World Academy of Science, Engineering and Technology
International Journal of Information and Communication Engineering
Vol:17, No:2, 2023

AI-Based Approaches for Task Offloading, Resource Allocation and Service Placement of IoT Applications: State of the Art

Fatima Z. Cherhabil, Mammar Sedrati, Sonia-Sabrina Bendib

***Abstract*—**In order to support the continued growth, critical latency of IoT applications and various obstacles of traditional data centers, Mobile Edge Computing (MEC) has emerged as a promising solution that extends the cloud data-processing and decision-making to edge devices. By adopting a MEC structure, IoT applications could be executed locally, on an edge server, different fog nodes or distant cloud data centers. However, we are often faced with wanting to optimize conflicting criteria such as minimizing energy consumption of limited local capabilities (in terms of CPU, RAM, storage, bandwidth) of mobile edge devices and trying to keep high performance (reducing response time, increasing throughput and service availability) at the same time. Achieving one goal may affect the other making Task Offloading (TO), Resource Allocation (RA) and Service Placement (SP) complex processes. It is a nontrivial multi-objective optimization problem to study the trade-off between conflicting criteria. The paper provides a survey on different TO, SP and RA recent Multi-Objective Optimization (MOO) approaches used in edge computing environments, particularly Artificial Intelligent (AI) ones, to satisfy various objectives, constraints and dynamic conditions related to IoT applications.

***Keywords*—**Mobile Edge Computing, Multi-Objective Optimization, Artificial Intelligence Approaches, Task Offloading, Resource Allocation, Service Placement.

I. Introduction

EVEN though IoT devices (e.g., sensors, actuators, mobile phones and other smart devices) are becoming powerful, the available local resources cannot meet and guarantee the required high performance of time-critical IoT-enabled applications (e.g., high-quality video streaming, interactive mobile gaming, augmented-reality, and mission-critical applications). Each application has its own requirements in terms of sensitivity on latency, computing and reliability [1], [2].

The high network delay for sending data over public internet alters the benefits of the powerful computing resources that are available at a cloud data center. Therefore, the trend of Edge Computing (EC) has arisen as promising approach to overcome this obstacle by providing the benefits of cloud computing in the proximity of the end-users [3]. However, the EC nodes are more heterogeneous and have fewer capability resources (e.g., processing, memory and storage resources) compared to cloud data centers [4]. For that reason, there are various mapping possibilities and several feasible solutions to whether execute those IoT applications locally, deploy them on an edge server, different fog nodes or distant cloud data centers [5]. Finding the optimal deployment scheme (which is known as TO and the service/application placement) is computationally complex and there is not any exact solution for this. Thus, the problem belongs to NP-Hard class [6]. The focus is on how to execute IoT applications efficiently with edge networks capabilities and especially MEC associated requirements.

The TO strategy is to decide whether, what and where the task generated by the user terminal needs to be offloaded [7]. It has a great impact on IoT applications, since they are usually resource-constrained, by reducing task execution time, response time and energy consumption [8], [9]. A result of such a decision may be [10]:

- Local execution: The whole computation is done locally at the mobile device with no offloading at all to the MEC server.
- Full offloading: The whole computation is offloaded and processed by the MEC server.
- Partial offloading: A portion of the computation is performed locally, and the remaining is transferred to the MEC server. The advantage of this type of offloading is that it can benefit from both local and remote resources.

Next, if a decision on the full or partial offloading of an application is taken, a proper allocation of computation resources has to be done. Application placement involves finding the available resources in the network (nodes and links) that satisfy the application requirements (locality and delay sensitivity), satisfy the constraints (CPU, RAM, Storage, latency, bandwidth, etc.) and optimize the objectives (if any) [11].

The process of application/SP is affected by the capacity of the offloaded application to be parallelized or partitioned. If the application cannot be split into several parts, only one physical node may be allocated. In contrast, resources distributed over several computing nodes may process the offloaded application.

The success of the TO, SP and RA of IoT-enabled applications has attracted the researchers since the concept of EC was proposed. They are very complex processes and depend on many contextual parameters, e.g., Mobility, Availability, Dynamicity, Energy consumption, Cost, Performance, and

Fatima Z. Cherhabil and Dr. Sonia-Sabrina Bendib are with Lamie Laboratory, Dr. Mammar Sedrati is with Lastic Laboratory, Department of Computer Science, University of Mustafa Benboulaid, Batna 2. 53, Constantine Avenue. Fésdis, Batna (e-mail: f.cherhabil@univ-batna2.dz, ss.bendib@univ-batna2.dz, m.sedrati@univ-batna2.dz).

