*Intelligent and Converged Networks*
ISSN 2708-6240
2023, 4(4): 273–279
DOI: 10.23919/ICN.2023.0022

# Enhancing resource allocation in edge and fog-cloud computing with genetic algorithm and particle swarm optimization

Saad-Eddine Chafi*, Younes Balboul, Mohammed Fattah, Said Mazer, and Moulhime El Bekkali

**Abstract:** Evolutionary algorithms have gained significant attention from researchers as effective solutions for various optimization problems. Genetic Algorithm (GA) is widely popular due to its logical approach, broad applicability, and ability to tackle complex issues encountered in engineering systems. However, GA is known for its high implementation cost and typically requires a large number of iterations. On the other hand, Particle Swarm Optimization (PSO) is a relatively new heuristic technique inspired by the collective behaviors of real organisms. Both GA and PSO algorithms are prominent heuristic optimization methods that belong to the population-based approaches family. While they are often seen as competitors, their efficiency heavily relies on the parameter values chosen and the specific optimization problem at hand. In this study, we aim to compare the runtime performance of GA and PSO algorithms within a cutting-edge edge and fog cloud architecture. Through extensive experiments and performance evaluations, the authors demonstrate the effectiveness of GA and PSO algorithms in improving resource allocation in edge and fog cloud computing scenarios using FogWorkflowSim simulator. The comparative analysis sheds light on the strengths and limitations of each algorithm, providing valuable insights for researchers and practitioners in the field.

**Key words:** particle swarm optimization; genetic algorithm; performance evaluation; edge and fog cloud; FogWorkflowSim

## 1 Introduction

The idea of heuristic optimization is not new. Although the Particle Swarm Optimization (PSO) was originally suggested in 1995[1] and the first Genetic Algorithm (GA) was created in 1975[2], both are still regarded as recent approaches as seen in Ref. [3]. These two are among the most recognizable figures in the most esoteric class of optimization approaches, the majority of which get their inspiration from occurrences in nature. They are typically used for nonlinear problems with expansive and intricate design spaces or discontinuous objective functions that are extremely challenging or impossible to solve using conventional techniques. Both GA and PSO are population-based methods that guide a group of potential answers to the given problem toward an ideal solution. Using bio evolutionary principles including crossover, mutation, and selection based on fitness, GA replicates the natural development of organisms. PSO is based on the social behavior of big groups, such fish schools or swarms of flying birds. Finding the ideal shape for a structure that must bear specific loads and adhere to specified boundary conditions is the focus of structural optimization. In recent years, researchers' interest in the use of GA[3–9] and PSO[10–12] in structural optimization issues has grown significantly. Previous studies[13–15] have compared the effectiveness of GA

• Saad-Eddine Chafi, Younes Balboul, Said Mazer, and Moulhime El Bekkali are with the Laboratory of Artificial Intelligence, Data Sciences and Emerging Systems, National School of Applied Sciences, Sidi Mohamed Ben Abdellah University, Fez 30000, Morocco. E-mail: saad.chafi@usmba.ac.ma; younes.balboul@usmba.ac.ma; said.mazer@usmba.ac.ma; moulhime.elbekkali@usmba.ac.ma.
• Mohammed Fattah is with the LIA Laboratory, Moulay Ismail University, Meknes 50050, Morocco. E-mail: m.mohammedfattah@umi.ac.ma.
∗ To whom correspondence should be addressed.
Manuscript received: 2023-04-10; accepted: 2023-05-30

and PSO in structural optimization[12–14]. The choice of these factors is crucial to the effectiveness of the approach since they have a significant impact on how the methods behave.

This research suggests a study in which these crucial variables are adjusted throughout a range, and only the most effective configurations of the two approaches are evaluated in terms of performance. It is important to note that several researchers have investigated the idea of combining GA and PSO in a single algorithm in order to benefit from both the evolutionary elements of GAs and the individual data exchange capabilities unique to PSO[14, 15]. The authors claimed that this integrated strategy can result in more effective methods. After this brief introduction, the remaining sections of the paper are arranged as follows: section 2 gives a summary of the of related work. Section 3 presents the operating principle of the GA algorithm. Section 4 gives an overview of the PSO algorithm and its implementation. Section 5 summarizes the runtime performance evaluation of the two algorithms in an edge & fog cloud architecture. The last section presents the conclusion of this article.

