International Journal of Grid and Distributed Computing
Vol. 9, No. 7 (2016), pp.91-100
http://dx.doi.org/10.14257/ijgdc.2016.9.7.10

An Enhanced Task Scheduling Algorithm on Cloud Computing Environment

Hussin M. Alkhashai[1], Fatma A. Omara[2]

[1](Department of Computer Science, Faculty, of Computers & Information, Cairo University, Cairo, Egypt
h.elkhashey@grad.fci-cu.edu.eg)
[2](Professor, Department of Computer Science, Faculty of Computers & Information, Cairo University, Cairo, Egypt
f.omara@fci-cu.edu.eg.com

Abstract

Cloud computing is the technology that moves the information technology (IT) services out of the office. Unfortunately, Cloud computing has faced some challenges. The task scheduling problem is considered one of the main challenges because a good mapping between the available resources and the users' tasks is needed to reduce the execution time of the users' tasks (i.e., reduce make-span), and increase resource utilization. The objective of this paper is to introduce and implement an enhanced task scheduling algorithm to assign the users' tasks to multiple computing resources. The aim of the proposed algorithm is to reduce the execution time, and cost, as well as, increase resource utilization. The proposed algorithm is considered an amalgamation of the Particle Swarm Optimization (PSO),the Best–Fit (BF), and Tabu-Search (TS) algorithms; called BFPSOTS. According to the proposed BFPSOTS algorithm, the BF algorithm has been used to generate the initial population of the standard PSO algorithm instead of to be random. The Tabu-Search (TS) algorithm has been used to improve the local research by avoiding the trap of the local optimality which could be occurred using the standard PSO algorithm. The proposed hybrid algorithm (i.e., BFPSOTS) has been implemented using Cloudsim. A comparative study has been done to evaluate the performance of the proposed algorithm relative to the standard PSO algorithm using five problems with different number of independent task, and Virtual Machines (VMs). The performance parameters which have been considered are the execution time (Makspan), cost, and resources utilization. The implementation results prove that the proposed hybrid algorithm (i.e., BFPSOTS) outperforms the standard PSO algorithm..

Keyword*: Cloud computing; Cloudsim; task scheduling; Particle Swarm Optimization; Tabu search*

1. Introduction

The Cloud computing is a way of providing the IT resources in demand and pay as you go from anywhere and anytime. In the Cloud computing, a typical datacenter consists of some computing machines connected by high-speed links. This environment is well suited to the computation of large and diverse groups of tasks. Tasks belonging to different users are no longer distinguished one from another. The scheduling challenge in such a context is to allocate more tasks to be executed on the available computing machines.

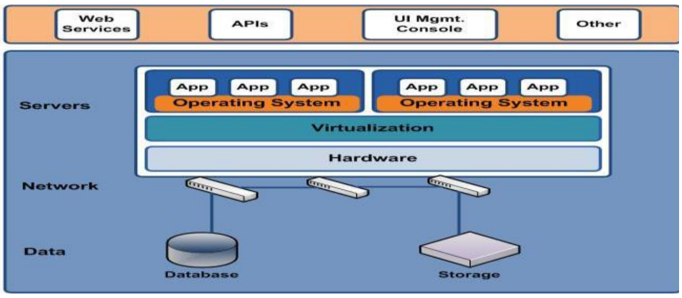
Three particular characteristics would describe the Cloud computing; 1) Unlimited computing resources; like computing power, data storage space, and applications are available on demand as needed to enable high degree of agility and scalability which meet business needs. 2) No long-term commitments; computing resources are immediately

ISSN: 2005-4262 IJGDC

available and they may be used as long as needed and then retired because they are acquired on a month-to-month or even a minute-to-minute basis. 3)Pay-as-you-go cost structure; because there are no long-term commitments where the cost of the Cloud computing resources is depended on the amount of usage[1].

On the other hands, Cloud computing offers three types of IT services; Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). SaaS provides the software service to the end users. PaaS provides the platform including operating system, programming language, execution environment , database, and web server to the developer such that they can develop their applications and deploy these applications quick and efficient .IaaS provides the infrastructure to the users for various purposes like storage system and computation resources(See Fig. 1)[2][3].

