

QoS based task scheduling algorithm in cloud computing

Arslan Nedhir Malti, ^{1,*}, and Badr Benmammar¹, and Mourad Hakem²

¹ LTT Laboratory of Telecommunication Tlemcen, UABT, Tlemcen, Algeria.

² DISC Laboratory, Femto-ST Institute, UMR CNRS, Université de Franche-Comté, Besançon, France

Abstract. Task scheduling is a critical topic that has a significant impact on the performance of the Cloud computing environment where cloud service providers and users have conflicting objectives and requirements. A good scheduler must provide an acceptable trade-off between these goals. Thus, cloud task scheduling becomes a multi-objective optimization problem. In this paper, our contribution is to address the problem of task scheduling in the Cloud computing environments in order to orchestrate the suitable assignment of the submitted tasks to the set of virtual machines. For this purpose, we have adapted the Flower Pollination Algorithm (FPA) metaheuristic by evaluating its objective function in term of three metrics which are makespan, cost and reliability. The simulation results achieved using CloudSim framework are very satisfactory and clearly show the interest of our FPA-based approach.

1 Introduction

Cloud computing is a modern technology in the computer field. It appeared with the increase in server power and the fall of their operating costs as well as with the appearance of the high-speed Internet. It allows to provide several flexible and scalable services allowing access to the network on demand. Resources are shared and the computing power is configurable according to the users' needs [1]. In the field of Cloud computing, task scheduling is an important issue that considerably influences the performance of this environment [2]. This problem is considered NP-complete requiring the use of heuristics and metaheuristics for its solution [2]. It is characterized by the fact that it includes complex and often conflicting objectives. This problem is classified as a multi-objective optimization problem, whose complexity grows with the increase of the constraints and the considered parameters [1]. Moreover, finding a good compromise between objectives remains a major challenge.

In this context, the objective of scheduling is to distribute tasks across the Cloud service provider's computational resources in order to optimize several quality of service (QoS) metrics. It's worth noting that most of the researches in the literature have often focused on the optimization of a single QoS metric which is the makespan [3].

Many researches have been conducted in the literature to address the task-scheduling problem in cloud computing environments. [4-10]. It should be noted that the Flower Pollination Algorithm (FPA) was previously applied for this problem. For instance, an efficient scheme of pollen representation, as well as a dedicated process for determining the assignment of submitted tasks to virtual machines (VMs) from a given

pollen is proposed in [11], so that only two criteria namely average cloud utilization and time makespan are optimized. However, it differs from our study in the sense that various models as well as a data set were used.

In this paper, we have adapted the FPA metaheuristic for multi-objective scheduling optimization in Cloud computing. Our objective is to find the best assignment of the submitted tasks in a reasonable amount of time according to three QoS metrics which are makespan, cost and reliability.

We chose two approaches to measure the objective function; on the one hand the aggregation approach and on the other hand, the Pareto principle [12] based on the TOPSIS [13] to pick the best Pareto front solution.

In the aggregation approach, we mapped the multi-objective scheduling issue into a single-objective problem by aggregating the different QoS metrics with a priori defined preferences. Whereas, in the Pareto approach, we treated the multi-objective scheduling problem using the Pareto principle to best meet all QoS metrics according to the TOPSIS obtained results.

The rest of this paper is organized as follows. Section 2 describes our approach based on the adaptation of the FPA. Section 3 demonstrates the proposed algorithm's performance based on three metrics which are makespan, cost and reliability. Our conclusions are drawn in the final section 4.

2 Adaptation of the FPA metaheuristic

Xin-She Yang in 2012 [14] proposed a natural bioinspired flower pollination algorithm drawing its metaphor from the pollination process of flowering plants. Pollination can be classified into two types: self-pollination and cross-pollination.

* Corresponding author: arslan.malti@univ-tlemcen.dz

The former, also known as local pollination, occurs when pollen from one flower pollinates the same flower or other flowers of the same plant with the assistance of environmental factors [14]. While cross-pollination, also known as global pollination, occurs over long distances when pollen is transferred from one flower to another flower in a different plant through direct or indirect intervention by pollinators following Levy flight behavior [15].