## 2 Related work

Holland[16] proposed a formal framework for tackling complex optimization problems that exhibit two key features significant difficulty and original uncertainty. The framework addresses the need to incorporate new information as it becomes available to improve average results at a rate that matches the pace of information acquisition. Additionally, it emphasizes the importance of utilizing new information to achieve the best possible results. The idea of optimizing nonlinear functions using the particle swarm approach was first presented by Kennedy and Eberhart[1]. The development of multiple paradigms is described, and one of the paradigms' implementations is addressed. Applications including neural network training and nonlinear function optimization are suggested, and benchmark testing of the paradigm is described. The area of Swarm Intelligence (SI), which arose from biological study and provides the many mathematical models of social insect collective behavior, has been thoroughly introduced by Engelbrecht, who also demonstrates how they may be utilized to solve optimization issues[17]. Nedjah et al.[18] put forth highly creative and intriguing propositions for leveraging and refining the theory and techniques of multi-objective swarm intelligence. Their work involves imitating social swarm behaviors to solve optimization problems based on multiple criteria. The ideas presented by them are creative and intriguing, making significant contributions to the field of swarm intelligence. According to a comparison study by Kumar et al.[19], the H-best Particle Swarm Optimization (HPSO) performs better than the global best (gbest) PSO in terms of localization that is quicker, more developed, and more precise. Singh et al.[20] suggested the use of different migratory forms of biogeography-based optimization approaches and swarm optimization, including PSO, to achieve optimal localization of randomly positioned sensors in a distributed fashion. The efficacy of the proposed techniques was assessed on the basis of the number of nodes identified, localization precision, and computational time, using experimental sensor network data.

The study[20] utilized a distributed iterative localization investigation approach. The component switching problem was addressed using a genetic algorithm by Maimon and Braha[21]. GA was appealing because of its resilience and simplicity, especially when paired with current processing power. This method overcomes the drawback of isolating the Printed Circuit Boards (PCB) by handling "look-ahead" consideration of component switching. Different optimization algorithms are presented by Singh et al.[22] with significant calculation time speedups.

## 3 Genetic algorithm

One significant type of evolutionary algorithms is the genetic algorithm. Prof. John Holland employed the genetic algorithm for the first time in 1975[16]. GA often offers rough answers to the different issues. GAs utilize various biological mechanisms, such as reproduction, crossover or recombination, mutation, and inheritance, to optimize solutions to complex problems. Instead, it operates directly on real-valued chromosomes, enabling faster convergence and improved search efficiency. This algorithm's different steps include the following:

(1) Create a starting population haphazardly or heuristically.

(2) Determine each person's fitness level within the population.

(3) Every member should have a selection probability that is proportionate to their fitness value.

(4) By choosing the ideal individuals to generate offspring, one may create the following generation from the present generation.

(5) Up until a workable solution is identified, repeat the processes.

(6) Based on GA principles, a population refers to a collection of particles, while a chromosome represents an individual particle. Once chromosomes are created, the subsequent step involves evaluating their fitness referred to as a fitness function. This cost function serves to assess the efficacy of the solution contained within each chromosome, with greater fitness corresponding to superior solutions. Frequently, the following are a few of the GA-related processes:

- **Selection.** According to the fitness criterion, this method is often employed to determine which chromosome will continue to reproduce.
- **Reproduction.** This process is used to create the following generation from the present one.
- **Crossover.** The genetic material between the chromosomes is exchanged through this method. Crossover can be done at one or more points.
- **Mutation.** One person's chromosomes alter as a result of this procedure. The algorithm is kept from becoming stuck at a specific point through mutation.
- **Stopping criteria.** The GA process ends with this. When the intended out-come is reached or the maximum number of cycles is reached, the iteration comes to an end.

**Implementation algorithm.** By utilizing a predetermined fitness function, GA causes the creation of the fittest members after each iteration. GA's fundamental flowchart is shown in Fig. 1.

**Applications.** A wide range of disciplines may make use of GAs. It is mostly employed to address optimization issues. Bioinformatics, computational science, electrical engineering, manufacturing, and phylogenetics are a few of the industries that employ GA.

