A Hybrid Meta-heuristic Algorithm for Edge Site Deployment with User Coverage Maximization and Cost Minimization

Xiaodong Xing[1],Ying Song[2], Bo Wang[3]
School of Computer and Network Engineering, Shanxi Datong University, Shanxi, China 037009[1]
Computer School, Beijing Information Science and Technology University, Beijing, China 100192[2]
Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou, China 450001[3]

***Abstract*—Recent years, edge computing has been getting increased attention due to its ultra-low delay service deliveries. Plenty of works have focused on the performance improvement of edge computing by e.g., edge server deployment, edge caching, and task offloading. While, there is a lack of work on improving the investment cost for building or upgrading the edge site deployment by making a decision on which places edge sites are deployed. In this paper, we focus on the edge site deployment problem (ESDP) to maximize user coverage with fewest edge sites. We first formulate ESDP into a binary nonlinear programming with two optimization objectives of user coverage maximization and edge site minimization, and prove that ESDP is NP-complete. Then, we propose a hybrid meta-heuristic algorithm to solve ESDP with polynomial time complexity, which combining the crossover and mutation operators of genetic algorithm with self- and social-cognition of particle swarm optimization. At last, we conduct extensive simulated experiment based on a real data set to evaluate the performance of our proposed algorithm. The results show that our algorithm achieves 100% user coverage with much fewer edge sites than other seven meta-heuristic algorithms, and has a good scalability.**

***Keywords*—Edge computing; edge deployment; GA; PSO; meta-heuristic**

I. Introduction

Over the past two decades, cloud computing has been applied in all fields, due to its numerous benefits, e.g., on-demand, flexibility, reliability [1]. But in recent years, cloud computing alone cannot satisfy real time requirements of many user requests [2], especially in mobile networks. This is mainly because cloud resources are shared by users around the world and cloud computing platforms provide services over Wide Area Networks (WAN) that generally have high latencies. In addition, more and more users request services by mobile devices nowadays [3] with the rapid development of communication and network technologies, which results in a highly dynamic locations where requests are initiated and thus a much fluctuating communication performance between users and cloud computing platforms.

Therefore, in recent years, edge computing is becoming more and more popular in both industry and academia, because it can efficiently compensate for the shortcomings of cloud computing [4], [5]. In edge computing, some computing and storage resources are deployed close to user devices, and thus ultra-low latency services can be provided for users. While, due to the distribution and heterogeneity of limited edge resources, it is a very challenging work to provide high quality services for all users. Many works have focused on addressing the challenge in various aspects including edge server placement [6], [7], edge caching [8], [9], task offloading [10], [11] and so on. While, all of works assume that edge sites have been deployed, where edge resources are placed and used for processing requests. For each request, it can be accepted by closely located edge sites, because edge sites provide Local Area Network (LAN) connections for users usually by wireless networks in a short distance. Thus, the locations of edge sites have an effect on the service quality, by deciding which edge resources can be used for processing every request. Therefore, in this paper, we focus on the edge site deployment problem (ESDP) that is deciding which of multiple candidate places to be edge sites.

For a user, if it is not covered by the network signal of any edge site, then it cannot communicate with any edge site. In such case, the user's requests cannot received by an edge site, and thus cannot be processed by edge resources, which can cause major performance degradation of these requests. Therefore, in this paper, we identify maximizing user coverage as a major objective, which is maximizing the number of users that are covered by the network signal of at least one edge site. As the profit maximization is the first aim of service providers, in this paper, we consider the deployment cost as the second optimization objective by minimizing the number of deployed edge site.

Due to geographical characteristics and urban planning, candidate places generally are dispersive. This leads to that ESDP is a discrete optimization problem and hard to be solved precisely. Therefore, in this paper, we exploit meta-heuristic algorithms for solving ESDP, due to their powerful search ability [12]. To be specific, we consider to exploit Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), which are most representative evolutionary algorithm and swarm intelligence, respectively. GA, inspired by Darwin's theory of evolution, has powerful global search ability by evolutionary operators but usually convergences slowly [13]. PSO is designed based on the movements of birds for foraging, which has high convergence speed but is easily trapping into local best positions [14]. Thus, GA and PSO are complement each other, and we design a hybrid meta-heuristic algorithm by combining their advantages for providing a ESDP solution with maximized user coverage and minimized monetary cost. In brief, the contributions of this paper are as followings.

