

# Non-Dominated Sorting Genetic Optimization-Based Fog Cloudlet Computing for Wireless Metropolitan Area Networks

Ali Salah Alasady<sup>1\*</sup>, Wid Akeel Awadh<sup>2</sup>, Mohammed S. Hashim<sup>3</sup>

<sup>1</sup>Department of Computer Science, University of Basrah, Basrah, Iraq

<sup>2</sup>Department of Computer Information Systems, University of Basrah, Basrah, Iraq

<sup>3</sup>Department of Computer science, Education College for Pure Sciences, University of Basrah, Basrah, 61004, Iraq

E-mail: Ali\_s.hashim@uobasrah.edu.iq<sup>1</sup>, wid.jawad@uobasrah.edu.iq<sup>2</sup>, moh.salah@uobasrah.edu.iq<sup>3</sup>

\*Corresponding author

**Keywords:** fog-cloud computing, multi-objective optimization, non-dominated sorting, wireless metropolitan area network (WMAN), energy consumption management

**Received:** August 22, 2023

*Fog cloud computing management is an emerging topic. It involves accepting generated tasks from users and assigning them in the fog nodes or sending them to the cloud with optimizing various objectives. One specific type of fog environment is cloudlet which indicates to local servers deployed in wireless metropolitan area network and leveraging the existing access points or base-station for communicating tasks between users and cloudlets. The existing methods of cloudlet-based computing optimization are numerous. This article provides cloudlet computing management process based on multi-objective optimization using the concept of non-dominated sorting. The considered objectives are latency, energy consumption at the cloudlets, energy consumption at the user, and cost which is calculated based on the number of cloudlets. Non-dominated sorting genetic algorithms (NSGA-II) and (NSGA-III) are used for comparison. The results reveal equivalent performance between them with slight superiority of NSGA-III over NSGA-II in terms of the number of non-dominated solutions.*

*Povzetek: Članek obravnava upravljanje računalništva v oblaku z večciljno optimizacijo in nedomirajočim razvrščanjem, primerjajoč NSGA-II in NSGA-III algoritme.*

## 1 Introduction

Cloud services have emerged as a popular solution for providing remote computing resources to meet the growing demands of users for data processing and storage. The cloud computing model provides a scalable, flexible, and cost-effective platform for businesses and individuals to store, manage and process large amounts of data. Despite the numerous advantages of cloud services, they still face several critical challenges, one of which is latency. Latency refers to the time it takes for a request to travel from the client to the server and back, and it can have a significant impact on the quality of service (QoS) of cloud services [1].

To address this challenge, fog computing has been proposed as a way of bringing computing and storage of data in closer proximity to the network's periphery [2]. Unlike cloud computing, which centralizes all computing and data storage in a single location, fog computing distributes computing and data storage across multiple nodes in the network, including edge devices and fog nodes, which are typically located at the network edge [3]. This approach helps to reduce latency and improve the QoS of cloud services, as the data does not have to travel as far to reach the server [4].

However, the deployment and optimization of fog nodes or cloudlets is still a significant challenge in fog computing. The placement of cloudlets is an important problem, as it can have a significant impact on the overall performance of the fog computing system. The

deployment of cloudlets must consider various factors, such as network topology, traffic patterns, and the availability of computing and storage resources. Furthermore, the deployment of clouds has a multi-objective nature. More specifically, it aims at optimizing more than one objective at the same time with handling the issues of self-conflicting between the objectives [5].

Some of the important objectives are latency, energy consumption at the cloudlet, energy consumption at the user device, and cost of deployment which is related to the number of cloudlets. The self-conflicting nature occur in various cases [6]. For example, reducing latency might require deploying more cloudlets closer to the end-users, which could increase the cost of deployment and energy consumption at the cloudlets. Similarly, reducing energy consumption at the user device might require more processing to be offloaded to the cloud, which could increase the energy consumption at the cloudlet and increase the latency. In order to reconcile these contradictory goals, cloud computing researchers and practitioners frequently employ multi-objective optimization techniques, which seeks to identify a collection of solutions that strike a balance between the competing objectives. One of the popular multi-objective optimizations is non-dominated sorting genetic algorithm that has two variants, namely, NSGA-II, NSGA-III. NSGA-II, NSGA-III are two variants of the NSGA algorithm [7].

