

# An Improved Grey Wolf Optimization Algorithm Based Task Scheduling in Cloud Computing Environment

Gobalakrishnan Natesan<sup>1</sup> and Arun Chokkalingam<sup>2</sup>

<sup>1</sup>Department of Information Technology, Sathyabama University, India

<sup>2</sup>Department of Electronics and Communication Engineering, R.M.K College of Engineering and Technology, India

**Abstract:** *The demand for massive computing power and storage space has been escalating in various fields and in order to satisfy this need a new technology known as cloud computing is introduced. The capability of providing these services effectively and economically has made cloud computing technology more popular. With the advent of virtualization, IT services being offered have started to shift to cloud computing. Virtualization had paved way for resource availability in an inexhaustible manner. As Cloud Computing is still at its unrefined form and to derive its full potential more analysis is needed. The way in which resources and tasks get allocated in cloud environment requires more analysis. This in turn accounts for the Quality of Services (QoS) of the services offered by cloud service providers. This paper proposes to simulate the Performance-Cost Grey Wolf Optimization (PCGWO) algorithm based to achieve optimization in the process of allocation of resources and tasks in cloud computing domain using CloudSim toolkit. The main purpose is to lower both the processing time and cost in accordance to objective function. The superiority of proposed technique is evident from the simulation results that show a comprehensive reduction in task completion time and cost. Also using this technique more no. of tasks can be efficiently completed within the deadline. Thus the results indicate that in accordance to performance the PCGWO method fares better than existing algorithms.*

**Keywords:** *Virtualization, cloud computing, GWO, task scheduling, optimization, resource, CloudSim and QoS.*

Received July 8, 2017; accepted September 13, 2017  
<https://doi.org/10.34028/iajit/17/1/9>

## 1. Introduction

Cloud computing is said to be the most prominent technology in IT sector. It is applicable to various fields such as, health care, business, smart system, mobile system, environmental computing. Due to the competitors and the necessity of the world, there has been rapid development in the field of cloud computing in recent time. By the concept of virtualization, the cloud computing resources are distributed within the clients. Several remote environments are gathered and clustered together and utilize to the fullest of its capability [5]. There are many ways to provide services like software as a service, platform as a service and infrastructure as a service [27]. The service provided makes it look like the cloud computing resources are inexhaustible. The services provided are charged by pay per use criteria. It becomes a major deciding factor for the migration of IT services to cloud environment. The resources are provided in a silver platter to organizations hassle free.

Task Scheduling is always considered as one of the broadly researched problems as it holds the ultimate key to unlocking the fullest potential of the cloud computing technology [23]. It offers endless amount of

possibilities to be explored. For more efficient ways to provide services, task scheduling provides the key to the productive optimal solution. The optimal key can be figured out only through heuristic methods as there are no exact means to derive a perfect solution for the NP-hard problem. The goal of task scheduling algorithm is to minimize the cost and the execution time [9, 11, 13, 30]. The algorithm makes the decision on which Virtualization Machine a task should be allocated.

In cloud computing environment, the processing capabilities and the characteristics are heterogeneous in nature. There are some deciding factors which are to be taken into consideration, such as, an execution time, flowtime, response time and cost. There are two methods which are proposed to clarify the problems related to task scheduling. They are heuristics based and meta heuristics based. Heuristics based approaches finds optimal solution under predefined constraints. The solution obtained from the heuristics based method relies too much on rules and size of the problem. This method is way too much expensive and exorbitant. Meta heuristics based methods provide a variety of solution instead of a single candidate solution like heuristics based method solutions. The

performance of metaheuristics based method are relatively better than heuristics methods. Some instances of Meta heuristics are Genetic Algorithm (GA) [6], Ant Colony Optimization (ACO) [8] and Symbiotic Organisms Search (SOS) [7]. Grey Wolf Optimization (GWO) algorithm outperforms other well-known meta-heuristic algorithms in terms of cost, execution time, etc.

