

An optimization approach for task scheduling in cloud computing environments

Ali Belgacem

*Ecole Militaire Polytechnique
Alger, Algeria
al.BLgacem@gmail.com*

Kadda Beghdad-Bey

*Ecole Militaire Polytechnique
Alger, Algeria
k.beghdadbey@gmail.com*

Sofiane Bouznad

*Ecole Militaire Polytechnique
Alger, Algeria
bouznad.sofiane@gmail.com*

Abstract— The evolution of the tech world has boosted the power of the cloud more than ever due to the changing ways of software use and has made them mere services. Since the services are paid per use, reducing cost is an important issue to focus on. Consequently, task scheduling is considered one of the predominant topics to overcome this issue. In this paper, we suggest a new hybrid heuristic algorithm based on particle swarm optimization and Electromagnetism Meta-heuristic Algorithm (PSO-EMA). PSO-EMA aims to minimize the cost of using virtual machines. The simulation results demonstrate that the proposed algorithm is better than the Max-min, FCFS, PSO, EMA, and DSOS methods in terms of cost.

Keywords—cloud computing, resource allocation, cost, task Scheduling, EMA algorithm.

I. INTRODUCTION

During the last few years, cloud computing has become one of the interesting paradigms in the realm of information technology (IT). It is a key IT megatrend that will take root in the future to replace the existing Internet. Currently, cloud options are growing quickly in private and professionals life, where there is an increase in using applications provided by this technology.

Virtualization technology is the main property of the cloud. It is the process of turning physical IT resources (Servers, Storage, Network, and Power) into virtual resources. Moreover, server consolidation and virtual machine (VM) migration functions, which are provided by virtualization allow the dynamic placement of the resources over physical machines. This way, different VMs can share one physical server. Hence, the management, configuration and administration of physical machines are necessary to exploit the maximum capacity of resources, and enhance hardware utilization, load balancing and optimization of available resources. Knowing that in cloud computing a typical datacenter consists of computing machines connected by high-speed network links; this environment is well suited for the computation of large and diverse groups of tasks [1].

Scheduling is another concept relating to cloud computing that refers resource allocation.

It is the process of assigning a set of tasks to available resources. However, allocation cost is the major challenging problem for scheduling tasks in cloud computing [2]. Cloud providers always seek to improve the services given to their users because the use of resources is payable and the amounts of physical resources are limited. For this reason, providers apply task scheduling algorithms to make the resources unlimited and used with the minimum cost. Every scheduling problem in the cloud has these important characteristics:

Domain description problem: We specify the context of our issues like mobile cloud computing, energy efficiency in data centers [3], load balancing, resource provisioning [4], big data, internet of things (IoT), etc.

Used resources: In cloud computing the resources are assembled to rent as a service, which allows providing many computing services over the Internet (servers, storage, databases, networking, software and more) with an outstanding performance [5], [6].

Machine configuration: It means the arrangement manner and parameters of resources in machine (single or multiple CPUs, memory capacity, etc) [7].

Optimization criterion: It is the metrics that must be improved to achieve the overall goal of objectives functions e.g. reducing makespan and cost, minimizing response time or resource consumption, etc [8], [9].

To allocate the computing power and resource capacity from a cloud provider, there are three different modes of resources allocation:

Best-effort: This is the most used model, resources are provisioned as soon as possible and the requests are placed in a queue.

Deadline Sensitive Leases (DSL): Some algorithms like [10], [11] take into account the deadlines, while others do not respect them (Best-effort leases with deadline). It is a limited time associated with each task that must finish its execution before that date.

Advance Reservation (AR): The resources are reserved in advance. It should be available at a specific time.

To overcome these, our main contribution in this paper is to minimize the resource allocation cost; more precisely, we use the PSO-EMA approach to schedule the tasks in an improved manner to reduce the price of using VMs.

The remainder of this paper is organized as follows: the approaches applied to resolve task scheduling problems in a cloud computing environment are illustrated in section II. Section III presents our proposition by designing the problem formulation. Description of proposed approach is given in section IV. Simulation and discussion results are in section V. Section VI concludes the paper.

II. RELATED WORKS

Recently, several researches have been conducted on task scheduling in the cloud [12].

For example, in [5], authors presented a technique for managing the tasks based on particle swarm optimization (PSO) algorithm. The algorithm takes into account both computation cost and data transmission cost to schedule applications. This implementation was applied to minimize the overall cost of execution. In another study [6] Wu, Zhangjun, et al. suggested a Revised Discrete Particle Swarm Optimization (RDPSO) approach to improve the services given to the cloud user by minimizing both data transmission cost and computation cost. However, the presented method takes a long time in the task execution. Other researches decreased overall resources allocation time by keeping the system state balanced, but without minimizing the cost of

resource reservation [7].