NSGA-II was proposed by Deb et al. in 2002 [8], and it is a widely used and highly effective algorithm for multi-objective optimization. NSGA-II uses a fast non-dominated sorting approach to rank candidate solutions based on their Pareto optimality, and it also uses crowding distance to maintain diversity in the population of solutions.

NSGA-III, on the other hand, was proposed by Deb et al. in 2014, and it is a more recent and advanced version of the algorithm. NSGA-III [9] uses a reference point-based approach to guide the search towards the desired region of the Pareto front. This approach enables NSGA-III to handle problems with more than three objectives, which is a limitation of NSGA-II.

The aim of this paper is to evaluate and contrast the NSGA-II and NSGA-III algorithms based on four criteria related to cloudlets optimization. These criteria include latency, energy usage at the cloudlets, energy usage by the user, and the cost determined by the quantity of cloudlets. Thus, the key takeaways of this paper are as follows:

- It provides cloudlet computing management process based on multi-objective optimization using the concept of non-dominated sorting. The considered objectives are latency, energy consumption at the cloudlets, energy consumption at the user, and cost which is calculated based on the number of cloudlets.
- It selects two popular multi-objective algorithms, namely, NSGA-II and NSGA-III and use them for comparing the Pareto front obtained by each of the two algorithms.
- The comparison provides different aspects of the performance including diversity which is represented by hyper-volume and delta metric and domination which is represented by set coverage.

The subsequent is the structure of this article: In the second section, an exhaustive literature review of related works is presented. Following this, Section 3 provides an overview of the methodology utilized in the research. Section 4 then presents the experimental evaluation and its corresponding results. Finally, Section 5 wraps up the article with the conclusion and a discussion on potential future works.

## 2 Related works

Edge computing has seen a surge in growth and attention recently, driven by the growing need for rapid and efficient data processing across a variety of applications. One primary challenge in this domain is the strategic positioning and optimization of edge servers. In response, numerous strategies and algorithms have emerged to facilitate the optimal deployment of edge servers across different situations and environments. This review aims to shed light on some of the leading research efforts in edge computing and the positioning of edge servers.

A significant portion of the initial research focuses on task offloading and its alignment with cloudlets. A variety of advanced algorithms have been put forward to address this issue. An example of this can be seen in the research discussed in reference [10], which examines a hybrid strategy for outsourcing tasks and deploying cloudlets in order to decrease task response time, energy consumption, and the overall count of active cloudlets. The research in [11] is centered on maximizing the computational frequencies and transmission capabilities of IoT devices, with an emphasis on reducing task completion duration and energy costs. Scholarly articles [12] and [13] employ novel methodologies to examine the implementation of edge servers in the context of the Internet of Vehicles and the placement of cloudlets within networks of wireless access points, respectively. The paper in [14] delves deep into the best practices for task offloading strategies and related parameters.

Another segment of research employs Linear Programming techniques to navigate the challenges of edge server deployment. For instance, [15] delves into a hybrid approach to cloudlet deployment over a specific network infrastructure, leveraging mixed-integer linear programming for its optimization endeavors. Other papers, like those from [16] to [23], tackle various aspects of the deployment challenge, from task delegation to load balancing, using a diverse set of methodologies.

In summary, the realm of edge computing and the strategic deployment of edge servers have been a hotspot of research focus in recent times. A myriad of methods, spanning from meta-heuristic algorithms and linear programming to machine learning, have been proposed to tackle deployment challenges. Particularly noteworthy is the fact that meta-heuristics are the area in which multi-objective optimization is found to function, especially with algorithms such as (NSGA-II, NSGA-III) taking the lead. While the use of previous algorithms for tasks akin to cloudlet optimization is clear, a direct comparative analysis of their efficacy is yet to be extensively explored in research. Such a comparison is crucial for two main reasons: understanding the differences in their optimal solutions and evaluating the similarity of their results. This comparative insight can guide decision-makers in selecting the most suitable algorithm for this and analogous optimization challenges.