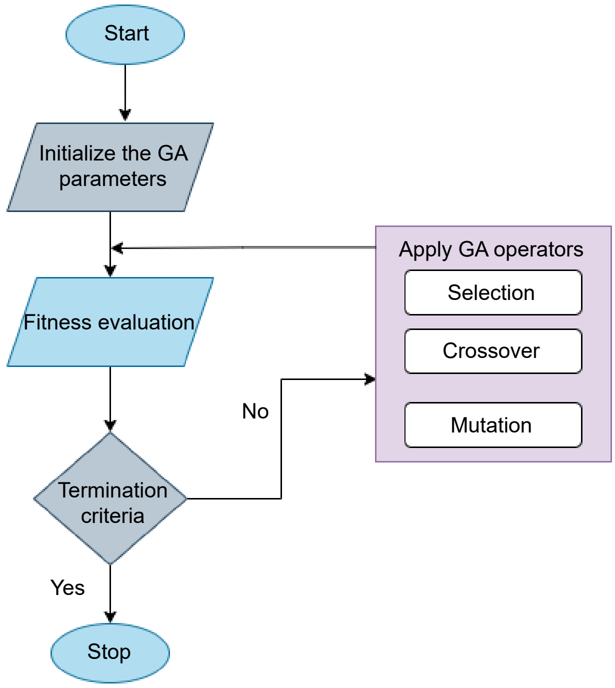


**Fig. 1 Basic implementation of GA.**

## 4 Particle swarm optimization

A swarm of particles is defined by a significant class of evolutionary algorithms called PSO. The swarm's particles are subsequently modified in accordance with the predetermined regulations. The prior position of the particle and its best-known position throughout the whole search area affect how the particle values change[23]. In order to progress towards the desired local best (pbest) and gbest location, each particle's position and velocity are adjusted after each iteration. The particles are first placed in a random location at the start of the search process. By weighing the acceleration coefficients with random factors, it is possible to increase the effectiveness solution[24]. For acceleration, the pbest and gbest sites both provide unique random numbers[25]. Let the $k$-th particle in the swarm be represented as in the context of the $n$-dimensional search space.

Let $X_k = \left(x_{k_1}, x_{k_2}, \ldots, x_{k_d}\right)$ and let $V_k$ be another $n$-

dimensional vector representing its velocity $(v_{k_1}, v_{k_2}, \ldots, v_{k_d})$. Let $P_k$ represent the best point that the $k$-th particle has visited $(p_{k_1}, p_{k_2}, \ldots, p_{k_d})$. The best particle generally is designated as $P_g$ and the best particle personally is marked as $P_g = (p_{g_1}, p_{g_2}, \ldots, p_{g_d})$, where $g$ and $p$ are particle indices. The following formulas can be used to update the particle's location and velocity:

$$X_{k_d}(t+1) = X_{k_d}(t) + V_{k_d}(t) \tag{1}$$

and

$$V_{k_d}(t+1) = \chi(v_{k_d}(t) + l_1 c_1 (P_{\text{pb}_d}(t) - X_{k_d}(t)) + l_2 c_2 (P_{\text{gb}_d}(t) - X_{k_d}(t))) \tag{2}$$

The learning factors, or $l_1$ and $l_2$, are non-negative constants in the equation above. Additionally, $c_1$ and $c_2$ are some randomly produced values in the [0, 1] range.

**PSO implementation.** PSO is an evolutionary method that needs random number generation. The amount and quality of the produced numbers have an impact on the PSO algorithm's performance. Figure 2 illustrates how PSO is fundamentally implemented.

The PSO algorithm's numerous stages are listed below:

• Set the particle's initial coordinates and velocities at random locations in the search space.

• Start computing the swarm particle's fitness function's associated value.

• Compare the evaluation of the fitness value to the particle's present pbest value. If the current value is better than the pbest value, the pbest position in the $n$-dimensional space should be changed to the current location.

• The next step is to compare the fitness value to the prior overall best. gbest is reset to the array index and value of the current particle if the current value is superior to it.

• Finally, apply these values to the swarm particle's matching position and speed.

