



A Multi-objective Optimal Task Scheduling in Cloud Environment Using Cuckoo Particle Swarm Optimization

T. Prem Jacob¹ · K. Pradeep²

Published online: 13 May 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

In cloud computing, varied demands are placed on the constantly changing resources. The task scheduling place very vital role in cloud computing environments, this scheduling process needs to schedule the tasks to virtual machine while reducing the makespan and cost. The task scheduling problem comes under NP hard category. Efficient scheduling method makes cloud computing services better and faster. In general, optimization algorithms are used to solve the scheduling issues in cloud. So, in this paper we combined two optimization algorithms namely called as Cuckoo Search (CS) and Particle Swarm Optimization (PSO). The new proposed hybrid algorithm is called as, CS and particle swarm optimization (CPSO). Our main purpose of the proposed paper is to reduce the makespan, cost and deadline violation rate. The performance of the proposed CPSO algorithm is evaluated using cloudsim toolkit. From the simulation results our proposed works minimize the makespan, cost, deadline violation rate, when compared to PBACO, ACO, MIN–MIN, and FCFS.

Keywords Cloud computing · Task scheduling · CPSO · Performance and multi-objective

1 Introduction

Cloud computing [1] has been developed in the field of processing and storing by providing endless way of accessing Information Technology in wide range of domains such as mobile system, network system, environmental computing, medicines, business. Thus leading to the decision of cloud computing at recent times because of its efficiency in computing provides IT services. It also includes the infrastructure based on pay-per-use model to eliminate the need of investing for the purpose of managing the IT infrastructure.

✉ T. Prem Jacob
premjac@yahoo.com

K. Pradeep
pradeepkrishnadoss@gmail.com

¹ Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India

² Department of Information Technology, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India

The services provided by the Cloud are categorized as follows: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). The users can access virtual resources by making use of these services. This, therefore, makes the resources flexible. The users have to pay for the services and thus the name “pay-per-use” model became another term for the Cloud. While using cloud services, optimal scheduling becomes a necessity as the users are directly dependent on it for physical resource usage.

In cloud computing, the widely spread problem is task scheduling that remains of the NP-hard problem. In IaaS cloud, a collection of servers are presented to provide the users with required resources. The IaaS cloud not only provides hardware to the users but also software which ensures properties of elasticity and efficient management of resources of the cloud. The subsystem of the resource management in IaaS is used to execute the scheduled tasks, used to map the tasks that are to be carried in a more efficient manner of VM characterizing dynamic and heterogeneous property. Good solution is achieved via heuristic methods to find the optimal solution [2, 3] of the NP-problem. The main aim is to schedule tasks to reduce the cost as well as the execution time. The VM has the capacity of heterogeneous processing that leads to load balancing among Virtual Machines. It is used to provide co-ordination, schedule the tasks and achieve lower make span. Thus, the execution time of VM and balancing load is performed by task scheduling algorithms.

In this paper, we proposed multi-objective-based task scheduling using a combination of cuckoo and particle swarm optimization algorithm. The multi-objective function used in this paper is makespan, cost and deadline violation rate. Based on the multi-objective function we obtain the near optimal scheduled task. The major contribution are made in the research for task scheduling process as follow,

- An approach namely CPSO is done for task scheduling, which has the advantages of quickly converging and easily realizing so that this scheduling approach is able to get a near optimal solution.
- Multi objective based task scheduling problems in cloud computing are the focus by considering minimization of makespan, cost and deadline violation rate in the heterogeneous cloud environment.
- The simulation is done by using cloudsim3.0 toolkit.

The rest of the paper is organized as follows; a detailed review of the related work based on task scheduling in cloud environments is described in Sect. 2. In Sect. 3 the problem definition and system model is presented in detail. The resource cost and task scheduling model is given in Sect. 4. In Sect. 5, The proposed technique for task scheduling using cuckoo particle swarm optimization is described. The experimental results and the performance evaluation discussion are provided in Sect. 6. The conclusion is summed up in Sect. 7.

2 Related Work

In [4], the scheduling and maintenance of the cloud resources present in the cloud environment are done by the deadline-based job scheduling and Particle Swarm Optimisation (PSO)-based resource allocation mechanism. Using these mechanisms, the Cloud Resource Broker is implemented. The number of jobs can be increased and the time, cost can be decreased by this technique. In [5], the multi-objective scheduling technique which

was proposed is found to be reducing the cost, makespan, and all the other objectives to a considerable rate. The Ant Colony Optimization (ACO) method is used for evaluation and for reviewing the performance and cost of budget. In [6], the multi objective function for scheduling is achieved by making use of a hybrid cuckoo search and gravitational search algorithm (CGSA). Their work focused on allocating entire task to an equivalent virtual machine while maximizing the profit the entire task was executed with low cost, less resource use, and less energy consumption. In [7], the work focused on six rules based heuristic algorithms namely, First Come First Serve (FCFS), Minimum Completion Time (MCT), Minimum Execution Time (MET), Max–min, Min–min and Suffrage to execute and schedule autonomous jobs in homogeneous and heterogeneous environment. Their aim was to compare processing in terms of cost, degree of imbalance, makespan and throughput.

In [8], the authors proposed scheduling algorithm named Global League Championship Algorithm (GBLCA) for global job scheduling of scientific applications in a cloud environment. This algorithm showed a remarkable process development rate on the make span that ranges between 14.44 and 46.41%. In addition, there was a reduction in the time taken to securely schedule the applications. In [1], The Simulated Annealing (SA)-based Symbiotic Organisms Search (SASOS) technique was presented so that the convergence rate and solution quality is increased. The objective is to optimize the scheduling of the jobs in the cloud. This technique is similar to PSO algorithm when it comes to efficiency of response time. Moreover, the utilization level of the VMs was taken into consideration for a fitness function in order to bring down the degree of imbalance. The work in [9] addressed the issue of job scheduling and resource provisioning for a set of jobs that executed on IaaS clouds. The proposed algorithm named as VM Capacity-Aware Scheduling implemented the jobs within given a budget, at the same time reducing the slow down due to the budget constraints.

