Article history:
Received 03 June 2025
Revised 10 September 2025
Accepted 17 September 2025
Published online 20 October 2025

Journal of Resource Management and Decision Engineering

Volume 4, Issue 4, pp 1-9

Firefly Optimization Algorithm for Multi-Objective Job Scheduling in Cloud Computing

Vahid. Mokhtari[1], Nasser. Mikaeilvand[2*], Abbas. Mirzaei[3], Babak. Nouri-Moghaddam[3], Sajjad. Jahanbakhsh Gudakahriz[4]

[1] Department of Computer, Faculty of Engineering, Qeshm International Branch, Islamic Azad University, Qeshm, Iran
[2] Department of Computer Science and Mathematics, CT.C., Islamic Azad University, Tehran, Iran
[3] Department of Computer Engineering, Ard.C., Islamic Azad University, Ardabil, Iran
[4] Department of Computer Engineering, Germi.C., Islamic Azad University, Germi, Iran

*** Corresponding author email address**: nasser.mikaeilvand@iau.ac.ir

Article Info

Article type:
Original Research

How to cite this article:
Mokhtari, V., Mikaeilvand, N., Mirzaei, A., Nouri-Moghaddam, B., & Jahanbakhsh Gudakahriz, S. (2025). Firefly Optimization Algorithm for Multi-Objective Job Scheduling in Cloud Computing. *Journal of Resource Management and Decision Engineering*, *4*(4), 1-9.
https://doi.org/10.61838/kman.jrmde.154

ABSTRACT

Due to the increasing use of the Internet of Things, efficient task scheduling in cloud computing has become increasingly important with the aim of maximizing the use of available resources, reducing energy consumption, and enhancing the quality of service (QoS). In this paper, we use the Firefly Optimization (FFO) algorithm to improve scheduling efficiency and minimize the overall completion time in cloud environments. For this purpose, twelve distinct scenarios were designed in the Cooja Contiki simulator environment with the perspective of computationally intensive, input/output intensive, and mixed workloads, and the overall completion time results obtained with the Min-Min and GA-PSO-Min methods were compared and the better performance of the method was confirmed.

Keywords: *Firefly Optimization Algorithm, Internet of Things, Cloud Computing, Job Scheduling, Total Time Spent, Efficiency in Energy Use, Scalability and Multi-Objective Optimization.*

1. Introduction

In the cloud-fog environment, different types of virtual machines are used for computing, which are executed in the fog or cloud depending on the importance of execution time and the volume of calculations. In the cloud sector; the scheduling problem is a critical component with the aim of maximizing resource efficiency, reducing the completion time of each task, reducing the overall execution time (span and flow time), and also the cost of execution while maintaining the quality of service. The continuity and dynamics of the continuous execution of these tasks have multiplied the importance of the problem. Since the tasks presented are often interdependent and do not have the same priority; examining and presenting a suitable method for continuous and rapid prioritization of tasks is one of the important and up-to-date issues studied by researchers. Since the cloud scheduling problem is an NP-complete problem, investigating and finding suitable solutions is computationally challenging.

Many of existing schedulers cannot cope with the dynamic nature of cloud systems and frequent changes in resource status. As a result, problems such as insufficient gap-filling during backfilling operations and project completion delays become more serious (Murad et al., 2024). For its efficient use, it is essential to design a scheduler that is not only more comprehensive but also more configurable.

Furthermore, with increasing user expectations for privacy, enhanced security while increasing or maintaining the quality of service (QoS) at the current level, and minimizing the number of service level agreements (SLAs) that involve compliance violations, the development of advanced scheduling strategies is essential to ensure cost-effective, scalable, and responsive computing infrastructures. This issue has impacted various industries, including IoT, smart homes, and wearable technologies (Khezri et al., 2024). Considerable efforts have been made to investigate task scheduling in cloud computing, with the aim of maximizing resource efficiency and increasing overall system effectiveness, with previous studies mainly focusing on various methods such as First-in-First-Serve (FCFS) and Shortest Job First (SJF) due to their ease of implementation (Paulraj et al., 2023).

The main objective of this research is to improve the efficiency of task scheduling in cloud environments by considering all the complexities of cloud scheduling that depend on the dynamic nature of resources and changing needs of users. This study proposes a novel approach that integrates multi-objective optimization to simultaneously minimize critical performance metrics such as completion time, flow time, and overall latency.

