Engineering Universe for Scientific Research and Management
ISSN (Online): 2319-3069
Vol. XVI Issue VIII
August 2024

Multi-Objective Task Scheduling for Performance and Security in Mobile Cloud Computing

Pooja Kumari Singh[1], Dr. Leena Shrivastava[2]
Research Scholar, SDU School of IT and Computer Science, Sona Devi University, Ghatsila, East Singhbhum, Jharkhand[1]
Professor, SDU School of IT and Computer Science, Sona Devi University, Ghatsila, East Singhbhum, Jharkhand2

Abstract
This paper introduces an advanced version of the Imperialist Competitive Algorithm (ICA) tailored for task scheduling in cloud computing environments. Addressing the ICA's common issue of rapid convergence to local optima, the proposed method integrates a uniform mixing process, similar to that used in Genetic Algorithms (GAs), into the absorption policy. This enhancement improves exploration capabilities and reduces the risk of local optima entrapment. The method employs a uniform recombination technique to generate new colony positions and ensure effective mutations. Simulation results, conducted using MATLAB, demonstrate that the proposed algorithm significantly outperforms the Genetic Algorithm (GA) in terms of task completion time and resource productivity. The inclusion of bandwidth considerations in the cost function further enhances scheduling efficiency. Compared to other scheduling methods, including GAs, Particle Swarm Optimization, and traditional ICA, the enhanced ICA approach offers superior performance, optimizing both task management and resource efficiency in cloud environments. This advancement highlights the effectiveness of the proposed method in addressing the complex challenges of performance and security in multi-objective cloud computing scenarios.

Keywords: ***Multi-Objective Task Scheduling, Imperialist Competitive Algorithm (ICA), Cloud Computing, Genetic Algorithms (GAs), Resource Efficiency.***

1. Introduction

Task scheduling in cloud computing environments presents a complex challenge due to the need to efficiently allocate resources while managing performance and security concerns. The Imperialist Competitive Algorithm (ICA) has emerged as a promising approach for this task, offering a meta-heuristic solution inspired by socio-political competition. However, a significant limitation of the ICA is its tendency to quickly converge to local optima, which can hinder its effectiveness in finding globally optimal solutions. To address this issue, this paper introduces an enhanced version of the ICA specifically designed for task scheduling in cloud environments. The proposed method incorporates a uniform mixing process similar to that used in Genetic Algorithms (GAs) into the absorption policy of the ICA. This integration aims to improve the algorithm's exploration capabilities and reduce the likelihood of getting trapped in local optima. By employing a uniform recombination technique, the method generates new colony positions and ensures more effective mutations. Simulation results, conducted using MATLAB, reveal that the advanced ICA outperforms the Genetic Algorithm (GA) in terms of task completion time and resource productivity. Additionally, the proposed cost function, which includes bandwidth considerations, significantly enhances scheduling efficiency. When compared to other scheduling methods, such as GAs, Particle Swarm Optimization, and traditional ICA, the enhanced ICA demonstrates superior performance, optimizing both task management and resource efficiency in cloud computing environments. This paper highlights the effectiveness of the proposed approach in addressing the multi-objective challenges of performance and security in mobile cloud computing.

2. Literature Review

Multi-objective task scheduling in mobile cloud computing has gained significant attention due to its critical role in balancing performance and security. As mobile devices increasingly rely on cloud resources, the challenge is to effectively schedule tasks while optimizing conflicting objectives such as response time, throughput, and data protection. Existing literature highlights various approaches to improve performance through algorithms like genetic algorithms and particle swarm optimization, yet many overlook the integration of robust security measures. Recent studies emphasize the need for frameworks that can simultaneously address both performance metrics and security concerns, reflecting the complex demands of modern mobile cloud environments.

This review explores the advancements and identifies gaps in achieving optimal multi-objective task scheduling.

