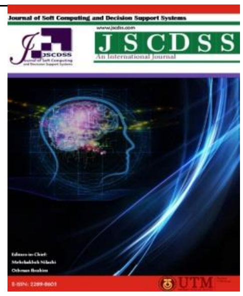
An International Journal
JSCDSS
http://www.jscdss.com
Vol.7 No.4 June 2020: 18-26
Article history:
Accepted 12 June 2020
Published online 12 June 2020

Journal of Soft Computing and Decision Support Systems



E-ISSN: 2289-8603

A Hybrid Swarm Particle Optimization Algorithm for Task Scheduling in Cloud Computing

Ehram Safari [a], Seyedeh OmSalameh Pourhashemi [b], Mohsen Gharahkhani [c]

[a] ICT Research Institute (Iran Telecommunication Research Center), Tehran, Iran
[b] Islamic Azad University Central Tehran Branch, Tehran, Iran
[c] Department of Finance, Iranian E-Institute of Higher Education, Tehran, Iran

*** Corresponding author email address**: pourhashemi1986@gmail.com

Abstract

Today, cloud computing experts seek internet-based service providing to share resources using service providing techniques. This environment provides users with an image of abundant resources. The present paper recommends a combination of particle swarm optimization algorithm and simulated annealing algorithm to obtain an improvement in the performance of task scheduling to resources considering the available bandwidth allocated to each virtual machine. The performance of the proposed algorithm is investigated by the use of the Cloudsim Simulator. Research results show that the proposed algorithm outperforms the Swarm Particle Optimization (SPO), bat, and raven roosting optimism algorithms in terms of task execution time, response time, and performance efficiency.

Keywords: Cloud computing, Cloudsim Simulator, Swarm Particle Optimization, Efficiency, Task scheduling

1. Introduction

At the present age, the speed in the advancement of science and the fast development of information technology had led to the need to use new computing methods. In the past, models such as high-performance computing, utility computing, and grid computing were presented (Zhang, Cheng, & Boutaba, 2010). Considering the developments in computing areas, many techniques such as data clustering, grid computing, and distributed database management system have been introduced to distribute resources and use data (Beloglazov, Buyya, Lee, & Zomaya, 2011). Today, cloud computing technology is very popular due to providing numerous facilities and advantages such as high computing power, flexibility, high performance, providing services proportionate with demands, users' ease of accessibility without time and place limitations and simultaneous access to multiple documents, and it is used in everyday life at a higher speed (Asadi, Nilashi, Husin, & Yadegaridehkordi, 2017; Mohammed, Ibrahim, Nilashi, & Alzurqa, 2017; Rimal, Choi, & Lumb, 2009; Yadegaridehkordi, Shuib, Nilashi, & Asadi, 2019). Cloud computing shares the data and provides its services clearly under the title of internet services(Wang et al., 2010) . Three sets should be taken into consideration when defining scheduling problems: tasks, processors, and resource sets. The relationships between the tasks and the time constraints of each task need to be determined. Scheduling means determining the allocation order of processor of set P and resources of set R to tasks set T such that all the tasks are completed under the predetermined and governing conditions and constraints. Scheduling is one of the issues where an increase in the size of the problem leads to an exponential growth in the problem-solving time. Such problems are called complete-NP (Buttazzo, 2011). With an increase in the number of cloud users, the tasks that need to be scheduled will increase. Cloud scheduling is considered a mechanism that allocates user tasks to appropriate resources to be performed and affects the cloud performance directly (Choudhary & Peddoju, 2012). Cloudsim simulator is a general simulator framework that can be developed and it allows researchers to model and simulate different scenarios in the cloud computing environment (Calheiros, Ranjan, Beloglazov, De Rose, & Buyya, 2011). In the cloud simulator, the data centers, virtual machines, tasks, network connections, different allocation policies, service broker, and virtualization management can be defined. Therefore, the proposed method is investigated in cloudsim environment and the simulations are modeled according to the information from the real data centers to achieve the actual results. The remaining sections of the paper are organized as follows. Section 2 describes the proposed method. Section 3 describes the proposed algorithm and the process steps of the proposed algorithm in detail. Section 4 presents the test and the empirical results and finally, Section 5 presents the conclusion and further studies.

2. Description of the Proposed Method

The strategy in this study is proposed to reduce the task execution time, reduce the response time and increase efficiency which performs based on the idea that it leads to significant improvements in the stated targets by the use of a meta-heuristic algorithm at task scheduling. The proposed method in the present study is suggested as follows:

Step One: the target scheduling problem is a dynamic scheduling problem for independent tasks. Since the environment that the process is being performed on over its lifespan is not clear and definite, the algorithm has little information regarding the requirements of the resources of a process, the jobs or resource sets are not fixed in this method and the time of tasks entrance is different, the processes in dynamic scheduling are allocated to a processor at the beginning of the execution and the allocation might be repeated while implementation (Chapin, Katramatos, Karpovich, & Grimshaw, 1999).