Security, etc.

Several surveys had been done focusing on different aspects [8], [10], [11]. This paper gives a survey on recent works aiming at using MOO approaches, emphasizing more on AI techniques, in TO, RA and SP in MEC context.

The rest of this paper is organized as follows: Section II reveals the computing paradigms and a brief comparison between them. Section III presents some objective metrics that need to be optimized in the context of MEC and different AI techniques used to solve the resulting MOO problems. A comparison of the TO, RA and SP existing works are summarized in Section IV. Finally, in Section V, the paper is concluded with directions for future work.

II. Computing Paradigms

Since the 1960s, computing has alternated between centralization and decentralization letting the emergence of different computing paradigms from distant cloud data centers to the edge of the network (such as: Cloudlets, Fog computing, Mobile EC). They are partially overlapping and complementary concepts [12] defined as follows:

- *Cloud Computing (CC):* A centralized infrastructure that aims to provide uninterrupted access to powerful cloud servers.
- *Mobile Cloud Computing (MCC):* Integration of mobile devices with CC technology
- *Cloudlets:* A kind of MCC that offers the necessary cloud resources closer to mobile devices.
- *Fog Computing (FC):* Defined by Cisco 2012 as the process of extending CC capabilities to the network edge [48].
- *EC:* A lightweight and primitive type of FC that resides at the edge of the network and closer to intelligent terminals.
- *Mobile Edge Computing (MEC):* Dedicated to help wireless networks with CC-like capabilities to deliver low-latency, context-aware services directly from the network edge. It was first defined by the European Telecommunications Standard Institute (ETSI) in 2014 as a new platform that "provides IT and cloud computing capabilities within the Radio Access Network (RAN) in close proximity to mobile subscrDibers" [13].

Table I summarizes materials, location restrictions, latency, security and system management of each computing paradigm [8], [14], [15].

TABLE I
Comparison of Computing Paradigms

Computing Paradigms	Materials	Location restrictions	Available resources	Latency	Security	System Management
CC	Large scale data centers and high-level computing servers.	Independent of the type of the device and the location of the end-user.	Practically infinite capacity.	Not always a feasible solution for providing low latency communication.	Sending data to the cloud over the Internet can be susceptible to attacks and increases the surface of threat.	Centralized
MCC	Distant high performance computing server or cluster.	Support thin client user interactions with the application over the Internet.				
Cloudlets	A group of computers or multiple multi-core hardware equipment directly connected to an Access Point (AP) or Base Station (BS).	Provide services to devices located in the same geographical area (a local "mini cloud").	Mostly limited (comprised of lower capability micro-data centers compared with cloud ones).	The use of those infrastructures is the most favorable way to reduce the communication delay with different levels accordingly.	Since the data of end devices usually stay within the local network, the necessary security and confidentiality can be attained.	Distributed
FC	Heterogeneous devices, such as routers, switches, industrial controllers or access points, which leads to flexibility.	Could be located anywhere between end-users and the cloud.				
EC	It uses micro-computers and micro-controllers to feed into nodes of FC.	On the same level with the end devices				
MEC	Small-scale data-centers, with moderate resources, deployed at a 3G Radio Controller or an LTE Macro BS.	Assist wireless mobile devices at the edge of a mobile network.				

III. Optimized Criteria

The optimization can be addressed to maximize or minimize the metric value. Those metrics can be seen as challenges faced in order to have a full satisfaction of all involved parties in the TO, RA and SP process [8], [11], [15].

- *Delay/Latency* means task execution time taking into consideration the transmission and propagation time in case the task was offloaded to an edge server or to the cloud. Most commonly, the task execution delay is associated with the response time (i.e., the time duration between user requests and service lunch).
- *Energy Consumption* is the major concern in the case of EC since they are power limited compared with the cloud.
- *Load balancing* aims at minimizing the overall resource usage by carefully allocating and scheduling the available resources.
- *Cost* can include networking, monetary and execution cost.
- *Mobility* represents a substantial challenge for realizing pervasive and reliable computing (i.e., without interruptions and errors). Users unpredicted moving among different cells will cause severe interference, which will greatly degrade the communication performance. It can be insured with model accuracy.
- *Security* is processing data near end-users protects user privacy. However, MEC servers become more vulnerable for both logical and physical intrusions.