Table 1 Related work

Author	Work Done	Findings
Masadeh et al. (2021)	Task scheduling on cloud computing using sea lion optimization algorithm.	Presented a sea lion optimization algorithm for task scheduling, improving efficiency in cloud computing.
Iranmanesh & Naji (2021)	Developed DCHG-TS, a hybrid genetic algorithm for deadline-constrained and cost-effective scientific workflow scheduling in cloud computing.	The DCHG-TS algorithm was effective in managing deadlines and costs for scientific workflows in cloud environments.
Najafizadeh et al. (2020)	Surveyed task scheduling in fog computing.	Provided a comprehensive overview of task scheduling strategies in fog computing, highlighting various approaches and challenges.
Hosseinioun et al. (2020)	Proposed an energy-aware task scheduling approach using a hybrid meta-heuristic algorithm in fog computing.	The proposed method achieved better energy efficiency for task scheduling in fog computing environments.
Elashri & Azim (2020)	Focused on energy-efficient offloading of real-time tasks using cloud computing.	Achieved significant improvements in energy efficiency for real-time task offloading in cloud settings.
Ghobaei-Arani et al. (2020)	Introduced a task scheduling approach using the moth-flame optimization algorithm for cyber-physical systems in fog computing.	The moth-flame optimization algorithm enhanced task scheduling efficiency for cyber-physical systems in fog computing.
Huang et al. (2020)	Implemented task scheduling in cloud computing using particle swarm optimization with time-varying inertia weight strategies.	The particle swarm optimization approach with time-varying inertia weights improved scheduling performance in cloud computing.
Mohammad Taisir Masadeh et al. (2019)	Applied the humpback whale optimization algorithm based on vocal behavior for task scheduling in cloud computing.	The humpback whale optimization algorithm showed improved task scheduling efficiency by incorporating vocal behavior patterns.
Xu et al. (2019)	Improved particle swarm optimization for workflow scheduling in cloud-fog environments.	Enhanced particle swarm optimization methods were effective in managing workflows across cloud and fog environments.
Wang & Li (2019)	Developed a hybrid heuristic algorithm for task scheduling in smart production lines with fog computing.	The hybrid heuristic algorithm optimized task scheduling for smart production lines, improving overall system efficiency.
Li et al. (2019)	Jointly optimized data placement and scheduling to enhance user experience in edge computing.	The joint optimization approach improved user experience by effectively managing data placement and task scheduling in edge computing environments.

3. Research gap

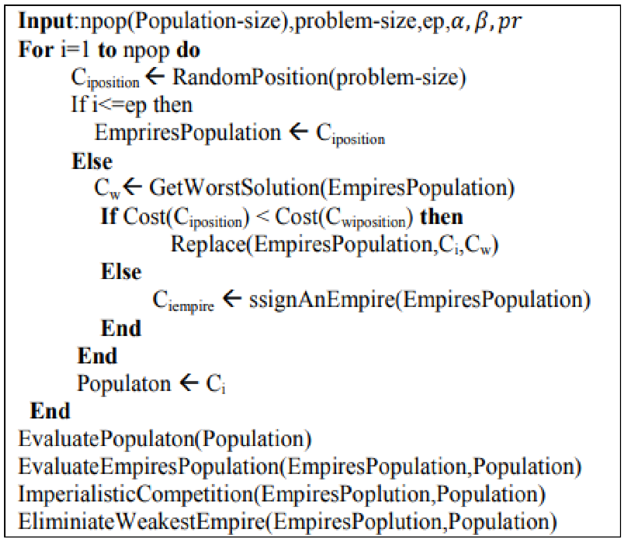
While multi-objective task scheduling in mobile cloud computing addresses performance and security, gaps remain in balancing conflicting objectives like response time and data protection. Existing approaches often prioritize either performance or security, neglecting the optimal trade-off between them. Further research is needed to develop robust models that simultaneously optimize both performance and security across diverse mobile cloud environments.

4. Methodology

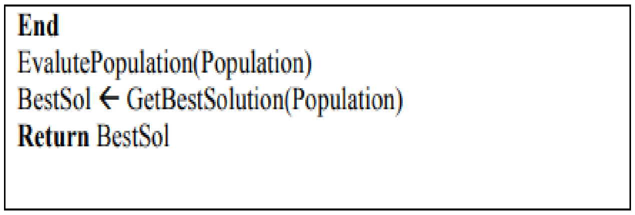
A common drawback of the Imperialist Competitive Algorithm (ICA) is its tendency to converge quickly to local optima. To mitigate this issue and enhance the algorithm's exploration capabilities, a process akin to the fixed mixing used in Genetic Algorithms (GAs) is incorporated into the absorption policy. Specifically, after applying the absorption policies within the empire, a uniform recombination technique is used to generate new positions for colonies. This involves combining the positions of both the colonizer and the colonized. To prevent random exploration and ineffective mutations, a condition is introduced in the algorithm: if the new position is an improvement over the previous position of a colony, the current position is updated accordingly. The recombination process is defined by the formula:

$$\text{Colony}y_k = \text{a}(\text{Colony}y_{ik}) + (1\text{-a})^* (\text{imperialist}t_k) \quad (1)$$