- First, ESDP is formulated as a binary non-linear programming, and its hardness is proven. The objectives include the user coverage maximization and the monetary cost minimization.
- Then, a hybrid meta-heuristic algorithm is proposed for solve ESDP in polynomial time complexity by combining GA and PSO. The hybrid algorithm uses the swarm evolving framework of PSO to exploit the self- and social-cognitions, and employs the mutation and crossover operators of GA to ensure the population diversity for a powerful global search ability.
- At last, the performance of proposed algorithm is evaluated by extensive simulated experiments that are designed based on a real data set. Experiment results show that the proposed algorithm achieves 100% user coverage and requires much less edge sites than several other meta-heuristic and hybrid algorithms.

The rest of this paper is organized as followings. Section II presents the formulation of ESDP. Section III illustrates the proposed meta-heuristic algorithm, and section IV shows the performance evaluation results. Section V discusses the related works. At last, Section VI concludes this work.

II. Problem Statement

In this paper, we consider to select part of multiple candidate places to deploy edge sites for providing edge computing services. Assuming that there are P candidate places represented as $p_i, 1 \leq i \leq P$. For candidate place p_i, its location is $(l^P_{i,1}, l^P_{i,2})$, which can be either latitude and longitude in geographic coordinate or horizontal and vertical values in Cartesian coordinate. When an edge site is decided to be deployed on p_i, there will be m_i monetary cost. For each edge site, the maximal distance of network signal is S. This is meaning that a user is covered by an edge site or a candidate place if and only if their distance is smaller than S. The edge site deployment decisions can be indicated by binary variables $x_i, 1 \leq i \leq P$, where $x_i = 1$ means p_i is selected for deploying an edge site, and $x_i = 0$ means not. In this case, the total monetary cost C can be calculated by Eq. (1).

$$C = \sum_{i=1}^{P} (m_i x_i) \tag{1}$$

In the considered edge computing, there are U users ($u_j, 1 \leq i \leq U$). The location of u_j is $(l^U_{j,1}, l^U_{j,2})$. Then, the distance (d) between users and candidate places can be calculated by Eq. (2) and (3), respectively, when using geographic and Cartesian coordinate systems. Where R is the earth radius, which is usually set as 6371.393 kilometres.

$$\begin{aligned} d_{i,j} = \quad & R \arccos(\cos l^U_{i,1} \cos l^P_{j,1} \cos(l^P_{i,2} - l^U_{j,2}) \\ & + \sin l^P_{i,1} \sin l^U_{j,1}) \end{aligned} \tag{2}$$

$$d_{i,j} = \sqrt{(l^P_{i,1} - l^U_{j,1})^2 + (l^P_{i,2} - l^U_{j,2})^2} \tag{3}$$

Then, the cover between users and candidate places can be achieved by Eq. (4), where $c_{i,j} = 1$ represents u_j is covered by p_i, and u_j can be served by the edge site deployed on p_i during the operation of edge computing. And, we can get the set of covered users for each candidate places, $\mathbb{C}_i = \{u_j | c_{i,j} = 1\}$, and the set of all covered tasks by selected candidate places, $\mathbb{C} = \cup_{x_i=1} \mathbb{C}_i$. Now, we can calculate the overall user coverage by Eq. (5).

$$c_{i,j} = \begin{cases} 1, & \text{if } d_{i,j} \leq S \\ 0, & \text{if } d_{i,j} > S \end{cases} \tag{4}$$

$$Q = \frac{|\mathbb{C}|}{U} \times 100\% \tag{5}$$

Based on above formulations, we can now model ESDP as following optimization problem. The two objectives are maximizing the user coverage (Eq. (6)) and minimizing the cost of deployed edge sites (Eq. (7)) , respectively. Decision variables include $x_i, 1 \leq i \leq P$, which are all binary. As the nonlinear of Eq. (5), the ESDP is a binary nonlinear programming.

$$\text{maximizing} \quad Q, \tag{6}$$

$$\text{minimizing} \quad C. \tag{7}$$

In this paper, we consider coverage maximization as the major objective. Then, the two objectives Eq. (6) and Eq. (7) can be convert into one, as shown in Eq. (8). $UQ = |\mathcal{C}|$ is the number of covered users. $\sum_{i=1}^{P} m_i$ is the total cost when all candidate places are selected for deploying edge sites, which is greater than or equal to C, and thus $\frac{C}{\sum_{i=1}^{P} m_i} \leq 1$.