In this model, the resources are provisioned in the form of Virtual Machines (VMs) consisting of data, physical resources for fulfilling the requests. The task allocating problem concerns on allocating the users' tasks on the VMs. The optimal task allocating is considered NP-complete complexity. Therefore, heuristics are often applied as a sub-optimal algorithm to obtain relatively good solutions[4][5].



.

Figure 1. The Cloud Infrastructure[4]

Kennedy and Eberhart[6]have presented an adaptive global search based optimization algorithm, called Particle Swarm Optimization (PSO), in terms of cost, time and resources. PSO algorithm has become popular because of its simplicity and its effectiveness in a wide range of applications. Some applications that have used PSO to solve the NP-Hard problems are task scheduling and allocation problem, the data mining problem, and environmental engineering problem[7].

In this paper, a hybrid BF-PSO-TS (BFPSOTS) algorithm based on PSO algorithm has been proposed to find a suitable scheduling of the users' tasks on the Cloud computing environment. The main issues of this hybrid algorithm are reducing execution time (Makspan), and execution cost, as well as, maximizing resources utilization.

According to the hybrid (BFPSOTS) algorithm, the BF algorithm has been merged into PSO for generating the initial population instead of generating it randomly. Then, TS algorithm has been merged to PSO to improve local research by avoiding the trap of the local optimality. This leads to improve the performance.

The proposed hybrid algorithms; BFPSOTS; has been implemented using the open source Cloudsim3.0.3 simulator with a different number of independent tasks and different number of VMs.

The rest of this paper is organized as follows; in Section 2, related work is discussed. In Section 3, the problem has been explained. In Section 4, the problem statement is discussed. describe the proposed algorithm, its implementation and performance evaluation are discussed in Sections 5, 6. Finally, section 7 presents the conclusions and future research directions.

2. Related Work

The task scheduling is an important issue and considered one of the main challenges in the Cloud Computing because the service provider has to serve many users' applications at different times and from different places. Therefore, schedule these tasks to be executing in the Cloud computing environment has to be more flexible and quick way.

Many meta-heuristic task scheduling algorithms based on *PSO* and *GA* have been proposed[8][9][10]. Because the proposed task scheduling is based on *PSO* algorithm, related work concerning *PSO* will be discussed.

.***Ali Al-maamari, Fatma A. Omara***[11]have proposed a task scheduling algorithm. According to this proposed algorithm, the *Cuckoo* algorithm has been merged into the *PSO* algorithm to generate an optimal task schedule on Cloud computing environment in order to complete the tasks in a minimum execution time, as well as, tosatisfy resources utilization.

SolmazAbdi, Seyyed Ahmad Motamedi, and SaeedSharifia[5]have proposed a task schedule algorithm based on *PSO* algorithm. The main principle of this algorithm is that the existed tasks and processors are sorted in ascending order. The existed processors have been sorted based on their processing power. After sorting the tasks and processors, jobs are assigned using one to one mapping. After that, the Fastest Job Processor (*SJFP*) algorithm is applied to generate initial population for the *PSO* algorithm.

Shaobin Zhan andHongyingHuo Shenzhen[12] have proposed an enhanced task scheduling algorithm based on *PSO* algorithm by merging the simulated annealing algorithm with its fast random global searching scheduling strategy on the Cloud Computing environment. The experiment results show that this algorithm can reduce the average running time, and raise the availability rate of resources.

P. Visalakshi , S. N Sivanandam[13] have proposed a task schedule by using *PSO* with Elitism. Elitismis is the process of selecting better individuals with a bias towards the better ones. Elitism is important since it allows the solutions to get better over the time. By using Elitism, the performance of *PSO* can be rapidly increased because it prevents losing the best-found solution.

3. Scheduling Probleme

Assuming that there are ***n*** tasks whose execution time on each processing machine is known in advance and they should be processed on ***m*** computational resources. The number of tasks is considered more than the number of resources (i.e., ***n***>>***m***), and tasks cannot be assigned to different resources, i.e. tasks are not allowed to migrate between resources. The goal of the task scheduling is to minimize the tasks' execution time, and the execution cost, as well as, increase resource utilization.

