thống. Trong đó, bài toán giảm tải hệ thống đóng vai trò then chốt trong các mô hình xử lý dữ liệu, tính toán hay bảo mật. Tuy nhiên, bài toán này thuộc dạng NP-Hard do đặc tính đa mục tiêu và đa ràng buộc của chúng. Cụ thể, các nhiệm vụ yêu cầu giảm tải cần đạt đồng thời hai mục tiêu là năng lượng và độ trễ thực thi. Vì vậy, tiếp cận heuristics đã và đang là một xu hướng hiệu quả cho cả mục tiêu nghiên cứu lẫn triển khai. Nghiên cứu này sẽ trình bày một lược đồ giảm tải dựa trên thuật toán PSO để vượt qua các thách thức hiện nay. Các kết quả mô phỏng số trong bài báo này sẽ chứng minh đề xuất của chúng tôi hiệu quả hơn đề xuất sử dụng thuật toán di truyền của các tác giả khác.

Từ khóa: Điện toán biên di động, bài toán giảm tải, các thuật toán kinh nghiệm, tối ưu bầy đàn.

Hoang Trong Minh received a bachelor's degree in physic engineering (1994) and electronic and telecoms engineering (1999) from HUST, a master's degree in electronic and telecommunication engineering (2003), and a Ph.D. degree in telecommunication engineering (2014) from PTIT. His research interests include routing, security, and network performance in mobile edge computing, flying ad hoc networks, wireless mobile networks, and beyond 5G.

Email: hoangtrongminh@ptit.ed.vn

Nguyen Thanh Tra received the Diploma of Engineer in Electronic Engineering and Telecommunication Technologies from Hanoi University of Science and Technology (HUST), and the Master of Engineering in Electronics and Telecommunications from Posts and Telecommunications Institute of Technology (PTIT) in 2000 and 2006, respectively. Currently, she is the senior lecturer at the Telecommunication Faculty 1 of PTIT

Email: trant@ptit.ed.vn