## 3 Methodology

This section outlines the methodology we've formulated. It's broken down into multiple sub-sections. Sub-section 3.1 introduces the mathematical model. This is followed by a detailed look at the widely recognized NSGA-II and a separate overview of the equally renowned NSGA-III in sub-sections 3.2. Subsequently, we delve into our multi-objective assessment that draws on both algorithms in the subsequent sub-section. Finally, the evaluation criteria employed in this paper are detailed in the concluding sub-section.

### 3.1 Mathematical model

We adopted the mathematical model presented in [10]. The mobile edge computing network in this architecture has  $M$  users and  $N$  Access Points (APs). To service the users, cloudlets are set up across the network. Each user may access the network via the AP that is linked with them since all APs are hooked together. We assumed that the creation of cloudlets is solely dependent on the locations of the Access Points (APs), with each AP having the capacity to facilitate the deployment of a solitary cloudlet. The work is described by using the exponential distributions to represent the input data size and computing resource requirement of a user's job. The average values of these parameters are denoted by  $\alpha_i$  (bits) and  $\beta_i$  (CPU cycle), respectively. The computing capacity and effective switched capacitance of a user's  $u_i$  CPU are denoted by  $\varphi_i^u$  and  $\xi_i$ , respectively. The model is depicted in Figure 1. Financial constraints of the service provider are denoted by the letter  $L$ , which also signifies the maximum count of cloudlets that can be set up.

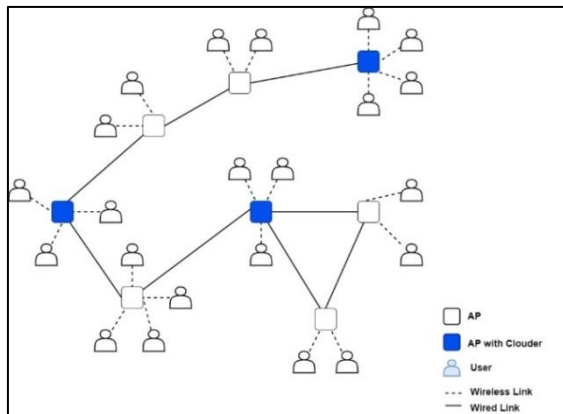


Figure 1: Graph representing an existing wireless metropolitan area network

### 3.2 Multi-objective optimization algorithms

The Non-dominated Sorting Genetic Algorithm (NSGA)-II and -III are two widely utilized optimization techniques for many objectives in many different domains.

- **Non-dominated sorting genetic algorithm (NSGA-II)**

NSGA-II is a genetic algorithm-based optimization technique that uses non-dominated sorting and crowding distance concept to find the Pareto-optimal set of solutions. The Pareto-optimal set is a set of solutions in which no solution can be improved in one objective without deteriorating at least one of the other objectives. NSGA-II operates in a generational manner, in which

each generation consists of selection, crossover, and mutation operators. Here is a rundown of the NSGA-II algorithm's operation.

**Algorithm – pseudocode of NSGA-II**

**Input:** populationSize ; numIterations;  
**Output:** non-dominated solutions  
**Start:**  
 Initialize the population with 'populationSize' solutions.  
 Evaluate the objective function of each individual.  
**For** each generation 'i' up to 'numIterations':  
     Perform non-dominated sorting on the population  
     Assign crowding distance to individuals in each front.  
     Create an empty offspring population.  
     **While** the size of the offspring population is less than the population size:  
         Select two parents from the population based on tournament selection.  
         Perform crossover and mutation to create offspring.  
         Evaluate the objective function of each offspring.  
         Add the offspring to the offspring population.  
     Merge the population and the offspring population.  
     Perform non-dominated sorting on the merged population.  
     Assign crowding distance to individuals in each front.  
     Truncate the merged population based on the crowding distance.  
**Return** the non-dominated solutions from the last generation.  
**End**