Step Two: the meta-heuristic optimization algorithms are used to solve the abovementioned scheduling problem because the meta-heuristic methods provide approximately optimal answers at the logical time. This type of algorithm is examined at each period of implementation according to the obtained value of the target function and follows a method according to the policies latent in the algorithm that leads them to the global optimal point in the next period (Karlin, 1955). A combination of meta-heuristic, particle swarm optimization and simulated annealing algorithms are used in the present study to solve the scheduling problem. Hence, the performance of both algorithms was improved and then the combination of the improved algorithms was applied on cloud computing for a better scheduling of the tasks.

Step Three: the presence of the termination condition is very important in meta-heuristic algorithms. In the current scenario, the highest number of repetitions is considered the termination condition. Every time the meta-heuristic algorithm is recalled, it is called a repetition of the algorithm. The termination condition that is determined according to the most number of repetitions is such that regardless of the value of the target function if the number of algorithm repetitions reaches n, the value of the target function is returned as the optimal response.

Step Four: the objective function is the evaluation function of the value of a solution in the proposed algorithm. Since the purpose of the present study is to present an appropriate scheduling algorithm to properly allocate tasks to the available and accessible resources, the response time reduces and efficiency increases by following the approach to reduce the execution time of each task. Hence, the execution time, response time, and efficiency are considered objective functions. Each of these objective functions considers a numerical value for the solution. Meanwhile, the solution with the highest value for the aforementioned functions is taken as the optimal response. The aforementioned objective functions are achieved by the use of the following equations (Buttazzo, 2011; Kalra & Singh, 2015):

$$Execution\ Time = \frac{lj}{vmmips_i} \quad (1)$$

where *l* is the length of *j*th task and vmmipsi is the value of the processing power of the ith virtual machine.

$$\text{Response Time} = (\mathrm{f_i} - \mathrm{a_i}) \quad (2)$$

where *fi* is the end time of task execution and ai is the task transfer time for execution.

$$throughput = \frac{\sum_{j=0}^{m} Tasks}{Makspan} \quad (3)$$

where $\sum_{j=0}^{m}$ Tasks represents the total number of tasks and Makspan represents the end time of the last task which is calculated using the following equation:

$$Makespan = max_{i \in tasks}\{f_i\} \quad (4)$$

3. The Proposed Algorithm

The parameters of the proposed algorithm are presented in Table 1.

Table 1
The parameters of the proposed algorithm

PSOSA	$\mathrm{w} = 1, \mathrm{C_1} = \mathrm{C_2} = 2, \mathrm{P} = 0.1, \mathrm{wdamp} = \alpha = 0.99, \mathrm{T_0} = 10$

3.1. Processing Steps of the Proposed Algorithm

1. Definition of the solution space [VarMax, VarMin], fitness function and population size
2. Creating an initial population from particles and initializing the parameters of position (X_{old}), velocity (V), the best global position (P_{gbest}) and the best personal position (P_{pbest})
3. Until the most repetitions are not finished (examination of termination condition), the following steps will continue for each particle at each repetition:

A. The position and velocity of the particle is updated.

The new velocity is calculated using the following equation:

$$V_{new} = w \times V_{old} + C_1 r_1 \times (P_{pbest} - x_{old}) + rand \times (x_k - x_{old}) + C_2 r_2 \times (P_{gbest} - x_{old}) \quad (5)$$

The symbols used in the equation are presented in Table 2.

The new position of the particle is updated through the following equation:

Updating the particle position, the random number (rand) is compared with parameter $P = 0.1$; if the random number is less than $P = 0.1$, then the new position of each particle is calculated through the following equation:

$$rand < P \quad x_{new} = (x_k - x_{old}) + T \times P_{gbest} \quad (6)$$

An International Journal JSCDSS
E-ISSN: 2289-8603

The symbols used in the above equation are presented in Table 3.

Table 2
The symbols of the new velocity of the particle

V_{new}	The new velocity of the particle
V_{old}	The old velocity of the particle
W	The inertia coefficient
c_1	Personal learning coefficient
c_2	Collective learning coefficient
P_{pbest}	The best personal position for each particle
P_{gbest}	The best global position
x_{old}	The current position of the particle
rand	A random number in the interval [0, 1]
x_k	Position of the k^{th} particle randomly selected from the population
r_1 and r_2	Random numbers with uniform distribution

Table 3
The symbols of the new position of the particle

x_{new}	The new position of the particle
x_k	The position of the k^{th} particle randomly selected from the population
x_{old}	The current position of the particle
T	Represents temperature which was considered 10 at step one of the present study and its value reduces at each repetition.
P_{gbest}	The best global position

If the random number (rand) is greater than $p = 0.1$, then the new position of each particle is calculated through the following equation:

$$rand > P \ \ x_{new} = x_{old} + \left(1 - \frac{it}{MaxIt}\right) \times V_{new} \qquad (7)$$

where,
it: current repetition
MaxIt: the maximum number of repetitions.