During the process of optimization, the FPA algorithm's search space is explored through biotic and cross-pollination where the movement of the pollen is represented by the markovian Lévy flight stochastic processes [15]. The latter is a random walk based on a random step from the Lévy distribution, causing a much longer movement from its current position.

The FPA pseudo code that has been adapted for the assignment of tasks to the different virtual machines is outlined in Algorithm I.

We have adapted the FPA metaheuristic to our scheduling problem by considering a set of VMs as the FPA population. The objective is to find the suitable VM for each task based on a multi-objective optimization of the following three parameters:

- **Makespan:** it represents the time difference between the submission of the first task and the receipt of the results (the finish time of the last executed task).
- **Cost:** the total execution cost is the sum of the costs related to the use of virtual machines for all the tasks. The execution cost of the task depends on both the duration of its execution on the machine and the unit price of running it on that machine.
- **Reliability:** it represents the probability that a task will be executed successfully, without any resource failure. Our model for measuring reliability is based on a failure rate which is an intrinsic property of the resource. It is calculated as follows.

$$Reliability = \exp^{-\sum_{i=1}^n TE(T_i) * \lambda_j} \quad (1)$$

Where, $TE(T_i)$ is the execution time of the task T_i and λ_j is the failure rate of the machine running the task.

The above three metrics are classified into two categories: the QoS metrics to be maximized (reliability) and those to be minimized (makespan and cost).

In a multi-objective optimization context, the objective function that results from the aggregation of the three previously criteria is defined as follows:

$$F = w_1 * Makespan + w_2 * cost + w_3 * \left(\frac{1}{Reliability}\right) \quad (2)$$

where $w = \{w_1, w_2, w_3\}$ is the weight vector that reflects the importance or requirement of the user with respect to each criterion.

As the function F is to be minimized and as the third parameter *Reliability* is to be maximized, we will take its

inverse $\frac{1}{Reliability}$ in order to minimize all the objective functions.

Algorithm I

Define a switch probability $p \in [0,1]$

For each task T_i in the list of cloudlets **do**

 Initialize the initial population with random VMs

 Find the best virtual machine V_{best} for task T_i in the initial population

While (stop criterion) **do**

For each virtual machine V_i in the population **do**

If $rand() < p$ **then**

 Draw a step size L that obeys a Lévy distribution

 Do global pollination via $V_i' = V_i + L(V_i - V_{best})$

Else

 Draw ε from a uniform distribution in $[0, 1]$

 Randomly choose two virtual machines V_j and V_k from all solutions.

 Do local pollination via $V_i' = V_i + \varepsilon(V_j - V_k)$

End if

 Evaluate the virtual machine V_i'

 If the new generated machine is better, replace V_i with V_i'

End for

 Update V_{best} (the best machine in the population)

End while

End for

Note that a pre-simulation stage using the appropriate FPA is necessary to find the best assignments of cloudlets to VMs before sending them to the broker.

3 Performance evaluation

For FPA, two parameters have to be defined: the size of the population and the switching probability p which determine the search's diversification and intensification percentage. For all the simulations, we used a population size equal to 20% of the created VMs (this percentage gave us better performances). Concerning the switching probability, we took $p = 0.8$ since this value is suitable for most applications [14]. For the stop criterion, we considered the convergence of the FPA algorithm. The simulations were performed in a heterogeneous environment as listed in Table 1 defining the value associated with each parameter.

We used the CloudSim framework [16] for the various simulations. This tool was chosen for its modeling qualities. It is a generalized and extensible toolkit that supports system modeling and the behavior of Cloud computing components including DataCenters, VMs, cloudlets (tasks) and brokers.

For the assessment of the addressed multi-objective scheduling problem, a comparison was performed using two versions for the evaluation of the solutions provided by FPA, namely the Pareto principle based on the TOPSIS technique and the weighted sum method by aggregating the different QoS metrics according to the previously stated objective function (2).