Figure 1 illustrates the overall stages of the proposed algorithm, presented in pseudo-code, for multi-objective task scheduling aimed at enhancing performance and security in mobile cloud computing.



```
Input:npop(Population-size),problem-size,ep,$\alpha, \beta, pr$
For i=1 to npop do
    $C_{iposition}$ ← RandomPosition(problem-size)
    If i<=ep then
        EmpriresPopulation ← $C_{iposition}$
    Else
        $C_w$ ← GetWorstSolution(EmpiresPopulation)
        If Cost($C_{iposition}$) < Cost($C_{wiposition}$) then
            Replace(EmpiresPopulation,$C_i$,$C_w$)
        Else
            $C_{iempire}$ ← ssignAnEmpire(EmpiresPopulation)
        End
    End
    Populaton ← $C_i$
End
EvaluatePopulaton(Population)
EvaluateEmpiresPopulation(EmpiresPopulation,Population)
ImperialisticCompetition(EmpiresPoplution,Population)
EliminiateWeakestEmpire(EmpiresPoplution,Population)
```



```
End
EvalutePopulation(Population)
BestSol ← GetBestSolution(Population)
Return BestSol
```

Fig. 1 The proposed algorithm for pseudo-code

Simulation Results: To evaluate the efficiency of the proposed method, it is compared with the Genetic Algorithm (GA) in terms of task completion time and resource productivity, using MATLAB for simulations. The data used for this assessment is presented in Table 1, while Table 2 outlines the parameters utilized for the Imperialist Competitive Algorithm (ICA).

Table 1 Simulation parameters

Parameter	**Values**
Number of work	60~30
Virtual machine	12~6
The time of tasks	6000~1000 (MI)
Tasks size	200~50 (MB)
Bandwidth	500~100 (Mbps)
Processor speed	500~100 (MIPS)

Table 1 outlines the simulation parameters used in the study. The number of work units ranges from 60 to 30, indicating a variable workload size. Virtual machines are set between 12 and 6, suggesting different levels of computational resources. Task durations vary from 6000 to 1000 milliseconds (MI), reflecting a range in processing time. Task sizes span from 200 to 50 megabytes (MB), which impacts the data handling requirements. Bandwidth ranges from 500 to 100 megabits per second (Mbps), affecting data transfer rates. Finally, processor speeds vary from 500 to 100 million instructions per second (MIPS), influencing computational performance. These parameters provide a comprehensive view of the simulation's operational scope, highlighting the effects of resource variability on task scheduling.

Table 2 Imperialist competitive parameters

Parameter	**Values**
Maximum frequency	100
Population size	50
Number of empires	10
(The absorption coefficient) β	2
The possibility of revolution	0.1
Rate of revolution	0.05
(Colonies the cost factor) δ	0.1
(Selection pressure) η	10

Table 2 presents the parameters for the Imperialist Competitive Algorithm used in the study. The maximum frequency is set to 100, indicating the upper limit of iterations or evaluations during the optimization process. The population size is fixed at 50, determining the number of potential solutions considered. The number of empires is set to 10, influencing the division of solutions into different groups for competition. The absorption coefficient (β) is 2, affecting how solutions are integrated into empires. The probability of revolution is 0.1, with a revolution rate of 0.05, reflecting the likelihood and frequency of introducing new solutions. The cost factor for colonies (δ) is 0.1, and the selection pressure (η) is 10, which together control the emphasis on selecting superior solutions. These parameters collectively guide the

Engineering Universe for Scientific Research and Management
ISSN (Online): 2319-3069 Vol. XVI Issue VIII
August 2024

algorithm’s search and optimization processes, impacting solution diversity and convergence.
Given the stochastic nature of meta-heuristic algorithms and their reliance on random initial positions, each algorithm was executed 10 times, and the average results were used as the final outcome and benchmark for comparison. Figure 2 presents a comparison between the proposed method and the Genetic Algorithm (GA) in terms of standard task completion time, excluding distribution time.

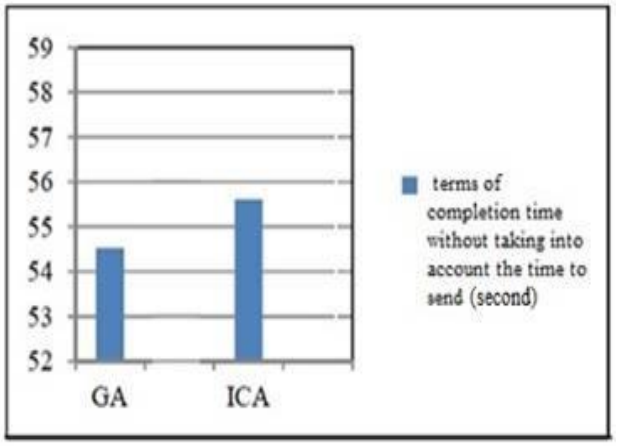


Fig. 2 Chart of methods comparison in terms of completion time withouttaking into account the time to send

Figure 3 shows a comparison chart of the Imperialist Competitive Algorithm (ICA) with the Genetic Algorithm (GA) in a cloud environment, based on the fitness function described in formula (1). The proposed cost function takes into account the time required to send tasks from the scheduler to the sources, which is determined by the bandwidth parameter.