B. Evaluation of the fitness function

The new position of the particle is compared with its old position through the following equation:

$$\delta = F(x_{new}) - F(x_{old}) \qquad (8)$$

In this equation, δ represents the fitness difference between the new and old positions. If the value of δ is less than or equal to 0, then the old position is replaced by the new position which is new and good.

Otherwise, if δ is greater than 0, then the new position is accepted with a probability (only when the value of the random number (rand) is less than or equal to p_1, then the new and bad position is accepted and replaced for the current position:

$$P_1 = e^{\left(-\frac{\delta \times \sigma}{T}\right)} \qquad (9)$$

where, T represents temperature and its value reduces at each repetition and σ is an example of changing Morlet wavelet which is achieved using the following equation (Awad et al., 2016):

$$\sigma = \frac{1}{\sqrt{a}} e^{-\left(\frac{rand}{a}\right)^2/2} cos\left(5\left(\frac{rand}{a}\right)\right) \qquad (10)$$

Such that is a positive number and a scale parameter to change the wavelet and it is achieved using equation (Masdari, Salehi, Jalali, & Bidaki, 2017):

$$a = e^{\left(T \times \left(1 - \frac{it}{MaxIt}\right)\right)} + \alpha \qquad (11)$$

C. Updating the best personal position:

If the fitness of the new position is better than the best personal position, then the best personal position is replaced by the new position.

D. Updating the best global position

If the fitness of the new position is better than the best global position, then the best global position is replaced by the new position.

E. All the above-mentioned steps (from A to D) are performed on each and every particle at each repetition and finally the best global position along with its fitness is saved as the best position for that repetition.

F. At the end of each repetition, the inertia coefficient (W) and temperature (T) will reduce as much as wdamp and alpha, respectively.

$$W = W * wdamp \qquad (12)$$

$$T = T * alpha \qquad (13)$$

Where, wdamp represents the change rate of the inertia coefficient at each repetition, and alpha shows the temperature change rate at each repetition.

If the termination condition does not hold and the next repletion is going to happen for the algorithm, the tendency of each particle to change the current position and moving to the new position needs to increase at each repetition. That is, the inertia coefficient should reduce as much as wdamp, in which situation the particle does not tend to keep its current position and can take a new position. Besides, the temperature should reduce as much as alpha at each repetition so that gradually by the reduction in the temperature, the algorithm can converge toward a global optimal response (position).

G. If the number of repetitions reaches its maximum (termination condition), then the best global position along with its fitness created until this stage will be returned as the best optimal response (position). Otherwise, the next repetition will be taken and all the above-mentioned steps will be repeated.

Two concepts of the ability to search (exploration) and the ability to extract (exploitation) are put forward in the optimization algorithms (Masdari et al., 2017). In other words, the algorithm should be able to nurture the ideas created in the problem to make use of them. If the algorithm is merely about searching, the method turns into random searching and if there is mere extraction, then the method turns into local search and this leads the problem to the direction that local optimizations will be trapped. None of these two abilities is suitable per se for a heuristic algorithm and both abilities need to be present in the heuristic techniques to an adequate level. The mode that is usually taken in problem-solving and it is also followed here is such that first the search level is considered high so that the random conditions might lead to finding the appropriate responses quicker and then as the algorithm reaches its end, the search level is reduced and extraction increases. Therefore, a very good balance will be between extraction and search in the proposed algorithm according to the aforementioned equations. To have a better balance between extraction and search in the proposed algorithm, σ

An International Journal
JSCDSS
E-ISSN: 2289-8603

was used which is an example of wavelet change. Accordingly, in the first repetitions of the algorithm where the temperature is high, the probability of admitting new and bad responses increases and it leads to an increase in the search. While by the increase in the number of algorithm repetitions and reduction of the temperature, the probability of admitting new and bad positions reduces and extraction increases and the algorithm converges towards a global optimal response (position). The pseudo-code of the proposed algorithm is presented in Table 4.