- *Dynamicity* means make on-line decisions and try to capture the network changes (e.g. addition, failure or removal of an equipment).
- *Other metrics* such as resource availability, Quality of service (QoS) application requirements, Quality of Experience (QoE), etc.

Objective metrics are taken usually as individual objectives. However, to be more realistic, they should be considered simultaneously in the objective function even if they come with a trade-off.

Since our problem of optimization belongs to NP-Hard category, meta-heuristic AI algorithms are engaged to solve this kind of problems. In fact, they are becoming successful alternatives for solving optimization problems that include the mathematical formulation of uncertain, stochastic and dynamic information, thus making them excellent candidates for TO, RA and SP problems [8], which need to make a trade-off between conflicting objectives at the same time [16].

MOO deals with multiple conflicting objectives. Among MOO approaches, Artificial Intelligence-based ones ensure good compromise between conflicting objectives.

Population-based and physics-based methods are AI techniques that include a wide range of nature-inspired algorithms and provide close to optimal solutions in combinatorial problems. In the following, we conduct a summary of how some of those algorithms works.

A. Genetic Algorithm (GA)

Evolutionary algorithms are based on Darwinian Theory of survival of the fittest. The method follows a sequence of generations (a generation represents an algorithmic iteration while a gene is comparable to a component of the design vector), where the best design points in the population (represents a group of potential solution points) are considered to be the most 'fit' and are allowed to survive and reproduce [17]. The process is composed of several steps namely: Encoding scheme, Generation of initial population, Reproduction and introducing variations into the population of designs (by using crossover or mutation) [18], [19].

GAs combine the use of random numbers and information from previous iterations to evaluate and improve a population of points (a group of potential solutions) rather than a single point at a time.

B. Simulated Annealing (SA)

This method tries to mimic the process of annealing solids in order to optimize complex systems. It consists of two steps [20], [21]:

- Increase the temperature of the heating bath to a maximum value at which the solid melt.
- Decrease carefully the temperature of the heating bath until the particles arranging themselves in the ground state of the solid.

The algorithm of SA is seen as a sub set of GAs with a population of one individual and a changing mutation rate. It starts with an initial design, then, new designs are generated randomly according to some algorithms.

C. Artificial Bee Colony (ABC) algorithm

This algorithm simulates the intelligent foraging behavior of a honeybee swarm. It consists of three essential components [22]:

- *Food sources*: The quality of a food source can be represented by its closeness to the hive, richness of the energy, taste of its nectar, etc.
- *Employed foragers:* These are bees searching the food resource visited by themselves and sharing it with other bees waiting in the hive.
- *Unemployed foragers:* It is a bee that looks for a food source to exploit, it can be an onlooker (bees waiting for the information given by the employed bee to choose a food source) or a Scout (bees carrying out random search).

The position of a food source denotes a possible solution to the optimization problem and its quality (fitness) symbolizes the nectar amount of this food source. Additionally, the number of the employed bees or the onlooker bees and the number of solutions are equal in the population [22].

D. Particle Swarm Optimization (PSO) Algorithm

PSO is an experimental optimization method developed from the swarm intelligence. It is based on the research of birds and the fish flock program behavior. While birds are looking for food from place to another, they trace the best places where the food can be found [23], [24].

E. Whale Optimization Algorithm (WOA)

It mimics the social behavior of humpback whales. The algorithm takes its inspiration from the bubble-net hunting technique and includes three operators (the search for prey, encircling prey, and bubble-net foraging behavior of humpback whales) [25].

F. Ant Colony Optimization (ACO)

The basic principle is to release pheromone with different levels on the path of natural ants (based on the path's priority). The selection of the path can be determined by pheromone concentration and local heuristic values calculated [26], [27].

G. Cuckoo Search Algorithm (CSA)

The idea of CSA is derived from the effort to survive among cuckoos. The survived cuckoo society immigrates to a better environment and start reproducing and laying eggs [28].

IV. Comparison of TO, RA and SP Works

A. Service Placement

Some works address the SP problem in the context of EC and try to formulate it as a single objective (SO) optimization problem [12], [29]. However, a shortcoming of those works is that they are optimizing just one metric, which can be usually on the expense of the others. MOO is more realistic and practical way of solving this kind of problems since it ensures the simultaneous and systematic optimization of a collection of objective functions.