The grey wolf algorithm exhibits the actual survival instincts of wolves, their cooperative nature while hunting for their prey and its leadership hierarchy and hunting nature. There are four types of grey wolves for exhibiting the leadership hierarchy. Further, it implies the three main steps of hunting such as searching for prey; encircling prey and attacking prey are enforced in it. Thus, the unrealized possibility of grey wolf optimization algorithm in finding a universal solution to optimization problems displayed so far as made it attractive for the further analysis and development. The purpose is to achieve optimal schedules by reducing execution time and the cost of task.

The main contributions of this paper are:

1. The objective of optimum scheduling of tasks on VMs is made clear under the utilization level of VMs In order to reduce the Makespan, Total Cost and the Maximum number of task completion within the deadline.
2. Applying Performance-Cost Grey Wolf Optimization (PCGWO) to find the nearest optimal solution to realize the potential of the proposed algorithm.
3. The proposed method is implemented in CloudSim.
4. Performance analysis between the traditional algorithms with the proposed algorithm by the aspects of Makespan, Total Cost and the Maximum number of task completion within the deadline.

Section 2 presents the related literature work. The illustration and objective of the problem is described in section 3. Section 4 represents task scheduling and the estimated resource cost. The scheduled tasks are explained using PCGWO algorithm in section 5. Section 6 elaborates on the simulation details. The conclusion is provided in section 7 and section 8 provides references.

## 2. Related Works

This paper primarily focuses on Task Scheduling and Resource Allocation. Task scheduling is considered to be an NP-problem [22] and there are many heuristics and meta-heuristics algorithms which have been designed for making an effective scheduling in computing environments. Some of the algorithms are designed based on the inspiration towards the nature and non-nature characteristics [12]. It considers the makespan, flowtime, fairness, cost and deadline. To solve task assignment problem and to reduce

makespan, total cost and resource utilization, the meta-heuristic algorithms [4, 10, 15, 17, 21, 25] are implemented.

The economy based grid scheduling mechanism for parameters sweep applications in computational grid environments was proposed by Abramson *et al.* [3] and it accounts for resource cost, price and deadline. The task selection and resource allocation framework in a cloud environment was suggested by Song *et al.* [19]. It is classified by two conditions, namely heavy workload and light workload condition. It is mainly based on resource utilization. The optimized resource scheduling mechanism for open-source cloud systems using the Improved Genetic Algorithm (IGA) was proposed by Zhong *et al.* [28]. A Particle Swarm Optimization (PSO) based scheduling mechanism for data-intensive workflow types of applications were presented by Pandey *et al.* [16]. The Execution cost and Data-transfer cost is considered by the scheduling mechanism and it was compared with Best Resource Selection (BRS) technique.

The Ant Colony Optimization (ACO) approach minimizes the makespan of tasks which is available in workflow applications. The metaheuristic technique of Chemical Reaction Optimization (CRO) for Task Scheduling in the dynamic Grid Environment was implemented by Xu *et al.* [24]. By taking the workload and reliability of the resources into account, the makespan and flowtime was minimized. The GA-based scheduling mechanism for scientific workflow applications on Utility Grids in accordance with the deadline and budget constraints was implemented by Yu and Buyya [26].

The work done by Somasundaram and Govindarajan [18] provides the framework for task scheduling and managing high performance computing application to minimize makespan cost job rejection ratio and maximize job meeting deadline. Zuo *et al.* [29] put forth the Multi-objective Optimization Scheduling method based on the resource cost model and Ant colony algorithm to minimize makespan, cost, deadline violation rate and resource utilization. Abdullahi *et al.* [2] introduced Discrete Symbiotic Organism Search algorithm for task scheduling. It targets makespan, response and degree of imbalance among VM's. It performs relatively better than the Particle Swarm Optimization (PSO).

Abdullahi *et al.* [1] suggested a task scheduling method based on the combination of simulated annealing and symbiotic organism search to reduce makespan and degree of imbalance among VM's. Tsai *et al.* [20] worked on the cost and time model using improved differential evolution algorithm to optimize task scheduling and resource allocation. It focuses on reducing on makespan and total-cost. The proposed paper will evaluate the quality of solution and the feedback from both performance and cost.