The wide variety of task requirements and fluctuations in resource availability lead to the increasing complexity of task scheduling in cloud environments and pose significant challenges in achieving optimal performance in terms of overall completion time and resource utilization. The main goal in designing scheduling optimization algorithms is to reduce the time required to complete large-scale distributed cloud tasks while maintaining computational efficiency.

Nature-Inspired Algorithms are among the most powerful optimization algorithms. The Firefly Optimization Algorithm (FOA) is one of the most powerful of these algorithms. An important feature of the Firefly Optimization Algorithm, which distinguishes it from some similar optimization algorithms, is its excellent performance in finding optimal solutions to multimodal problems and functions. Such an important feature of the Firefly optimization algorithm has made it an ideal choice for multi-mode optimization applications. In particular, it appears to be very powerful in solving NP-Hard problems, such as task scheduling, and converges to the global optimal solution in a very reasonable time.

In this paper, we will use the Firefly optimization algorithm to provide a scalable and practical solution that increases resource efficiency and speeds up the execution of activities, especially those with time-sensitive workloads in the cloud. The remaining parts of this work are structured as follows: In Section 2, we will review the related work that has been done on heuristic scheduling algorithms in cloud systems. We will also get acquainted with the history and applications of the Firefly optimization algorithm. The proposed Firefly optimization algorithm, including the algorithm architecture, will be presented in Section 3. In Section 4, the performance of the proposed technique is evaluated using data collected from experiments. In addition, a comparison is made between the total completion times of the proposed algorithm with Min-Min and GA-PSO-Min methods in a wide range of sparse and diverse situations, and figures and tables provide assistance based on the situation requirements. The results, contributions, and suggestions for further research are summarized in Section 5, which serves as the conclusion of the study.

2. Literature Review and Previous Work

This section defines and introduces the subject literature's concepts, methods, and standard definitions and terminology. Subsequently, previous studies are reviewed.

2.1. Task scheduling in Cloud-fog environments

The task scheduling problem in cloud-fog environments has become one of the main issues studied by many researchers due to the widespread use of the Internet of Things. For example, Abedinzadeh and Akiol (Abedinzadeh & Akyol, 2023) proposed the AEOSA algorithm to improve the efficiency of more heterogeneous datasets. Zhang et al. (Zhang et al., 2022) proposed the SCC-DSO algorithm to optimize queues based on storage location. Liu et al. (Liu et al., 2018) designed a framework for cloud-fog (IoT) for clustering and decentralized scheduling. Zhao et al. (Zhao et al., 2023) proposed a location-aware scheduling strategy for autonomous tasks to reduce the execution time by using data replication. Sobhanayak et al. (Sobhanayak et al., 2018) proposed a hybrid initiative influenced by biological processes and optimized energy generation and consumption time. Ghobaei-Arani et al. (Ghobaei-Arani et al., 2018) proposed a hybrid reinforcement learning and self-directed strategy to allocate resources and improve efficiency by using the MAPE cycle.

A Tabu-Harmony hybrid was developed to improve throughput and reduce task execution time in (Alazzam et al., 2019). In (Devaraj et al., 2020), the firefly algorithm and the improved multi-objective PSO (IMPSO) are combined to optimize response time and resource utilization while ignoring memory and cost considerations. Whale optimization is used in (Reddy & Kumar, 2017) to provide a better balance between resource utilization, energy consumption, quality of service, and better performance of alternatives in terms of energy efficiency.

A heuristic task scheduling strategy (TSO-MCR) was introduced in the field of cloud computing (Boroumand et al., 2025). A hybrid fuzzy meta-heuristic approach called IVPTS was designed for quality of service (QoS) and was introduced in (Long et al., 2025). In (Chen et al., 2025), a customer-oriented multi-task scheduling model for cloud production was proposed. The advanced Willow Cat optimization (AWCO) technique is introduced to improve task scheduling in cloud computing (Pan et al., 2025). In (Zade et al., 2025), Mohammad Hassanizadeh et al. proposed an improved multi-objective Beluga Whale optimization approach with ring topology (MO-IBWO-Ring) for multi-objective task scheduling in cloud computing, focusing on minimizing time and cost.