$$\text{maximizing } O = UQ - \frac{C}{\sum_{i=1}^{P} m_i} \tag{8}$$

Next, we proof that ESDP is NP-Complete, which means no polynomial algorithm can exactly solve it unless P=NP, and in the next section, we will present a hybrid meta-heuristic algorithm to solving it in a polynomial algorithm with global search abilities of GA and PSO. Considering an instance of ESDP, all user can be covered and costs of edge site deployments on all candidate places are identical. Then, the ESDP instance is to minimize the number of deployed edge sites with 100% user coverage, which can be formulated following optimization problem. Objective (9) is minimizing the number of selected candidate places for deploying edge sites. Constraints (11) require that every user must be covered by at least one selected candidate place or deployed edge site. Constraints (11) represent that decision variables are binary. Therefore, the ESDP instance is binary linear programming which has been proven as NP-complete [15]. Thus, ESDP is NP-complete.

$$\text{minimizing} \sum_{i=1}^{P} x_i, \tag{9}$$

subject to,

$$\sum_{i=1}^{P} c_{i,j} x_i \geq 1, 1 \leq j \leq U, \tag{10}$$

$$x_i \in \{0, 1\}, 1 \leq i \leq P. \tag{11}$$

III. Hybrid GA and PSO for Edge Site Deployment

In this section, we propose a hybrid meta-heuristic algorithm aiming to search a global best solution for ESDP. In this paper, we choose to employ GA and PSO, and will exploit other meta-heuristics for more powerful global search ability and efficiency. The reasons of choosing GA and PSO are two-fold. One is that both GA and PSO are most representative meta-heuristic algorithms and have widely used in various fields due to their good performance and easily implementations [14], [16]. Another is that one's advantage can make up another's disadvantage for GA and PSO, as GA has powerful global search ability but slow convergence speed, and on the contrary, PSO has fast convergence speed but is easily trapping into local optima. Fig. 1 gives the flow chart of our proposed hybrid meta-heuristic algorithm for ESDP, which is represented by PGSAO.

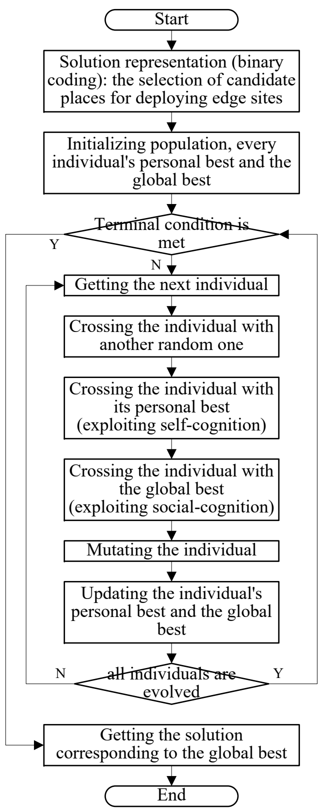

Fig. 1. The algorithm flow chart of the hybrid GA and PSO for edge site deployment.

As shown in Fig. 1, at first, PGSAO employs a binary coding method to establish the solution space for meta-heuristic algorithms' searching. A solution of ESDP is represented as a P-dimensional vector, $\langle x_1, x_2, ..., x_P\rangle$, where values in all dimensions are binary, indicating whether edge sites are deployed on the corresponding places. During the search, there is a fitness function used for evaluating the goodness for each individual/solution. In this paper, we identify user coverage maximization as the major objective and deployed edge site number as the minor objective. Thus, the fitness function of PGSAO is defined as Eq. (12).

$$f = |\mathbb{C}| + \frac{\sum_{i=1}^{P} x_i}{P}. \tag{12}$$

Given the solution space and the fitness function, PGSAO first initializes a population consisting of multiple individuals (chromosomes, particles), where every individual is a solution, by randomly setting a value on each dimension of every individual. Then, for each individual, PGSAO evaluates its fitness, and records itself as its personal best. After the fitness evaluation, PGSAO finds the individual with the highest fitness, and records it as the global best. Now, PGSAO proceeds to the evolutionary stage to upgrade individuals for retrieving better or even global best solutions.

In the evolutionary stage, PGSAO repeats upgrading the population by performing crossover and mutation operators on every individual as following steps until the terminal condition is met. (1) The individual is crossed with another one that is selected randomly with the crossover probability, which is same to GA. (2) The individual is crossed with its personal best with the crossover probability, which is exploiting the self-cognition of PSO. (3) The individual is crossed with the global best with the crossover probability for employing the social-cognition of PSO. (4) The individual is mutated with the mutation probability, as done by GA for increasing the population diversity. (5) For each individual, by three crossover operators (steps 1, 2, and 3) and one mutation operator (step 4), total seven offspring are produced (two for a crossover and one for a mutation). At the end of each individual's evolution, its personal best and global best are updated as the best offspring when an offspring has better fitness.