- **Non-dominated sorting genetic algorithm (NSGA-III)**

NSGA-III, or the Non-dominated Sorting Genetic Algorithm III, is a multi-objective optimization algorithm. Developed as an advancement from its predecessors, NSGA and NSGA-II, this iteration was specifically engineered to address optimization challenges involving more than two objectives [24].

NSGA-III uses a reference-point-based nondominated sorting approach to efficiently handle many-objective problems, i.e., problems with more than three objectives. We present the pseudocode of the algorithm that explains the basic structure of NSGA-III. Please note that the crossover and mutation operations

are problem-specific, as are the encoding of solutions and the definition of the objective functions. The reference point-based niching also needs specific implementations that aren't detailed here. You should also ensure you have adequate mechanisms in place for maintaining diversity in the population and for determining when to terminate the algorithm.

Remember, each objective function in Objective\_Functions [] must take a solution (a member of the population) as an input and output a fitness value. The Solution\_Space\_Boundary [] is used to constrain the solutions to a feasible space. The Population\_Size is the number of individuals in the population and Max\_Generation is the maximum number of iterations the algorithm will run for. Lastly, keep in mind this is a simplified version of NSGA-III. The actual algorithm can be more complex and may require fine-tuning and additional operations to work effectively on specific problems.

Algorithm – Pseudocode of NSGA-III
<p><b>Input:</b> Objective_Functions;Solution_Space_Boundary; Population_Size ;Max_Generation</p> <p><b>Output:</b> Pareto Front</p> <p><b>Start:</b> Initialize Population of size Population_Size within Solution_Space_Boundary. Evaluate Population using Objective_Functions. <b>For</b> generation = 1 to Max_Generation <b>do:</b></p> <p style="padding-left: 20px;">Select mating pool from Population. Generate Offspring by performing crossover and mutation on mating pool. Evaluate Offspring using Objective_Functions. Combine Population and Offspring into Combined_Population. Compute non-domination fronts of Combined_Population.</p> <p style="padding-left: 20px;"><b>if</b> size of first non-domination front &lt; Population_Size <b>then</b></p> <p style="padding-left: 40px;">Add members of first non-domination front to next Population. <b>For</b> each subsequent non-domination front <b>do:</b></p> <p style="padding-left: 60px;"><b>if</b> size of next Population + size of current non-domination front &lt;= Population_Size <b>then</b></p> <p style="padding-left: 40px;">Add members of current non-domination front to next Population.</p>

<b>Else</b>
Use reference point-based niching to select remaining members of next Population from current non-domination front. break.
<b>End if</b>
<b>End For</b>
<b>Else</b>
Use reference point-based niching to select members of next Population from first non-domination front.
<b>End if</b>
Population = next Population.
<b>End For</b>
<b>Return</b> Population as Pareto front
<b>End</b>

▪ **Multi-objective optimization based on NSGA-II and NSGA-III**

In our cloudlet optimization, we compare between the two stated algorithms, namely, NSGA-III, NSGA-II from the perspective of multi-objective optimization. Basically, we use various metrics for this purpose, namely, set coverage, hyper-volume, delta metric, number of non-dominated solutions.