**PSO variants.** By fusing the PSO algorithm with other evolutionary algorithms, different iterations of the PSO algorithm may be created. In order to enhance the algorithm's overall optimization, hybrid PSO algorithms are becoming more popular in research. The PSO algorithm has a few popular iterations,

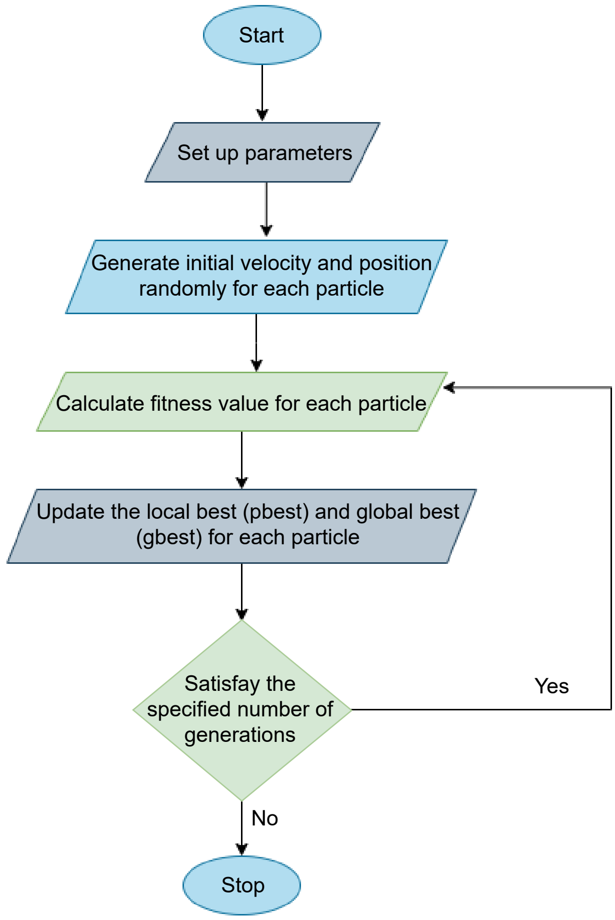


**Fig. 2 Basic application of the technique for particle swarm optimization.**

including[26]

• Distinct PSO;

• Coefficient of constriction;

• Bare-bones PSO;

• PSO fully informed.

**Applications.** Originally, PSO was developed for training neural networks. However, over time, PSO has found numerous applications in various industries, such as design[27], power systems, control, and many others. The usage of PSO algorithms in dynamic tracking, MinMax issues, and other optimization[28] issues are widespread.

## 5 Performance evaluation

The WorkflowSim was used to develop the GA and PSO algorithms in order to evaluate the suggested approach[29]. The WorkflowSim adds to the capabilities of the already-existing CloudSim simulator[30] by offering a greater level of workflow management and a suitable environment for implementing various scheduling algorithms. In order to verify our algorithm, metrics such as execution time, energy consumption, cost, and execution time of the algorithm are used. PSO and GA algorithms are developed and tested on the same configuration in order to make relevant comparisons. Tables 1 and 2 contain a list of the configuration's specifications and the configuration of the GA and PSO algorithms.

The initial level of the hierarchy in our simulations for the fog devices is the "MobileDevice", which is coupled to the sensors and actuators. Level 2 contains the gateway that connects the "MobileDevice" to Level 3, or the cloud[31] data center[32], through the operator access network. In the simulations, it is assumed that all sensors have an identical sensing frequency and that fog devices at the same hierarchical level are homogeneous. Additionally, the linkages between each network need to be configured. The architecture of the scenario is shown in Fig. 3.

After several test, we noted that the PSO algorithm takes less time to be executed compared to the GA algorithm about 50% of the PSO task execution, also for the energy consumption, we notice that the PSO

**Table 1 WorkflowSim simulation parameters.**

| Parameter | Cloud | Gateway | EndDevice |
|---|---|---|---|
| Million Instructions per Second (MIPS) | 1600 | 1300 | 1000 |
| RAM (MB) | 40 000 | 4000 | 1000 |
| Uplink bandwidth (KB/s) | 10 000 | 10 000 | 10 000 |
| Downlink bandwidth (KB/s) | 10 000 | 10 000 | 270 |
| Level hierarchy | 0 | 1 | 2 |
| Rate per MIPS | 0.96 | 0.48 | 0.05 |
| Uplink latency (ms) | None | 40 | 20 |