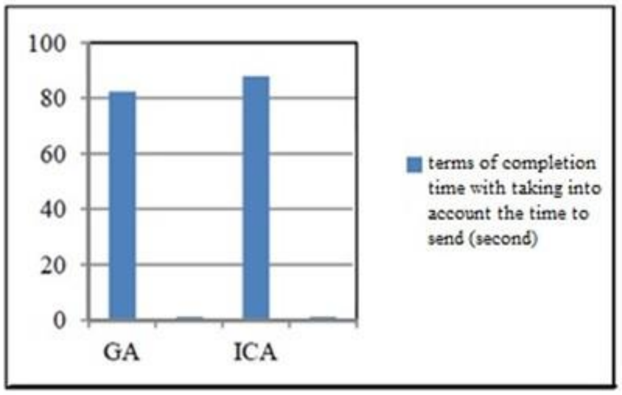


Fig. 3 The chart of comparing algorithms based on the cost function

Figure 4 demonstrates the efficiency of resources when using the Imperialist Competitive Algorithm (ICA) compared to the Genetic Algorithm (GA). The results indicate that the proposed algorithm outperforms other task scheduling methods in cloud environments, including GAs, Particle Swarm Optimization, and the standard ICA, showing a notable improvement in performance.

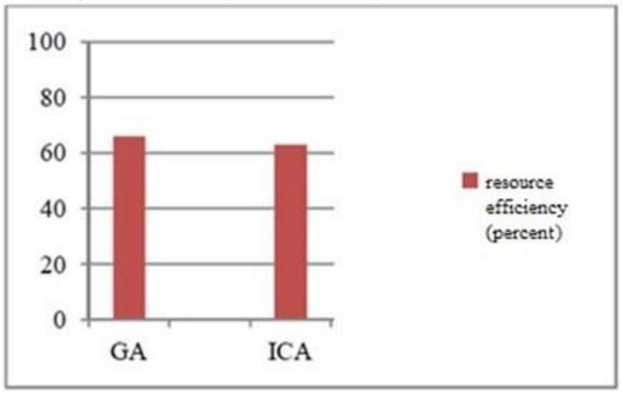


Fig. 4 The chart of comparing algorithms based on resource efficiency

In this section, an enhanced version of the Imperialist Competitive Algorithm (ICA) is proposed for task scheduling in cloud environments. Many existing algorithms for mapping tasks to resources in the cloud focus solely on task execution time and often overlook factors such as resource bandwidth and timing. This paper introduces a cost function that incorporates bandwidth considerations to optimize scheduling and improve task management efficiency in cloud computing environments. Additionally, to prevent the algorithm from getting stuck in local optima and to enhance its exploration capabilities, a process similar to the uniform mixing used in Genetic Algorithms (GAs) is applied to the empires after the absorption phase. The implementation and comparison with other task scheduling methods, including GA, Particle Swarm Optimization, and traditional ICA, demonstrate that the proposed approach achieves superior performance and more effective task scheduling.

5. Conclusion

This paper presents an enhanced version of the Imperialist Competitive Algorithm (ICA) designed for task scheduling in cloud environments, addressing its common drawback of rapid convergence to local optima. By incorporating a process akin to the uniform mixing used in Genetic Algorithms (GAs) into the absorption policy, the proposed method improves exploration capabilities and mitigates the risk of getting stuck in local optima. The uniform recombination technique, which combines the positions of both the colonizer and the colonized, is employed to generate new colony positions and ensure effective mutations. Simulation results, using MATLAB for comparison, indicate that the proposed method outperforms the Genetic Algorithm (GA) in terms of task completion time and resource productivity. The cost function introduced in this study, which includes bandwidth considerations, significantly enhances task scheduling efficiency. The proposed algorithm demonstrates superior performance when compared to other scheduling methods, including GAs, Particle Swarm Optimization, and the standard ICA. Overall, the enhanced ICA approach not only optimizes scheduling by integrating bandwidth factors but also achieves notable improvements in resource efficiency and task management within cloud computing environments. This advancement underscores the effectiveness of the proposed method in addressing the multi-objective challenges of performance and security in mobile cloud computing.