In this paper, to ensure the population diversity, PGSAO employs the uniform crossover and the uniform mutation operators. The uniform crossover operator is to swap the values on every dimension of two individuals with a probability, which represents exchanging the selection states of a candidate place between two solutions. By the uniform mutation operator for an individual, each dimension is changed from one value (0/1) to another one (1/0) with a probability, which represents changing the selections of candidate places on the solution.

PGSAO is terminated when the repeat time reaches the predefined threshold or there is no change on the global best a few times continuously. And PGSAO returns the solution corresponding to the global best.

IV. Performance Evaluation

To evaluate the performance of PGSAO, we conduct extensive simulated experiments based on a real data set, EUA [17], [18], which includes locations of 9318 LTE base station sites and 131312 users in the Melbourne CBD area. We consider LTE base station sites as candidate places in ESDP. In our experiment, we convert the geographic coordinate into a Cartesian coordinate system by setting one degree as 1000 metres in both longitude and latitude for just simplifying the calculation of distances, and set the coverage of each edge site as 100 metres referring to existing related works.

The performance metrics used for evaluating PGSAO include the user coverage and the number of selected candidate places for edge site deployment. For the first metric, it is better for a greater value, and 100% is the best value. For the second one, a smaller number is better, which indicates a lower cost for edge site deployment.

To prove the superiority of our method, we compare PGSAO with five classical and widely used meta-heuristic algorithms, GA [16], Differential Evolution (DE) [19], Artificial Bee Colony (ABC) [20], PSO [14] and Multi-Verse Optimizer (MVO) [21], and two hybrid meta-heuristic algorithms (GAPSO [22] and PSOGA) in solving ESDP. GAPSO is to perform GA in the first half of the evolutionary stage, and PSO in the second half. PSOGA reverses the order of GA and PSO for GAPSO.

A. Overall Performance

For each experiment, we repeat eight times and show results by the box-plot. In our experiment, all algorithms achieve 100% user coverage, which verifies the powerful search ability of meta-heuristics. In Fig. 2, we present the number of deployed edge sites when applying various algorithms. From the figure, we can see that PGSAO requires the fewest edge sites for full user coverage, and thus the minimal deployment cost. On average, PGSAO requires to deploy 16.0%–31.7% fewer edge sites than other methods. This verifies that the performance superiority of PGSAO in the global optimization, compared with other [hybrid] meta-heuristic algorithms. This is mainly because of the efficient fusion of GA and PSO in the following two aspects. First, hybrid meta-heuristic algorithms, GAPSO, PSOGA, and PGSAO, achieve better performance than single meta-heuristic algorithms, GA, DE, ABC, PSO, and MVO, in minimizing the number of deployed edge sites, which verifies the validity of combining different meta-heuristic algorithms for a better performance. Second, PGSAO needs fewer deployed edge sites than GAPSO and PSOGA. This proves that the high efficiency of the combination strategy exploited by PGSAO, as GAPSO, PSOGA, and PGSAO are all combining GA and PSO.

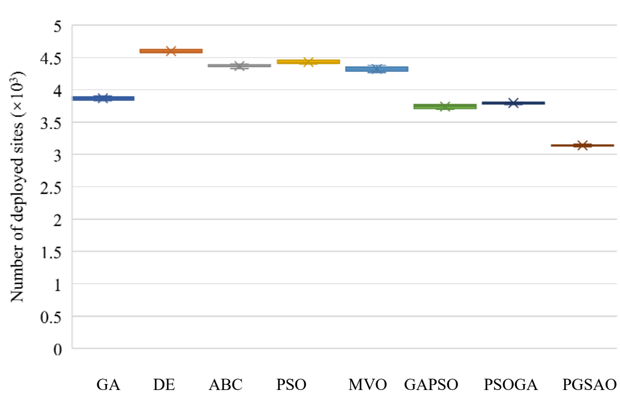

Fig. 2. The number of deployed edge sites for user coverage maximization when applying various algorithms.

We also perform t-test to verify the statistical difference of our method to others in optimizing deployed edge site number, and present the test results in Table I. As shown in the table, the p-values of all t-test are much smaller than 0.01. Thus, PGSAO has significantly different performance to other algorithms, which confirms the superiority of PGSAO further.