A study [8] conducted by Abdullahi, M. et al presented a Discrete Symbiotic Organism Search (DSOS) algorithm was used for enhancing the scheduling of tasks in order to minimize the utilization time of physical resources. Yet, they neglected the tax of resources usage. To our knowledge, the proposed algorithm gave a better result than the Particle Swarm Optimization (PSO). On the other hand, in [9] proposed a (S-PSO) approach to solve workflow scheduling problem. This approach is subjected to various QoS constraints like the deadline constraint, the budget constraint, and the reliability constraint. However, it does not keep pace with cloud system stats change. In [13], the initial population of the proposed algorithm is realized using a bidirectional control method.

It allows computing the priority of scheduling tasks in both horizontal and vertical direction in graph topology (according to the depth of the task in the critical path).

Instead, the authors in [14] employed a Markov Decision Process as a learning agent to optimally guide the task execution process depending on the environmental state. Besides, they used a genetic algorithm to increase workflow schedules. However, the proposed approach still needs to adapt to use in resource allocation.

Whereas, the study [15] presented a new scheduling strategy bas

ed on spacing distance to rank non-dominate solutions. It aims mainly to minimize both the makespan and cost of execution tasks on virtual machines (VMs). In [16] researchers presented a scheduling strategy to trade-off for the energy consumption and the makespan. But, they did not consider monetary costs associated with the use of VMs. Moreover, the work [17] aimed to minimize the execution cost and the execution time withing the deadline and budget constraints. For this end, the authors applied a Bi-Criteria Priority based Particle Swarm Optimization (BPSO) approach. But, they neglected the transmission costs.

In the work [18], authors improved cost performance using an algorithm based on the symbiotic organism search. Additionally, in [19], they proposed an optimization of task scheduling based on a metaheuristic electromagnetism algorithm to minimize the makespan. However, these works require more evaluations to prove their effectiveness. In the paper [20], the authors presented an optimized version of FCFS for scheduling tasks in the cloud. This algorithm needs more evaluation to prove its effectiveness in cloud computing. The research [21] presented a queue-based job scheduling algorithm (QHS) for the efficient execution of user jobs. QHS used FCFS, SJF, and RR algorithms were employed to improve resource allocation in terms of average waiting time and average response time. Though, it did not consider the cost of resource utilization. In turn, the study [22] improved the Max-min algorithm to minimize the completion time of tasks. In [23] researchers improved the Max-min based on the expected execution time instead of completion time. Nevertheless, these methods did not lead to optimal solutions.

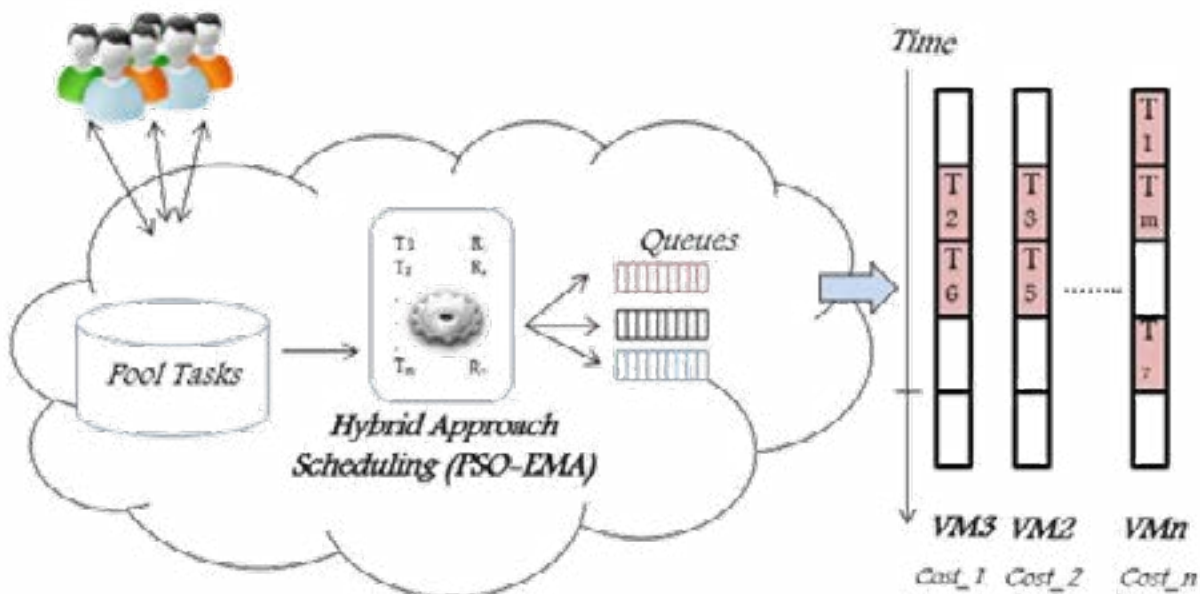


Fig.1. Hybrid approach scheduling (PSO-EMA) system architecture

III. TATEMENT OF THE PROBLEM

Figure 1 shows the system architecture of the proposed approach. The users send the requests to the cloud to express their needs. The scheduler picks out the tasks from requests and put them in the pool to run them over the available resources (VMs) in cloud. This, is achieved through using the PSO-EMA approach. Finally, the tasks execution process starts. In the studied problem, we consider the following notations:

- Let m the number of tasks and n the number of resources required for tasks. As well, each resource process on a virtual machine.
- Input: Set of VM = {VM1, VM2,...,VMn } with different processor configurations. Each machine is represented by an identifier (ID) and the parameter millions of instructions per second (MIPS). Also, set of independent tasks $T = \{T1, T2,...,Tm\}$ with different sizes.
- Output: The best mapping of T_i to VM_j (T_i, VM_j), in manner to reduce the resources allocation cost.
- Function Objective (Z): Reach a Tasks-VMs mapping, such as when estimating the total cost of execution all tasks is minimized (equations 1 and 2).

$$Z = \text{Min} \left[\sum_{i=1}^m f(x_{ij}) \right], \quad j \in [1, n] \quad (1)$$

$$f(x_{ij}) = \frac{T_{length_i}}{mips_j} \times price \quad (2)$$

IV. PROPOSED APPROACH

In order to treat the problem posed, we apply a hybridization between Particle Swarm Optimization algorithm (PSO) and Electromagnetism Metaheuristic Algorithm (EMA) (figure 2 and 3). The PSO and EMA are adequate approaches used widely in solving tough optimization problems. Table 1 lists the symbols used in the paper.

TABLE I. SYMBOLS

Symbols	Signification
Pn	Number of particles
Nvm	Number of VMs
Nt	Number of tasks
Pi	Initial population
Pf	Positive forces
Nf	Negative forces
v _i (.) u _i	Random numbers [0, Nvm]
f	Cost
β	$\in [0, 1]$
new_l	Index of the new virtual machine that running the task

A Particle Swarm Optimisation (PSO)

PSO algorithm was introduced by Kennedy (1995) as an evolutionary search algorithm inspired by social interaction observed among no evolution operators. The main advantage of PSO is its simplicity compared to other evolutionary algorithms such as GA [24]. Next, the PSO operators and equations used in this work are described.

The realization of PSO algorithm:

To realize this algorithm, we present the particles as tasks,

and their positions correspond to the index of virtual machines. In addition, the 'new_l' of each particle is the new VM of the task when searching for the optimal solution, as follow:

1) Initialize the Swarm: To initiate the PSO algorithm, we first define a group of the dimensions that are used in the scheduling operation such as number of VMs, number of tasks and number of particles. Then, we initialize randomly the position and velocity vectors.

2) Update the Swarm: To update the PSO particles, we follow these steps:

- Step 1: Calculate the fitness value of each particle using Eq.2.
- Step 2: If the fitness value is better than the previous value, set the current fitness value as the new best.
- Step 3: Select the best particle among all the particles as g^{best} .
- Step 4: The velocity of each particle is calculated using Eq.3 and 4. Then, its position is updated according to Eq.5 and 6.

3) The best Swarm: At the end, if our program reaches the maximum iteration or gets the ideal result (stable result), it stops. Otherwise, it repeats PSO processes from the initialization step. After that, we get a VMs-tasks mapping set, which will be enhanced using the EMA algorithm.

The PSO algorithm equations:

Indeed, there are different formulas of equations used to calculate velocity and position of the particle Swarm Optimisation algorithm. However, in this study the equations (3), (4), (5) and (6) are used as indicated in the work [6], but in two dimensions (x,y). At each iteration "k" and for each particle "i", we apply the following equations:

To update the particle velocity, the equations 3 and 4 are used:

$$V_x^{k+1} = wV_x^k + rC_1(x_{pbest}^k - x_k) + rC_2(x_{gbest}^k - x_k) \quad (3)$$

$$V_y^{k+1} = wV_y^k + rC_1(y_{pbest}^k - y_k) + rC_2(y_{gbest}^k - y_k) \quad (4)$$

Whereas for the particle position updating the equations 5 and 6 are used:

$$x^{k+1} = x^k + V_x^k \quad (5)$$

$$y^{k+1} = y^k + V_y^k \quad (6)$$

Thus, to determine the location of particle (defined with two variables) the following empirical equation is solved:

$$new_l = 4.33 - x(3 + 2\sqrt{y + y}) \quad (7)$$

Where:

g^{best} is the best particle in individuals of the current population.

- p^{best} is the best position of particle.
- V_x^k and V_y^k present the velocity of particle at iteration k.
- w is an inertia weight.
- C_1 and C_2 are acceleration coefficients.
- r is a random number between 0 and 1.
- x^k and y^k are a current position of particle at iteration k.