In [10], a Task Scheduling with error Tolerance in Grid Computing using Ant Colony Optimization was proposed to ensure that the tasks were implemented even when amenity error had occurred. This algorithm showed an improvement in the user's QoS over the existing ACO algorithm by making the use of the performance metrics: throughput and average turnaround time. In [11] paper focuses on solving VM placement problem with respect to the available frequency which is formulated as variable sized bin packing problem. Moreover, a new frequency allocation policy is developed and hybridized with a developed various of whale optimization algorithm (WOA) called improved Levy based whale optimization algorithm. Cloudsim toolkit is used in order to test the validity of the proposed algorithm on 25 various data sets that generated in various and compared with different optimization algorithms including: WOA, first fit, best fit, particle swarm optimization, genetic algorithm, and intelligent tuned harmony search.

In [12], the capabilities of the Taguchi method and the Differential Evolution Algorithm (DEA) were combined into a newly proposed algorithm called the Improved Differential Evolution Algorithm (IDEA). IDEA was presented so as to optimize the scheduling of tasks and allocation of resources in the cloud environment. Cost and time models are depicted in order to evaluated the costs and time during task scheduling. The performance and receiving cost is included in the cost model and the processing time, receiving time and waiting time is included in the time model. In [13] proposed a demanding task to find appropriate commutation among resource utilization, energy consumption, and Quality of service (QoS). Taking into account of both processing time and transmission time a Particle Swarm Optimization (PSO) based Adaptive Multi-Objective Task Scheduling -AMTS Strategy is proposed in this paper. It is carried out by formulation of task scheduling

followed by advancement of scheduling policy to produce ideal resource utilization, task completion time, average cost, and average energy consumption. To maintain particle variant adaptive acceleration coefficient is adopted.

In [14], Infrastructure as a Service (IaaS) is a service model that delivers computer infrastructure on a outsourced basis to support enterprise operations. It offers many benefits to create effective cost and easily scalable IT solutions. While maintaining the Quality of service (QoS) challenge arises on scheduling the task on peak demand. Proactive machine purchasing or cloud federation proposed previously is not economic and hardly feasible in practise and also requires an inter cloud-agreement. So a resource allocation framework is proposed where IaaS provider can outsource the tasks to External Clouds (EC's) when required to meet the challenge. The core issue is maximizing the profit of IaaS and guarantee QoS by allocating resource accordingly. The problem is serviced as an Integer Programming (IP) Model, and worked out by a Self-Adaptive Learning Particle Swarm Optimization (SLPSO) based scheduling approach.

In [15], Task scheduling plays a key role in cloud computing. The main reason behind scheduling tasks to the resources in accordance with the given time bound, which involves finding out a complete and best sequence in which various tasks can be executed by reducing the makespan and cost to give the best and satisfactory result to the user. This paper put forwards the task scheduling algorithm called W-Scheduler based on multi-criteria model and the whale optimization algorithm (WOA). It carries out its process by calculating the fitness value from the cost function of the central processing unit (CPU) and the memory. Then the fitness value is calculated by summing up the makespan and the budget cost function. Thus, the whale optimization algorithm can optimally schedule the tasks to the virtual machines by preserving the minimum makespan and cost.

In [16] scheduling algorithm is necessary to optimize the VMs resources. Trade-off is not usual in most algorithms. Modified Fractional Grey Wolf Optimizer for Multi-Objective Task Scheduling (MFGMTS) was proposed for use in cloud computing environment. Objectives such as execution time, execution cost, communication time, communication cost, energy consumption, and resource utilization are calculated by the usage of the epsilon-constraint and penalty cost function. The fitness function is reduced considerably. It is an inspiration from the Fractional Grey-Wolf Optimization (FGWO). In [17], an algorithm based upon Ant-Colony optimisation is proposed called the MOSACO. The measurable network of the public and private resources of the VMs in hybrid clouds are used to its best potential according to deadline and cost constraints. Features like completion time, time-first cost, cost-first single objective optimization, user QoS, etc. are increased to a great extent by the use of an entropy model. High level of optimization is provided by MASACO. In [18, 19] point out to multi-objective based scheduling like time, memory and cost.

The above literature reviews did not provide near optimum result when performance, cost and deadline violation rate are considered together. This proposed method achieves a minimal makespan and cost among VMs as performance metrics to optimize task and resource, using a Cuckoo Search and Particle swarm Optimization (CPSO) algorithm based on the proposed models in cloud. The proposed scheduling hybridizes two algorithms namely cuckoo search algorithm (CSA) and Particle swarm Optimization (PSO). This new hybrid algorithm optimizes the task and resources more efficiently.

3 The Problem Definition and Explanation

This paper describes the simple definition of the system framework, tasks and the resources. The meaning and the primary parameters are listed below in Table 1.

Firstly, in cloud computing of the current system assume there are N resources $R = \{R_1, R_2, \dots, R_j, \dots, R_n\}$ and K tasks $T = \{T_1, T_2, \dots, T_i, \dots, T_k\}$ and also the virtual resources are referred as cloud resources.

Definition 1 (Resources) Each resources on cloud defined by utilization of both CPU and memory, that is $R_j = (C_j; M_j)$.

Definition 2 (Tasks) $T_i = (C_i; M_i; D_i; B_i)$. The C_i and M_i is the CPU and memory utilization of user submitted task. While D_i represents the task deadline, and B_i represents the users budget cost. The task manager performs the tasks and the user submits the tasks.

Assumption 1 To proceed with the research, it is important to provide related assumptions for the definitions above mentioned. It is assumed that all the information submitted by the user is very accurate and also the information related to the resource demand is trusted.

In cloud computing, the resource usage is monitored via virtualization technology. During the process, the number of users exceeding while applying for the process implementation is performed by user-accessed resources such as the CPU and memory. During the implementation, the tasks may fail in case if the system cuts-off the task performance. Therefore, Assumption 1 is reasonable.