Table 4
The pseudo-code of the proposed algorithm

Procedure PSOSA	
1	**Define the solution space, fitness function, and population size**
	Initialization
2	Initialise current position(x_{old}), velocity(v), P_{pbest}, P_{gbest}
	Where P_{pbest} is the personal best position and P_{gbest} is the global best position
	For each particle
3	{
	3-1-Update velocity:
	$\mathbf{V_{new} = w \times V_{old} + C_1 r_1 \times (P_{pbest} - x_{old}) + rand \times (x_k - x_{old}) + C_2 r_2 \times (P_{gbest} - x_{old})}$
	Where V_{new} is the new velocity, $w = 1, C_1 = C_2 = 2$, r1, r2, rand ε [0, 1]
	x_k is the position of member who is randomly selected from the members of the population
	3-2-Update position
	P=0.1 P is the Possibility to update position
	If (rand(0,1) > P(0.1)) then
	$\mathbf{x_{new} = (x_k - x_{old}) + T \times P_{gbest}}$
	Where X_{new} is the new position and T is a temperature parameter that reduce in each iteration
	Else
	$\mathbf{X_{new} = X_{old} + \left(1 - \frac{it}{MaxIt}\right) \times V_{new}}$
	Where it is the current repetition and MaxIt is the maximum number of iterations
	End if
	3-3-Evaluate Fitness function(F)
	$\boldsymbol{\delta} = \mathbf{F(X_{new}) - F(X_{old})}$
	Where $\mathbf{F(X_{new})}$ is the Fitness of the new position and $\mathbf{F(X_{old})}$ is the Fitness of old position
	If δ< = 0 then
	$\mathbf{X_{old}=X_{new}}$
	Else
	$\mathbf{P_1 = e^{\left(-\frac{\delta \times \sigma}{T}\right)}}$
	Where T is a temperature parameter that reduces in each iteration and **σ** is the Morlet wavelet transform
	$\boldsymbol{\sigma} = \mathbf{\frac{1}{\sqrt{a}} e^{-\left(\frac{rand}{a}\right)^2/2} \cos\left(5\left(\frac{rand}{a}\right)\right)}$
	Where a is a scale parameter in wavelet transform and it is a positive number
	If rand(0,1)<=P1 then
	$\mathbf{F(X_{old}) = F(X_{new})}$
	$\mathbf{X_{old}=X_{new}}$
	End if
	End if
	3-4-**If F ($\mathbf{X_{new}}$) >= F ($\mathbf{P_{Pbest}}$)** then
	$\mathbf{P_{pbest} = X_{new}}$
	Where F (P_{Pbest}) is the Fitness of personal best position
	End if
	3-5- **if F ($\mathbf{X_{new}}$) >= F ($\mathbf{P_{gest}}$)**
	$\mathbf{P_{gbest} = X_{new}, F(P_{gbest}) = F(x_{new})}$
	Where F(P_{gbest}) is the Fitness of global best position
	End if
	}
	3-6-Next particle // repeat step of 3-1 until 3-5
4	**W= W*wdamp T= T*alpha**
	if No iteration = max. no. iteration then
	Solution is $\mathbf{P_{gbest}}$
	Else
	Next iteration

3.2. The Proposed Algorithm to Solve the Scheduling Problem

According to the following proposed model, the broker acts as a medium between the users and the data center and is responsible for receiving requests and sending them to the data center (see Fig. 1).