In (Khaledian et al., 2025), a hybrid Markov chain-based dynamic scheduling architecture is proposed to improve load balancing in cloud-fog conditions. Modified parallel particle swarm optimization (MPPSO) is designed to achieve the goal of improving task scheduling in cloud computing (Pradhan et al., 2025). Archimedes optimization algorithm with deadline and budget constraints (ADB) is developed for scheduling workflows in cloud computing (Kushwaha & Singh, 2025). A hybrid GA-PSO-Min approach is proposed in (Mokhtari et al., 2025) for multi-objective optimization or adaptation to changing cloud conditions.

2.2. Min-Min Scheduling strategy

The Min-Min scheduling heuristic is used in parallel and distributed computing to map tasks to processors. The algorithm works as follows:

1. The completion time of each task is calculated on each available machine.
2. The task with the minimum completion time is selected among all machines and assigned to the machine where it completes faster.
3. The above two steps are repeated iteratively, and the completion time is calculated after each assignment until all tasks are scheduled.

The goal of this approach is to minimize the overall completion time (total completion time) by prioritizing tasks that can be completed quickly and thus reduce idle time on machines.

Despite the relatively simple implementation of Min-Min, because its effectiveness depends on the characteristics of tasks and machines, and especially in heterogeneous environments where machines have different processing capabilities and tasks have different computational loads; it does not lead to finding a global optimum. Therefore, it is always recommended to consider hybrid or adaptive approaches that change strategies based on workload characteristics.

2.3. Firefly optimization algorithm

To date, about 2,000 different species of fireflies have been recorded in the world, and most of them produce short, rhythmic flashes of light. Usually, each species of firefly produces a unique and unique flashing light pattern. The flashing light emitted by fireflies is produced by a biological process called bioluminescence, which causes the fireflies to glow. Researchers have found that fireflies use flashing lights as a protective mechanism to send warnings to other fireflies in the environment. The rhythm or frequency of the flashing light, the rate of the light flashing, and the duration of the light flashing by the fireflies form different parts of the communication system between the worms.

One thing to remember about the light flashing pattern of fireflies is that the “Light Intensity” at a given distance, r, from the “Light Source” follows the “Inverse Square Law.” In addition, the “Air” absorbs light, which in turn causes the light intensity to become weaker and weaker with increasing distance. The combination of these two important factors makes fireflies visible only from a certain distance.

The flashing light produced by fireflies can be formulated to correspond to an “objective function” that is to be optimized by optimization algorithms; this allows researchers to formulate and implement new optimization algorithms.

The firefly optimization algorithm proposed by Xin-She Yang (Yang, 2008) in 2008. The firefly optimization algorithm is one of the most reliable optimization techniques implemented on various problem domains and achieves good accuracy (Adaniya et al., 2015; Adaniya et al., 2012; Ahmed & Maheswari, 2017; Mahdi & Hassan, 2018; Tuba et al., 2018). Although it has some inefficiency in terms of parameter dependence and computing complexity (Yu, 2020), it has been used in various applications because of its reliability and accuracy, such as power economic dispatch problems and spectrum access (Kolias et al., 2015; Lakshmana Rao et al., 2021; Liaquat et al., 2020; Shandilya et al., 2023).

To implement the Firefly Optimization Algorithm, the characteristic features related to the behavior of fireflies and the flashing light pattern produced by them will be formulated. To simplify the formulation of the firefly algorithm, the following rules are used:

1. All fireflies are “unisex”. In other words, fireflies, regardless of their gender, will be attracted to other fireflies in the problem space.

2. In the Firefly Algorithm (FA), the “Attractiveness” of a firefly will be proportional to its “Brightness”. In other words, for every two flashing fireflies, the one with less light will be attracted to the one with more light. Therefore, the attractiveness of a firefly will be proportional to its brightness. When the distance between two worms increases, their Attractiveness and Brightness decrease. In other words, when the distance between two fireflies increases, not only does their attractiveness to each other decrease, but their (visible) brightness (to each other) also decreases. If a particular firefly is brighter than the others, it will move randomly in the environment (it will not be attracted to any of the others).

3. The brightness of a firefly is affected by or determined by the characteristic features of the objective function. In "Maximization" problems, the brightness can be specified in proportion to the value of the "Fitness Function". It is worth noting that it is possible to define the brightness of fireflies in a similar way to the fitness function in "Genetic Algorithms".