- 1) SET\_COVERAGE\_MEASURE (C) which compares the Pareto sets  $P_{s1}$  and  $P_{s2}$  as follows [25]:

$$c(P_{s1}, P_{s2}) = \frac{|\{y \in P_{s2} | \exists x \in P_{s1} : x > y\}|}{|P_{s2}|} \quad (1)$$

- 2) HYPER-VOLUME MEASURE (HV): Evaluation of the efficacy of search algorithms in evolutionary multi-objective optimization has made extensive use of it. When compared to the reference point, which represents the worst solution, the volume of the dominated region in the objective space is calculated. This region is formed by joining the solutions  $x$  from the Pareto set PS that are positioned at the reference point along the diagonal of the hypercube PS [25]. If this indicator has higher values, it indicates that the solutions being considered are more desirable. The following equation can be used to get HV:

$$HV = volume \left( \bigcup_{x \in P_s} HyperCube(x) \right) \quad (2)$$

- 3) DELTA MEASURE: or the diversity metric  $\Delta$  represents the degree to which spread is accomplished among the many solutions that were produced. The function accepts as input the set of solutions that are not dominated, and it generates the delta measure using the subsequent equation [25]:

$$\Delta = \frac{d_f + d_i + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_i + (N-1)\bar{d}} \tag{3}$$

$N$  represents the count of solutions  $d_i$ . The Euclidean distances between the extreme solutions and the boundary solutions are represented by  $d_f$  and  $d_l$ , respectively. Meanwhile,  $\bar{d}$  is the mean of the sequential distances  $d_i$ , ranging from  $i = 1$  to  $N - 1$ .

- 4) 4) The cardinality of PS may be determined as follows to determine how many non-dominated solutions the optimization technique produces [25]:

$$NDS(N) = |P_s| \tag{4}$$

A higher value for NDS is preferable, signifying the presence of a sufficient number of solutions.

## 4 Experimental evaluation and results

For evaluation, we run each of NSGA-II and NSGA-III based on the parameters depicted in table 1. As it is depicted in the table, we used range for the parameters, namely, number of iterations and population size. This is because each algorithm is operated for more than one time to capture the statistical behavior of the results generated from each algorithm and to reflect this in boxplot. Furthermore, the simulation parameters are detailed in Table 2.

Table 1: The parameters of the algorithms used for evaluation

	NSGA-II	NSGA-III
Number of iterations min	100	100
Number of iterations max	200	200
Population size min	100	100
Population size max	200	200

Table 2: The parameters of the simulation used for evaluation

Parameter name	Min	Max
Area	100 × 100	100 × 100
Number of users	100	200
Number of base stations	20	20
Number of cloudlets	5	10

The experimental results are presented bases on the set coverage between NSGA-III, NSGA-II. The set coverage indicates to the percentage of domination of one algorithm over the other. We find from the results that the domination of NSGA-II over NSGA-III has gotten a median value equal to 0.25 compared with almost the same median value for NSGA-III over NSGA-II. In addition, we find that several experiments have generated higher domination of NSGA-III over NSGA-II and are plotted as outliers in the boxplot as shown in Figure 2. This means that NSGA-III has more potential of providing more dominant solutions than NSGA-II.

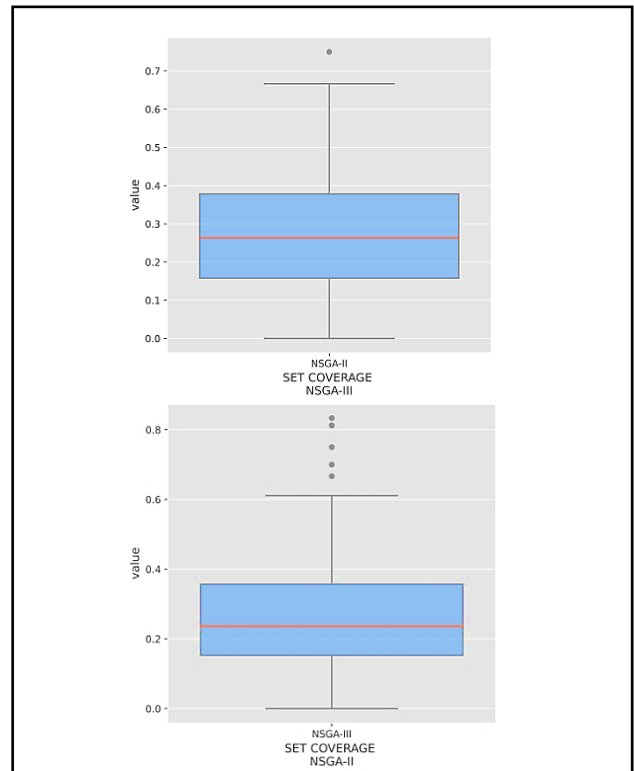


Figure 2: Boxplot visualization of the domination between NSGA-II, NSGA-III using set coverage.