To formulate the problem, consider the set of tasks is defined as $\boldsymbol{Ti = \{1, 2, \dots n\}}$where ***n*** is the number of independent tasks and$\boldsymbol{Rj = \{1, 2, \dots m\}}$is the set of computational resources. Suppose the execution of task ***i*** on VM ***j*** is defined as ***CTij***, the completion time (***Makspan)*** is determined using Equation (1)[5],[14]:

$$akspan = CTmax\ (i, j), i \in T,$$

$$i = 1,2, \dots n\ and\ j \in VM, J = 1,2, \dots m \qquad (1)$$

Where, ***CTmax*** is the maximum time for completing task***i*** on a virtual machine ***j.***

The total cost of processing all tasks is represented as[15]:

$$Total\ Cost = \frac{Task\ length\ *\ Cost\ per\ seconds}{VM\ mips} + Processing\ Cost \qquad (2)$$

Resources Utilization (***U***)for processing all tasks on VMs is represented by Equation (3) [16][11].

$$U = \frac{final\,VM\ available\ time}{\#VMs * schedulingtime} * 100 \qquad (3)$$

4. Problem Statement

According to the work in this paper, the main problem of task scheduling is to define the best mapping of ***n*** independent tasks into ***m*** heterogeneous resources with the goal of minimizing the execution time (***Makspan***) and cost of resources, as well as, maximizing resources utilization. Therefore, the goal of task scheduling is to create a mapping ($\boldsymbol{f}$)to allocate tasks (***T***) into resources (***R***). i.e.;$\boldsymbol{f}: \boldsymbol{T} \rightarrow \boldsymbol{R}$**.**

On the other hand, the following assumptions have been considered:

1- *Tasks are independent*

2- *The available resources (VMs) are heterogeneous*

3- *The available resources are exclusive usage and cannot be shared among different tasks (i.e.,* the resources (i.e., VMs) cannot consider other tasks until the completion of the current one.

5. The Proposed Best-Fit –PSO-Tabu Search (BFPSOTS) Algorithm

The proposed *BFPSOTS* algorithm for the task scheduling problem is based on *PSO* algorithm and considered an amalgamation of *PSO*, Best-Fit, and Tabu Search Algorithms. Therefore, the basics of the standard *PSO* algorithm will be introduced first.

A. The Standard PSO Algorithm:

The basic idea of the *PSO* algorithm is that if one of the birds finds a good path to the prediction location, more followers would be attracted. This helps to improve the search for best path because all populations are involved. According to the *PSO*, every particle in the swarm behavior has two characters; (1) A position ***x*** which notates the suggested location. (2) A velocity ***v*** which defines the amount of moving. Each particle travels over the whole search space and remembers the best position found. The communication is made between particles such that each bird could determine the location and velocities based on the best solutions discovered by others. Each position and velocity is scored by a fitness function ***f*** to quantify the best solution. Specifically, at iteration ***k***, particles keep two values for each track; local best position ***(P_LB***) and global best position ***(P_GB***), both of which are calculated from ***f***. Therefore, the update function is defined using Equations(4) a

$$V(t+1) = wV(t) + c_1 * R1 * (pbest(t) - p(t)) + c_2 * R2 * (gbest(t) - p(t)) \qquad (4)$$

$$p(t+1) = p(t) + V(t+1) \qquad (5)$$

Wher***e V*** is the particle velocity, ***W*** is the inertia weight factor, ***p*** is the current solution, ***pbest*** is a local solution, ***gbest***is a global solution, and ***R1***,***R2*** are the random numbers uniformly distributed on the interval [0,1].The variables ***C1, C2*** are the learning factors. The pseudo code of the standard *PSO* algorithm is as follows [7].