Adyson in [30] had conducted a system model where multiple replicas of an application can be placed in different

network parts to have requests (load) distributed among these replicas. Then, he formulated it as a MOO problem, and solved it with a combination of Biased Random-Key Genetic Algorithm (BRKGA) and Non-dominated Sorting Genetic Algorithm II (NSGA-II is the most prominent and popular genetic algorithm used in MOO [31]). The proposal had outperformed other benchmark algorithms in terms of response deadline violation, cost and availability. Later, the author had improved the performance of his previous solutions by including heuristics in the initialization of the proposed meta-heuristic based on GA in another work [32]. The proposed MOGA (Multi-Objective Genetic Algorithm) had achieved values close to the optimum of the MILP (Mixed Integer Linear Programming) formulation in terms of deadline violation, and outperformed the benchmark heuristics for the other analyzed objectives.

In another work using GAs [33], authors were looking for an efficient decision-making algorithm for virtual machine (VM) placement that should be able to work dynamically throughout the application lifecycle. Thus, they had formulated a MOO problem for generating application deployment plans that minimize the following metrics: cost, CPU, memory, user-node and inter-node distance and then solved it with a GA. Simulation had demonstrated that the algorithm is efficient and can provide optimal solutions for VM placement decision making.

Seeking to optimize the completion time, energy consumption and economic cost, the work presented in [34] is one more article using NSGA-II in resource and IoT application management that considers both computation and communication aspects for executing IoT applications in a heterogeneous fog infrastructure. Based on both simulated and real-world testbeds tailored for a set of medical IoT application case studies, the previous metrics had been optimized compared to benchmark approaches.

A conceptual computing framework based on fog-cloud control middleware was proposed for optimal IoT SP is cited in [35]. They had formulated the problem as an automated planning model for managing service requests and solved it with an evolutionary approach based on PSO as a MOO problem using a scalarization method. Then, they had optimized the following metrics: cost, latency (response time), throughput and utilization of fog resources in a three-layered ecosystem (IoT devices, FC layer, and CC layer).

Morkevicius et al. in [36] had chosen to use the PSO algorithm with the Pareto dominance concept. Also in a fog architecture, a dynamic service orchestration to provide an efficient SP inside fog nodes was achieved by using a two-stage MOO method (IMOPSO: integer multi-objective particle swarm optimization and AHP: analytical hierarchy process) taking the security as an optimizing metric along with CPU, RAM, power utilization and range.

By using another AI MOO technique, which is a multi-objective cuckoo search algorithm (MOCSA), in [16], the authors had introduced an algorithm for the deployment of IoT application components on fog nodes to meet reliable deployment for user requests. Simulation results proved improvement of proposed MOCSA in terms of power consumption and total latency against the above algorithms (NSGA-II, MOPSO and other AI methods).

TABLE II
SP STUDIES

Techniques	Cited in	Infrastructure	Optimization Type	Delay / Latency	Energy	Load Balancing	Cost	Availability	Other Metrics	Mobility	Dynamicity	Security	Multi-User	Multi-Server
BRKGA	[12]	EC	SO	✓					-				✓	
Lyapunov Optimization	[29]	MEC	SO	✓					-	✓	✓		✓	✓
Combination of BRKGA and NSGA-II	[30]	MEC	MO (Pareto)	✓			✓	✓	-				✓	✓
Meta-Heuristic Based on GA	[32]	MEC	MO (Pareto)	✓			✓	✓	-				✓	✓
GA	[33]	Edge-Cloud	MO (Scalarization)				✓		CPU, Memory, User-Node & Inter-Node Distance		✓			
NSGA-II	[34]	Fog	MO (Pareto)	✓		✓	✓		-					
PSO	[35]	Fog-Cloud	MO (Scalarization)	✓			✓		Throughput and Utilization of Fog Resources		✓			
IMOPSO and AHP	[36]	Fog	MO (Pareto)	✓					CPU, Ram, Range	✓	✓	✓		
MOCSA	[16]	Fog	MO (Pareto)	✓	✓				-					

Table II summarizes the techniques used for each work, the metrics taken into consideration and if it supports multi-server or multi-user scenarios. In multi-user scenario, multiple tasks of multiple users can be offloaded to the edge-computing server for execution in a time slice. In the opposite side, each MEC server only holds one application or task of a user device. In multi-server scenario, a task can be split into different parts in order to be computed by different servers. Yet, in other side, each application or task can be assigned to only one MEC server.