The second metric that is visualized is the delta metric which indicates how much the solutions are equally distributed in the environment.

The lower the value of the delta metric, the more fairness of solutions distributions in the Pareto front. Observing Figure 3, we find that NSGA-III could generate some Pareto fronts with lower values of delta metric reaching below 0.75. Hence, NSGA-III had more potential of generating a more equally distributed Pareto front.

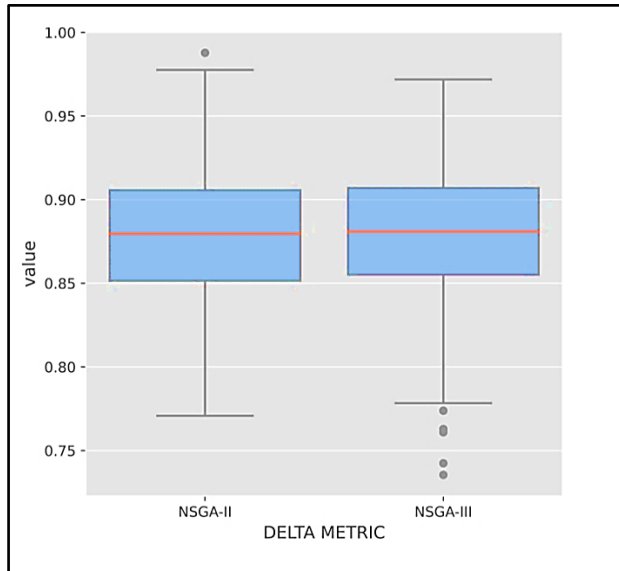


Figure 3: Delta metric visualization of NSGA-II, NSGA-III.

The third metric that is visualized is the number of non-dominated solutions. For this metric, we find that NSGA-II has generated some Pareto fronts with higher values of the number of non-dominated solutions reaching 35 compared with one Pareto front generated from NSGA-III with a value of 34 in Figure 4. Another observation is that the median values of the two algorithms are almost 20. However, NSGA-III has generated a higher value of the third percentile.

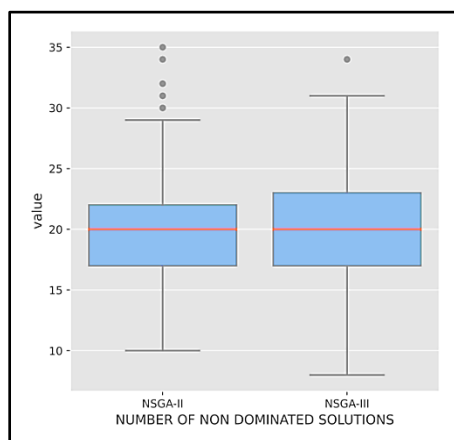


Figure 4: Number of non-dominated solutions metric visualization of NSGA-II, NSGA-III.

The last metric that is visualized is hyper-volume which indicates to the allocated volume of the solutions

in the objective space. Observing the results from Figure 5 provides that both algorithms have generated the same value of hyper-volume which is zero. This might be interpreted by the nature of solutions distributions in the space which have taken a straight-line. In other words, the solutions for all of experiments have at least two objective values that have never changed their values. This might be interpreted the nature of the optimization surface.