PSO Algorithm

```
1. Set of particle dimension as equal to the size of ready tasks in {ti}∈ T
2. Initialize particles position randomly from PC= 1...j and velocity vi randomly.
3. For each particle, calculate its fitness value as in Equations (3)and (4)
4. If the fitness value is better than the prior-best, set, the current fitness value as the new p-best.
5. After Steps 3 and 4 for all particles, select the best particle as g-best.
6. For all particles, calculate the velocity using equation (3) and update their positions using equation (4).
7. If the stopping criteria or maximum iteration is not satisfied, repeat starting from step 3.
```

The flowchart of the standard *PSO* is presented in Fig. 2.The computational cost of the *PSO* algorithm involves the initialization time (T_{ini}), evaluation time (T_{eva}), and velocity and position update (T_{upd}) for each particle[17]. Therefore, the time complexity (Tc) of *PSO* the algorithm can be estimated as:

$$\mathrm{Tc} = Tini + Teva + Tupd \qquad (6)$$

$$\mathrm{Tc} = \mathrm{O}(N^2 + N + 2^N)(7)$$

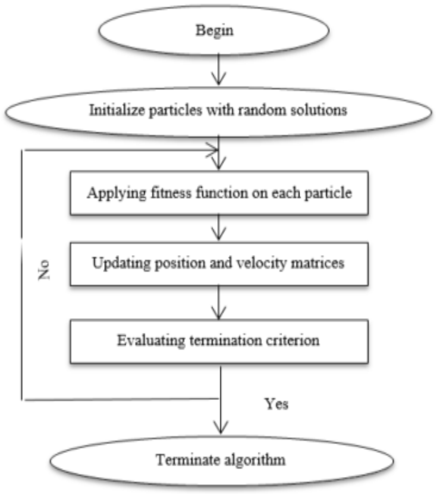

Begin
Initialize particles with random solutions
Applying fitness function on each particle
Updating position and velocity matrices
Evaluating termination criterion
No
Yes
Terminate algorithm

Figure 2. Flowchart of Standard PSO Algorithm[5]

B. The Proposed *BFPSOTS* Algorithm

In the standard *PSO* algorithm, initial particles are created randomly. Unfortunately, randomness could decrease the chance of the algorithm to converge to the best solution. In order to improve the behavior of the standard *PSO* algorithm, the Best-Fit (*BF*) algorithm has been merged into *PSO* algorithm (i.e., instead of generating initial population randomly; it is created according to the *BF* algorithm). All other steps of the standard *PSO* algorithm are not changed. On the other hand, TS algorithm is a neighborhood search algorithm which employs intelligent search and flexible memory technique. It is designed to guide other algorithms to escape from the trap of local optimally. Therefore, it has been applied to solve task scheduling and other optimization problems [15] [18].So, *TS* algorithm has been merged into *PSO* algorithm to avoid being trapped at the local optimum and speed up the search process.

The proposed *BFPSOTS* algorithm consists of three phases:

1. The initial population of the *PSO* will be generated using *BF* algorithm.

2. *PSO* algorithm starts to be executed, till the stopping condition will be satisfied.
3. The output results of the *PSO* will be sent to *TS* algorithm. The *TS* algorithm applies the neighborhood technology, adds the result to the Tabu list, compares the result and selects the best solutions. This situation is repeated until the best solution is obtained[19],[20].

According to the proposed *BFPSOTS* algorithm, the stopping criteria for the *PSO* and *TS* phases are supposed as follows:

$$Stopping\ Criteria = \begin{cases} The\ difference\ between\ the\ current\ solution\ and\ former\ one\ is\ \leq 0.0012 \\ or \\ No.of\ Iteration = 300 \end{cases} \tag{8}$$

The flowchart of the proposed *BFPSO* algorithm is shown in Fig.3. In the *BF* phase, the initial population is selected and the tasks are sorted according to its priority. After that, *PSO* algorithm starts to execute by allocating the tasks on to the suitable resources and sending the result to *TS* algorithm to apply the neighborhood technology.