Table 1. CloudSim simulation parameters.

Data Center	Number of Data Centers	4
Host	Number of Hosts	8
	PES	4
	MIPS	6 000
	RAM	20 GB
	Bandwidth	10 GB
	Storage	1 TB
Virtual Machine	Number of VMs	40 – 60
	MIPS	1 000 to 5 000
	RAM	1 GB to 5 GB
	Bandwidth	100 MB to 500 MB
	Storage	10 GB
Cloudlets	Number of cloudlets	500 – 1 000 – 1 500
	Length	3 000 to 10 000
	Type	Heterogeneous
	Submission time	Poisson distribution of parameter λ

For both versions, we used two vectors with different weights $\{0.5,0.2,0.3\}$ and $\{0.2,0.5,0.3\}$ matching respectively to the criteria $\{Makespan, Cost, Reliability\}$. The first vector $\{0.5,0.2,0.3\}$ favors makespan, whereas in the second vector $\{0.2,0.5,0.3\}$ privileges cost objective.

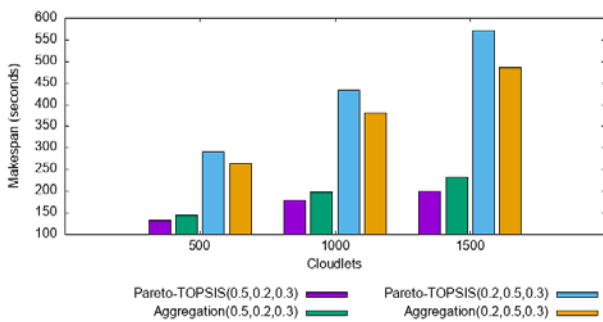


Fig. 1. Comparison in terms of makespan for 40 VMs.

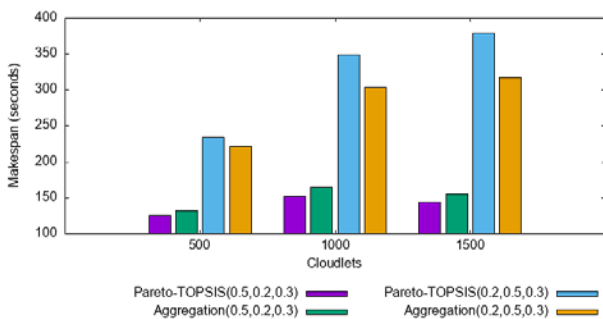


Fig. 2. Comparison in terms of makespan for 60 VMs.

First, we evaluate the time makespan achieved by the compared approaches. From Figs. 1 and 2, we can observe that, no matter the number of tasks and the number of virtual machines, we deal conflicting objectives. Indeed, Pareto-TOPSIS approach minimizes

well the time makespan with the weight vector $\{0.5,0.2,0.3\}$ which favors the time makespan objective. In contrast, the weighted sum method produces better performance for the vector weight $\{0.2,0.5,0.3\}$ which privileges the execution cost objective. This experiment show that Pareto-TOPSIS strategy alleviates this problem by finding better compromise solutions between the considered objectives. This can be explained by the *pareto dominance* concept which does not allow one objective to dominate the other.

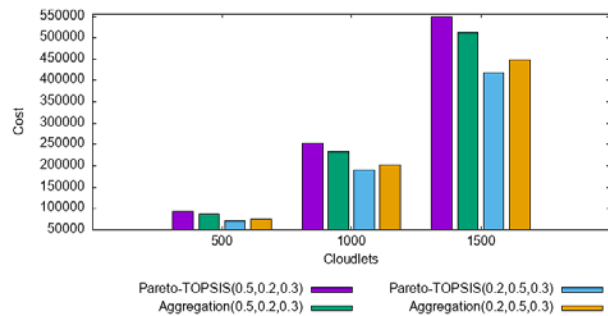


Fig. 3. Comparison in terms of cost for 40 VMs.