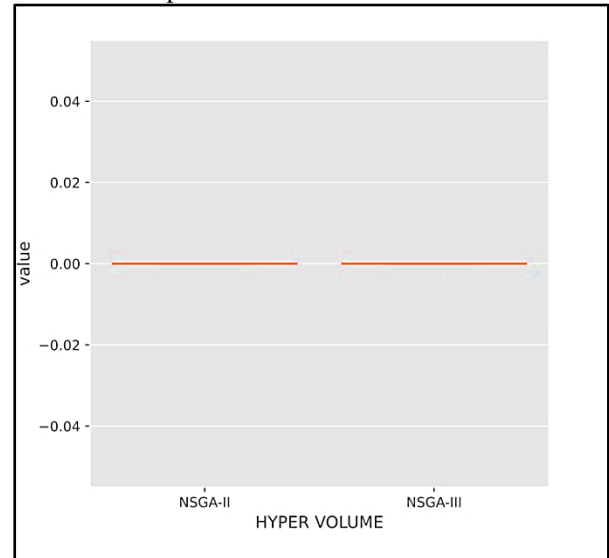


Figure 5: Hyper volume metric visualization of NSGA-II, NSGA-III.

## 5 Conclusion and future works

The article focuses on fog cloud computing management, which involves assigning tasks to fog nodes or sending them to the cloud while optimizing various objectives. The authors specifically examine cloudlets, which are local servers deployed in wireless metropolitan areas that communicate tasks between users and cloudlets. They propose a cloudlet computing management process based on multi-objective optimization using non-dominated sorting, with latency, energy consumption at the cloudlets and user, and cost (based on the number of cloudlets) as considered objectives.

Set coverage, delta measure, number of non-dominated solutions, and hyper-volume are some of the experimental metrics used to compare the two NSGA versions in this article. The set coverage metric shows that NSGA-III has more potential to provide dominant solutions than NSGA-II. The delta metric indicates that NSGA-III can generate Pareto fronts with lower values of delta metric, meaning more equally distributed solutions. While NSGA-III has produced greater values of third percentiles, the number of non-dominated solutions metric indicates that NSGA-II is capable of generating Pareto fronts with a greater quantity of non-dominated solutions. Both algorithms have generated the same value of hyper-volume, which is zero, possibly due to the optimization surface's nature. Overall, the results suggest that NSGA-III has some advantages over

NSGA-II in terms of providing more dominant solutions and equally distributed Pareto fronts.