B. Task Offloading

Several studies concerning computation offloading in the context of MEC scenarios were done. Yet, the majority of them provide a single objective optimization such as [37] where the problem was linearized and solved using Lagrangian duality algorithm in order to find a trade-off between completion time and communication costs in a device-to-device (D2D) offloading with assistance of base station (BS) and presence of user mobility. The algorithm had provided structural insights of the optimal trade-off but had a small gap with the optimal solutions. In [38], a collaborative task offloading model was formulated to offload the tasks to UAVs (Unmanned Aerial Vehicles) or the BS selectively by improving a GA. The simulation results revealed the efficiency of using UAV-assisted MEC infrastructures in TO process. The authors in [39] solved the tradeoff between energy efficiency and service delay for multi-user multi-server MEC-enabled IoT systems by a Lyapunov optimization when provisioning offloading services in a user mobility scenario. Liu et al. in [26] used the ACO algorithm to find the optimal scheduling solution in a reduced search space of resources by dividing it with a fuzzy clustering.

Among the few MOO existing works in MEC, authors in [40] had tried to find a trade-off between energy consumption and latency when offloading the intensive computing tasks to edge servers by modifying the NSGA-II algorithm. The work provided a better way than existing studies, the ones using SO optimization, and had offered more flexible choices based on the requirements of different IoT applications. Similarly, in works presented in [41], a multi-objective offloading strategy MOPCA had been designed based on NSGA-II to solve the problem. The approach was able to obtain best trade-offs among energy consumption, task delay and price and results showed its effectiveness and efficiency.

Another AI MOO solution using a centralized algorithm SMOSA (Stochastic Multi-Objective Simulated Annealing) was cited in [42]. The goal was to ensure the availability of offloading and reduce both energy consumption and the amount of data transmitted through cellular access links. Experimental results had revealed near-optimal solutions for several studied scenarios.

Compared with some typical approaches such as NSGA-II, the MOWOA2 [43], an improved Multi-Objective Whale Optimization Algorithm, by using the gravity reference point method, had performed better in terms of the quality of the final solutions and had significantly lower energy consumption.

A different optimization approach, the bidding model, had been considered in [7]. The goal was to minimize the migration overhead of the computing task and optimize the scheduling decision of the computing task with the user's minimum performance guarantee as the constraint. Even though the work had provided new ideas for task scheduling in EC, but by using the scalarization method, the optimization result is greatly influenced by the configuration of the weights.

TO studies are summarized in Table III with the optimization type and metrics supported by each one.

C. Joint TO and RA

Limited number of works had combined the problem of task offloading with RA. The study cited in [44] had addressed a two-tier strategy for multi-user multi-MEC-servers in a 5G heterogeneous networks in order to minimize the total computing overhead of mobile devices. The problem was formulated as a mixed-integer nonlinear program (MINLP) problem of NP-hard complexity and divided into two sub-problems: computational RA and computation offloading decisions and then solved using PSO meta-heuristic algorithm. The results of simulations indicated that the proposed algorithm outperformed several baseline schemes in terms of total computing overhead.

Regarding processing time, mobility and service cost, [45] also had presented a dynamic task scheduling and load-balancing technique based on an integrated accelerated particle swarm optimization (APSO) algorithm with dynamic programming. The proposed method was associated with reducing service cost and waiting time compared to the other algorithms and improvements in the fitness function value.

Working again with NSGA-II, Yue et al. in [46] had examined the problem of computing offload and RA to balance energy and time delay of task execution (trade-off) in an EC wireless network. The results had shown that the unloading decision of NSGA-II can reach the best and can be distributed in a wider range. However, the number of generations can cause a wastage of computing resources to a certain extent.

MOO for Joint TO, Power Assignment, and RA in MEC was a recent study [47] that aims to minimize delay (response time), energy consumption, and cost of mobile device users. The experimental results indicated that the proposed strategy obviously outperforms the baseline method. However, it did not take into account neither the mobility and dynamicity nor the security of network equipment.

Table IV summarizes the previous studies, metrics used and if they support mobility, dynamicity, security, multi-user and multi-server scenarios.

V. Conclusion

The use of MEC can improve efficiency and flexibility of IoT applications. However, conflicting criteria will appear as a key challenge. In fact, achieving one goal may affect the other. As a solution, MOO procedures are becoming primordial in order to handle crucial operations such as intelligent computation offloading, RA and service continuity.

The scope of the article was to point up some recent works on TO, SP and RA used in MEC environments that are applying

MOO techniques. Specific emphasis was given on the AI approaches to satisfy the IoT applications requirements.

As a future work, the goal is to use a combination of some MOO intelligent techniques to find a set of compromise solutions that minimize energy consumption and simultaneously maximize performance as much as possible.