### 3. The Description of Problem

The system architecture model and its terminologies are listed above in Table 1. The chore and resources are defined and constructed with Table 1 that holds notations and denotations that is used all over the segment.

#### 3.1. The Definition of Tasks and Resources

First, it is assumed that there are  $N$  tasks,  $T=\{T_1, T_2, \dots, T_N\}$  and  $M$  resources  $R=\{R_1, R_2, \dots, R_M\}$  in the current system of cloud environment. Here, cloud resources denote virtual resources.

- **Definition 1 (Tasks):**  $T_i=(CPU_i, Storage_i, Memory_i, Bandwidth_i, DL_i, BC_i)$ . The usage of CPU, storage, memory and bandwidth are accounted with four parameters as user applied. Where the deadline of the task and budget cost are symbolized as  $(DL_i, BC_i)$  by user. And this information comes from the task manager and relented by users.
- **Definition 2 (Resources):** The main four parameters (CPU, Storage, Memory and Bandwidth) are delineated as virtual resource cloud. Say,  $R_j=(CPU_j, Storage_j, Memory_j, Bandwidth_j)$ . These four parameters are the representative of CPU utilization, storage, memory and bandwidth usage.

Table 1. Notation and descriptions.

Notation	Description
$T_i$	the task $i, 1 \leq i \leq N$
$R_j$	the resource $j, 1 \leq j \leq M$
$M, N$	the amount of resources and tasks
$CPU_i, Storage_i, Memory_i, Bandwidth_i$	CPU, Storage, Memory and Bandwidth of $T_i$
$DL_i$	the deadline of task $T_i$
$BC_i$	the budget cost of task $T_i$
$CPU_j, Storage_j, Memory_j, Bandwidth_j$	CPU, Storage, Memory and Bandwidth of $R_j$
$CPU_{cost}(j), Storage_{cost}(j), Memory_{cost}(j), Bandwidth_{cost}(j)$	the cost of CPU, Storage, Memory and Bandwidth of $R_j$
$CPU_{base}$	the base cost of CPU under the lowest usage
$Storage_{base}$	the base cost of storage under the lowest usage
$Memory_{base}$	the base cost of memory under 1GB memory
$Bandwidth_{base}$	the base cost of bandwidth under the lowest usage
$t_{ij}$	the duration time of task $T_i$ in resource $R_j$
$CPU_{Trans}, Storage_{Trans}, Memory_{Trans}, Bandwidth_{Trans}$	the transmission cost associated with CPU, Storage, Memory and Bandwidth
$DL$	the deadline of the task

**Assumption 1:** In order to progress the research, we need to provide relevant prediction to the definition. By hope that user as relented the information rightly. In other way, the information of resource relented by user is precise.

The system framework model of task scheduling and resource allocation is shown in Figure 1. The task manager work is to accept and manage the task requests submitted by user and then pushes the information to the scheduler.

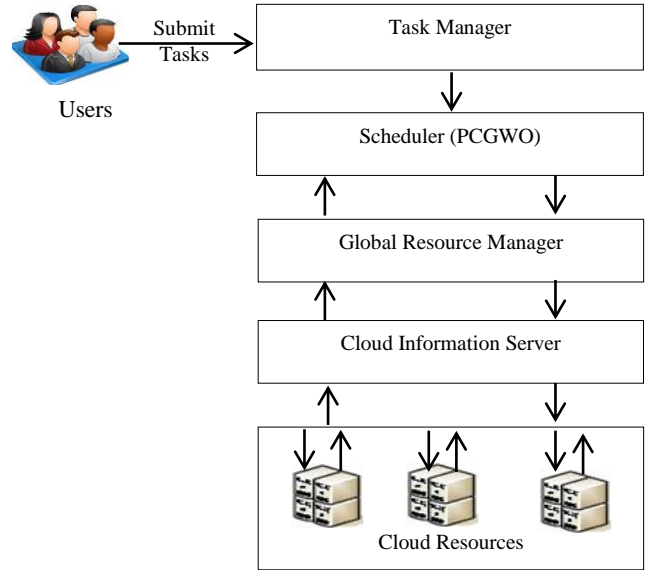


Figure 1. Framework model of the system.