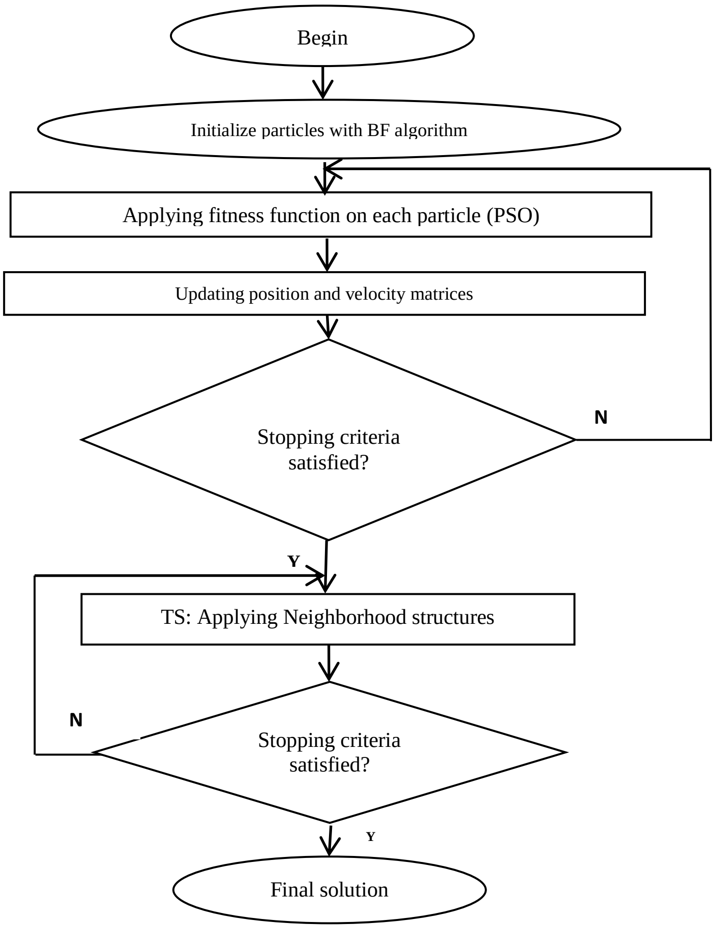

Begin
Initialize particles with BF algorithm
Applying fitness function on each particle (PSO)
Updating position and velocity matrices
Stopping criteria satisfied?
N
Y
TS: Applying Neighborhood structures
N
Stopping criteria satisfied?
Y
Final solution

Figure 3. Flowchart of BFP*SOTS Algorithm*

To explain the effect of the *TS* algorithm, suppose the distribution of tasks among VMs due to *PSO* phase is represented in Table [1].In the *TS* phase, T1 is moved from VM0 to VM4. Also, T5 is moved from VM0 to VM3. This has been happening after applying the minimum local execution time for particle (1) (as an example) according to the neighborhood technology. This will be repeated for all particles which are coming from *PSO* phase.

Table 1. Show the Distribute the Task's on VM's with Tabu Search

Instances of particle 1	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	TIME
Instance 0	-	VM0	-	VM0	-	VM0	VM0	-	-	-	**18.4**
Instance 1	-	-	VM1	-	-	-	-	-	VM1	-	13.07
Instance 2	VM2	-	-	-	-	-	-	-	-	VM2	9.13
Instance 3	-	-	-	-	-	-	-	VM3	-	-	**4.86**
Instance 4	-	-	-	-	VM4	-	-	-	-	-	**3.14**

On the other hand, the Time Complexity (TC) of the proposed *BFPSOTS* algorithm depends on the time complexity of the *BF*,*PSO* and *TS* algorithms.

The time complexity of the *BF* algorithm is[21]:

$$Tc = O(N^2) \tag{9}$$

Where ***N*** is the number of the tasks will be sorted.

The *TS* heuristic for the quadratic assignment problem consists of repeatedly swapping locations of two tasks. A single iteration of the heuristic consists of [22]:

I. Define the task will be moved to decrease the execution time.
II. Move the task
III. Recalculating the new execution time

The process is repeated for a specified number of iterations. Traditional implementations of *TS* require $O(N^2)$ operations per iteration. Therefore, the *TS*'s time complexity is:

$$Tc = O(N^2) \tag{10}$$

Where, ***N*** is the number of tasks.