3. Methodology / Proposed Method

3.1. Description of the Proposed Method or Algorithm

Since the task scheduling problem in cloud environments is a dynamic and constantly evolving problem, providing a suitable task scheduling algorithm with reasonable speed is a major challenge. Since the Min-Min algorithm performs well in static scenarios; in this paper, we use the optimized Min-Min algorithm (Muradi et al., 2022) to generate the initial population for the Firefly optimization algorithm with the aim of minimizing the total completion time as a critical performance criterion and also ensuring scalability and compatibility in dynamic cloud systems. The end result is a robust hybrid task scheduling algorithm that outperforms the techniques used alone in various distributed scenarios.

3.2. Technical Details (e.g., Algorithm, Model, or Architecture)

Suppose that n jobs are considered for processing in m virtual machine in the cloud at the beginning of the algorithm and tij (i=1,…, n, j=1,…,m) is the time to complete each job i on the j machine. In the proposed hybrid Min-Min-FOA algorithm, an initial population is generated using the Min-Min method based on the completion of each job on the entire machine, which accelerates the convergence process by providing a starting point with superior quality. Then, the firefly optimization algorithm uses the aforementioned initial population to find the global optimal solution with the feature of using the fewest machines to perform the most work and in the shortest possible time. This feature also ensures a reduction in energy consumption. The solution provided is in the form of an array that specifies the order in which the machines are placed in that order to start the work. The length of this array is proportional to the number of machines in use. During the optimization process, the main step, i.e., using the firefly optimization algorithm, is repeated over and over again.

The pseudocode below outlines the Min-Min-FOA algorithm, integrating the described phases:

Table 1

Pseudo-code for Min-Min-FOA algorithm, combining Min-Min strategy and FOA algorithm.

```
Define Objective function I=f(x)(I is light intensity)
For all jobs
        For all virtual machined
       Calculate completion time
Choose the job with the least time, remove the desired job and VM from the list.
Go to 1 until all jobs mapped to Vms.
Choose sorted jobs as initial population of fireflies
Define absorption coefficient γ
While (t<MaxGeneration):
                For i=1 to n (fireflies)
                    For j=1 to i:
                        If Ii>Ij:
                          Vary attractiveness with distance r via exp(-γr)
                          Move firefly I towards j
                          Evaluate new solutions and update light intensity
                      Rank fireflies and find the current best
```

4. Experiments / Results and Discussion

4.1. *Parameter Settings and Sensitivity Analysis*

This method leads to further energy savings and associated costs by further reducing the task completion time. Faster task execution minimizes the system execution time and therefore improves performance. In the FOA implementation, we initialized a population of 50 fireflies, set MaxGen to 1000 iterations, and defined the attractiveness parameter (β) as 0.5 with a light absorption coefficient (γ) of 0.01. To ensure exploration, a Gaussian distribution was used for the random parameter (ε). Convergence was controlled by tracking the minimum value of the objective function and achieving stability after 800 iterations. The initial population was selected using the EHD strategy. The simulation parameters were selected empirically after sensitivity analysis, ensuring robust evaluation of the proposed method.

To ensure stability, we tuned the FOA parameters based on initial experiments with the Cooja simulator. The attractiveness parameter (β=0.5), which balances exploration and exploitation, and the light absorption coefficient (γ=0.01), which ensures gradual convergence under hyperdynamic conditions, were experimentally tested by performing sensitivity analysis by varying β (0.1 to 1.0) and γ (0.001 to 0.1) in 10 runs with 50 fireflies and 10 virtual machines. The results showed that β=0.5 and γ=0.01 minimize the energy consumption (33.28%) and latency (18% reduction) with convergence in 1000 iterations, while higher β values (e.g., 1.0) increase the computational overhead by 18% and lower γ (e.g., 0.001) delay convergence. These findings confirm the selected parameters as optimal parameters.