An International Journal
JSCDSS
E-ISSN: 2289-8603

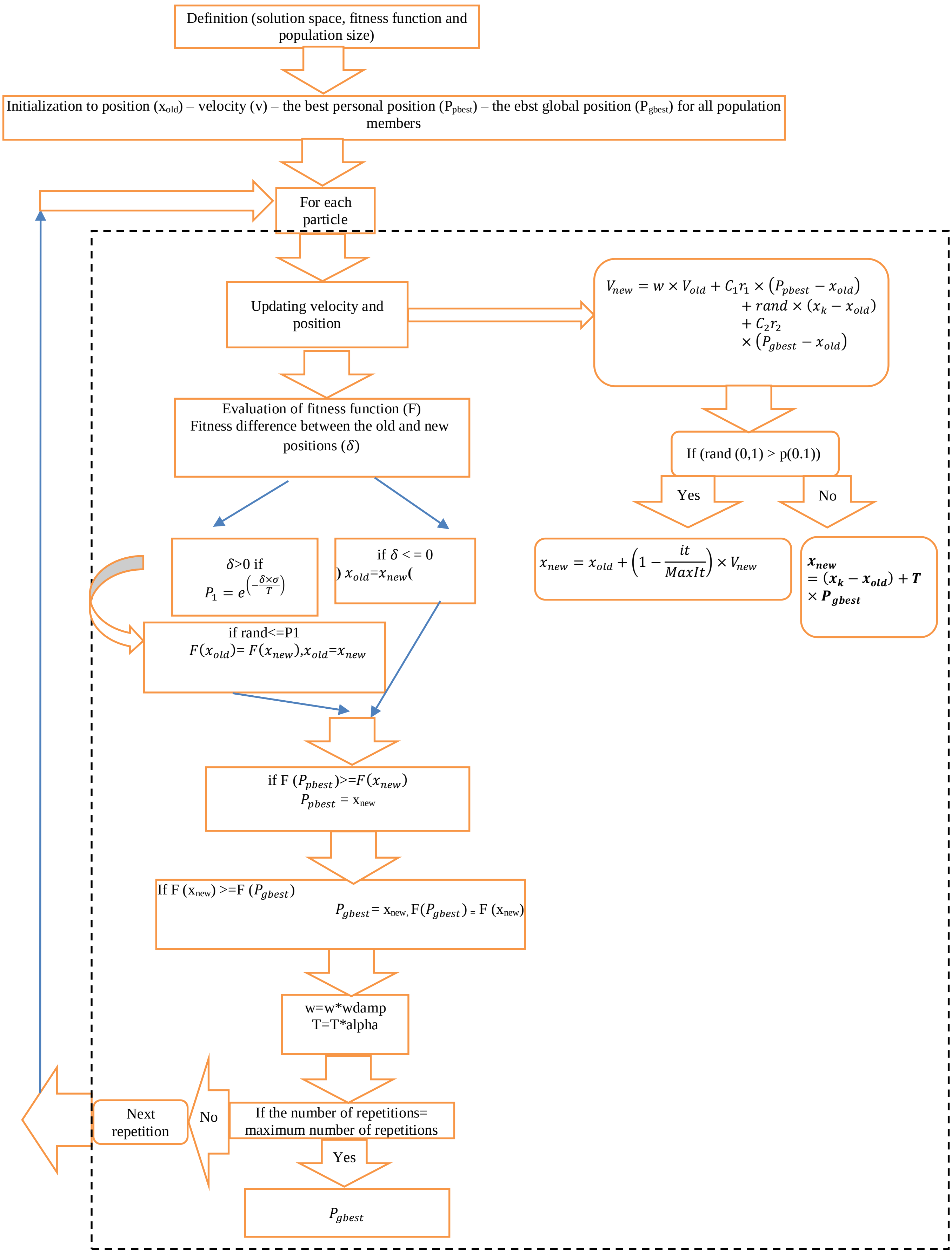


Fig. 1. The overview of the proposed algorithm

An International Journal JSCDSS
E-ISSN: 2289-8603

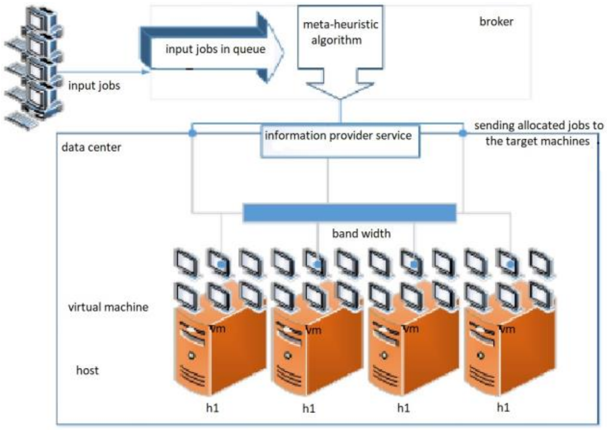


Fig. 2. System structure

Users' requests are considered as a set of tasks. After processing and analyzing the requirements of each task, the broker selects the appropriate virtual machine to perform the task. To make the decisions, the data related to the available virtual machines in the data center is used. After the entrance of users' requests to the cloud computing centers, the request for scheduling the tasks is made on the resource at the cloud computing environment, in the data center to the task scheduling policy which is in order of the particle swarm optimization algorithm with simulated annealing. The data center sends the tasks and virtual machines in the form of two separate lists along with their specifications to obtain an appropriate allocation design to the proposed method. Fig. 3 shows an example of a map and obtained a response from the algorithm to allocate the tasks to resources (virtual machines).

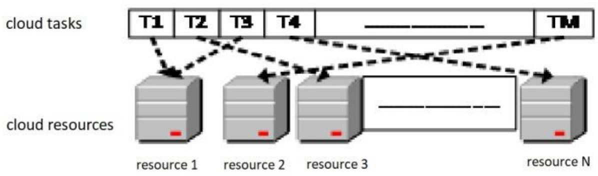


Fig. 3. Scheduling problem in cloud computing to allocate tasks to resources

4. Test and Results

To test the performance of the proposed compound algorithm, it was run on a 2.67 GHz with a 2GB RAM computer using MATLAB Version 2015. This software is widely used for method implementation (Ahani, Nilashi, Ibrahim, Sanzogni, & Weaven, 2019; Ahani, Nilashi, Yadegaridehkordi, et al., 2019; Nilashi, Ahani, et al., 2019; Nilashi, Ahmadi, Manaf, et al., 2020; Nilashi, Ahmadi, Shahmoradi, Ibrahim, & Akbari, 2019; Nilashi, Ahmadi, Sheikhtaheri, et al., 2020; Nilashi, Ibrahim, et al., 2019; Nilashi, Rupani, et al., 2019; Yadegaridehkordi et al., 2020). The set of benchmark functions CEC2017 was used (Awad, Ali, Liang, Qu, & Suganthan, 2016). The set includes 30 benchmark functions and all the functions are for minimization problems. Since the cloudsim is a package containing a series of predetermined classes and examples, there is a need for special software or compilers such as Apache Ant, Netbeans, and Eclips to run and use the files. The simulation tool used in the present study is similar to version 3.0.3 of the cloudsim simulator. This simulator supports the behavior modeling of components such as data centers, virtual machines, and resource preparation policies. Also, it can model virtual environments, resource allocation, and management according to the demands, simulation of resource allocation methods, scheduling policies that are adaptable with the objectives and properties of the research system. The parameters and fixed values used in the proposed algorithm were used in tests according to Table 5.