The scheduler is considered to play important in this. Since it is core component and responsible for allocating tasks to resources of the system by applying PCGW scheduling method and it schedules task based on the information to the resources. Initially, it collects the task and resource information from task manager and the global resource manager. Later, it checks whether the resource  $R_j$  meet the requirement of the task  $T_i$ . At last, the resource  $R_j$  allocated by scheduler.

The work of cloud resource manager is to control and maintain resource nodes and frequently controlling virtual resources that receive their Central Processing Unit (CPU), memory load, storage and bandwidth information. Later it is pushed to global resource manager.

The global resource manager work is to frequently collect and update information from cloud information server. Furthermore, it work is to control the duration tie of task running on resource and then relented to global manager and which computes the resource cost by information relented from resource model.

### 4. Representations of Task Scheduling and Resource Cost

This segment defines the resource cost model in order to build bondage between resource costs and user budget. Based on the resource cost model, the multi-objective optimization scheduling model is suggested to achieve multi-objective optimization scheduling in cloud computing.

#### 4.1. The Resource Cost Model

In the field of cloud computing, tasks and resources are contradictory with each other. For instance, some task involve while others need some extern bit of storage. There is a variance in costs for different resources.

This leads to the difference in task costs. Thus, if we were to consider the variance in demand of resources for the tasks, it will be productive to emulate the details of the costs of the task briefly and explain the link between the resource costs & user budget costs. To approach this problem, a resource cost model is proposed in this paper which separates the resource cost model into bilateral parts of CPU and memory. Based on the resource definition the model has bilateral parts of the CPU and memory.

The CPU cost is defined as follows:

$$CPU_{cost}(j) = CPU_{base} * CPU_j * t_{ij} + CPU_{Trans} \quad (1)$$

When a resource is utilized by the lowest utilization the base cost is incurred. This base cost is represented by  $CPU_{base}$ . The task  $T_i$  runs on resource  $R_j$  in the duration time  $t_{ij}$ . The cost incurred by the CPU transmission is represented by  $CPU_{Trans}$ .

The storage cost is defined by the Equation as follows:

$$Storage_{cost}(j) = Storage_{base} * Storage_j * t_{ij} + Storage_{Trans} \quad (2)$$

The task  $T_i$  runs on resource  $R_j$  in the duration time  $t_{ij}$ . The cost incurred by the storage transmission is represented by  $Storage_{Trans}$ .

The memory cost is defined by the Equation as follows:

$$Memory_{cost}(j) = Memory_{base} * Memory_j * t_{ij} + Memory_{Trans} \quad (3)$$

Equivalently,  $Memory_{base}$  is the base cost when memory is 1 GB. The task  $T_i$  runs on resource  $R_j$  in the duration time  $t_{ij}$ . The cost incurred by the memory transmission is represented by  $Memory_{Trans}$ .

The bandwidth cost is defined by the Equation as follows:

$$Bandwidth_{cost}(j) = Bandwidth_{base} * Bandwidth_j * t_{ij} + Bandwidth_{Trans} \quad (4)$$

The task  $T_i$  runs on resource  $R_j$  in the duration time  $t_{ij}$ . The cost incurred by the bandwidth transmission is represented by  $Bandwidth_{Trans}$ .

On the basis of the mentioned models of CPU, storage, memory and bandwidth the cost functions are acquired as follows:

$$CPU(j) = \sum_{j=1}^M CPU_{cost}(j) \quad (5)$$

$$Storage(j) = \sum_{j=1}^M Storage_{cost}(j) \quad (6)$$

$$Memory(j) = \sum_{j=1}^M Memory_{cost}(j) \quad (7)$$

$$Bandwidth(j) = \sum_{j=1}^M Bandwidth_{cost}(j) \quad (8)$$

## 4.2. The Scheduling Optimization Model Based on Performance and Cost Constraints

In the field of cloud computing, the deciding factors of scheduling efficiency are the cost of the user budget and the scheduling performance. In this paper, on the

basis of the resource cost model and by the definition of tasks and resources a scheduling optimization model is provided. At first,  $T = \{ T_1, T_2, \dots, T_i, \dots, T_N \}$  and  $M$  resources  $R = \{ R_1, R_2, \dots, R_j, \dots, R_M \}$  are assumed as tasks in the cloud computing system. The scheduling algorithm considered as optimization problem: to achieve optimal span and to schedule  $k$  tasks to  $N$  resources. Simultaneously, there is a need to consider the constraints like deadlines and budget costs. It is optimization problem and it is defined as Equations (9, 10, 11, and 12).