There are numbers of users where the tasks are submitted to the task manager to accept and manage the task information. This task manager submits tasks via budget, cost, memory, and deadline. Then, the tasks are scheduled and mapped by the scheduler. It is mapped through the resources of tasks. These are mapped via the global resource manager. Global

Table 1 Notations used in proposed CPSO task scheduling

Symbol	Definitions
T_i	The task $i, 1 < j < K$
U_j	The resource $j, 1 < i < K$
N, K	The amount of resources and tasks
C_i, M_i	CPU and memory of t
D_i	The deadline of task T
B_i	The budget cost of task T
C_j, M_j	CPU and memory of R_i
$C^{cost(j)}, M^{cost(j)}$	The costs of CPU and memory of R_i
C^{base}	The base cost of CPUs lowest utilization
M^{base}	The base cost of memory under 1 GB memory
T_{ij}	The duration time of task $T1$ in resource R_i
$C^{Transmission}, M^{Transmission}$	The transmission cost associated with CPU and memory
D_v	The deadline violation rate
N_d	The number violating the deadline time in K tasks

resource manager controls the resources globally. It is used to collect the cloud resource. The global resource manager monitors the time duration task on the resource. In addition to this process, it also calculates the cost of the resource. The overall process is done in cloud. This cloud allows the various physical nodes. There are also local resource manager for the servers. Each local resource manager allows various virtual machines. The cloud provides overall control of the physical nodes and the local resource manager's.

3.1 The System Model

Here, the role of task manager is to accept the tasks, manage the tasks also requests the users to submit those information to the scheduler. The task scheduling architectural diagram shown in Fig. 1.

The each physical node in cloud computing environments managed and monitor by local resource manager. The node's CPU, memory load information and individual task execution time on resources are collected periodically sent global resource manager to calculate resource cost.

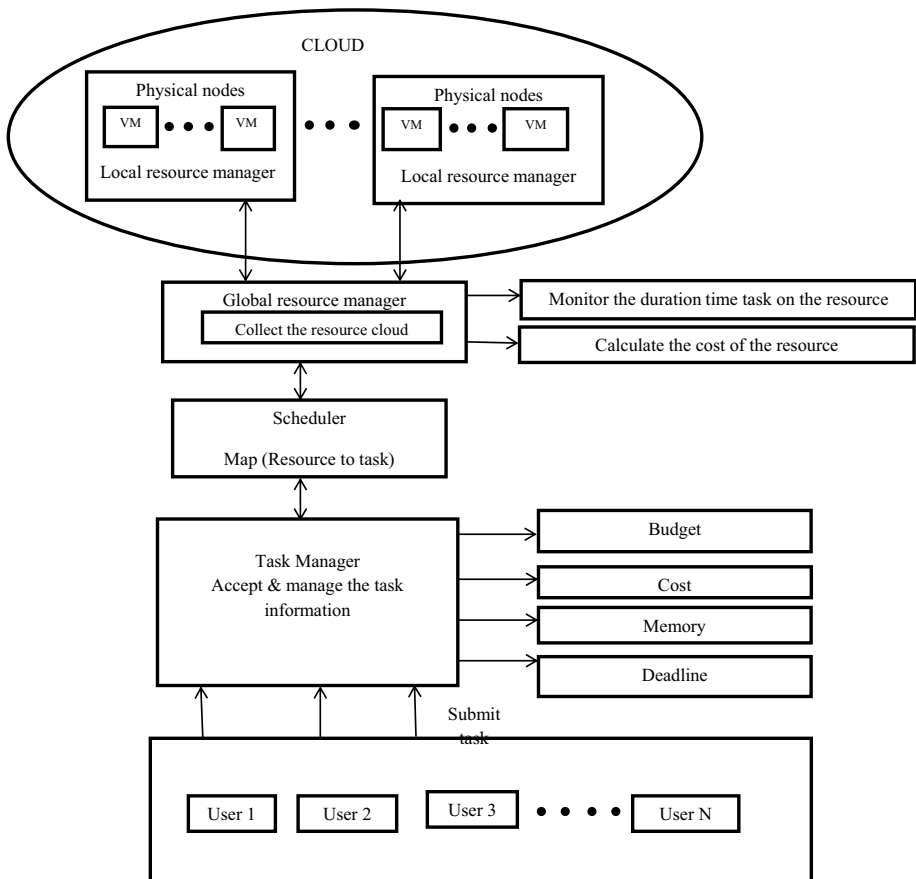


Fig. 1 Architecture diagram of task Scheduling using CPSO

The major architectural component in cloud environments is scheduler, Which is used to schedule task efficiently using Hybridization method propose in this paper. To archive efficient scheduling, the scheduler collect task and resource information from task manger and global resource manager. Based on the collected information such as deadline, cost, the algorithms fits each and every task to the properly judged resource to complete on time (Fig. 2).

4 Resource Cost and Task Scheduling Models

The relation between the user budget and cost of the resources from the resource cost model is depicted in the below given section. The multi-objective optimization scheduling model is built upon the resource cost model so as to optimize scheduling in the cloud.

4.1 The Resource Cost Model

In cloud computing, Tasks are of different natured either it can be more of CPU or storage utilized. In addition to it, the costs for different resources vary with each other; accordingly, task costs are also different. The different between the various task demands for each resource is considered to obtain connection between resource cost and user budget cost.