Table 5
The specifications of the proposed algorithm parameters

Parameter	Value
Number of loop repetition (termination condition)	100
Inertia coefficient (W)	1
Personal learning coefficient (c_1)	2
Collective learning coefficient (c_2)	2
Probability of updating particle position (P)	0.1
Maximum temperature (Temp)	$T_0 = 10$
Cooling rate	$\alpha = 0.99$
Inertia coefficient cooling rate	wdamp = 0.99

The proposed strategy in the present study was compared in terms of task execution time, response time and in three methods of 1) particle swarm optimization algorithm, 2) bat algorithm, and 3) raven roosting optimism algorithm and the results were examined in two tests.

4.1. Description of Test One

The purpose of designing this test was to investigate the performance of the proposed method compared to the introduced methods in terms of the change in the number of tasks. The number of virtual machines and hosts in this test was fixed and equal to 100. However, the number of tasks changed from 400 to 1000. In the following, the obtained results from this test are investigated in terms of each criterion.

4.2. Description of Test Two

The purpose of designing this test was to investigate the performance of the proposed method with an increase in the number of virtual machines compared to the introduced methods. The number of tasks in this test was assumed fixed and equal to 1500. However, the number of virtual machines and hosts increased from 100 to 400. In the following, the obtained results from this test are investigated in terms of each criterion.

An International Journal JSCDSS
E-ISSN: 2289-8603

Table 6
The values of task execution time with an increase in the number of tasks

Tasks	PSO	PSOAS	BAT	RRO
400	460.0919	439.8541	457.3291	459.092
500	617.0191	542.1624	622.4523	630.0075
600	738.5204	643.3436	734.2813	761.719
700	758.6603	665.3458	768.7497	765.0219
800	877.3357	754.0178	866.4432	875.2581
900	984.9341	847.7675	993.4194	1009.112
1000	1092.935	955.7432	1106.75	1091.151

Table 7
The values of response time to tasks with an increase in the number of tasks

Tasks	PSO	PSOAS	BAT	RRO
400	2.81E+03	2.64E+03	2.72E+03	1.52E+04
500	3.66E+03	3.46E+03	3.62E+03	2.37E+04
600	4.27E+03	5.07E+03	4.07E+03	3.37E+04
700	4.49E+03	5.29E+03	4.59E+03	3.50E+04
800	5.09E+03	5.17E+03	4.99E+03	5.02E+04
900	5.27E+03	6.78E+03	5.37E+03	6.03E+04
1000	5.49E+03	5.59E+03	5.79E+03	7.11E+04

Table 8
The values of efficiency with an increase in the number of tasks

Tasks	PSO	PSOAS	BAT	RRO
400	29.35295	46.9668	30.99375	29.52966
500	31.98066	49.78859	34.42883	34.32139
600	36.74956	53.68364	41.98358	42.02432
700	40.85618	60.19983	45.48298	44.78204
800	48.48795	69.98997	54.25057	53.99383
900	56.67417	73.66199	65.0274	64.40663
1000	67.33056	78.60474	78.88904	73.57745

Table 9
The values of task execution time with an increase in the number of virtual machines

VMs	PSO	PSOAS	BAT	RRO
100	1426.98	1197.99	1438.368	1449.944
150	1425.424	1195.057	1433.781	1445.958
200	1419.7	1193.001	1433.182	1442.707
250	1418.238	1191.35	1427.848	1438.501
300	1417.661	1189.6	1425.792	1439.226
350	1414.9	1185.8	1424.939	1438.557
400	1414.175	1185.948	1421.567	1414.216

An International Journal JSCDSS
E-ISSN: 2289-8603

Table 10
The values of response time to tasks with an increase in the number of virtual machines

VMs	PSO	PSOAS	BAT	RRO
100	8.37E+03	5.17E+03	6.77E+03	6.87E+03
150	7.82E+03	4.01E+03	4.31E+03	4.50E+03
200	5.86E+03	4.18E+03	4.40E+03	4.53E+03
250	3.81E+03	3.55E+03	3.74E+03	4.74E+03
300	3.40E+03	3.10E+03	3.52E+03	3.80E+03
350	4.28E+03	2.89E+03	3.19E+03	3.29E+03
400	3.10E+03	2.68E+03	3.08E+03	3.06E+03