$$\text{Minimize} \sum_x H(x) = BC(x), MS(x) \quad (9)$$

$$BC(x) = CPU(x) + Storage(x) + Memory(x) + Bandwidth(x) \quad (10)$$

$$BC(x) \leq \sum_{i=1}^k BC_i \quad (11)$$

$$MS(x) \leq \sum_{i=1}^k DL_i \quad (12)$$

## 5. Scheduling Based on PCGWO Algorithm

In the process of finding an efficient way to tackle the problems of computing, many began to illustrate the characteristics of nature in problem solving to achieve better results. Among those nature inspired algorithms, the idea of a new population based swarm intelligence approach known as GWO was put forth by Mirjalili *et al.* [14] in 2014. The Algorithm was inspired by the nature of grey wolves. The algorithm GWO imitates the social leadership hierarchy and the hunting behaviour of grey wolves. In order to mathematically represent these social leadership of the grey wolves the best solution is considered to be alpha ( $\alpha$ ) wolf and the second and the third best solutions are represented by beta ( $\beta$ ) and delta ( $\delta$ ) wolves respectively. The rest of the candidate solutions are represented by omega ( $\omega$ ). The hunting (optimization) is guided by the  $\alpha$ ,  $\beta$  and  $\delta$  wolves. In order to search the optimum path the  $\omega$  wolves follows the socially dominant wolves. The hunting behaviour of these grey wolves are mainly categorised into three parts: Tracking, Encircling and attacking the prey. The encircling behaviour can be mathematically represented by the following Equations:

$$DS = MOD(N \cdot X_{Pray}(t) - M \cdot X_W(t)) \quad (13)$$

$$X_W(t+1) = X_{Pray}(t) - M \cdot DS \quad (14)$$

Where  $[DS]$  denotes the distance between the position vectors of both the prey  $[X_{Pray}]$  and a wolf  $[X_W]$  and  $t$  denotes the current iteration number.  $[M]$  and  $[N]$  are coefficient vectors and they are calculated by the following:

$$M = 2a \cdot ran_1 \cdot a \quad (15)$$

$$N = 2 \cdot ran_2 \quad (16)$$

The elements of 'a' are declined linearly from 2 to 0 in the course of iteration and  $ran1$ ,  $ran2$  are random vectors in [0, 1].

- **Fitness Function (FF):** Fitness value denotes the capacity of the organism to adapt to the environment. Makespan and budget cost can be minimized only when the efficient group of tasks is fed to the Virtual Machines (VMs). Therefore, the fitness evaluation function  $FF$ ,

$$FF(x) = \lambda e^{-MS(x)} + \gamma e^{-BC(x)} \quad (17)$$

Where  $MS(x)$  means makespan and  $BC(x)$  means budget cost. Here  $\lambda$  denotes weight factor of performance and  $\gamma$  denotes that of cost. Execution time and cost are two major scheduling objectives minimized

While hunting, the first three best solution ( $\alpha$ ,  $\beta$  and  $\delta$ ) acquired are stored and compel other search agents (including the omega) to update their position based on the best search agent. The Equation are:

$$DS_{\alpha} = MOD(N_1 \cdot X_{\alpha} - X_w) \quad (18)$$

$$DS_{\beta} = MOD(N_2 \cdot X_{\beta} - X_w) \quad (19)$$

$$DS_{\delta} = MOD(N_3 \cdot X_{\delta} - X_w) \quad (20)$$

$$X_{w1} = X_{\alpha} - M_1 \cdot (DS_{\alpha}) \quad (21)$$

$$X_{w2} = X_{\beta} - M_2 \cdot (DS_{\beta}) \quad (22)$$

$$X_{w3} = X_{\delta} - M_3 \cdot (DS_{\delta}) \quad (23)$$

$$X_{w(t+1)} = (X_{w1} + X_{w2} + X_{w3})/3 \quad (24)$$

The position is updated based on the alpha, beta, and delta. The alpha, beta and delta access the position of the prey and other wolves rely on this to update their position around the prey and the final position is limited to the circle.