According to Equations 7, 9 and 10, the time complexity of the proposed *BFPSOTS* algorithm is:

$$BFPSOTS(Tc) = O(3N^2 + \mathrm{N} + 2^N) \tag{11}$$

3. The Performance Evaluation

To evaluate the performance of the proposed algorithm, a comparative study has been conducted between the standard *PSO* and the proposed *BFPSOTS* algorithms.

A. Experimental Settings

Cloudsim3.0.3 is an open source simulator which has been developed by Grid bus project team and the grid Laboratory of the University of Melbourne in Australia. The Cloudsim can run on Linux and Windows systems [6]. Cloudsim has been used to implement and compare the proposed *BFPSOTS* task scheduling algorithm with respect to the standard *PSO* algorithm. This simulation mainly validates the advantages of the Makspan, cost and the resource utilization among these scheduling algorithms in the Cloud computing environment.

B. Performance Evaluation

To evaluate the performance of the standard *PSO, and BFPSOTS* algorithms, 15 Virtual machines are considered with 25, 50, 75,100, and 125 cloudlets (i.e., tasks).

the simulation results of the Makspan, resources utilization, and resources cost of the two algorithms, *PSO*, *BFPSOTS*, using 15 Virtual machines and 25,50,75,100,125 tasks, are depicted in Tables[2], [3], [4],and Figs(4),(5),(6) and (7) respectively.

Table 2: Completion Time (Makspan) in sec Using 15 VMs

No Of VM	*No of Task*	*BFPSOTS*	*PSO*	Improve BFPSOTS Vs PSO
15	25	3.09205235	4.0920524	0.2443762
	50	6.46920192	8.0497707	0.1963495
	75	8.10439915	10.060102	0.1944019
	100	13.1493105	15.0574	0.126721
	125	15.1493105	18.327754	0.1734224

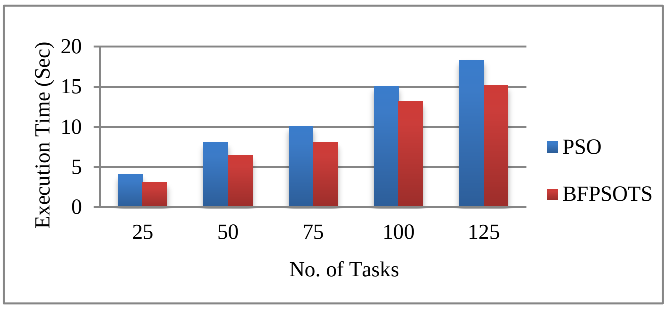


Figure 4. Comparison of Makspan Using PSO and BFPOSTS Algorithms

Table 3. Resource Utilization using 15 VMs

No Of VM	*No of Task*	*BFPSOTS*	*PSO*	Improve BFPSOTS Vs PSO
15	25	0.425427	0.419474	0.0139927
	50	0.585427	0.460671	0.2482034
	75	0.61276	0.481335	0.1778051
	100	0.721029	0.590684	0.1807768
	125	0.751963	0.6001	0.2019556

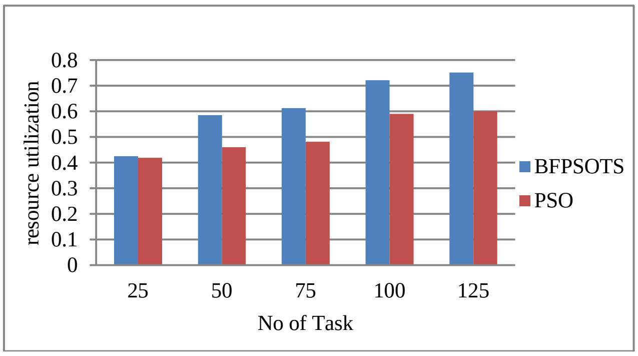


Figure 5. Comparison of Resource Utilization Using PSO and BFPOSTS Algorithms

Table 4. Cost of Executing Tasks using 15 VMs

No Of VM	*No of Task*	*BFPSOTS*	*PSO*	Improve BFPSOTS Vs PSO
15	25	1948.94813	2447.8	0.203796
	50	3118.94813	3459.6	0.0984657
	75	5030.6	6348.2	0.2075549
	100	6891.26439	7656.6	0.0999576
	125	8335.79351	9158.6	0.0898398