**Table 2 Parameters of GA and PSO.**

| Parameter | | Value |
|---|---|---|
| PSO | Number of particles | 20 |
| | Number of iterations | 100 |
| | Learning factor $c_1$ | 1 |
| | Learning factor $c_2$ | 1 |
| | Inertia weight | 0.73 |
| GA | Population size | 20 |
| | Number of iterations | 100 |
| | Cross rate | 0.8 |
| | Mutation rate | 0.01 |

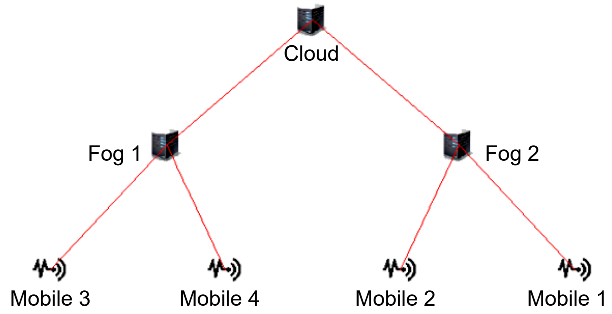


**Fig. 3 Simulated architecture on WorkflowSim.**

algorithm consumes a little bit more energy compared with the GA algorithm about 5% less. For the execution of the two algorithms, we see that the PSO takes 746 ms to be executed in contrast of the GA who takes 859 ms to be executed about 14% of the difference. Furthermore, for the cost the PSO can be implemented with a minimal cost compared with the GA algorithm about 42% of the resource need to be used. The comparison is shown in Fig. 4.

PSO algorithm outperforms GA algorithm in terms of task execution time and algorithm execution. However, GA algorithm exhibits lower energy consumption and cost. The choice between the two algorithms depends on the specific requirements and trade-offs of the application or problem being addressed.

## 6 Conclusion

PSO and GA are both evolutionary search techniques, meaning that they go from one set of points to another inside an iteration while clearly improving over the prior values utilizing a combination of probabilistic and deterministic principles. From the simulation results, it is clear that PSO algorithm gives lower cost and fast

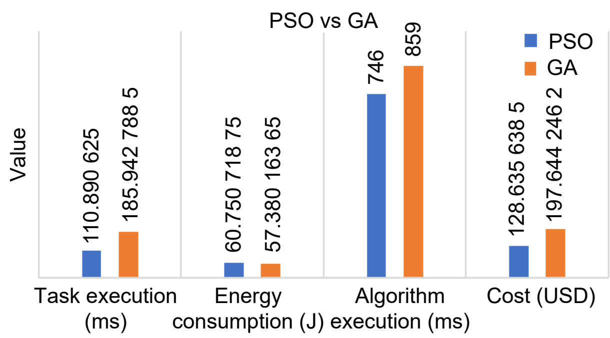


**Fig. 4 Comparison between GA and PSO algorithms.**

execution compared to GA algorithm. On the other hand, GA algorithm offers slightly reduced energy consumption compared to PSO. Although GA and PSO are both significant components of evolutionary optimization methods, they both have drawbacks that restrict their use to a small number of issues. A combination of GA and PSO can be utilized to solve these issues and boost performance in general.

## References

[1] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proc. Int. Conf. Neural Networks* (*ICNN'95*), Perth, Australia, 1995, pp. 1942–1948.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*: *An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[3] R. Cazacu, Comparison between the performance of GA and PSO in structural optimization problems, *Am. J. Eng. Res.*, vol. 5, no. 11, pp. 268–272, 2016.

[4] O. Buiga and C. O. Popa, Optimal mass design of a single-stage helical gear unit with genetic algorithms, *Proc. Rom. Acad. Ser. A Math. Phys. Tech. Sci. Inf. Sci.*, vol. 13, no. 3, pp. 243–250, 2012.

[5] R. Cazacu and L. Grama, Steel truss optimization using genetic algorithms and FEA, *Procedia Technol.*, vol. 12, pp. 339–346, 2014.

[6] K. Deb and S. Gulati, Design of truss-structures for minimum weight using genetic algorithms, *Finite Elem. Anal. Des.*, vol. 37, no. 5, pp. 447–465, 2001.