The pseudo code of PSO is presented as algorithm 1.

Algorithm 1: PSO

Input: Pn, Nvm, Nt

Result: Initiale population Pi

iter :=0; max_iter:=100; ideal_result= false;

While(iter < max_iter or ideal_result = false) do

 function INITIALIZESWARM()

 function UPDATEFITNESSLIST()

 iter ++

end

Return the best Swarm.

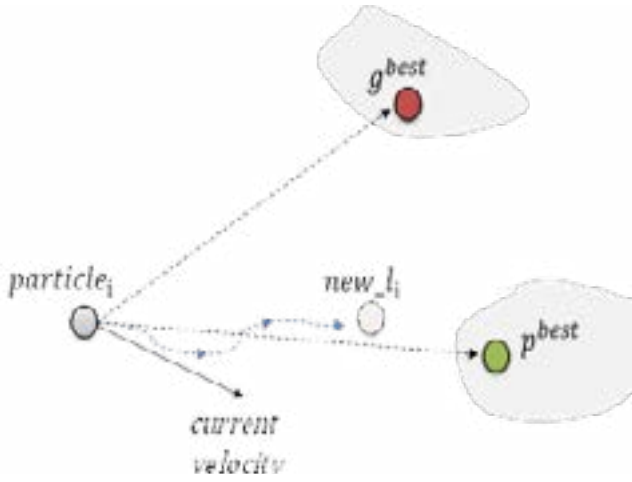


Fig.2. The principle of PSO algorithm

B. Electromagnetism algorithm (EMA)

Electromagnetism algorithm (EMA) is a heuristic technique for global optimization, it was proposed by Birbil and Fang (2003). The method is based on the principle of attraction-repulsion (Coulomb's Law) to move the sample points towards the optimality. The principle of this algorithm is that inferior solution points will prevent a move in their direction by repelling other solution points in the population, and that attractive points will facilitate moves in their direction [25].

EMA consists of four main phases: initialization of the points' positions, calculation of the total force exerted on each point, movement along the direction of the force, and application of neighborhood search to exploit the local minima. In this study, EMA operators are used as shown in [26].

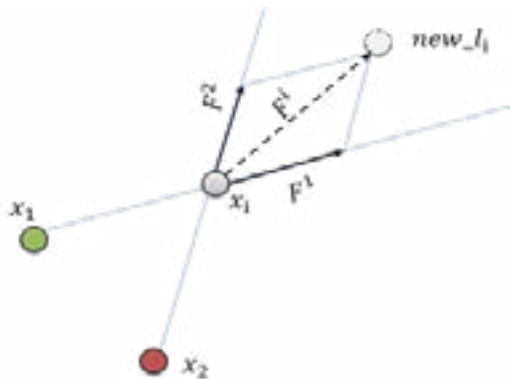


Fig.3. The principle of EMA algorithm

Realization of EMA algorithm:

In our case, we consider the points as tasks and their positions as selected virtual machines. The EMA steps are:

1) Initialization: We use the above mentioned algorithm to obtain the first population of EMA algorithm. So that, the sort population of PSO considered as the initial points of EMA, as shown in the figure 4 and algorithm 2.

2) Evaluation of points: After the initial positioning of the tasks over the VMs, we examine the fitness $f(x_i)$ of each bind by using the equation 2.

3) Compute charges: The point charges are calculated according to equation 8.

4) Compute forces: The point forces are calculated according to equation 9 and 10.

5) Calculate the reference force (rf):

We calculate the reference force according to the flowing empirical formula:

$$rf = \rho \frac{highforce - lowforce}{N_t}$$

$$\text{Where: } \rho \in \left[0, \frac{1}{4}\right]$$

6) Classification of forces: At this step, we arrange and classify the forces to positive Pf and negative forces Nf. If the value of force F^i is inferior to reference force (rf), the force sign is positive. Whereas, if it is greater than or equal to the reference force (rf), the force sign is negative.

7) Moved: The points are moved according to the equation 11. In the case where the position value given by equation 11 is greater than the numerical index of VM in the pool, the task is assigned to VM which gives the minimum cost. The algorithm continues looping until it reaches the stop criteria (maximum number of iterations).