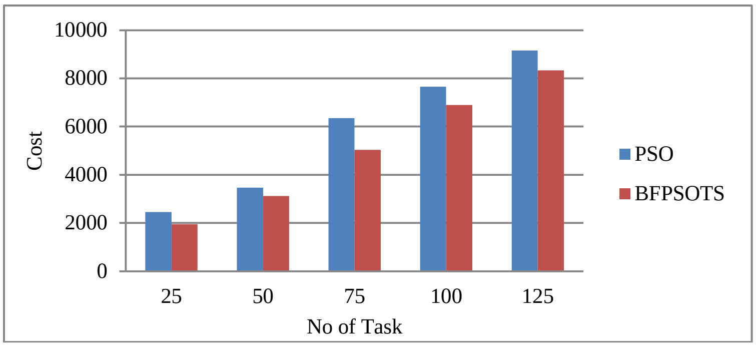


Figure 6. Comparison of Cost Using PSO and BFPOSTS Algorithms

According to the implementation results, it is found that the proposed *BFPSOTS* algorithm outperforms the standard *PSO* algorithm by 18.71%, 16.45%, and 13.99% in average with respect to Makspan, resource utilization, and execution cost respectively. In spite of the time complexity of the proposed *BFPSOTS* algorithm, it outperforms the standard *PSO* algorithm with respect to Makspan, execution cost, and resource utilization.

4. Conclusion and Future Work

In this paper, the problem of task scheduling in the Cloud computing environment is concerned. *BF*, *PSO*, and *TS* algorithms are most famous algorithms for scheduling tasks on cloud computing environment. In order to improve the performance of the standard *PSO* algorithm, the modified *PSO* algorithm is proposed .Because the first step of the *PSO* algorithm is random (i.e., initial population is generated randomly), this might produce wrong path. In order to resolve this situation, BF algorithm is merging into standard *PSO* algorithm for generating initial population in order to obtain a good initial selection. On the other hand, *TS* algorithm has been merged to *PSO* algorithm to avoid trapped and make an optimal local search to get a good solution by reducing the execution time (*Makspan*), the cost of processing, and increasing resources utilization. The implementation results prove that the proposed algorithm (i.e., *BFPSOTS*) outperforms the standard *PSO* with respect to the Makspan, resource utilization, and cost. Unfortunately, this improvement has been satisfied at the expense of fulfillment of the time complexity. In the future work, we plan to improve PSO algorithm using other greedy algorithms (e.g., Worst-Fit), and considering dependent tasks instead of independent tasks to achieve the high performance of the overall system.

References

[1] M. Hugos, C. Emeritus, S. Advisor, C. T. Officer, T. Weather, C. Interactive, C. I. Officer, C. E. Officer, and E. V. President, Business in the Cloud,book, *John Wiley Sons, Inc., Hoboken, New Jersey. USA*, (**2011**).

[2] A. J. Staring and G. Karagiannis, “Cloud computing models and their application in LTE-based cellular systems,” *2013 IEEE Int. Conf. Commun. Work. ICC,IEEE*, (**2013**), pp. 750–755.

[3] S. Abdi, S. A. Motamedi, and S. Sharifian, “Task Scheduling using Modified PSO Algorithm in Cloud Computing Environment,”, Dubai (UAE),(**2014**),pp. 37–4

[4] A. J. Staring and G. Karagiannis, “Cloud computing models and their application in LTE-based cellular systems,” Proc. of the IEEE Int. Conf. Commun. Work. ICC,IEEE,(2013), pp. 750–755.

[5] F. T. Frédéric Magoulès, and Jie Pan, Cloud Computing: Data-Intensive Computing and Scheduling,CRC Press, 2012.

[6] S. Zhan and H. Huo, “Improved PSO-based Task Scheduling Algorithm in Cloud Computing,” Journal of Information & Computational Science.",(2012), pp. 3821–3829.