[7] N. Noilublao and S. Bureerat, Simultaneous topology, shape and sizing optimisation of a three-dimensional slender truss tower using multiobjective evolutionary algorithms, *Comput. Struct.*, vol. 89, no. 23&24, pp. 2531–2538, 2011.

[8] J. E. Rodriguez, A. L. Medaglia, and J. P. Casas, Approximation to the optimum design of a motorcycle frame using finite element analysis and evolutionary algorithms, in *Proc. 2005 IEEE Design Symp., Systems and Information Engineering*, Charlottesville, VA, USA, 2005, pp. 277–285.

[9] K. Sourabh, S. S. Chauhan, and V. Kumar, A review on genetic algorithm: Past, present, and future, *Multimed. Tools Appl.*, vol. 80, pp. 8091–8126, 2021.

[10] P. C. Fourie and A. A. Groenwold, The particle swarm optimization algorithm in size and shape optimization, *Struct. Multidiscip. Optim.*, vol. 23, no. 4, pp. 259–267, 2002.

[11] R. E. Perez and K. Behdinan, Particle swarm approach for structural design optimization, *Comput. Struct.*, vol. 85, no. 19-20, pp. 1579–1588, 2007.

[12] S. Saremi, S. M. Mirjalili, and S. Mirjalili, Unit cell topology optimization of line defect photonic crystal waveguide, *Procedia Technol.*, vol. 12, pp. 174–179, 2014.

[13] R. Cazacu and L. Grama, Structural optimization with genetic algorithms and particle swarm optimization, *Ann. ORADEA UNIVERSITY Fascicle Manag. Technol. Eng.*, vol. 12, no. 22, pp. 19–22, 2013.

[14] R. C. Eberhart and Y. Shi, Comparison between genetic algorithms and particle swarm optimization, in *Proc. 7th Int. Conf. Evolutionary Programming*, San Diego, CA, USA, 1998, pp. 611–616.

[15] M. R. Maheri, M. Askarian, and S. Shojaee, Size and topology optimization of trusses using hybrid genetic-particle swarm algorithms, *Iran. J. Sci. Technol. Trans. Civ. Eng.*, vol. 40, no. 3, pp. 179–193, 2016.

[16] J. H. Holland, Genetic algorithms and the optimal allocation of trials, *SIAM J. Comput.*, vol. 2, no. 2, pp. 88–105, 1973.

[17] J. Kennedy, Review of engelbrecht's fundamentals of computational swarm intelligence, *Genet. Program. Evolvable Mach.*, vol. 8, no. 1, pp. 107–109, 2007.

[18] N. Nedjah, L. dos Santos Coelho, and L. de Macedo de Mourelle, Multi-objective swarm intelligent systems: Theory & experiences, Heidelberg, Germany: Springer, 2009.

[19] A. Kumar, A. Khosla, J. S. Saini, and S. Singh, Meta-heuristic range based node localization algorithm for Wireless Sensor Networks, in *Proc. 2012 Int. Conf. Localization and GNSS*, Starnberg, Germany, 2012, pp. 1–7.

[20] S. Singh, Shivangna, and E. Mittal, Range based wireless sensor node localization using PSO and BBO and its variants, in *Proc. 2013 Int. Conf. Communication Systems and Network Technologies*, Gwalior, India, 2013, pp. 309–315.

[21] O. Z. Maimon and D. Braha, A genetic algorithm approach to scheduling PCBs on a single machine, *Int. J. Prod. Res.*, vol. 36, no. 3, pp. 761–784, 1998.

[22] S. Singh, J. Kaur, R. S. Sinha, A comprehensive survey on various evolutionary algorithms on GPU, in *Proc. Int. Conf. Communication, Computing & Systems* (*ICCCS–2014*), Punjab, India, 2014, pp. 83–88.

[23] D. D. Ramírez-Ochoa, L. A. Pérez-Domínguez, E. A. Martínez-Gómez, and D. Luviano-Cruz, PSO, a swarm intelligence-based evolutionary algorithm as a decision-making strategy: A review, *Symmetry*, vol. 14, no. 3, p. 455, 2022.