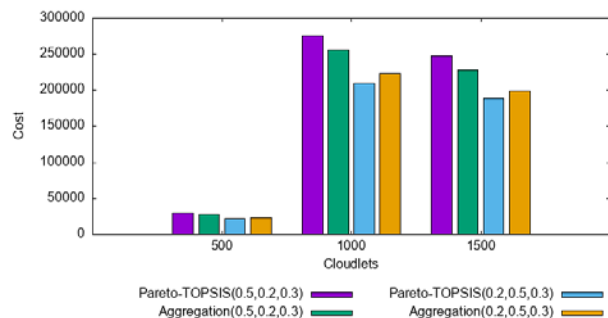


Fig. 4. Comparison in terms of cost for 60 VMs.

Comparing the results of the weighted sum to the results of Pareto-TOPSIS in terms of execution cost, it can be shown in Figs. 3 and 4 that the later gives the best performance with a vector of weights that favors the execution cost. This is because the Pareto-TOPSIS based approach tries to ensure the three considered objectives owing to their importance. This leads to efficient cost minimisation during the scheduling process. However, in terms of reliability optimization, we can see from Figs 5 and 6 that the difference between the achieved performances of the compared methods are almost similar regardless of the number of tasks and the number of virtual machines.

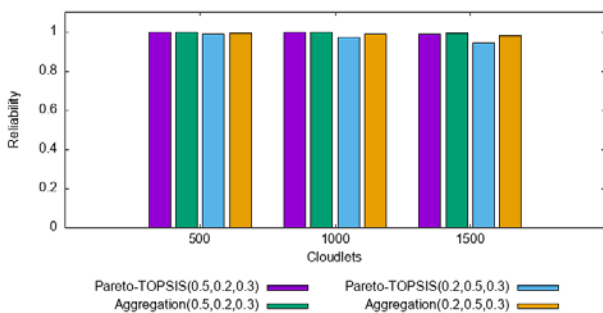


Fig. 5. Comparison in terms of reliability for 40 VMs.

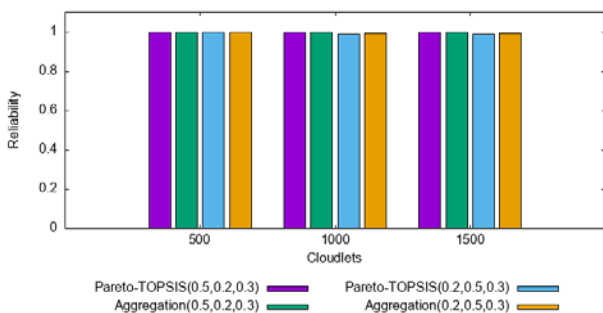


Fig. 6. Comparison in terms of reliability for 60 VMs.

Taken as a whole, concerning the two objectives makespan and cost, the best is to use Pareto-TOPSIS with the adapted weight vector. Indeed, the weighted sum is a simple projection from the multi-objective to the mono-objective. A purely multi-objective approach like Pareto-TOPSIS is more recommended in this context. For reliability, this is a somewhat special metric because in the Cloud, VMs are considered reliable since existing Cloud providers offer services with too small failure rates that can go down to 10^{-7} or 10^{-8} . For this reason, the used method does not have much impact on this objective.

4 Conclusion

In this paper, we have addressed the task scheduling problem in Cloud computing environments. The main objective is to orchestrate the best assignment of the submitted tasks to VMs while satisfying user's requirements. To this end, we proposed a multi-objective

optimization approach that combined flower pollination metaheuristic with TOPSIS-based Pareto principle to select the Pareto front's most compromising solution. Based on CloudSim Framework, the obtained results are more interesting for TOPSIS according to the Pareto principle by comparing them with the aggregation especially for the favored criterion, in order to offer a better QoS to the users according to their needs.