The search pattern of the grey wolves solely depends upon the position of alpha, beta and delta. They diverge from each other while searching. In theory, to make the search agent to deviate from the prey  $M$  differs with random values greater than 1 or less than -1. This inturn enables exploration and permit GWO algorithm to search. If  $|M| > 1$ , the grey wolves deviates from the grey wolves to find a suitable prey.

When the prey comes to halt, the grey wolf ends its hunt by attacking it. If  $|M| < 1$ , grey wolves coincides towards the prey and strikes it. The vector  $M$  is a random value in the interval  $[-a, a]$ . This method is called as exploitation.

## 5.1. Pseudo Code of Proposed PCGWO Algorithm

The proposed algorithm is terminated when a stated (maximum) number of iterations are completed and then the current solutions are the task chore solutions.

*Algorithm 1: proposed PCGWO*

*Input:*

*Requirement of PCGWO algorithm*

*Specification of Task Scheduling & Resource allocation.*

*Tasks  $T_1, T_2, \dots, T_N$  and Resources  $R_1, R_2, \dots, R_M$  and  $Max_{itr}$*

*Output: Set of tasks mapped to VMs*

1. *Procedure: PCGWO*

2. {

*//Parameters Initialization*

3. *Initialize the number of task, number of resources, Set the initial values of the cluster size n, parameter a, coefficient vectors M, N and the maximum number of iterations  $Max_{itr}$*

4. *Set  $t = 0$  {counter initialization}.*

*//Population initialization*

5.  $i = 1$

6. *while( $i \leq n$ ) do*

7. {

a. *Bring about an initial population  $X_{wi}(t)$  randomly.*

b. *Appraise the fitness evaluation function of each hunt wolf (solution)  $FF(x)$*

8. *} end while*

*//Assign the best three solutions*

9. *Accredit the values of the first, second and third near optimum solution  $X_{\alpha}, X_{\beta}$  and  $X_{\delta}$  correspondingly.*

10. *reiteration*

11.  $i = 1$

12. *while( $i \leq n$ ) do*

13. { *Update each hunt agent in the location as shown in equation 14*

a. *Cutback the parameter 'a' from 2 to 0.*

b. *Update the coefficient M & N as shown in equation 15, 16 correspondingly*

c. *Appraise the fitness function of each hunt agent  $FF(x)$*

14. *} end While*

15. *update the vectors  $X_{\alpha}, X_{\beta}$  and  $X_{\delta}$*

16. *Set  $t=t+1$  (iteration counter increasing)*

17. *//Termination criteria*

18. *until ( $t < Max_{itr}$ ). (Termination criteria satisfied)*

19. *//Best solution*

20. *Produce the optimum solution  $X_{\alpha}$*

21. *} //end-procedure*

## 6. Simulation Details

The performance of the suggested technique was assessed by CloudSim. The toolkit that is used for modeling cloud computing scenarios is CloudSim. Three datacenters were developed each consisting of 2 hosts correspondingly. Every host has 12GB/s bandwidth, 1.5TB storage, 30GB RAM and space-shared VM scheduling algorithm. Xen Virtual Machine Monitor (VMM), Linux operating system and cumulative processing power of 1000000 MIPS. One host is quad-core machine and the other host is the octa-core machine each with X86 architecture. 30 VM's were generated each VM image size of 12GB, 1.2GB/s bandwidth, 1 processing unit, 0.6GB memory. The power of the processing of VMs ranges from 100-6000 MIPS correspondingly. Xen VMM and time-shared cloudlet scheduler were used for all the VMs. The results are correlated with alternative prominent

Metaheuristic approaches of standard GWO algorithm and the traditional technique First Come First Served (FCFS), Min-Min method. The simulation approach uses random number generator that generates 100-500 tasks requests in an arbitrary that prefers the resources based on the scope of developing cloud resources.