References

[1] Masadeh, R., Alsharman, N., Sharieh, A., Mahafzah, B.A., and Abdulrahman, A. “Task Scheduling on Cloud Computing Based on Sea Lion Optimization Algorithm.” International Journal of Web Information Systems, vol. 17, no. 1, 2021, pp. 99–116.

[2] Iranmanesh, A., and Naji, H.R. “DCHG-TS: A Deadline-Constrained and Cost-Effective Hybrid Genetic Algorithm for Scientific Workflow Scheduling in Cloud Computing.” Cluster Computing, vol. 24, no. 2, 2021, pp. 667–681.

[3] Najafizadeh, A., Salajegheh, A., Rahmani, A.M., and Sahafi, A. “Task Scheduling in Fog Computing: A Survey.” Journal of Advanced Computing Research, vol. 11, no. 1, 2020, pp. 1–10.

[4] Hosseinioun, P., Kheirabadi, M., Kamel Tabbakh, S.R., and Ghaemi, R. “A New Energy-Aware Task Scheduling Approach in Fog Computing Using Hybrid Meta-Heuristic Algorithm.” Journal of Parallel and Distributed Computing, vol. 143, 2020, pp. 88–96.

[5] Elashri, S., and Azim, A. “Energy-Efficient Offloading of Real-Time Tasks Using Cloud Computing.” Cluster Computing, vol. 23, no. 4, 2020, pp. 3273–3288.

[6] Ghobaei-Arani, M., Souri, A., Safara, F., and Norouzi, M. “An Efficient Task Scheduling Approach Using Moth-Flame Optimization Algorithm for Cyber-Physical System Applications in Fog Computing.” Transactions on Emerging Telecommunications Technologies, vol. 31, no. 3, 2020, pp. 1–17.

[7] Huang, X., Li, C., Chen, H., and An, D. “Task Scheduling in Cloud Computing Using Particle Swarm Optimization with Time Varying Inertia Weight Strategies.” Cluster Computing, vol. 23, no. 2, 2020, pp. 1137–1147.

[8] Masadeh, R., Sharieh, A., Mahafzah, B.A., and Sharieh, A. “Humpback Whale Optimization Algorithm Based on Vocal Behavior for Task Scheduling in Cloud Computing.” International Journal of Advanced Science and Technology, vol. 13, 2019.

[9] Xu, R., Wang, Y., Cheng, Y., Zhu, Y., Xie, Y., Sani, A.S., and Yuan, D. “Improved Particle Swarm Optimization Based Workflow Scheduling in Cloud-Fog Environment.” Lecture Notes in Business Information Processing, Springer, Berlin, 2019, pp. 337–347.

[10] Singh, Harsh Pratap, et al. "AVATRY: Virtual Fitting Room Solution." 2024 2nd International Conference on Computer, Communication and Control (IC4). IEEE, 2024.

[11] Singh, Nagendra, et al. "Blockchain Cloud Computing: Comparative study on DDoS, MITM and SQL Injection Attack." 2024 IEEE International Conference on Big Data & Machine Learning (ICBDML). IEEE, 2024.

[12] Singh, Harsh Pratap, et al. "Logistic Regression based Sentiment Analysis System: Rectify." 2024 IEEE International Conference on Big Data & Machine Learning (ICBDML). IEEE, 2024.

[13] Naiyer, Vaseem, Jitendra Sheetlani, and Harsh Pratap Singh. "Software Quality Prediction Using Machine Learning Application." Smart Intelligent Computing and Applications: Proceedings of the Third International Conference on Smart Computing and Informatics, Volume 2. Springer Singapore, 2020.

[14] Pasha, Shaik Imran, and Harsh Pratap Singh. "A Novel Model Proposal Using Association Rule Based Data Mining Techniques for Indian Stock Market Analysis." Annals of the Romanian Society for Cell Biology (2021): 9394-9399.

[15] Md, Abdul Rasool, Harsh Pratap Singh, and K. Nagi Reddy. "Data Mining Approaches to Identify Spontaneous Homeopathic Syndrome Treatment." Annals of the Romanian Society for Cell Biology (2021): 3275-3286.

[16] Vijay Vasanth, A., et al. "Context-aware spectrum sharing and allocation for multiuser-based 5G cellular networks." Wireless Communications and Mobile Computing 2022 (2022).

[17] Wang, J., and Li, D. “Task Scheduling Based on a Hybrid Heuristic Algorithm for Smart Production Line with Fog Computing.” Sensors, vol. 19, no. 4, 2019, p. 1023.

[18] Li, C., Bai, J., and Tang, J. “Joint Optimization of Data Placement and Scheduling for Improving User Experience in Edge Computing.” Journal of Parallel and Distributed Computing, vol. 125, 2019, pp. 93–105.