TABLE I. p-VALUES OF T-TEST ON THE STATISTICAL EQUALITY OF PGSAO TO OTHER ALGORITHMS

Method	p-value
GA	1.33×10^{-15}
DE	2.60×10^{-20}
ABC	3.41×10^{-20}
PSO	9.72×10^{-19}
MVO	3.89×10^{-14}
GAPSO	1.32×10^{-12}
PSOGA	7.94×10^{-21}

In the next, we compare the time consumed by various algorithms, where the results are shown in Fig. 3. The time is tested on a personal computer with Window 11, Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz, and 24 GiB RAM. As shown in the figure, PGSAO consumes about five times time of others, where the increased time is mainly consumed for exploiting self- and social cognitions by crossover operators of PGSAO. Even so, it is absolutely a worthwhile trade-off for a much lower deployment cost, noticing the following observation. The edge deployment strategy is stable. There are mainly two cases in which service providers need edge deployment solutions, building or upgrading edge computing on some new or original areas when providers decide to develop new business or user traffics are increased greatly. For both cases, it generally takes a long time, e.g., months or years, for changing the edge deployment solution. Thus, service providers prefer to consume more hours for an edge deployment solution with much lower cost.

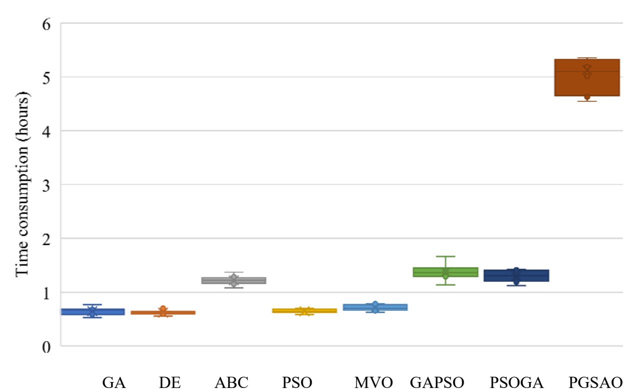

Fig. 3. The time consumed by different algorithms.

B. Performance Varied with User Number

Now, we exam the performance variations with the user number for different algorithms. Fig. 4 and 5 give such variations when the user number is changed from 10,000 to 100,000. The user coverage is always 100% for every algorithm in any case. As shown in Fig. 4, we can see that the number of required edge site for the full user coverage is stable with varied user number for every algorithm. This is mainly because the area size instead of the user number determine the number of required edge site for a full coverage, given the fixed coverage range of each edge site. From Fig. 4, PGSAO always requires minimal number of edge sites than others, and thus has the best performance in minimizing the deployment cost.

As shown in Fig. 5, we can see that every algorithm consumes time increased linearly with the increase of the user number in overall. This is mainly because all users are traversed once for calculating the fitness for each individual. This verifies that all algorithms including ours have good scalability in solving various scale ESDP, and thus have good usability.

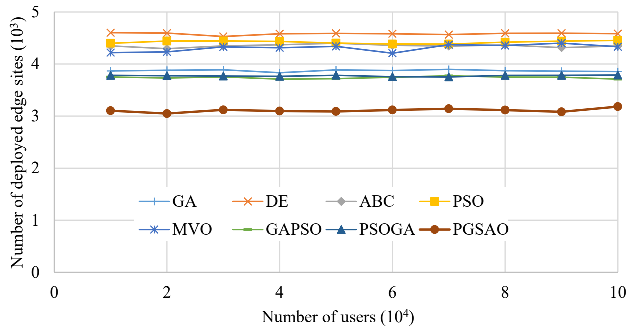

Fig. 4. The number of deployed edge sites required by various algorithms with varied user numbers.

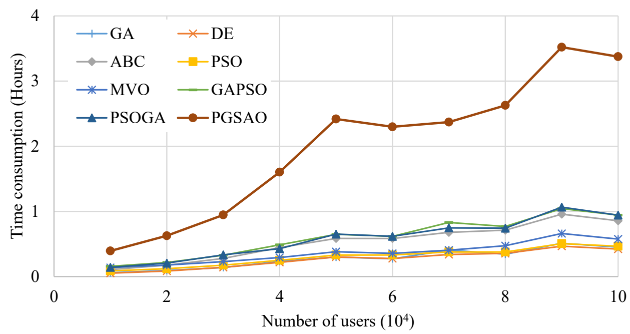

Fig. 5. The time consumed by different algorithms with varied user numbers.