The EMA algorithm equations:

A function local was used to examine the existing neighbourhood of an individual point. Also, a function named compute forces to calculate the total force was exerted on each point by all other points; knowing that the charge of each point x_i is determined by its fitness function value $f(x_i)$ in relation to the fitness value of the current best point x^{best} in the population. Hence, the charge Q^k of particle x_i is determined according to equation 8:

$$Q_i = \exp \left(-m \frac{f(x_i) - f(x^{best})}{\sum_{l=1}^{m-1} (f(x_l) - f(x^{best}))} \right) \quad (8)$$

Where m denotes the size of the population, a set of force vectors F^i ($i = 1, 2, \dots, m-1$) that are exerted on the corresponding point x_i is determined as follows:

If $(f(x_i) < f(x^l))$, where $l \in [1, m]$ and $l \neq i$

$$F^i = \sum_{l=1}^{m-1} (x^l - x_i) \frac{Q^l Q_i}{\| (x^l - x_i)^2 \|} \quad (9)$$

If $(f(x_i) \geq f(x^l))$

$$F^i = \sum_{l=1}^{m-1} (x^l - x_i) \frac{Q^l Q_i}{\| (x^l - x_i)^2 \|} \quad (10)$$

Fig.4. Hybridization of PSO-EMA

The movement according to the resulting forces is performed using the function apply forces, which generates a new population. The magnitude of each move is determined randomly by the parameter β ; which is generated from a uniform distribution from the interval $[0, 1]$.

$$\text{if } F^i > 0 \quad \text{new_}l_i = x_i + \beta \frac{F^i(\mu_i - x_i)}{\|F^i\|} \quad (11)$$

$$\text{if } F^i \leq 0 \quad \text{new_}l_i = x_i + \beta \frac{F^i(x_i - \nu_i)}{\|F^i\|} \quad (12)$$

The pseudo code of EMA is presented in algorithm 2.

Algorithm 2: PSO-EMA

Input : Algorithm1 results

Result: An optimized generated schedule

Call Algorithm1

while iter < max_iter do

 foreach x_i do

 if $f(x_i) < f(x^{\text{best}})$ then

$x^{\text{best}} \leftarrow x_i$

 End

 // Compute charges

 Foreach x_i do

 Calculate Q_i according to equation (8)

 End

 // Compute forces

 foreach x_i do

 Calculate F^i according to equations (9 and 10)

 if $F^i \geq 0$ then

$Pf \leftarrow F^i$

 if $F^i < 0$ then

$Nf \leftarrow F^i$

 end

 // Apply forces

 foreach F^i do

 if $F^i < rf$ then

 Calculate the new position x_c of x_i according to equation

(11)

$x_i \leftarrow x_c$

 end

 end



V. PERFORMANCE EVALUATION

A.Tools and Platform

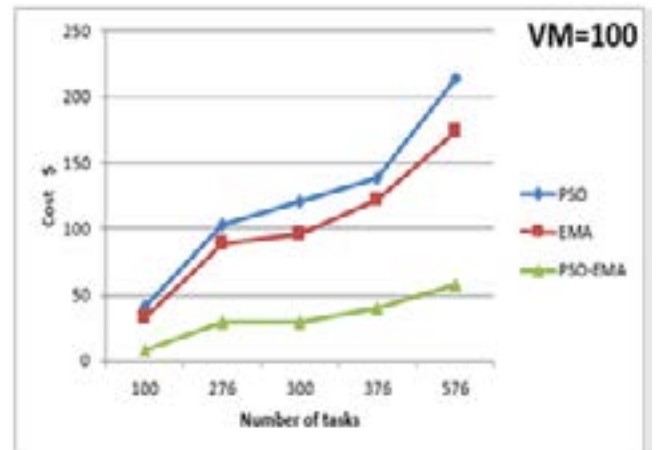
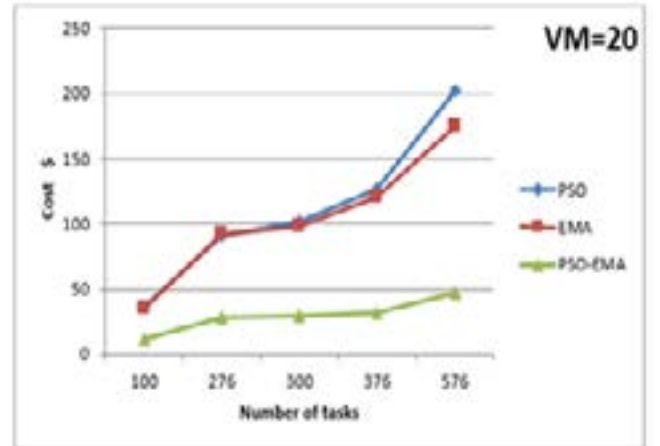
The platform used to execute experiments was an Intel I5 3320M 2.60 GHz equipped with a 4GB RAM and an OS windows 7 professional. The experiments programmed with JAVA language in Eclipse IDE Luna release 4.4.2 using Cloudsim 3.0 toolkit, which is a toolkit for simulating Cloud computing scenarios [27]. In addition, the tests were done on one data center containing fifty hosts. Each host has 20 GB of RAM, 1 TB of storage and 10 GB/s of bandwidth.