In future work, we intend to extend the studied optimization problem to other QoS metrics (load balancing, energy, fault tolerance ...). It would also be interesting to introduce a new notion of optimality in addition to Pareto and aggregation as well as the application of other metaheuristics such as firefly algorithm [17], bacterial foraging for optimization [18] or dragonfly algorithm [19] in order to compare their results with the FPA's performances.

References

1. J. PRAVEENCHANDAR, A. TAMILARASI, *Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing*. Journal of Ambient Intelligence and Humanized Computing, **12.3**, pp. 4147-4159 (2021)
2. E. H. Houssein, A. G. Gad, Y. M. Wazery, *Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends*, Swarm and Evolutionary Computation, pp. 100841 (2021)
3. A. Semmoud, M. Hakem, B. Benmammar, and J.-C. Charr, *Load balancing in cloud computing environments based on adaptive starvation threshold*, Concurrency and Computation: Practice and Experience, **32.11**, pp. e5652 (2020)
4. N. Miglani, G. Sharma. *Modified particle swarm optimization based upon task categorization in cloud environment*. IJEAT **8.4**, pp. 67-72 (2019)
5. C. Y. Liu, C. M. Zou, P. Wu, *A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing*. In 2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science. IEEE, pp. 68-72 (2014)
6. N. Bacanin, T. Bezdán, E. Tuba, I. Strumberger, M. Tuba, M. Zivkovic, *Task scheduling in cloud computing environment by greywolf optimizer*, in 2019 27th Telecommunications Forum (TELFOR), pp. 1-4 (2019)
7. I. Strumberger, N. Bacanin, M. Tuba, E. Tuba, *Resource scheduling in cloud computing based on a hybridized whale optimization algorithm*. Appl. Sci. **9.22**, pp. 4893 (2019)
8. R. Medara, R. S. Singh. *Energy-aware workflow task scheduling in clouds with virtual machine consolidation using discrete water wave optimization*. Simulation Modelling Practice and Theory, **110**, pp. 102323 (2021)

9. S. Ijaz, E. U. Munir, S. G. Ahmad, M. M. Rafique, O. F. Rana. *Energy-makespan optimization of workflow scheduling in fog-cloud computing*. Computing, pp. 1-27 (2021)
10. N. Rizvi, R. Dharavath, D. R. Edla. *Cost and makespan aware workflow scheduling in IaaS clouds using hybrid spider monkey optimization*. Simulation Modelling Practice and Theory, **110**, pp. 102328 (2021)
11. I. Gupta, A. Kaswan, P. K. Jana, *A flower pollination algorithm based task scheduling in cloud computing*. In : International Conference on Computational Intelligence, Communications, and Business Analytics. Springer, Singapore, pp. 97-107 (2017)
12. B. Benmammam, Y. Benmouna, and F. Krief, *A pareto optimal multi-objective optimisation for parallel dynamic programming algorithm applied in cognitive radio ad hoc networks*, International Journal of Computer Applications in Technology, **59.2**, pp. 152–164, (2019)
13. C.-L. Hwang, K. Yoon, *Methods for multiple attribute decision making*, in Multiple attribute decision making. Springer, pp. 58–191 (1981)
14. X.-S. Yang, *Flower pollination algorithm for global optimization*, in International conference on unconventional computing and natural computation. Springer, pp. 240–249 (2012)
15. I. Pavlyukevich, *L'evy flights, non-local search and simulated annealing*, Journal of Computational Physics, **226.2**, pp. 1830–1844 (2007)
16. Cloudsim: A framework for modeling and simulation of cloud computing infrastructures and services, <http://www.cloudbus.org/cloudsim/>, accessed: 2021-12-28
17. X.S. Yang. *Firefly algorithms for multimodal optimization*. In : International symposium on stochastic algorithms. Springer, pp. 169-178 (2009)
18. K.M. Passino, *Biomimicry of bacterial foraging for distributed optimization and control*. IEEE control systems magazine, **22.3**, pp. 52-67 (2002)
19. S. Mirjalili, *Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems*. Neural Computing and Applications, **27.4**, pp. 1053-1073 (2016)