Into rectify the problem, the resource cost model is used by dividing the resource cost into CPU and memory accordingly to resource definition Formula 1 defines the cost of CPU.

$$C^{cost}(j) = C^{base} \times C_j \times t_{ij} + C^{Transmission} \tag{1}$$

Here, C^{base} defines the base cost of resource's lowest utilization. t_{ij} defines the time duration of the task T_i run time of the task in resource R_j . $C^{Transmission}$ defines the CPU transmission associated with the cost. Formula (2) defines the memory cost.

$$M^{cost(j)} = M^{base} \times M_j \times t_{ij} + M^{Transmission} \tag{2}$$

Similarly, M^{base} represents the base cost while the memory space is 1 GB. t_{ij} represents the time duration of the task T_i running in resource R_j . $M^{Transmission}$ represents the memory transmission of the associated cost. Formula (3) and formula (4) represents the cost model of CPU and memory which are obtained from the cost functions.

$$CPU(j) = \sum_{j=1}^N CPU^{cost}(j) \tag{3}$$

$$Memory(j) = \sum_{j=1}^N M^{cost}(j) \tag{4}$$

4.2 The Scheduling Optimization Model Based on Performance and Budget Constraints

The efficiency of scheduling algorithms, in cloud computing, depends on the performance as well as the budget cost. On the basis of how the tasks, resources and costs are defined, an optimization model for scheduling is presented.

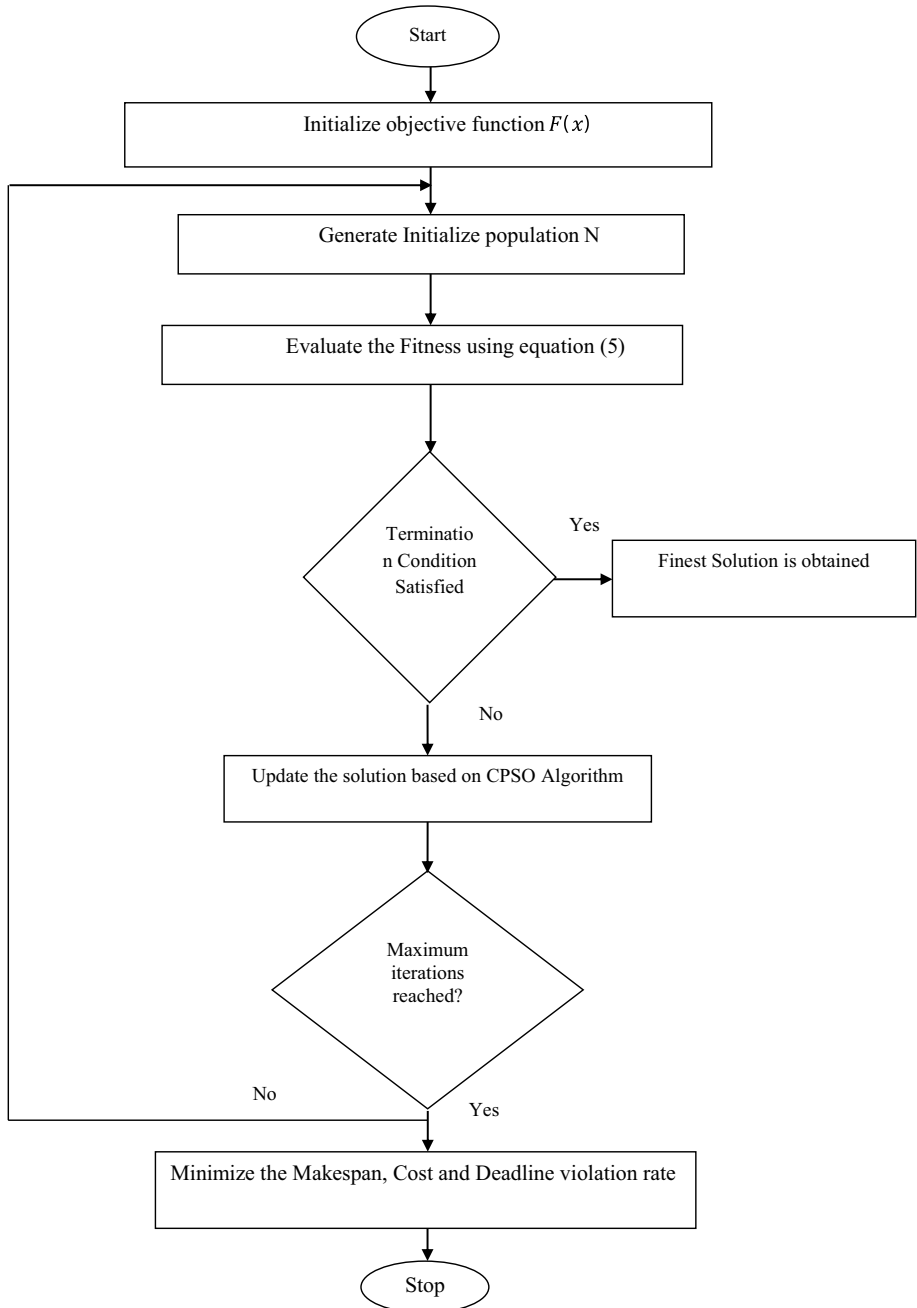


Fig. 2 Flow chart of proposed CPSO algorithm

Initially, consider K tasks $T = \{T_1, T_2, \dots, T_i, \dots, T_k\}$ and N resources $R = \{R_1, R_2, \dots, R_j, \dots, R_n\}$. The proposed optimization model schedules K tasks to N resources so as to achieve optimization. In addition to the optimization, it is also necessary to consider the deadline and budget cost constraints.

A multi-objective optimization problem is as described in formulas 5–8.

$$Fitness\ Function(x) = Minimize \sum_x MAkspan(x), Budget(x) \tag{5}$$

$$Budget(x) = CPU(x) + Memory(x) \tag{6}$$

$$Budget(x) = \sum_{i=1}^k B_i \tag{7}$$

$$Makespan(x) = \sum_{i=1}^k D_i \tag{8}$$

Here, the performance of the objectives considered is denoted by the above defined function $Makespan(x)$ where x denotes a feasible solution. The other function $Budget(x)$ defines the budget costs and memory for the respective costs. The issue of arriving at an optimal solution is solved by the multi-objective optimization problem. This can also be handled by the use of the ant colony algorithm.

5 Particle Swarm Optimization

Inspired by the bird flocking, Dr. Kennedy and Dr. Eberhart proposed a high level procedure algorithm known as the Particle Swarm Optimization for finding the best optimum solution. It is the collection of localised object to find the best effective value. The techniques of PSO are as follows.