TABLE III
TO STUDIES

Technique	Cited in	Infrastructure	Optimization Type	Delay / Latency	Energy	Load Balancing	Cost	Availability	Other Metrics	Mobility	Dynamicity	Security	Multi-User	Multi-Server
Lagrangian Duality Algorithm	[37]	MEC	SO		✓		✓		-	✓			✓	
Improved GA	[38]	UAV-Assisted MEC	SO	✓	✓				-				✓	
Lyapunov Optimization	[39]	MEC	SO	✓	✓				-	✓	✓		✓	✓
Fuzzy Clustering and the ACO Algorithm	[26]	MEC	SO						Total Profit				✓	
A Modified NSGA-II	[40]	MEC	MO (Pareto)	✓	✓				-				✓	
A Multi-Objective Offloading Strategy MOPCA Based on NSGA-II	[41]	MEC	MO (Pareto)	✓	✓		✓		-				✓	
SMOSA	[42]	MEC	MO (Pareto)		✓			✓	Data Traffic in Access Links	✓			✓	
An Improved MOWOA	[43]	MEC	MO (Pareto)	✓	✓				-				✓	
The Bidding Model	[7]	MEC	MO (Scalarization)	✓	✓	✓							✓	

TABLE IV
JOINT TO AND RA STUDIES

Technique	Cited in	Infrastructure	Optimization Type	Delay / Latency	Energy	Load Balancing	Cost	Availability	Other Metrics	Mobility	Dynamicity	Security	Multi-User	Multi-Server
PSO	[44]	MEC	MO (Scalarization)	✓	✓				-				✓	✓
APSO Algorithm with Dynamic Programming	[45]	MEC	MO (Pareto)	✓			✓		-	✓	✓		✓	
NSGA-II	[46]	EC	MO (Pareto)	✓	✓				-					
MOEA	[47]	MEC	MO (Pareto)	✓	✓		✓		-				✓	✓

REFERENCES

[1] Bebortta, Sujit, Singh, Amit Kumar and Senapati, Dilip, "Performance analysis of multi-access edge computing networks for heterogeneous IoT systems," Materials Today: Proceedings, vol. 58, pp. 267-272, 2022.

[2] Al-Fuqaha, Ala, Guizani, Mohsen, Mohammadi, Mehdi, Aledhari, Mohammed and Ayyash, Moussa, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE communications surveys & tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.

[3] El-Sayed, Hesham, Sankar, Sharmi, Prasad, Mukesh, Puthal, Deepak, Gupta, Akshansh, Mohanty, Manoranjan and Lin, Chin-Teng, "Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," IEEE Access, vol. 6, pp. 1706-1717, 2017.

[4] Shi, Weisong, Cao, Jie, Zhang, Quan, Li, Youhuizi and Xu, Lanyu, "Edge computing: Vision and challenges," IEEE internet of things journal, vol. 3, no. 5, pp. 637-646, 2016.

[5] H. Liu, F. Eldarrat, Alqahtani, Hanen, Reznik, Alex, De Foy, Xavier and Zhang, Yanyong, "Mobile edge cloud system: Architectures, challenges, and approaches," IEEE Systems Journal, vol. 12, no. 3, pp. 2495-2508, 2017.

[6] Azimi, Shelernaz, Pahl, Claus and Shirvani, Mirsaeid Hosseini, "Particle Swarm Optimization for Performance Management in Multi-cluster IoT Edge Architectures," in CLOSER, 2020.

[7] Shi, Zheng and Shi, Zhiguo, "Multi-node Task Scheduling Algorithm for Edge Computing Based on Multi-Objective Optimization," in Journal of Physics: Conference Series, 2020.

[8] Saeik, Firdose, Avgeris, Marios, Spatharakis, Dimitrios, Santi, Nina, Dechouniotis, Dimitrios, Violos, John, Leivadeas, Aris, Athanasopoulos, Nikolaos, Mitton, Nathalie and Papavassiliou, Symeon, "Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions," Computer Networks, vol. 195, p. 108177, 2021.

[9] Zhang, Guanglin, Zhang, Wenqian, Cao, Yu, Li, Demin and Wang, Lin, "Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices," IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4642-4655, 2018.