4.2. *Description of experiments or simulations*

To evaluate the performance of the proposed algorithm and given our intention to compare the proposed method with the results of the methods Min-Min and Min-PSO-GA, we performed simulations using a set of 512 tasks mapped to 16 virtual machines (VMs) in a simulated cloud environment, to have the same conditions and a more realistic comparison. The tasks were represented as fireflies, where each entry represents the virtual machine assigned to that task. The simulations were performed in the Contiki Cooja simulator environment, which was traditionally designed for IoT networks but we had made some changes to simulate cloud activities, and the reason for choosing Cooja is its flexibility in modeling distributed systems such as cloud scheduling. Our emphasis on scalability and flexibility in dynamic cloud environments is consistent with this approach, which, although unusual, enables rapid prototyping of scheduling algorithms across different resource configurations. Task execution times and virtual machine capacities were artificially generated to reflect twelve distinct scenarios categorized into compute-intensive (c), input/output-intensive (i), and mixed (p) workloads, each with task-virtual machine heterogeneity levels of high-high (hihi), high-low (hilo), low-high (lohi), and low-low (lolo), as shown in Table 2. To ensure realism, we adopted a uniform distribution for task execution times, ranging from 10 to 1000 time units for computationally intensive tasks (simulating CPU-intensive tasks), 1000 to 10000 units for I/O-intensive tasks (reflecting data transfer latencies), and 50 to 5000 units for mixed workloads. The capacities of the virtual machines were similarly varied: high-capacity virtual machines (1000 units/s) for "high" scenarios and low-capacity virtual machines (100 units/s) for "low" scenarios, calibrated based on benchmarks from previous cloud studies (Paulraj et al., 2023; Pradhan et al., 2025). This distribution mimics realistic workload patterns, such as bursty I/O demands or computationally intensive scientific applications, and ensures testing scenarios for the stability of the proposed algorithm under diverse cloud conditions.

5. Results

A summary of the findings from the experiment can be seen in Table 2, and the examples can be seen in Figures 1 through 3. Table 2 also contains the outcomes of the experiment. Table 2 displays, for each of the twelve instances that are shown, the total amount of time that is necessary to complete GA-PSO-Min, Min-Min and proposed algorithm. Furthermore, this reveals that proposed algorithm is superior than Min-Min and GA-PSO-Min on a constant basis. In the case of the compute-intensive high-high scenario (c_hihi), for example, Min-Min-FOA was able to achieve a completion time of 223.47 units, which is a drop of about 1.83% in comparison to the completion time of GA-PSO-Min, which was 227.65 units. In the I/O-intensive low-low scenario (i_lolo), Min-Min-FOA made a considerable improvement to the completion time by decreasing it from 29,064,893.5 units to 28491511.7 units. This represents an improvement of about 1.98%. The figures 1 through 5 provide a graphical representation of a comparison of the completion times across all of the instances. It is evident that GA-PSO-Min consistently plots below Min-Min, which exemplifies the performance advantage that it has. Table 2 graphically presented in Figure 1.