Table 11
The values efficiency with an increase in the number of virtual machines

VMs	PSO	PSOAS	BAT	RRO
100	4.457192	16.92934	41.90879	27.89442
150	4.335444	34.14416	43.4211	28.04442
200	7.339073	59.08638	42.61331	29.01293
250	9.590787	58.17946	42.44051	29.11634
300	11.75537	58.08877	43.09255	29.96059
350	14.89044	66.92782	42.86076	29.52621
400	15.04901	73.1218	43.15534	30.13512

5. Conclusion

Cloud computing is one of the key issues in operational research. The present study attempted to present an effective solution to the job scheduling problem in cloud computing using the combination of PSO and simulated annealing algorithm. The set of benchmark functions CEC2017 was used to evaluate the proposed algorithm which contained 30 functions. The results of the evaluation were compared with the Artificial Bee Colony (ABC) Algorithm, PSO Algorithm, Bat, and Whale Optimization Algorithm (WOA). Friedman Test was also used for rating the algorithms. Friedman test results showed that the performance of the proposed algorithm has been more successful than other algorithms in solving benchmark functions CEC2017. In the next step, the scheduling problem was referred to as the main research problem. Task scheduling methods and techniques were investigated in cloud computing and strategies were proposed for resource allocation at cloud infrastructure using the approach of reducing the task execution time, reducing response time and increasing efficiency. The task scheduling system was introduced in the cloud environment in Cloudsim Simulator Tool and it was evaluated. Research results were examined and the performance of the proposed algorithm was compared with the results of Particle Swarm Optimization (PSO) algorithm, Bat and Raven Roosting Optimization (RRO) in terms of task execution time, response time and efficiency. Results of the comparison at totally the same conditions indicated the better performance of the proposed algorithm. The correlation between the tasks was not taken into consideration in the present study and it can be investigated in further studies. Besides, criteria such as task execution time, response time, and efficiency were considered to test the proposed strategy. Apart from these parameters, there are other parameters of importance at cloud infrastructure including operational capacity, accessibility, reliability, error tolerance, etc. Investigation of these parameters for a more accurate evaluation of the proposed strategy can be referred to in further studies. Moreover, the use of incremental learning (Akbari et al., 2016; Nilashi, Bin Ibrahim, Mardani, Ahani, & Jusoh, 2018; Nilashi, Jannach, bin Ibrahim, & Ithnin, 2015; Nilashi, Samad, et al., 2019) and ensemnble learning (Nilashi, Bagherifard, Rahmani, & Rafe, 2017; Nilashi, Cavallaro, et al., 2018; Nilashi, Samad, et al., 2019) approaches may improve the effectiveness of the proposed method.

Reference

Ahani, A., Nilashi, M., Ibrahim, O., Sanzogni, L., & Weaven, S. (2019). Market segmentation and travel choice prediction in Spa hotels through TripAdvisor's online reviews. International Journal of Hospitality Management, 80, 52-77.

Ahani, A., Nilashi, M., Yadegaridehkordi, E., Sanzogni, L., Tarik, A. R., Knox, K., . . . Ibrahim, O. (2019). Revealing customers' satisfaction and preferences through online review analysis: The case of Canary Islands hotels. Journal of Retailing and Consumer Services, 51, 331-343.

Akbari, E., Buntat, Z., Nilashi, M., Afroozeh, A., Farhang, Y., & Zeinalinezhad, A. (2016). ISVR modeling of an interferon gamma (IFN-γ) biosensor based on graphene. Analytical methods, 8(39), 7217-7224.

Asadi, S., Nilashi, M., Husin, A. R. C., & Yadegaridehkordi, E. (2017). Customers perspectives on adoption of cloud