[10] Mach, Pavel and Becvar, Zdenek, "Mobile edge computing: A survey on architecture and computation offloading," IEEE Communications

Surveys & Tutorials, vol. 19, no. 3, pp. 1628-1656, 2017.
[11] Salaht, Farah Ait, Desprez, Frédéric and Lebre, Adrien, "An overview of service placement problem in fog and edge computing," ACM Computing Surveys (CSUR), vol. 53, no. 3, pp. 1-35, 2020.
[12] Maia, Adyson M, Ghamri-Doudane, Yacine, Vieira, Dario and de Castro, Miguel F, "Optimized placement of scalable iot services in edge computing," in 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2019.
[13] Kekki, Sami, Featherstone, Walter, Fang, Yonggang, Kuure, Pekka, Li, Alice, Ranjan, Anurag, Purkayastha, Debashish, Jiangping, Feng, Frydman, Danny and Verin, Gianluca, "MEC in 5G networks," ETSI white paper, vol. 28, no. 2018, pp. 1-28, 2018.
[14] Ramzanpoor, Yaser, Hosseini Shirvani, Mirsaeid and Golsorkhtabaramiri, Mehdi, "Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure," Complex & Intelligent Systems, vol. 8, no. 1, pp. 361-392, 2022.
[15] Cui, Yunfei, Geng, Zhiqiang, Zhu, Qunxiong and Han, Yongming, "Multi-objective optimization methods and application in energy saving," Energy, vol. 125, pp. 681-704, 2017.
[16] Marler, R Timothy and J. S. Arora, "Survey of multi-objective optimization methods for engineering," Structural and multidisciplinary optimization, vol. 26, no. 6, pp. 369-395, 2004.
[17] S. Forrest, "Genetic algorithms," ACM Computing Surveys (CSUR), vol. 28, no. 1, pp. 77-80, 1996.
[18] J. Andersson, "A survey of multiobjective optimization in engineering design," Department of Mechanical Engineering, Linktjping University. Sweden, 2000.
[19] Bertsimas, Dimitris and Tsitsiklis, John, "Simulated annealing," Statistical science, vol. 8, no. 1, pp. 10-15, 1993.
[20] Karaboga, Dervis and Akay, Bahriye, "A comparative study of artificial bee colony algorithm," Applied mathematics and computation, vol. 214, no. 1, pp. 108-132, 2009.
[21] Kumar, Parasuraman and Silambarasan, Karunagaran, "Enhancing the performance of healthcare service in IoT and cloud using optimized techniques," IETE Journal of Research, pp. 1-10, 2019.
[22] Zhang, Yudong, Wang, Shuihua and Ji, Genlin, "A comprehensive survey on particle swarm optimization algorithm and its applications," Mathematical problems in engineering, vol. 2015, 2015.
[23] S. Mirjalili and L. Andrew, "The whale optimization algorithm," Advances in Engineering Software, vol. 95, p. 51–67, 2016.
[24] Liu, Jianwei, Wei, Xianglin, Wang, Tongxiang and Wang, Junwei, "An Ant Colony Optimization Fuzzy Clustering Task Scheduling Algorithm in Mobile Edge Computing," in International Conference on Security and Privacy in New Computing Environments, 2019.
[25] Dorigo, Marco, Birattari, Mauro and Stutzle, Thomas, "Ant colony optimization," IEEE computational intelligence magazine, vol. 1, no. 4, pp. 28-39, 2006.
[26] Yang, Xin-She and Deb, Suash, "Cuckoo search via Lévy flights," in 2009 World congress on nature & biologically inspired computing (NaBIC), 2009.
[27] Maia, Adyson M, Ghamri-Doudane, Yacine, Vieira, Dario and de Castro, Miguel F, "A multi-objective service placement and load distribution in edge computing," in 2019 IEEE Global Communications Conference (GLOBECOM), 2019.
[28] Deb, Kalyanmoy, Pratap, Amrit, Agarwal, Sameer and Meyarivan, TAMT, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE transactions on evolutionary computation, vol. 6, no. 2, pp. 182-197, 2002.
[29] Maia, Adyson M, Ghamri-Doudane, Yacine, Vieira, Dario and de Castro, Miguel Franklin, "An improved multi-objective genetic algorithm with heuristic initialization for service placement and load distribution in edge computing," Computer Networks, vol. 194, p. 108146, 2021.
[30] Aryal, Ram Govinda and Altmann, Jörn, "Dynamic application deployment in federations of clouds and edge resources using a multiobjective optimization AI algorithm," in 2018 Third international conference on fog and mobile edge computing (FMEC), 2018.