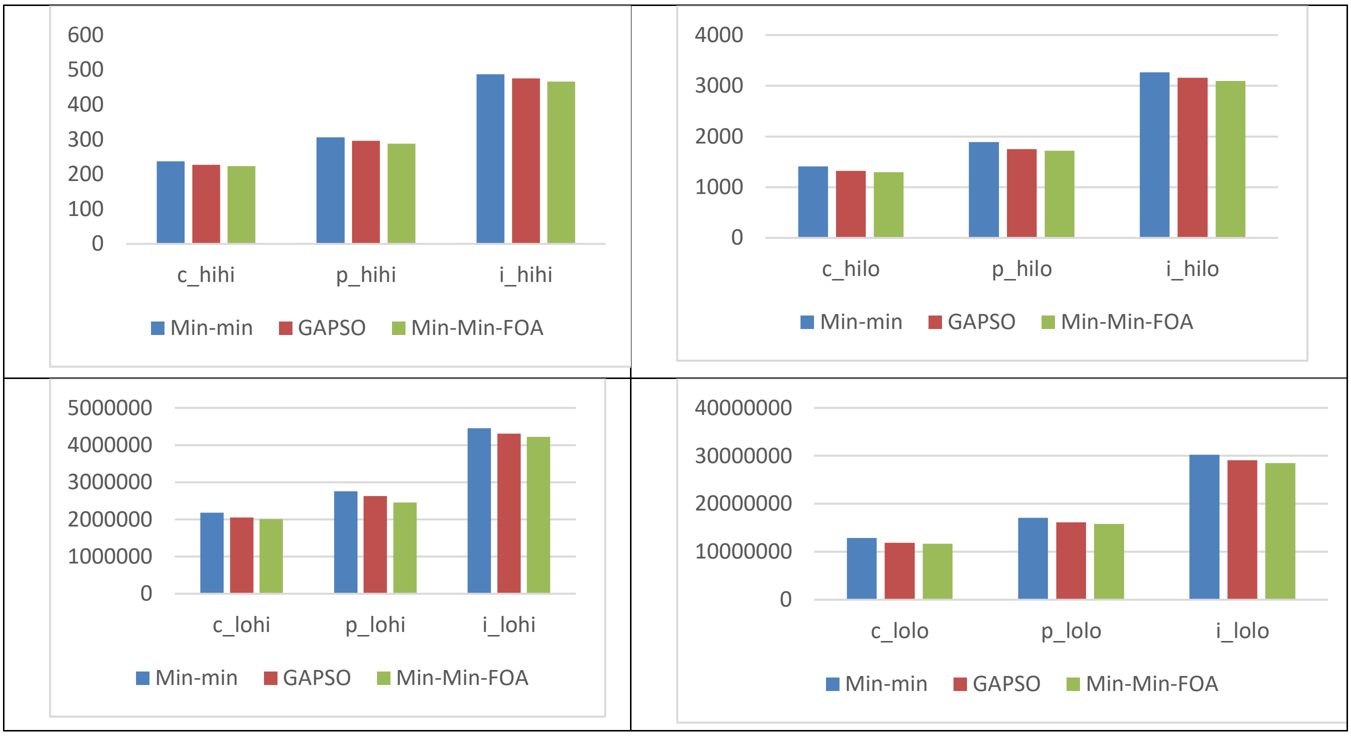
Table 2

All states and all distributed environments

Instance	Min-min	GAPSO	Min-Min-FOA
c_hihi	237.75	227.65	223.47
c_hilo	1411	1319.55	1294.16
c_lohi	2182578.65	2052542.3	2011489.46
c_lolo	12854160.55	11874263.4	11646775.14
i_hihi	487.3	476	466.53
i_hilo	3265.4	3157.05	3092.8
i_lohi	4456769.5	4310440.35	4224218.55
i_lolo	30258306.9	29064893.5	28491511.7
p_hihi	306	296.35	288.43
p_hilo	1890.1	1754	1718.8
p_lohi	2759795.95	2631302.05	2456676
p_lolo	17113178.8	16173294.95	15848821.2

Figure 1

Performance comparison between GA-PSO-Min and the Min-Min heuristic in scenarios with both heterogeneous tasks and virtual machines (VMs)



6. Discussion and Conclusion

The experimental results of this study underscore the efficacy of the Min-Min-FOA algorithm in addressing the complexities of job scheduling within dynamic cloud environments. Across twelve diverse scenarios, Min-Min-FOA consistently outperformed the Min-PSO-GA, reducing total completion time by 1.5–3%. This improvement stems from the hybrid architecture, which capitalizes on Min-Min's efficient initial solution, FOA's robust global exploration. These gains highlight the algorithm's adaptability to varying task-VM interactions. This adaptability positions Min-Min-FOA as a promising solution for modern cloud systems, where resource states and user demands fluctuate unpredictably.

Authors' Contributions

Authors contributed equally to this article.

Declaration

In order to correct and improve the academic writing of our paper, we have used the language model ChatGPT.

Transparency Statement

Data are available for research purposes upon reasonable request to the corresponding author.

Acknowledgments

We would like to express our gratitude to all individuals helped us to do the project.

Declaration of Interest

The authors report no conflict of interest.

Funding

According to the authors, this article has no financial support.

Ethics Considerations

Not applicable.

References

Abedinzadeh, M. H., & Akyol, E. (2023). A multidimensional opinion evolution model with confirmation bias. 59th Annual Allerton Conference on Communication, Control, and Computing (Allerton),

Adaniya, M. H., Carvalho, L. F., Zarpelão, B. B., Sampaio, L. D., Abrão, T., Jeszensky, P. J. E., & Proença, M. L., Jr. (2015). *Firefly Algorithm in Telecommunications*. Bio-Inspired Computation in Telecommunications. https://doi.org/10.1016/B978-0-12-801538-4.00003-3

Adaniya, M. H., Lima, M. F., Rodrigues, J. J., Abrão, T., & Proença, M. L. (2012). Anomaly detection using dsns and firefly harmonic clustering algorithm. Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada.