An International Journal
JSCDSS
E-ISSN: 2289-8603

computing in banking sector. Information Technology and Management, 18(4), 305-330.
Awad, N., Ali, M., Liang, J., Qu, B., & Suganthan, P. (2016). Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter. Nanyang Technological University, Singapore, Tech. Rep.
Beloglazov, A., Buyya, R., Lee, Y. C., & Zomaya, A. (2011). A taxonomy and survey of energy-efficient data centers and cloud computing systems Advances in computers (Vol. 82, pp. 47-111): Elsevier.
Buttazzo, G. C. (2011). Hard real-time computing systems: predictable scheduling algorithms and applications (Vol. 24): Springer Science & Business Media.
Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and experience, 41(1), 23-50.
Chapin, S. J., Katramatos, D., Karpovich, J., & Grimshaw, A. S. (1999). The legion resource management system. Paper presented at the Workshop on Job Scheduling Strategies for Parallel Processing.
Choudhary, M., & Peddoju, S. K. (2012). A dynamic optimization algorithm for task scheduling in cloud environment. International Journal of Engineering Research and Applications (IJERA), 2(3), 2564-2568.
Kalra, M., & Singh, S. (2015). A review of metaheuristic scheduling techniques in cloud computing. Egyptian informatics journal, 16(3), 275-295.
Karlin, S. (1955). The structure of dynamic programing models. Naval Research Logistics Quarterly, 2(4), 285-294.
Masdari, M., Salehi, F., Jalali, M., & Bidaki, M. (2017). A survey of PSO-based scheduling algorithms in cloud computing. Journal of Network and Systems Management, 25(1), 122-158.
Mohammed, F., Ibrahim, O., Nilashi, M., & Alzurqa, E. (2017). Cloud computing adoption model for e-government implementation. Information Development, 33(3), 303-323.
Nilashi, M., Ahani, A., Esfahani, M. D., Yadegaridehkordi, E., Samad, S., Ibrahim, O., . . . Akbari, E. (2019). Preference learning for eco-friendly hotels recommendation: A multi-criteria collaborative filtering approach. Journal of Cleaner Production, 215, 767-783.
Nilashi, M., Ahmadi, H., Manaf, A. A., Rashid, T. A., Samad, S., Shahmoradi, L., . . . Akbari, E. (2020). Coronary Heart Disease Diagnosis Through Self-Organizing Map and Fuzzy Support Vector Machine with Incremental Updates. International Journal of Fuzzy Systems, 1-13.
Nilashi, M., Ahmadi, H., Shahmoradi, L., Ibrahim, O., & Akbari, E. (2019). A predictive method for hepatitis disease diagnosis using ensembles of neuro-fuzzy technique. Journal of Infection and Public Health, 12(1), 13-20.
Nilashi, M., Ahmadi, H., Sheikhtaheri, A., Naemi, R., Alotaibi, R., Alarood, A. A., . . . Zhao, J. (2020). Remote Tracking of Parkinson's Disease Progression Using Ensembles of Deep Belief Network and Self-Organizing Map. Expert Systems with Applications, 113562.
Nilashi, M., Bagherifard, K., Rahmani, M., & Rafe, V. (2017). A recommender system for tourism industry using cluster ensemble and prediction machine learning techniques. Computers & industrial engineering, 109, 357-368.
Nilashi, M., Bin Ibrahim, O., Mardani, A., Ahani, A., & Jusoh, A. (2018). A soft computing approach for diabetes disease classification. Health Informatics Journal, 24(4), 379-393.
Nilashi, M., Cavallaro, F., Mardani, A., Zavadskas, E. K., Samad, S., & Ibrahim, O. (2018). Measuring country sustainability performance using ensembles of neuro-fuzzy technique. Sustainability, 10(8), 2707.
Nilashi, M., Ibrahim, O., Samad, S., Ahmadi, H., Shahmoradi, L., & Akbari, E. (2019). An analytical method for measuring the Parkinson’s disease progression: A case on a Parkinson’s telemonitoring dataset. Measurement, 136, 545-557.
Nilashi, M., Jannach, D., bin Ibrahim, O., & Ithnin, N. (2015). Clustering-and regression-based multi-criteria collaborative filtering with incremental updates. Information Sciences, 293, 235-250.
Nilashi, M., Rupani, P. F., Rupani, M. M., Kamyab, H., Shao, W., Ahmadi, H., . . . Aljojo, N. (2019). Measuring sustainability through ecological sustainability and human sustainability: A machine learning approach. Journal of Cleaner Production, 240, 118162.
Nilashi, M., Samad, S., Yadegaridehkordi, E., Alizadeh, A., Akbari, E., & Ibrahim, O. (2019). Early Detection of Diabetic Retinopathy Using Ensemble Learning Approach. Journal of Soft Computing and Decision Support Systems, 6(2), 12-17.
Rimal, B. P., Choi, E., & Lumb, I. (2009). A taxonomy and survey of cloud computing systems. Paper presented at the 2009 Fifth International Joint Conference on INC, IMS and IDC.
Wang, L., Von Laszewski, G., Younge, A., He, X., Kunze, M., Tao, J., & Fu, C. (2010). Cloud computing: a perspective study. New Generation Computing, 28(2), 137-146.
Yadegaridehkordi, E., Nilashi, M., Shuib, L., Nasir, M. H. N. B. M., Asadi, S., Samad, S., & Awang, N. F. (2020). The impact of big data on firm performance in hotel industry. Electronic Commerce Research and Applications, 40, 100921.
Yadegaridehkordi, E., Shuib, L., Nilashi, M., & Asadi, S. (2019). Decision to adopt online collaborative learning tools in higher education: A case of top Malaysian universities. Education and Information Technologies, 24(1), 79-102.
Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications, 1(1), 7-18.

An International Journal
JSCDSS
E-ISSN: 2289-8603