1. It is used to set the value of a random population called the swarm. It is used to evaluate the fitness function which are to be optimized and the velocities that direct the flying of the particles.
2. The problem space of the particles is used to find the best optimum solution.
3. It is used to find the pbest value (personal best) and the gbest value (global best) to find the personal and the overall best value of the particular group.

Assume that D is the problem space dimension, and N is the number of particles. The particle Position ‘ i ’ is represented mathematically as $\vec{Y}_i = (\vec{Y}_{i1}, \vec{Y}_{i2}, \dots, \vec{Y}_{iD})^T$. The velocity of the particle is given by $\vec{VP}_i = (VP_{i1}, VP_{i2}, \dots, VP_{iD})^T$. The best position (best previous performance) of that of the particle, in history, is termed as the personal best (P_i). The previously computed value of the best position of all the particles in the group is called the global best g^{best} . The updates of the swarm particles are accomplished using the following equations.

$$vp_{ij}(k + 1) = vp_{ij}(k) + c_1r_1 \left(p_{ij}^{best}(k) - y_{ij}(k) \right) + c_2r_2 \left(g_j^{best}(k) - y_{ij}(k) \right) \tag{9}$$

$$y_{ij}(k + 1) = y_{ij}(k) + vp_{ij}(k + 1) \tag{10}$$

where $i = 1, 2, \dots, N, j = 1, 2, \dots, D$. k is the current generation. $C1$ is the cognitive memory factor, $C2$ is the social memory factor, and they are two constants. $r1$ and $r2$ ($r1, r2 \in [0, 1]$) are random numbers.

$$p_i^{best}(k + 1) = \begin{cases} \bar{y}_i(k + 1), & \text{if } f(\bar{y}_i(k + 1)) < f(\bar{p}_i^{best}(k)); \\ \bar{p}_i^{best}(k), & \text{otherwise} \end{cases} \tag{11}$$

$$\bar{g}^{best}(k + 1) = \operatorname{argmin} f(\bar{p}_i^{best}(k + 1)) \tag{12}$$

where $f(\bar{y}_i(k + 1))$ is the fitness value of the current position of particle i .

5.1 Cuckoo Search Optimization

The cuckoo search algorithmic is a high level procedural language that was proposed by Yang and Deb to generate sufficient good solution. Female Cuckoo lays each egg in randomly chosen nest called host nest. If the host finds the cuckoo egg it simply abandon and build other nest or pushes the cuckoo egg away. Each single egg in a nest represents a possible outcome. The cuckoo egg represents the new solution. The aim is to get the best host solution for cuckoo egg to survive and make it to next generation.

1. Host nest carries its own eggs (i.e. host eggs) with one cuckoo egg in it.
2. Host nest is fixed and host bird finds the cuckoo egg on probability (0, 1).
3. If host bird finds it either pushes the egg away or simply leave the nest.
4. In general animals use Levy’s flight called a random a walk to search food or to find new location.

$$\bar{x}_i(k + 1) = \bar{x}_i(k) + a \oplus Levy \tag{13}$$

where $\bar{x}_i(k)$ represents the current solution, k is the current generation, and $\alpha (\alpha > 0)$ represents the step size

$$Levy(s, \lambda) \sim s^{-\lambda}, \quad (1 < \lambda \leq 3) \tag{14}$$

5.2 Proposed Cuckoo Particle Swarm Optimization Algorithm

- Step 1 Initialize objective function $F(x)$.
- Step 2 Generate Initialize population N .
- Step 3 Evaluate the Fitness using Eq. (5).
- Step 4 Repeat the steps 4 to 6 until Ready task (K) less than maximum number of task.
- Step 5 For each task (T) find the new solution for CS search using the formula (13–14).
- Step 6 For each task (T) find the new solution for PSO search using the formula (9–10).
- Step 7 Find the hybrid solution using the formula

$$\bar{Z}_i(k + 1) = d \times \bar{x}_i(k + 1) + (1 - d) \times \bar{y}_i(k + 1)$$

- Step 8 If the value of hybrid new solution $Z(k + 1)$ is less than value of $f(z(k))$.
Then

Replace the solution of i with current solution.

Step 9 Choose any resource R among N .

Step 10 If the execution time of the resource R based on P_a is high then eliminate the Resource R and choose another resource among N .

Step 11 Assign $\bar{x}_i(k+1) = \bar{z}_i(k+1), \bar{y}_i(k+1) = \bar{z}_i(k+1)$

Step 12 Update $p_i^{best}(k+1)$ and $g_i^{best}(k+1)$ using (11–12).

Step 13 Retain and rank the best solution.

Step 14 End.

6 Simulation Experiment

Using cloudsim toolkit 3.0, a data centre is generated by the experiment. The below Table 2 indicate the simulation environmental details.

6.1 Makespan Evaluation

In this section the performance of the task execution time is evaluated of the proposed work. To evaluating the performance, we consider 100 to 500 tasks with various arrival rates such as 10, 40 and 80. The proposed technique results are compared with PBACO, ACO, MIN–MIN, and FCFS. The simulation results are shown in Figs. 3, 4 and 5. Figure 3 shows the task execution time with arrival rate is 10. When the task is 200, the execution time of CPSO, PBACO, ACO, MIN–MIN, and FCFS are 26.08, 30.69, 34.61, 29.05 and 67 respectively. The task execution time with arrival rate 40 represented in Fig. 4. The execution time of 400 tasks are CPSO, PBACO, ACO, MIN–MIN, and FCFS are 47.41, 56.448, 79.2, 63.98 and 148.8 respectively. Figure 5 shows the task execution time with arrival rate is 80. when task is 500 the execution time of CPSO, PBACO, ACO, MIN–MIN, and FCFS are 70.02, 78.68, 171.20, 105.88 and 244.5 respectively.