Ahmed, A. A., & Maheswari, D. (2017). Churn prediction on huge telecom data using hybrid firefly based classification. *Egypt. Inform. J.*, *18*, 215-220. https://doi.org/10.1016/j.eij.2017.02.002

Alazzam, H., Alhenawi, E., & Rizik, A. (2019). A hybrid job scheduling algorithm based on Tabu and Harmony search algorithms. *The Journal of Supercomputing*, *75*, 7994-8011. https://doi.org/10.1007/s11227-019-02936-0

Boroumand, A., Hosseini Shirvani, M., & Motameni, H. (2025). A heuristic task scheduling algorithm in cloud computing environment: an overall cost minimization approach. *Cluster Computing*, *28*(2), 137. https://doi.org/10.1007/s10586-024-04843-3

Chen, M., Xu, J., Zhang, W., & Li, Z. (2025). A new customer-oriented multi-task scheduling model for cloud manufacturing considering available periods of services using an improved hyper-heuristic algorithm. *Expert Systems with Applications*, *269*, 126419. https://doi.org/10.1016/j.eswa.2025.126419

Devaraj, A. F. S., Elhoseny, M., Dhanasekaran, S., Lydia, E. L., & Shankar, K. (2020). Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments. *Journal of Parallel and Distributed Computing*, *142*, 36-45. https://doi.org/10.1016/j.jpdc.2020.03.022

Ghobaei-Arani, M., Jabbehdari, S., & Mohammad Ali, P. (2018). An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach. *Future Generation Computer Systems*, *78*(1), 191-210. https://doi.org/10.1016/j.future.2017.02.022

Khaledian, N., Razzaghzadeh, S., Haghbayan, Z., & Völp, M. (2025). Hybrid Markov chain-based dynamic scheduling to improve load balancing performance in fog-cloud environment. *Sustainable Computing: Informatics and Systems*, *45*, 101077. https://doi.org/10.1016/j.suscom.2024.101077

Khezri, E., Yahya, R. O., Hassanzadeh, H., Mohaidat, M., Ahmadi, S., & Trik, M. (2024). DLJSF: data-locality aware job scheduling IoT tasks in fog-cloud computing environments. *Results in Engineering*, *21*, 101780. https://doi.org/10.1016/j.rineng.2024.101780

Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2015). Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset. *IEEE Commun. Surv. Tutor.*, *18*, 184-208. https://doi.org/10.1109/COMST.2015.2402161

Kushwaha, S., & Singh, R. S. (2025). Deadline and budget-constrained archimedes optimization algorithm for workflow scheduling in cloud. *Cluster Computing*, *28*(2), 117. https://doi.org/10.1007/s10586-024-04702-1

Lakshmana Rao, K., Sireesha, R., & Shanti, C. (2021). On the convergence and optimality of the firefly algorithm for opportunistic spectrum access. *Int. J. Adv. Intell. Paradig.*, *18*, 119. https://doi.org/10.1504/IJAIP.2021.112900

Liaquat, S., Saleem, O., & Azeem, K. (2020). Comparison of Firefly and Hybrid Firefly-APSO Algorithm for Power Economic Dispatch Problem. Proceedings of the IEEE 2020 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP), Bandung, Indonesia.

Liu, Z., Zhang, J., Li, Y., Bai, L., & Ji, Y. (2018). Joint jobs scheduling and lightpath provisioning in fog computing micro datacenter networks. *J. Opt. Commun. Netw.*, *10*(7), B152-B163. https://doi.org/10.1364/JOCN.10.00B152

Long, G., Wang, S., & Lv, C. (2025). QoS-aware resource management in cloud computing based on fuzzy meta-heuristic method. *Cluster Computing*, *28*(4), 1-35. https://doi.org/10.1007/s10586-024-05021-1

Mahdi, M. S., & Hassan, N. F. (2018). Design of keystream Generator utilizing Firefly Algorithm. *J. Al-Qadisiyah Comput. Sci. Math.*, *10*, 91. https://jqcsm.qu.edu.iq/index.php/journalcm/article/view/441

Mokhtari, V., Mikaeilvand, N., Mirzaei, A., Nouri-moghaddam, B., & Gudakahriz, S. J. (2025). GA-PSO-MIN: A HYBRID HEURISTIC ALGORITHM FOR MULTI-OBJECTIVE JOB SCHEDULING IN CLOUD COMPUTING. *Archives for Technical Sciences*, *2*(33), 22-46. https://doi.org/10.70102/afts.2025.1833.022

Murad, S. A., Azmi, Z. R. M., Muzahid, A. J. M., Bhuiyan, M. K. B., Saib, M., Rahimi, N., & Bairagi, A. K. (2024). SG-PBFS: Shortest gap-priority based fair scheduling technique for job scheduling in cloud environment. *Future Generation Computer Systems*, *150*, 232-242. https://doi.org/10.1016/j.future.2023.09.005