**6.1. Simulation Results and Discussion**

This section explains the simulation results as the PCGW algorithm is compared to other scheduling algorithms like the standard GWO, Min-Min and FCFS to substantiate its superiority. The GWO algorithm backs the fittest solution based on an optimized fitness function. The Min-Min algorithm aims to complete the tasks with minimum workload first and then proceeds to longer tasks. The FCFS algorithm simply follows the first come first serve algorithm. The attributes considered are the makespan, cost and the number of tasks meeting the deadline. It is evident from the simulation results that the PCGW algorithm gives a much improved makespan, cost and increased number of tasks meeting the deadline. The detailed results are given below:

**6.1.1. Comparison of Makespan Values**

The main objective of any cloud scheduling algorithm is to produce a reduced makespan and PCGW follows the same order. The tasks were executed in the order of 100-500, corresponding to three different arrival rates of 30, 60, 90 (in Figures 2, 3, and 4 respectively) and the makespan of PCGW is compared with other algorithms. The makespan values produced by the algorithms under study were obtained by varying the number of resources allocated to complete the tasks. It can be seen that the proposed PCGW algorithm induces minimum makespan among the four algorithms under study. The PCGW algorithm shows 40% decrease in the makespan values when compared to the makespan shown by the FCFS algorithm. The contrast is drawn against the number of tasks and the corresponding makespan values for a specific number of resources. Even when the number of tasks is increased to 500, the makespan produced by the PCGW algorithm progressively increases and yet still remains the best.

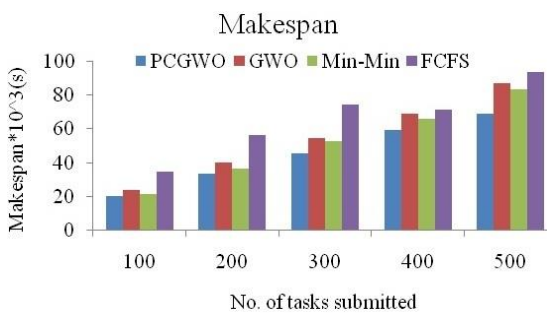


Figure 2. Makespan, the arrival rate=30.

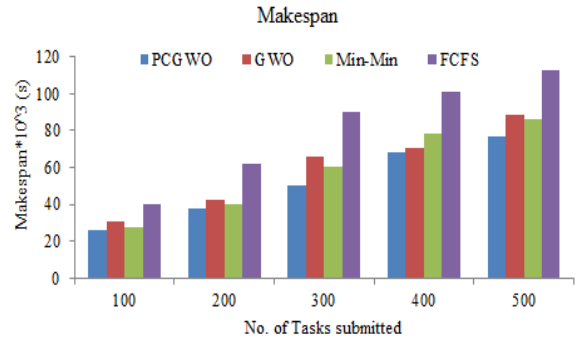


Figure 3. Makespan, the arrival rate=60.

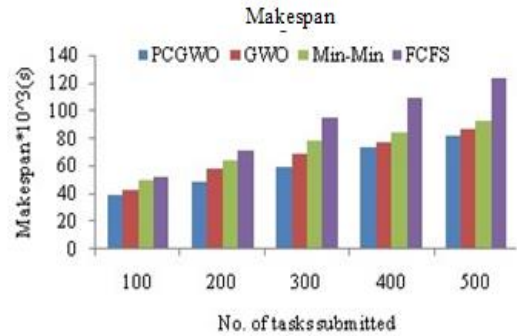


Figure 4. Makespan, the arrival rate=90.

**6.1.2. Comparison of Cost Varied Deadline**

The maximum cost is calculated based on the cost of the number of jobs that are completed within the deadline which is inclusive of the execution cost and the job transfer cost. The cost values are obtained for the four algorithms under observation, the PCGW, Min-Min, FCFS and the standard GWO by varying the deadline and are compared against the number of tasks to be executed. Figures 5, 6, and 7 shows the cost comparison for four different methods considering the different deadline values for 100, 300 and 500 tasks.