Table 2 Experimental setup details

Entity type	Parameters	Value
Task	Number of tasks	100–500
	Length	[400,1000]MIPS
	File size	[200,1000]MB
	Output size	[20,40]MB
Host	Host memory(RAM)	1860 MIPS, 2660 MIPS
	Host storage	10 GB
	Host bandwidth	100 M/s
Virtual machine	Number of VMs	50
	Type of policy	Time-shared
	VM RAM	512 MB
	VMM	Xen
	OS	Linux
Data center	Number of CPUs	1 on each
	Number of data center	10
	Number of hosts	10

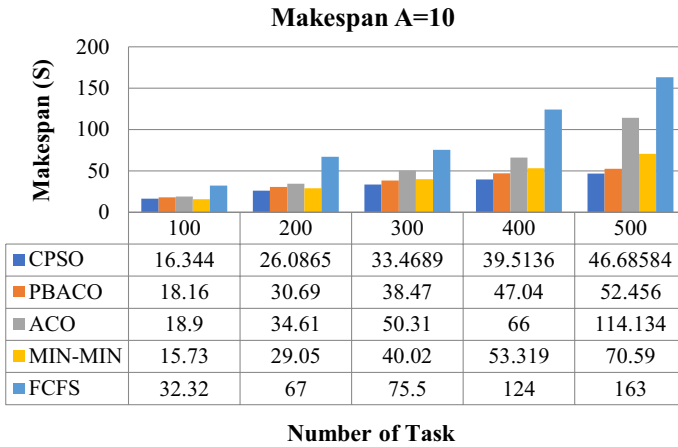


Fig. 3 Makespan at arrival rate = 10

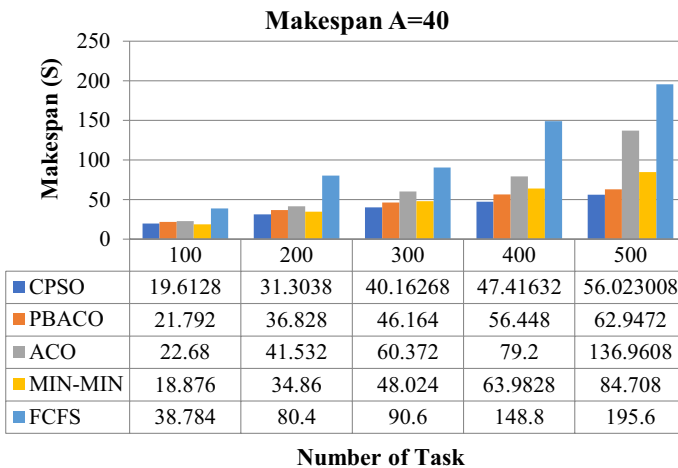


Fig. 4 Makespan at arrival rate = 40

6.2 Cost Evaluation

In this section the cost evaluation is performed. In this simulation we consider 200,400 & 500 task with various deadlines. The deadline ranges are from 10 to 100. The obtained results are shown in Figs. 6, 7 and 8. In Fig. 6 shows, 200 tasks cost values. 400 task cost values is shown in Fig. 7, similarly 500 task cost value is represented in Fig. 8. The proposed algorithm CPSO cost obtained values is compared with PBACO, ACO, MIN-MIN, and FCFS. From the simulation results, it is evident that the proposed CPSO achieves minimal cost in all the three set of task.

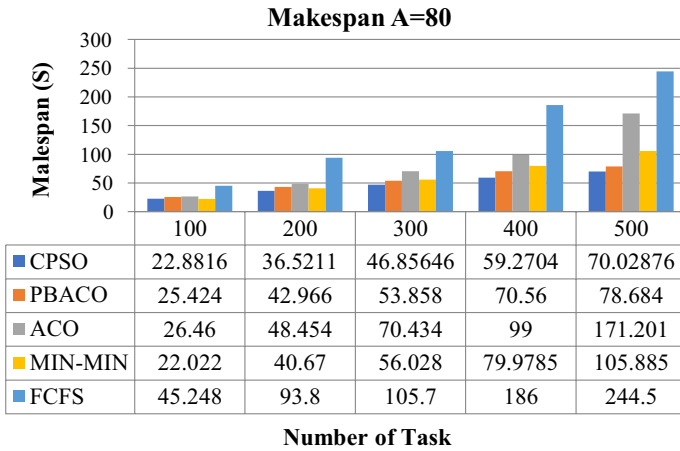


Fig. 5 Makespan at arrival rate = 80

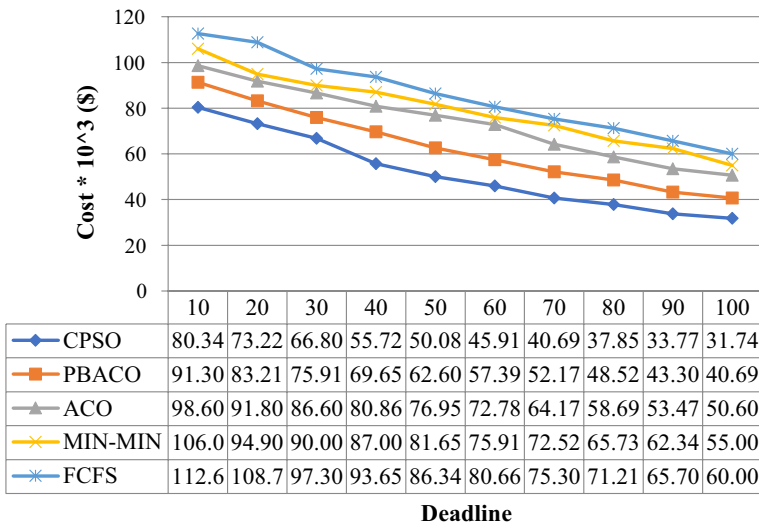


Fig. 6 Cost for 200 tasks with varied deadline

6.3 Deadline Violation Rate

The deadline violation rate is evaluated for verifying the Quality of service (QoS) of the scheduling. In this experiment evaluation we considered 200,400 and 500 tasks. Figures 9, 10 and 11 shows the deadline violation rate of 200, 400 and 500 tasks with varied deadlines. When the lower number of tasks is consider minimum number of tasks violating the deadline. Using CPSO algorithm, minimum numbers of tasks are violating the deadline when compare to PBACO, ACO, MIN-MIN, and FCFS.