[31] Mehran, Narges, Kimovski, Dragi and Prodan, Radu, "MAPO: a multi-objective model for IoT application placement in a fog environment," in Proceedings of the 9th International Conference on the Internet of Things, 2019.
[32] Salimian, Mahboubeh, Ghobaei-Arani, Mostafa and Shahidinejad, Ali, "An Evolutionary Multi-objective Optimization Technique to Deploy the IoT Services in Fog-enabled Networks: An Autonomous Approach," Applied Artificial Intelligence, pp. 1-34, 2022.
[33] Morkevicius, Nerijus, Venčkauskas, Algimantas, Šatkauskas, Nerijus and Toldinas, Jevgenijus, "Method for dynamic service orchestration in fog computing," Electronics, vol. 10, no. 15, p. 1796, 2021.
[34] Ouyang, Tao, Zhou, Zhi and Chen, Xu, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," IEEE Journal on Selected Areas in Communications, vol. 36, no. 10, pp. 2333-2345, 2018.
[35] Ahani, Ghafour and Yuan, Di, "BS-assisted task offloading for D2D networks with presence of user mobility," in 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), 2019.
[36] H. Wang, "Collaborative Task Offloading Strategy of UAV Cluster Using Improved Genetic Algorithm in Mobile Edge Computing," Journal of Robotics, vol. 2021, pp. 1-9, 2021.
[37] Hu, Han, Song, Weiwei, Wang, Qun, Hu, Rose Qingyang and Zhu, Hongbo, "Energy Efficiency and Delay Tradeoff in an MEC-Enabled Mobile IoT Network," IEEE Internet of Things Journal, 2022.
[38] Cui, Laizhong, Xu, Chong, Yang, Shu, Huang, Joshua Zhexue, Li, Jianqiang, Wang, Xizhao, Ming, Zhong and Lu, Nan, "Joint optimization of energy consumption and latency in mobile edge computing for Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4791-4803, 2018.
[39] Chen, Yidan, Wang, Xueyi, Ma, Lianbo and Zhou, Ping, "Multi-objective Optimization-Based Task Offloading and Power Control for Mobile Edge Computing," in International Conference on Intelligent Computing, 2021.
[40] Zhao, Xuhui, Shi, Yan and Chen, Shanzhi, "TS-SMOSA: A Multi-Objective Optimization Method for Task Scheduling in Mobile Edge Computing," Journal of Internet Technology, vol. 20, no. 4, pp. 1057-1068, 2019.
[41] Huang, Mengxing, Zhai, Qianhao, Chen, Yinjie, Feng, Siling and Shu, Feng, "Multi-objective whale optimization algorithm for computation offloading optimization in mobile edge computing," Sensors, vol. 21, no. 8, p. 2628, 2021.
[42] Huynh, Luan NT, Pham, Quoc-Viet, Pham, Xuan-Qui, Nguyen, Tri DT, Hossain, Md Delowar and Huh, Eui-Nam, "Efficient computation offloading in multi-tier multi-access edge computing systems: A particle swarm optimization approach," Applied Sciences, vol. 10, no. 1, p. 203, 2019.
[43] Alfakih, Taha, Hassan, Mohammad Mehedi and Al-Razgan, Muna, "Multi-Objective Accelerated Particle Swarm Optimization With Dynamic Programing Technique for Resource Allocation in Mobile Edge Computing," IEEE Access, vol. 9, pp. 167503-167520, 2021.
[44] Ma, Yue, Li, Xin and Li, Jianbin, "An Edge Computing Offload Method Based on NSGA-II for Power Internet of Things," Internet of Things and Cloud Computing, vol. 9, no. 1, pp. 1-9, 2021.
[45] Wang, Peng, Li, Kenli, Xiao, Bin and Li, Keqin, "Multi-objective optimization for joint task offloading, power assignment, and resource allocation in mobile edge computing," IEEE Internet of Things Journal, pp. 1-12, 2021.
[46] Hamdan, Salam, Ayyash, Moussa and Almajali, Sufyan, "Edge-computing architectures for internet of things applications: A survey," Sensors, vol. 20, no. 22, p. 6441, 2020.
[47] Mao, Yuyi, You, Changsheng, Zhang, Jun, Huang, Kaibin and Letaief, Khaled B, "A survey on mobile edge computing: The communication perspective," IEEE communications surveys & tutorials, vol. 19, no. 4, pp. 2322-2358, 2017.
[48] Bonomi, Flavio, Milito, Rodolfo, Zhu, Jiang and Addepalli, Sateesh, "Fog computing and its role in the internet of things," in Proceedings of the first edition of the MCC workshop on Mobile cloud computing, 2012.