C. Performance Varied with Candidate Site Number

We also evaluate the performance variations with the candidate place number for each algorithm on solving ESDP, and present results in Fig. 6 and 7, where the number of candidate places is in the range from 1000 to 9000. Same to previous results, all algorithms can achieve 100% user coverage, confirming the effectiveness of their search strategies. From Fig. 6, we can see that as the number of candidate places increases, more edge sites are required for deployment for every algorithm, i.e., the performance is degraded. This is mainly because as the number of candidate places increases, the search space is exponentially increased, and thus it is more and more difficult to retrieve the global best solution.

As shown in Fig. 7, the consumed time is stable as the number of candidate places is varied for every algorithm, as the main time overhead is consumed for the population evolution and thus mainly decided by population size and iteration number. This phenomenon further confirms the good scalability of our algorithm.

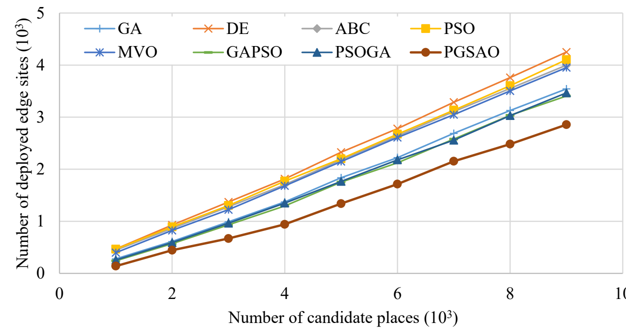

Fig. 6. The number of deployed edge sites required by various algorithms with varied candidate places.

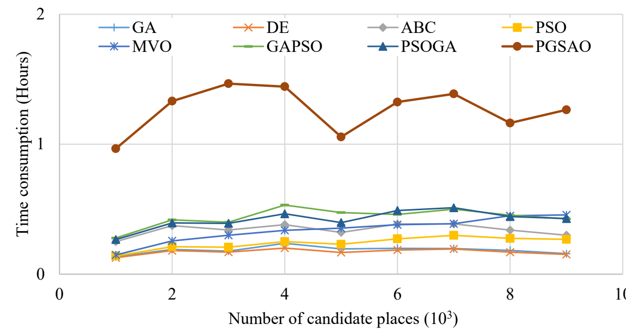

Fig. 7. The time consumption changed with varied number of candidate places for each algorithm.

V. Related Works

Edge-cloud computing has attract more and more attention from both industry and academia due to its huge advantage by combing edge and cloud computing. Several works have aimed at designing edge server placement approaches, edge service/data caching strategies, and task offloading algorithms to improve the efficiency and effectiveness of edge-clouds, which are discussed as follows, respectively.

Given the edge site deployment solution, edge server placement is to decide which sites edge servers (resources) are placed to maximize the overall performance with restricted edge resources. Zhang et al. [23] proposed a niche PSO to minimize the overall respond time considering the placement as a multimodal optimization problem, which divides similar individuals into a niche during the population upgrade. Li et al. [24] and Zhang et al. [25] employed K-means++ to cluster edge sites into k classes, and deploy k edge servers on the sites closest to theses classes' centres. These works assumed that all edge sites have been deployed, and thus our work is complementary to them. These works only considered to improve request processing performance with fixed number of edge servers, without concerning the investment cost improvement for building edge computing platforms.

During operation of edge computing, as resources configured in each edge site are restricted, there is no enough room for storing all services or data requested by users in the edge. Therefore, an edge caching strategy decides which service data that are deployed (cached) on every edge site for processing corresponding requests, and affect the performance by determining whether or not a request can be processed in edge servers. There mainly two kinds of edge caching, static and dynamic strategies. Static strategies provide a solution that the placement of services or data on edge sites is not changed, e.g., [26], [27], [28], while dynamic strategies adjust the placement over time, e.g., [29], [30], [31].

Another key work to tune the performance of operating edge computing is task offloading/scheduling that decides the resource that every request task is processed on. Laboni et al. [32] proposed a two-layer hyper heuristic algorithm to determinate the server and router path for each request's processing for optimizing processing delay and load balance. They exploited ant colony optimization as the high-level algorithm for selecting the low-level algorithm employed for each population evolution, and Whale optimization, sine-cosine algorithm as well as Henry gas solubility optimization as low-level algorithms. Zhang and Yu [33] proposed a hybrid task offloading algorithm by combining ABC and PSO, to improving processing delay and energy consumption. Concerned about service caching, Zhang et al. [34] formulated the task offloading problem into a mixed-integer non-linear programming, and exploited non-cooperative game to iteratively solving the problem by the interaction between wireless characteristics and mobile users. Both edge caching and task offloading are used for improving performance during the operation of edge computing, and thus requires that the edge sites and resources have been deployed. Therefore, our work focusing on ESDP is complement with these existing related works.