B.Simulation and Results

In order to illustrate the effectiveness of this approach among the other scheduling algorithms, PSO-EMA is compared to that of the original Max-min, FCFS and PSO. The comparison is done with the DSOS algorithm because the cost has a proportional relationship with the time that the task takes to use the resource, where the DSOS gave a better performance in this area. Its main steps were implemented as mentioned in reference [8], but in a constructive manner to minimize the VMs reservation cost. Therefore, we applied two types of tests, an empirical test and Wilcoxon signed rank test to verify the performance of PSO-EMA.

Empirical test: The above scheduling approach was tested using a common criteria as resource allocation cost, where we consider the reservation cost of each VM equal to 4 \$/hour. The test was applied to three cases: in the first case, it was assumed that the number of VMs is 20. In the second case the number of VMs is 100 and in the last case 200 VMs were identified.

The MIPS of each VM is between [1000. 2000]. The PSO parameters are set at $c_1 = 2.0$, $c_2 = 2.0$ and $w_i = \frac{ITERATION_i}{ITERATION_{MAX}}$, these values are empirically chosen.



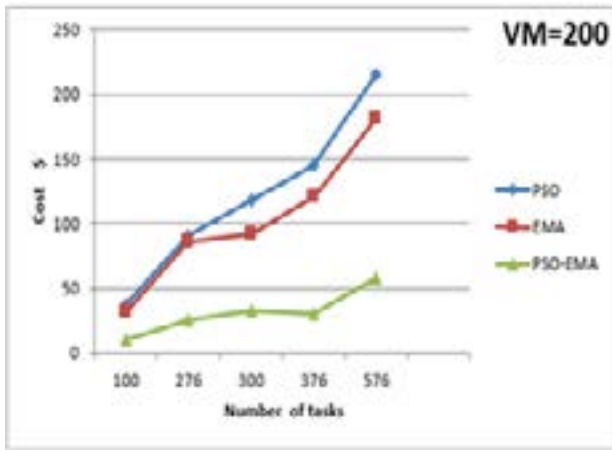


Fig.5. The variation in the total cost of VMs reservation for EMA, PSO and PSO-EMA algorithms (for 20, 100 and 200 VMs).

From figure 5, it is clear that cost increases as the number of tasks increased within the three cases, whereas the PSO-EMA graph is located under the other graphs which proves the effectiveness of our approach. On the other hand, the proposed algorithm keeps the same variation of results despite the different numbers of VMs, which reflect its stability.

The experiment was repeated by changing the MIPS value of VMs at different intervals, for various numbers of tasks and VMs. It is also suggested that the reservation cost of each VM is like 0.04 \$/hour. The simulation is as is depicted in table II.

TABLE II. SIMULATION RESULTS OF COST FOR MAX-MIN, FCFS, PSO, EMA, DSOS AND PROPOSED ALGORITHM

Cases	Tasks	Length	VMs	MIPS	Max-min	FCFS	PSO	EMA	DSOS	PSO-EMA
					Cost					
Case 1	130	[200,300]	245	[500,1000]	932.234	19.578	1.791019	1.383	1.932	1.260
	387				2680.927	55.283	5.451	3.820	2.111	1.480
	599				4274.34	84.667	8.125	2.181	1.856	5.831
Case 2	130	[900,2345]	344	[1234,4567]	883.764	34.143	3.329	2.105	3.112	2.107
	387				2743.854	107.542	9.969	6.564	3.668	2.184
	599				4233.925	152.406	14.102	9.799	3.183	2.617
Case 3	130	[8000,27853]	400	[5672,34596]	929.184	69.749	5.032	4.079	10.910	5.404
	387				2763.32	171.085	13.927	8.795	10.535	8.719
	599				4187.545	285.466	26.978	13.757	8.025	2.986

The results illustrated in table II indicate clearly that the proposed scheduling approach performs well compared with the other algorithms; it gave significant improvements in terms of cost. The cost of PSO-EMA is the minimum among all the other algorithms because it hired the cheapest VMs that achieved the minimum cost.

Wilcoxon signed ranks test: The Wilcoxon signed rank test is a non-parametric test, used to examine whether there is a statistically significant difference between two related groups (algorithms). The test essentially computes the differences between each set of pairs and analyses these differences [28]. It necessitates a null hypothesis H_0 and an alternate hypothesis H_1 defined as:

H_0 : The median difference is zero versus

H_1 : The median difference is not zero

During the test, a level of significance is set to 0.05. If the p-value is less than (or equal to) 0.05, then H_0 is rejected in favor of H_1 . We applied this test on a sample of the results obtained in the table II (case 1). Fifteen samples are used and the cost is calculated according to equation 1. The steps of this test are as follows:

$$w^- = 7 + 9 = 16$$

$$w^+ = 1+2+3+4+5+6+8+10+11+12+13+14+15 = 104$$

W

The total of $(w^- + w^+)$ should be equal to $\frac{M(M+1)}{2}$

is the number of observed pairs in the sample). Therefore, we have:

$$\frac{15 \times 16}{2} = 120 \quad w = \max(W^-, W^+)$$

We have one group of two tied ranks. So we must reduce the variance (∇) by: $\frac{8-2}{48} = 0.125$. We get

$$\nabla = \frac{104 - \frac{15 \times 16}{4}}{\sqrt{\frac{15 \times 16 \times 31}{24} - 0.125}} = 2.5$$

This gives a two-sided p-value of $p = 0.012419$, where the result is significant at $p < 0.05$. Hence, there is strong evidence that PSO-EMA reduces the VM reservation cost more than any other compared method.