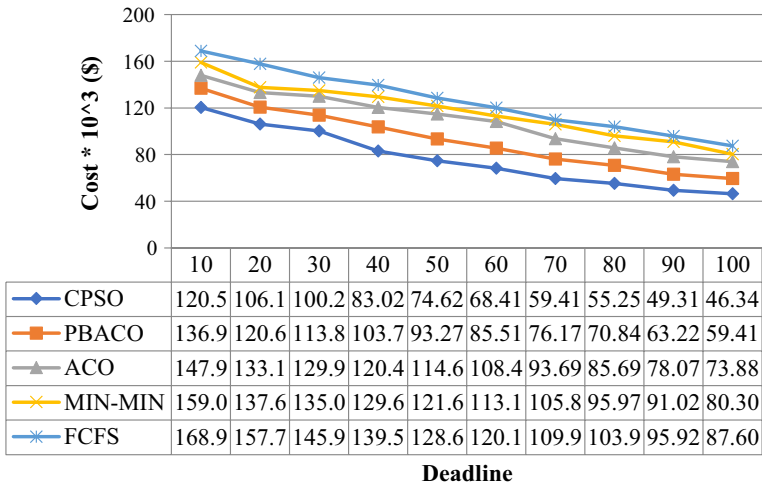


Fig. 7 Cost for 400 tasks with varied deadline

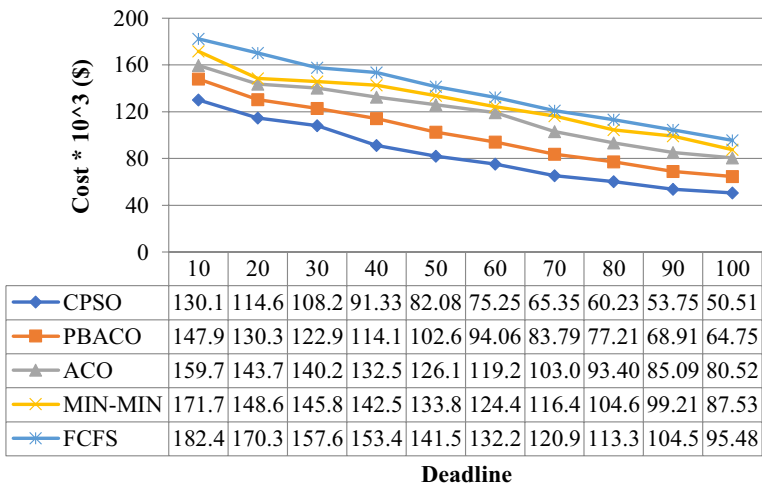


Fig. 8 Cost for 500 tasks with varied deadline

7 Conclusion

With the relationship between cost of resources and tasks in mind, a multi-objective optimisation scheduling method was presented. This method was based on cuckoo search and Particle Swarm Optimization (CPSO) algorithm. The aim of the proposed method is to improve and optimize the scheduling performance and costs. The proposed algorithm reduces all the cost factors such as performance cost and user costs. Also this model

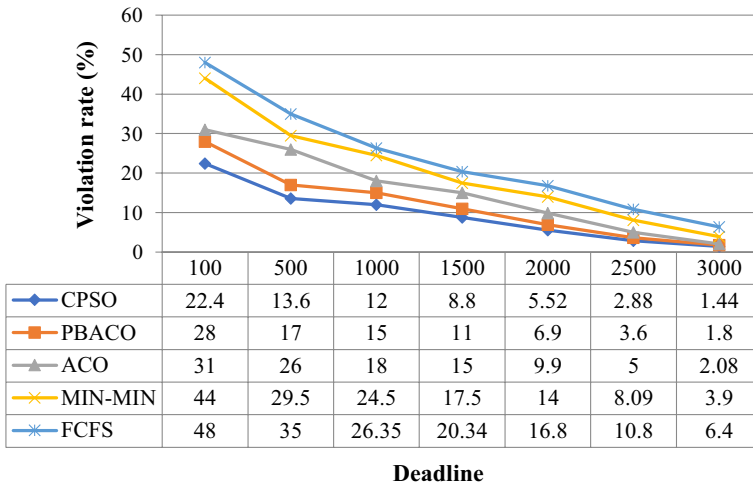


Fig. 9 Deadline violation rate for 200 tasks

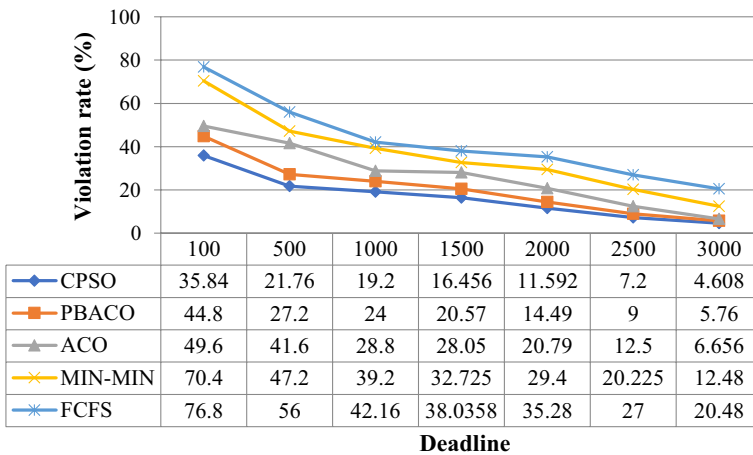


Fig. 10 Deadline violation rate for 400 tasks

relatively minimizes the makespan value of tasks. The proposed CPSO algorithm achieves minimal deadline violation rate when compared with algorithms such as PBACO, ACO, MIN-MIN, and FCFS. Application of the proposed algorithm to optimizes the other QoS parameters are considered in the future work.