[24] K. Habib, X. Lai, A. Wadood, S. Khan, Y. Wang, and S. Xu, Hybridization of PSO for the optimal coordination of directional overcurrent protection relays, *Electronics*, vol. 11, no. 2, p. 180, 2022.

[25] C. Davi and U. Braga-Neto, PSO-PINN: Physics-informed neural networks trained with particle swarm optimization, arXiv preprint arXiv: 2202.01943, 2022.

[26] S. Javed and K. Ishaque, A comprehensive analyses with new findings of different PSO variants for MPPT problem under partial shading, *Ain Shams Eng. J.*, vol. 13, no. 5, p. 101680, 2022.

[27] B. Younes, F. Mohammed, M. Saïd, and M. El Bekkali, 5G uplink interference simulations, analysis and solutions: The case of pico cells dense deployment, *Int. J. Electr. Comput. Eng. IJECE*, vol. 11, no. 3, p. 2245, 2021.

[28] S. E. Chafi, Y. Balboul, M. Fattah, S. Mazer, M. El Bekkali, and B. Bernoussi, Resource placement strategy optimization for IoT oriented monitoring application, *TELKOMNIKA Telecommun. Comput. Electron. Contr.*, vol. 20, no. 4, p. 788, 2022.

[29] W. Chen and E. Deelman, WorkflowSim: A toolkit for simulating scientific workflows in distributed environments, in *Proc. 2012 IEEE 8th Int. Conf. E-Science*, Chicago, IL, USA, 2013, pp. 1–8.

[30] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Softw. Pract. Exp.*, vol. 41, no. 1, pp. 23–50, 2011.

[31] M. El Bekkali, B. Bernoussi, M. Fattah, S. Mazer, Y. Balboul, and S. E. Chafi, Cloud computing services, models and simulation tools, *Int. J. Cloud Comput.*, vol. 10, nos. 5&6, p. 533, 2021.

[32] C. Saad-Eddine and B. Younes, Performance & energy consumption metrics of A data center according to the energy consumption models cubic, linear, square and square root, in *Proc. 2019 7th Mediterranean Congress of Telecommunications* (*CMT*), Fez, Morocco, 2019, pp. 1–5.

**Saad-Eddine Chafi** is currently a PhD candidate at the National School of Applied Sciences, Sidi Mohamed Ben Abdellah University (USMBA), Morocco. He is a member of the Laboratory of Artificial Intelligence, Data Science and Emerging Systems, USMBA, Morocco. His research interests include cloud-computing, quality of service of cloud, and Internet of Things.

**Younes Balboul** received the PhD degree in telecommunications from Sidi Mohamed Ben Abdellah University (USMBA), Morocco in 2016. He is currently a professor at the National School of Applied Sciences, USMBA, Morocco. He is a member of Artificial Intelligence, Data Sciences and Emerging Systems Laboratory, USMBA, Morocco.

**Said Mazer** received the PhD degree in electronics and signal processing from the University of Marne-La-Vallée, France in 2006. He is currently a full professor with the National School of Applied Sciences, Sidi Mohamed Ben Abdellah University (USMBA), Morocco. He is member of Artificial Intelligence, Data Sciences and Emerging Systems Laboratory (LIASSE), USMBA, Morocco. His research interests include the development of microwave-photonics devices for radio-over fibre and wireless applications, and he is involved in network security.

**Mohammed Fattah** received the PhD degree in telecommunications from Sidi Mohamed Ben Abdellah University, Fez, Morocco, in 2011. He is a professor in the Electrical Engineering Department, High School of Technology, Moulay Ismail University (UMI), Morocco. He is a member of the information processing and transmission research team, LIA Laboratory, UMI.

**Moulhime El Bekkali** received the PhD degree from USTL, France in 1991. USTL University - Lille 1- France. He was a professor at the Higher School of Technology of Fez (ESTF), Morocco and he was a member of the Transmission and Data Processing Laboratory. In 1999, he received a second PhD degree in electromagnetic compatibility from Sidi Mohamed Ben Abdellah University (USMBA), Morocco. During 2009–2018, he was the vice-president of Research and Cooperation, Sidi Mohamed Ben Abdellah University, Morocco. Currently, he is a professor at the National School of Applied Sciences, and a member of the LIASSE Laboratory, USMBA, Morocco.