It is believed that PSO-EMA performance improvement is due to the hybridization manner of the two algorithms. The PSO

processes contribute to forming an improved population that is an initial population for the EMA algorithm (Fig.4). After that, EMA operators manipulate these individuals to obtain optimal solutions.

TABLE III. WILCOXON COST DIFFERENCE RESULTS

	Pair	Diff.	Pair	Diff.	Pair	Diff.
Case 1 of table II	DSOS/ PSO-EMA	0.672	DSOS/ PSO-EMA	0.631	DSOS/ PSO-EMA	-3.65
	PSO/ PSO-EMA	3.886	PSO/ PSO-EMA	12.821	PSO/ PSO-EMA	16.342
	FCFS/ PSO-EMA	0.752	FCFS/ PSO-EMA	4.608	FCFS/ PSO-EMA	3.604
	EMA/ PSO-EMA	0.749	EMA/ PSO-EMA	0.26	EMA/ PSO-EMA	-3.975
	Max-min/ PSO-EMA	930.974	Max-min/ PSO-EMA	2679.447	Max-min/ PSO-EMA	4231.308

TABLE IV. LCOXON SIGNED RANKS TEST RESULTS

Diff.	0.26	0.631	0.672	0.749	0.752	0.26
Rank	1	2	3	4	5	1
Sign	+	+	+	+	+	+
Diff.	3.604	-3.65	3.886	-3.975	4.608	3.604
Rank	6	7	8	9	10	6
Sign	+	-	+	-	+	+
Diff.	12.821	16.342	930.974	2679.447	4231.308	12.821
Rank	11	12	13	14	15	11
Sign	+	+	+	+	+	+

VI. CONCLUSION

This paper presented the PSO-EMA approach which is designed for the cloud task scheduling problem. In order to satisfy the Service Quality requirements given to users, we considered a scheduling model that minimizes the cost of VMs usage. The experiment results show that the proposed approach has a promising performance in terms of cost optimization, especially it gives a better result than Max-min, FCFS, EMA, PSO, and DSOS methods. In the future, we look forward to implementing this algorithm with more measurable factors in real-time (e.g., elasticity, workload, etc.). We would even try to improve the proposed approach under various constraints.

REFERENCES

- [1] Raju, I.R.K., Varma, P.S., Sundari, M.R., Moses, G.J.: Deadline aware two stage scheduling algorithm in cloud computing. *Indian Journal of Science and Technology* 9(4) (2016)
- [2] Alkhanak, E.N., Lee, S.P., Khan, S.U.R.: Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. *Future Generation Computer Systems* 50, 3–21 (2015)
- [3] Yang, H.: Improved ant colony algorithm based on pso and its application on cloud computing resource scheduling. In: *Advanced Materials Research*. vol. 989, pp. 2192–2195. *Trans Tech Publ* (2014)
- [4] Ghribi, C., Hadji, M., Zeghlache, D.: Energy efficient vm scheduling for cloud data centers: Exact allocation and migration algorithms. In: *Cluster, Cloud and Grid Computing (CCGrid)*, 2013 13th IEEE/ACM International Symposium on. pp. 671–678. IEEE, Delft, Netherlands (13–16 May 2013)
- [5] Pandey, S., Wu, L., Guru, S.M., Buyya, R.: A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In: *Advanced information networking and applications (AINA)*, 2010 24th IEEE international conference on. pp. 400–407. IEEE, Perth, Western Australia (20–23 April 2010)
- [6] Wu, Z., Ni, Z., Gu, L., Liu, X.: A revised discrete particle swarm optimization for cloud workflow scheduling. In: *Computational Intelligence and Security (CIS)*, 2010 International Conference on. pp. 184–188. IEEE, Nanning, Guangxi, China (11–14 December 2010)
- [7] Kapur, R.: A cost effective approach for resource scheduling in cloud computing. In: *Computer, Communication and Control (IC4)*, 2015 International Conference on. pp. 1–6. IEEE, Medi-Caps Group of Institutions A.B. Road Pigdamber Rau, Indore Indore, India (10 Sep - 12 Sep 2015)
- [8] Abdullahi, M., Ngadi, M.A., et al.: Symbiotic organism search optimization based task scheduling in cloud computing environment. *Future Generation Computer Systems* 56, 640–650 (2016)
- [9] Chen, W.N., Zhang, J.: A set-based discrete pso for cloud workflow scheduling with user-defined qos constraints. In: *Systems, Man, and Cybernetics (SMC)*, 2012 IEEE International Conference on. pp. 773–778. IEEE, COEX Seoul, Korea (South) (14 Oct - 17 Oct 2012)
- [10] Calheiros, R.N., Buyya, R.: Meeting deadlines of scientific workflows in public clouds with tasks replication. *IEEE Transactions on Parallel and Distributed Systems* 25(7), 1787–1796 (2014)
- [11] Zuo, L., Shu, L., Dong, S., Zhu, C., Hara, T.: A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. *IEEE Access* 3, 2687–2699 (2015)
- [12] Belgacem, A., Beghdad-Bey, K., Nacer, H.: Task scheduling in cloud computing environment: A comprehensive analysis. In: *International Conference on Computer Science and its Applications*. pp. 14–26. Springer, Algiers, Algeria (24–25 April 2018)
- [13] Aryan, Y., Delavar, A.G.: A bi-objective workflow application scheduling in cloud computing systems. *Int J Integr Technol Educ* 3, 51–62 (2014)
- [14] Barrett, E., Howley, E., Duggan, J.: A learning architecture for scheduling workflow applications in the cloud. In: *Web Services (ECOWS)*, 2011