Muradi, S. S., Badeel, R., Abdulkarim Alsandi, N. S., Alshaaya, R. F., Ahmed, R. A., Muhammed, A., & Derahman, M. (2022). Optimized MIN-MIN Task Scheduling Algorithm For Scientific Workflows In a Cloud Environment. *Journal of Theoretical and Applied Information Technology*, *100*(2). https://www.researchgate.net/profile/Sallar-Murad/publication/358461191_OPTIMIZED_MIN-MIN_TASK_SCHEDULING_ALGORITHM_FOR_SCIENTIFIC_WORKFLOWS_IN_A_CLOUD_ENVIRONMENT/links/620385d4c2d279745e763540/OPTIMIZED-MIN-MIN-TASK-SCHEDULING-ALGORITHM-FOR-SCIENTIFIC-WORKFLOWS-IN-A-CLOUD-ENVIRONMENT.pdf

Pan, J. S., Yu, N., Chu, S. C., Zhang, A. N., Yan, B., & Watada, J. (2025). Innovative Approaches to Task Scheduling in Cloud Computing Environments Using an Advanced Willow Catkin Optimization Algorithm. *Computers, Materials and Continua*, *82*(2), 2495-2520. https://doi.org/10.32604/cmc.2024.058450

Paulraj, D., Sethukarasi, T., Neelakandan, S., Prakash, M., & Baburaj, E. (2023). An efficient hybrid job scheduling optimization (EHJSO) approach to enhance resource search using Cuckoo and Grey Wolf Job Optimization for cloud environment. *PLoS One*, *18*(3). https://doi.org/10.1371/journal.pone.0282600

Pradhan, A., Das, A., & Bisoy, S. K. (2025). Modified parallel PSO algorithm in cloud computing for performance improvement. *Cluster Computing*, *28*(2), 131. https://doi.org/10.1007/s10586-024-04722-x

Reddy, G. N., & Kumar, S. P. (2017). Multi objective task scheduling algorithm for cloud computing using whale optimization technique. Proc. Int. Conf. Next Gener. Comput. Technol.,

Shandilya, S. K., Choi, B. J., Kumar, A., & Upadhyay, S. (2023). Modified Firefly Optimization Algorithm-Based IDS for

Nature-Inspired Cybersecurity. *Processes*, *11*, 715. https://doi.org/10.3390/pr11030715

Sobhanayak, S., Kumar, T. A., & Sahoo, B. (2018). Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm. *Future Computing and Informatics Journal*, *3*(2), 210-230. https://doi.org/10.1016/j.fcij.2018.03.004

Tuba, E., Tuba, M., & Beko, M. (2018). Two stage wireless sensor node localization using firefly algorithm. Smart Trends in Systems, Security and Sustainability, Singapore.

Yang, X. S. (2008). *Nature-Inspired Metaheuristic Algorithms* (Vol. 12). Luniver Press. https://books.google.com/books?hl=fa&lr=&id=iVB_ETlh4ogC&oi=fnd&pg=PR5&dq=Nature-Inspired+Metaheuristic+Algorithms&ots=DyfBqfCJrf&sig=CtlwBSGqpLY9SO8TbNGfoL_q1oc

Yu, G. (2020). A modified firefly algorithm based on neighborhood search. *Concurr. Comput. Pract. Exp.*, *33*, e6066. https://doi.org/10.1002/cpe.6066

Zade, B. M. H., Mansouri, N., & Javidi, M. M. (2025). An improved beluga whale optimization using ring topology for solving multi-objective task scheduling in cloud. *Computers & Industrial Engineering*, *200*, 110836. https://doi.org/10.1016/j.cie.2024.110836

Zhang, H., Zou, Q., Ju, Y., Song, C., & Chen, D. (2022). Distance-based support vector machine to predict DNA N6-methyladenine modification. *Curr. Bioinf.*, *17*(5), 473-482. https://doi.org/10.2174/1574893617666220404145517

Zhao, Y., Liang, H., Zong, G., & Wang, H. (2023). Event-based distributed finite-horizon $ H_\infty $ consensus control for constrained nonlinear multiagent systems. *IEEE Syst. J.*, *32*(12), 82-98. https://ieeexplore.ieee.org/abstract/document/10284586/