[7] S. Pandey, L. Wu, S. M. S. M. S. M. Guru, and R. Buyya, “A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments,” The 4th IEEE Int. Conf. Adv. Inf. Netw. Appl.,(210), pp. 400–407

[8] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, “The eucalyptus open-source cloud-computing system,” *2009 9th IEEE/ACM Int. Symp. Cluster. Comput. Grid, CCGRID 2009*, (**2009**),pp. 124–131.

[9] X. Wang, C. S. Yeo, R. Buyya, and J. Su, “Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm,” abook chapter:Future Generation Computer Systems,ELSEVIER,, vol. 27, no. 8, (**2011**),pp. 1124–1134.

[10] J. Singh, “Efficient Tasks scheduling for heterogeneous multiprocessor using aGenetic algorithm with Node duplication,” *J. Comput. Sci.*, vol. 2, no. 3, (**2011**),pp. 402–410.

[11] A. Al-maamari and F. A. Omara, “Task Scheduling using Hybrid Algorithm in Cloud Computing Environments,”Journal of Computer Engineering (IOSR-JCE), vol. 17, no. 3, (**2015**),pp. 96–106.

[12] H. Zhan, Shaobin Huo, “Improved PSO-based Task Scheduling Algorithm in Cloud Computing,” *J. Inf. Comput. Sci.*, vol. 13,(**2012**), pp. 3821–3829.

[13] P. Visalakshi and S. N. Sivanandam, “Dynamic Task Scheduling with Load Balancing using Hybrid Particle Swarm Optimization,” *Int. J. Open Probl. Compt Math*, vol. 2, no. 3,(**2009**), pp. 475–488.

[14] S. Saha, “A Survey on Resource Management in Cloud Computing,” The International Journal of Computer Science and Information Technologies(IJCSIT),vol. 5, no. 3, (**2014**),PP. 3887–3889,.

[15] R. Sahal and F. A. Omara, “Effective Virtual Machine Configuration for Cloud Environment,” The9th International Conference on the Informatics and Systems (INFOS), Cairo, Egypt,(**2014**), PP. 15–20,.

[16] G. Zhao, “Cost-Aware Scheduling Algorithm Based on PSO in Cloud Computing Environment,” The International Journal of Grid and Distributed Computing,vol. 7, no. 1,(**2014**),PP. 33–42.

[17] Y. J. Gong, J. J. Li, Y. Zhou, Y. Li, H. S. H. Chung, Y. H. Shi, and J. Zhang, “Genetic Learning Particle Swarm Optimization,” *IEEE Trans. Cybern.*, Vol. PP, Issue 99 ,(**2015**),PP. 1–14.

[18] P. G. Dive, Prof Kranti Kulkarni, “Implementation of Modified Genetic Algorithm Based on the Sub GraphFormation of Travelling Salesman Problem,”International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 11, , (2013,)pp. 541–546.

[19] B. Yi, Pan Ding, Hui Ramamurthy, “A Tabu search based heuristic for optimized joint resource allocation and task scheduling in Grid/Clouds,” The IEEE International Conference on Advanced Networks and Telecommunications Systems , Kattankulathur, (**2013**),PP. 4–6.

[20] S. S. Manimegalai, “Task Scheduling Using Two-Phase Variable Neighborhood Search Algorithm on Heterogeneous Computing and Grid Environments,” King Fahd University of Petroleum andMinerals(*Arab. J. Sci. Eng.)*, vol. 40, (**2015**),pp. 817–844.

[21] S. Jadoon, S. Solehria, and M. Qayum, “Optimized Selection Sort Algorithm is faster than Insertion Sort Algorithm: a Comparative Study,” The International Journal of Electrical & Computer Sciences (IJECS-IJENS), vol. 11 no: 02. (**2011**),PP. 18-23, April.

[22] G. Paul, “An efficient implementation of the robust tabu search heuristic for sparse quadratic assignment problems,” *European Journal of Operational Research*, *ELSEVIER,*vol. 209, no. 3, (**2011**),pp. 215–218.