- Ninth IEEE European Conference on. pp. 83–90. IEEE, 14-16 Sept., in Lugano, Switzerland (2011)
- [15] Belgacem, A., Beghdad-Bey, K., Nacer, H.: A new task scheduling approach based on spacing multi-objective genetic algorithm in cloud. In: International Conference on Computer Science and Information Systems. pp. 189–195. Pozna, Poland (9 - 12 September 2018)
 - [16] Juarez, F., Ejarque, J., Badia, R.M.: Dynamic energy-aware scheduling for parallel task-based application in cloud computing. *Future Generation Computer Systems* 78, 257–271 (2018)
 - [17] Verma, A., Kaushal, S.: Bi-criteria priority based particle swarm optimization workflow scheduling algorithm for cloud. In: Engineering and Computational Sciences (RAECS), 2014 Recent Advances in. pp. 1–6. IEEE, Chandigarh, India (6-8 March 2014)
 - [18] Ali Belgacem, Kadda Beghdad-Bey, and Hassina Nacer. Enhancing cost performance using symbiotic organism search based algorithm in cloud. In 2018 International Conference on Smart Communications in Network Technologies (SaCoNeT), pages 306–311, 27-31 Oct. 2018, in El Oued, Algeria, 2018. IEEE.
 - [19] Ali Belgacem, Kadda Beghdad-Bey, and Hassina Nacer. Task scheduling optimization in cloud based on electromagnetism metaheuristic algorithm. In 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), pages 1–7, 24-25 Oct. 2018, in Tebessa, Algeria, 2018. IEEE.
 - [20] Shahram Behzad, Reza Fotuhi, and Mehdi Effatparvar. Queue based job scheduling algorithm for cloud computing. *International Research Journal of Applied and Basic Sciences* ISSN, 37853790, 2013.
 - [21] Upendra Bhoi, Purvi N Ramanuj, et al. Enhanced max-min task scheduling algorithm in cloud computing. *International Journal of Application or Innovation in Engineering and Management (IIAEM)*, 2(4):259–264, 2013.
 - [22] OM Elzeki, MZ Reshad, and MA Elsoud. Improved max-min algorithm in cloud computing. *International Journal of Computer Applications*, 50(12), 2012.
 - [23] Aditya Marphatia, Aditi Muhnot, Tanveer Sachdeva, Esha Shukla, and Lakshmi Kurup. Optimization of fcfs based resource provisioning algorithm for cloud computing. *Journal of Computer Engineering (IOSRJCE)*, 10(5), 2013.
 - [24] Mirzayi, S., Rafe, V.: A hybrid heuristic workflow scheduling algorithm for cloud computing environments. *Journal of experimental & theoretical artificial intelligence* 27(6), 721–735 (2015)
 - [25] Debels, D., Vanhoucke, M.: The electromagnetism meta-heuristic applied to the resource-constrained project scheduling problem. In: International Conference on Artificial Evolution (Evolution Artificielle). pp. 259–270. Springer, Lille, France (26-28 October 2005)
 - [26] S Ilker Birbil and Shu-Chering Fang. An electromagnetism-like mechanism for global optimization. *Journal of global optimization*, 25(3):263–282, 2003.
 - [27] Calheiros, R.N., et al., CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software - Practice and Experience*, 2011. 41(1): p. 23-50.
 - [28] García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences* 180(10), 2044–2064 (2010)