VI. Conclusion

In this paper, we focus on the edge site deployment problem (ESDP) to improve the build/upgrade cost to provide ESDP solution for service providers to expand their businesses. To achieve this goal, we first formulate ESDP into a binary non-linear programming problem with two objectives, user coverage maximization and edge site number minimization. And we prove the NP-complete of ESDP by establishing an ESDP instance that is binary linear programming. Then, we propose a hybrid GA and PSO algorithm aiming to retrieve the global best solution of ESDP. The proposed algorithm combines the global search ability of GA and cognitions exploited by PSO, to achieve a more efficient search strategy. In the end, we evaluate the performance of our proposed algorithm based on a real data set, and confirm the superiority of our algorithm in various aspects.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. The research was supported by the key scientific and technological projects of Henan Province (Grant No. 232102211084, 222102210137), the Natural Science Foundation of Henan (Grant No. 222300420582), the National Natural Science Foundation of China (Grant No. 61872043, 61975187, 62072414), Qin Xin Talents Cultivation Program, Beijing Information Science and Technology University (No. QXTCP B201904), and the fund of the Beijing Key Laboratory of Internet Culture and Digital Dissemination Research (Grant No. ICDDXN004).

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, p. 50–58, Apr 2010. [Online]. Available: https://doi.org/10.1145/1721654.1721672

[2] T. Wang, Y. Liang, X. Shen, X. Zheng, A. Mahmood, and Q. Z. Sheng, "Edge computing and sensor-cloud: Overview, solutions, and directions," *ACM Computing Surveys*, vol. 55, no. 13s, Jul 2023. [Online]. Available: https://doi.org/10.1145/3582270

[3] X. Wang, J. Li, Z. Ning, Q. Song, L. Guo, S. Guo, and M. S. Obaidat, "Wireless powered mobile edge computing networks: A survey," *ACM Comput. Surv.*, vol. 55, no. 13s, jul 2023. [Online]. Available: https://doi.org/10.1145/3579992

[4] S. Ahmad, I. Shakeel, S. Mehfuz, and J. Ahmad, "Deep learning models for cloud, edge, fog, and iot computing paradigms: Survey, recent advances, and future directions," *Computer Science Review*, vol. 49, p. 100568, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013723000357

[5] B. Wang, C. Wang, W. Huang, Y. Song, and X. Qin, "A survey and taxonomy on task offloading for edge-cloud computing," *IEEE Access*, vol. 8, pp. 186 080–186 101, 2020.

[6] B. Bahrami, M. R. Khayyambashi, and S. Mirjalili, "Edge server placement problem in multi-access edge computing environment: models, techniques, and applications," *Cluster Computing*, vol. 26, no. 5, p. 3237–3262, 2023.

[7] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731518304398

[8] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2525–2553, 2019.

[9] M. Reiss-Mirzaei, M. Ghobaei-Arani, and L. Esmaeili, "A review on the edge caching mechanisms in the mobile edge computing: A social-aware perspective," *Internet of Things*, vol. 22, p. 100690, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660523000136

[10] Y. Sang, J. Wei, Z. Zhang, and B. Wang, "A hybrid heuristic service caching and task offloading method for mobile edge computing," *Computers, Materials & Continua*, vol. 76, no. 2, pp. 2483–2502, 2023. [Online]. Available: http://www.techscience.com/cmc/v76n2/54015

[11] B. Wang, Z. Zhang, Y. Song, M. Chen, and Y. Chu, "Application of quantum particle swarm optimization for task scheduling in device-edge-cloud cooperative computing," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107020, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197623012046

[12] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, no. 4, p. 2191–2233, 2019.

[13] S. Mirjalili, "Genetic algorithm," *Evolutionary Algorithms and Neural Networks: Theory and Applications*, pp. 43–55, 2019. [Online]. Available: https://doi.org/10.1007/978-3-319-93025-1_4

[14] J. Nayak, H. Swapnarekha, B. Naik, G. Dhiman, and S. Vimal, "25 years of particle swarm optimization: Flourishing voyage of two decades," *Archives of Computational Methods in Engineering*, vol. 30, no. 3, p. 1663–1725, 2023.

[15] E. M. Gurari and O. H. Ibarra, "An np-complete number-theoretic problem," in *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, ser. STOC '78. New York, NY, USA: Association for Computing Machinery, 1978, p. 205–215. [Online]. Available: https://doi.org/10.1145/800133.804349

[16] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 4, p. 8091–8126, 2021.