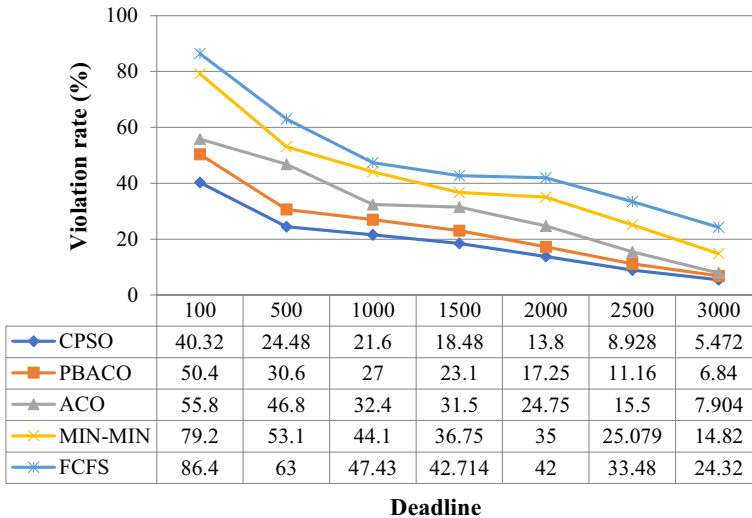


Fig. 11 Deadline violation rate for 500 tasks

References

1. Abdullahi, M., & Ngadi, M. A. (2016). Hybrid symbiotic organisms search optimization algorithm for scheduling of tasks on cloud computing environment. *PLoS ONE*, *11*(6), e0158229.
2. Dr, T., Jacob, P., & Pradeep, K. (2018). A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment. *Wireless Personnel Communications*, *101*(4), 2287–2311.
3. Dr, T., Jacob, P., & Pradeep, K. (2018). OCSA: Task scheduling algorithm in cloud computing environment. *International Journal of Intelligent Engineering and Systems*, *11*(3), 271–279.
4. Somasundaram, T. S., & Govindarajan, K. (2014). CLOUDRB: A framework for scheduling and managing High-Performance Computing (HPC) applications in science cloud. *Future Generation Computer Systems*, *34*, 47–65.
5. Zuo, L., et al. (2015). A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. *IEEE Access*, *3*, 2687–2699.
6. Pradeep, K., Dr, T., & Jacob, P. (2017). CGSA scheduler: A multi-objective-based hybrid approach for task scheduling in cloud environment. *Information Security Journal: A Global Perspective*, *27*(2), 77–91.
7. Madni, S. H. H., et al. (2017). Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. *PLoS ONE*, *12*(5), e0176321.
8. Latiff, M. S., Abd, G. A.-S., & Madni, S. H. H. (2016). Secure scientific applications scheduling technique for cloud computing environment using global league championship algorithm. *PLoS ONE*, *11*(7), e0158102.
9. Thanasias, V., et al. (2016). VM capacity-aware scheduling within budget constraints in IaaS clouds. *PLoS ONE*, *11*(8), e0160456.
10. Idris, H., et al. (2017). An improved ant colony optimization algorithm with fault tolerance for job scheduling in grid computing systems. *PLoS ONE*, *12*(5), e0177567.
11. Abdel-Basset, M., Abdle-Fatah, L., & Sangaiah, A. K. (2018). An improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment. *Cluster Computing*. <https://doi.org/10.1007/s10586-018-1769-z>.
12. Tsai, J.-T., Fang, J.-C., & Chou, J.-H. (2013). Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. *Computers & Operations Research*, *40*(12), 3045–3055.
13. He, H., et al. (2016). AMTS: Adaptive multi-objective task scheduling strategy in cloud computing. *China Communications*, *13*(4), 162–171.

14. Zuo, X., Zhang, G., & Tan, W. (2014). Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IaaS cloud. *IEEE Transactions on Automation Science and Engineering*, 11(2), 564–573.
15. Sreenu, K., & Sreelatha, M. (2017). W-Scheduler: whale optimization for task scheduling in cloud computing. *Cluster Computing*. <https://doi.org/10.1007/s10586-017-1055-5>
16. Sreenu, K., & Malempati, S. (2017). MFGMTS: Epsilon constraint-based modified fractional grey wolf optimizer for multi-objective task scheduling in cloud computing. *IETE Journal of Research*, 1–15.
17. Zuo, L., Shu, L., Dong, S., Chen, Y., & Yan, L. (2017). A multi-objective hybrid cloud resource scheduling method based on deadline and cost constraints. *IEEE Access*, 5, 22067–22080.
18. Gobalakrishnan, N., & Arun, C. (2018). A new multi-objective optimal programming model for task scheduling using genetic gray wolf optimization in cloud computing. *The Computer Journal*, 61(10), 1523–1536.
19. Natesan, G., & Chokkalingam, A. (2018). Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. *ICT Express*. <https://doi.org/10.1016/j.ict.2018.07.002>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Dr. T. Prem Jacob received the B.E. degree in Computer Science and Engineering from C.S.I. Institute of Technology, Manonmaniam Sundaranar University, Nagercoil, India in 2004, M.E degree in Computer Science and Engineering from Sathyabama University, Chennai, India in 2006 and Ph.D. degree from Sathyabama University, Chennai, India. He is an Associate Professor of Computer Science and Engineering in Sathyabama Institute of Science and Technology, Chennai. He has participated and presented many Research Papers in International and National Conferences. His area of interests includes Software Engineering, Data mining and Data warehouse and Cloud computing.



K. Pradeep pursued his Bachelor's degree in Computer Science and Engineering at Madras University, Tamil Nadu, India in 2004. Then he obtained his Master's degree in Computer Science and Engineering from Anna University, Tamil Nadu, India in 2011. Currently, he is a research scholar in Sathyabama Institute of Science and Technology, Chennai, India and he is an Assistant Professor in the Department of Information Technology, St. Joseph's College of Engineering, Tamil Nadu, India. His current research interests are Cloud computing and Virtualization.