## References

- [1] G. Caiza, M. Saeteros, W. Oñate, and M. V. Garcia, "Fog computing at industrial level, architecture, latency, energy, and security: A review," *Heliyon*, vol. 6, no. 4, p. e03706, 2020, doi: 10.1016/j.heliyon.2020.e03706.
- [2] H. Sabireen and V. Neelaranayanan, "A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges," *ICT Express*, vol. 7, no. 2, pp. 162–176, 2021, doi: 10.1016/j.ict.2021.05.004.
- [3] I. A. Najm *et al.*, "OLAP Mining with Educational Data Mart to Predict Students' Performance," *Inform.*, vol. 46, no. 5, pp. 11–19, 2022, doi: 10.31449/inf.v46i5.3853.
- [4] J. Singh, P. Singh, and S. S. Gill, "Fog computing: A taxonomy, systematic review, current trends and research challenges," *J. Parallel Distrib. Comput.*, vol. 157, pp. 56–85, 2021, doi: 10.1016/j.jpdc.2021.06.005.
- [5] J. U. Ren, D. Zhang, S. He, Y. Zhang, and T. A. O. Li, "A Survey on End-Edge-Cloud Orchestrated Network Computing Paradigms: Transparent Computing, Mobile Edge Computing, Fog Computing, and Cloudlet," *ACM Comput. Surv.*, vol. 52, no. 6, 2019.
- [6] W. A. Awadh, A. S. Alasady, and A. K. Hamoud, "Efficiently Secure Data Communications Based on CBC-RC6 and the Overflow Field of Timestamp Option in an IPv4 Packet," *Inform.*, vol. 46, no. 6, pp. 125–133, 2022, doi: 10.31449/inf.v46i6.4005.
- [7] J. Chen, T. Du, and G. Xiao, "A multi-objective optimization for resource allocation of emergent demands in cloud computing," *J. Cloud Comput.*, vol. 10, no. 1, 2021, doi: 10.1186/s13677-021-00237-7.
- [8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002, doi: 10.1109/4235.996017.
- [9] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, 2014, doi: 10.1109/TEVC.2013.2281534.
- [10] X. Zhu and M. C. Zhou, "Multiobjective Optimized Cloudlet Deployment and Task Offloading for Mobile-Edge Computing," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15582–15595, 2021, doi: 10.1109/JIOT.2021.3073113.
- [11] H. Guo, J. Zhang, J. Liu, and H. Zhang, "Energy-aware computation offloading and transmit power allocation in ultradense IoT networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4317–4329, 2019, doi: 10.1109/JIOT.2018.2875535.
- [12] B. Cao *et al.*, "Large-Scale Many-Objective Deployment Optimization of Edge Servers," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3841–3849, 2021, doi: 10.1109/TITS.2021.3059455.
- [13] Z. Wang, F. Gao, and X. Jin, "Optimal deployment of cloudlets based on cost and latency in Internet of Things networks," *Wirel. Networks*, vol. 26, no. 8, pp. 6077–6093, 2020, doi: 10.1007/s11276-020-02418-9.
- [14] J. Xue and Y. An, "Joint Task Offloading and Resource Allocation for Multi-Task Multi-Server NOMA-MEC Networks," *IEEE Access*, vol. 9, pp. 16152–16163, 2021, doi: 10.1109/ACCESS.2021.3049883.
- [15] C. He, R. Wang, D. Wu, H. Zhang, and Z. Tan, "QoS-aware hybrid cloudlet placement over joint fiber and wireless backhaul access network," *Opt. Switch. Netw.*, vol. 45, no. August 2020, p. 100678, 2022, doi: 10.1016/j.osn.2022.100678.
- [16] E. El Haber, T. M. Nguyen, and C. Assi, "Joint Optimization of Computational Cost and Devices Energy for Task Offloading in Multi-Tier Edge-Clouds," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3407–3421, 2019, doi: 10.1109/TCOMM.2019.2895040.
- [17] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective Optimization for Computation Offloading in Fog Computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, 2018, doi: 10.1109/JIOT.2017.2780236.
- [18] A. Khalili, S. Zarandi, and M. Rasti, "Joint Resource Allocation and Offloading Decision in Mobile Edge Computing," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 684–687, 2019, doi: 10.1109/LCOMM.2019.2897008.
- [19] L. Cui *et al.*, "Joint optimization of energy consumption and latency in mobile edge computing for internet of things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4791–4803, 2019, doi: 10.1109/JIOT.2018.2869226.
- [20] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C. H. Hsu, "Edge server placement in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 127, pp. 160–168, 2019, doi: 10.1016/j.jpdc.2018.06.008.
- [21] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient Algorithms for Capacitated Cloudlet Placements," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 10, pp. 2866–2880, 2016, doi: 10.1109/TPDS.2015.2510638.
- [22] Q. Fan and N. Ansari, "Workload Allocation in Hierarchical Cloudlet Networks," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 820–823, 2018, doi: 10.1109/LCOMM.2018.2801866.
- [23] Q. Fan and N. Ansari, "Application Aware Workload Allocation for Edge Computing-Based IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2146–2153, 2018, doi: 10.1109/JIOT.2018.2826006.

- [24] W. Mkaouer *et al.*, “Many-objective software remodularization using NSGA-III,” *ACM Trans. Softw. Eng. Methodol.*, vol. 24, no. 3, 2015, doi: 10.1145/2729974.
- [25] A. Metiaf, Q. Wu, and Y. Aljeroudi, “Searching with direction awareness: Multi-objective genetic algorithm based on angle quantization and crowding distance MOGA-AQCD,” *IEEE Access*, vol. 7, pp. 10196–10207, 2019, doi: 10.1109/ACCESS.2018.2890461.