[17] P. Lai, Q. He, M. Abdelrazek, F. Chen, J. Hosking, J. Grundy, and Y. Yang, "Optimal edge user allocation in edge computing with variable sized vector bin packing," in *16th International Conference on Service-Oriented Computing (ICSOC 2018)*, ser. ICSOC 2018, 2018, pp. 230–245. [Online]. Available: https://doi.org/10.1145/800133.804349

[18] Swin Edge, "EUA datasets: Edge server, user dataset for edge computing research," https://github.com/swinedge/eua-dataset, 2021.

[19] Bilal, M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, "Differential evolution: A review of more than two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103479, 2020.

[20] K. Thirugnanasambandam, M. Rajeswari, D. Bhattacharyya, and J. yoon Kim, "Directed artificial bee colony algorithm with revamped search strategy to solve global numerical optimization problems," *Automated Software Engineering*, vol. 29, no. 13, pp. 1–31, 2022.

[21] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, p. 495–513, 2016.

[22] H. Hafsi, H. Gharsellaoui, and S. Bouamama, "Genetically-modified multi-objective particle swarm optimization approach for high-performance computing workflow scheduling," *Applied Soft Computing*, vol. 122, p. 108791, 2022.

[23] X. Zhang, J. Zhang, C. Peng, and X. Wang, "Multimodal optimization of edge server placement considering system response time," *ACM Transactions on Sensor Networks*, vol. 19, no. 1, dec 2022. [Online]. Available: https://doi.org/10.1145/3534649

[24] W. Li, J. Chen, Y. Li, Z. Wen, J. Peng, and X. Wu, "Mobile edge server deployment towards task offloading in mobile edge computing: A clustering approach," *Mobile Networks and Applications*, vol. 27, no. 4, p. 1476–1489, Aug 2022.

[25] H. Zhang, J. Zhao, L. Yang, and Z. Zhang, "Mobile edge computing servers deployment with improved genetic algorithm in cellular internet of things," *China Communications (Early Access)*, 2023.

[26] J. Zhou, F. Chen, Q. He, X. Xia, R. Wang, and Y. Xiang, "Data caching optimization with fairness in mobile edge computing," *IEEE Transactions on Services Computing*, vol. 16, no. 3, pp. 1750–1762, 2023.

[27] S. Tang, K. He, L. Chen, L. Fan, X. Lei, and R. Q. Hu, "Collaborative cache-aided relaying networks: Performance evaluation and system optimization," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 3, pp. 706–719, 2023.

[28] X. Wei and Y. Wang, "Popularity-based data placement with load balancing in edge computing," *IEEE Transactions on Cloud Computing*, vol. 11, no. 1, pp. 397–411, 2023.

[29] C.-K. Huang and S.-H. Shen, "Enabling service cache in edge clouds," *ACM Transactions on Internet of Things*, vol. 2, no. 3, jul 2021. [Online]. Available: https://doi.org/10.1145/3456564

[30] T. Zong, C. Li, Y. Lei, G. Li, H. Cao, and Y. Liu, "Cocktail edge caching: Ride dynamic trends of content popularity with ensemble learning," *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, p. 208–219, aug 2023. [Online]. Available: https://doi.org/10.1109/TNET.2022.3193680

[31] S. Zhou, Z. Wang, C. Hu, Y. Mao, H. Yan, S. Zhang, C. Wu, and W. Zhu, "Caching in dynamic environments: A near-optimal online learning approach," *IEEE Transactions on Multimedia*, vol. 25, pp. 792–804, 2023.

[32] N. M. Laboni, S. J. Safa, S. Sharmin, M. A. Razzaque, M. M. Rahman, and M. M. Hassan, "A hyper heuristic algorithm for efficient resource allocation in 5g mobile edge clouds," *IEEE Transactions on Mobile Computing (In Press)*, pp. 1–13, 2022.

[33] W. Zhang and J. Yu, "Task offloading strategy in mobile edge computing based on cloud-edge-end cooperation," *Journal of Computer Research and Development (In Chinese)*, vol. 60, no. 2, pp. 371–385, 2023.

[34] X. Zhang, W. Wu, C. Zhang, Y. Cai, S. Yang, and X. Wang, "Energy-efficient computing offloading algorithm for mobile edge computing network," *Journal of Software (In Chinese)*, vol. 34, no. 2, pp. 849–867, 2023.