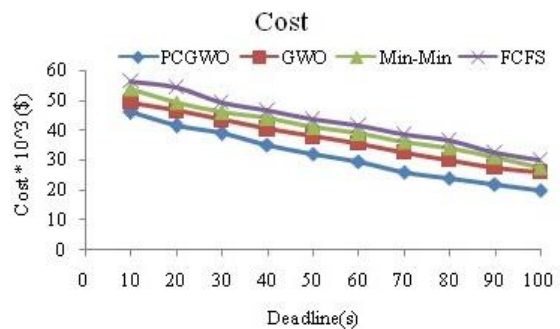


Figure 5. Costs with different deadline at task=100.

From the results it can be found that the PCGW algorithm continues to show reduced cost among all the algorithms even when the tasks increase. When we consider the values obtained when 90 resources are allocated for execution, the cost of standard GWO increases 24% as the number of tasks increase by 100, Min-Min shows an increase by 22%, FCFS shows an increase by 26% whereas PCGW shows an increase



of only 20%. Thus, the resulting values clearly show the dominance over the other algorithms.

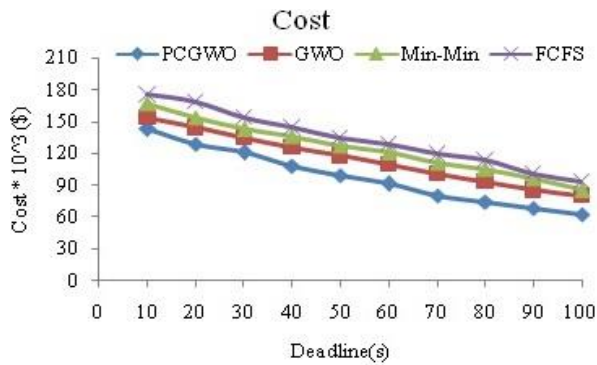


Figure 6. Costs with different deadline at task=300.

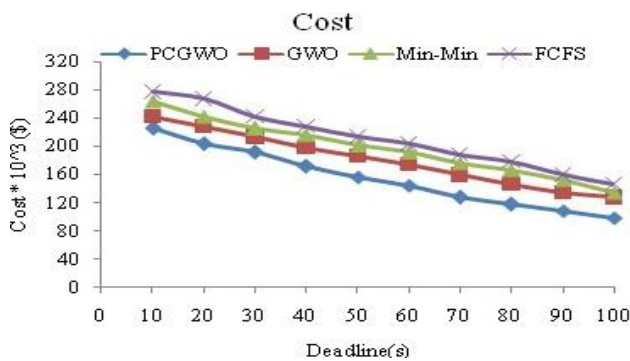


Figure 7. Costs with different deadline at task=500.

### 6.1.3. Comparison of Number of Tasks Meeting the Deadline

The number of tasks that are completed within the given deadline is analyzed. While the other algorithms do not work towards increasing the number of tasks that meet the deadline, the PCGWO algorithm has the deadline factor as one of its major objective. An increase in the number of tasks meeting the deadline is considered as a prime factor of customer satisfaction. The results are drawn on the number of tasks falling within the deadline versus the number of tasks allocated and are shown in Figures 8, 9, and 10.

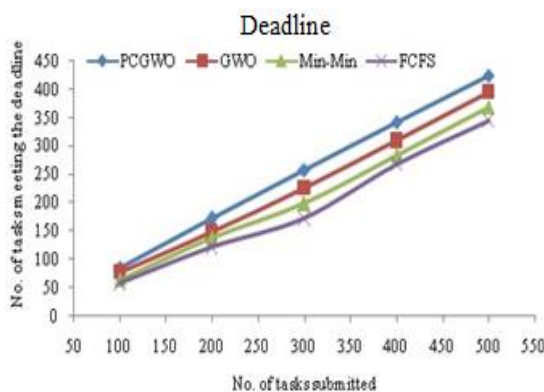


Figure 8. Comparison for no. of tasks meeting deadline, the arrival rate=25.



Figure 9. Comparison for no. of tasks meeting deadline, the arrival rate = 50.

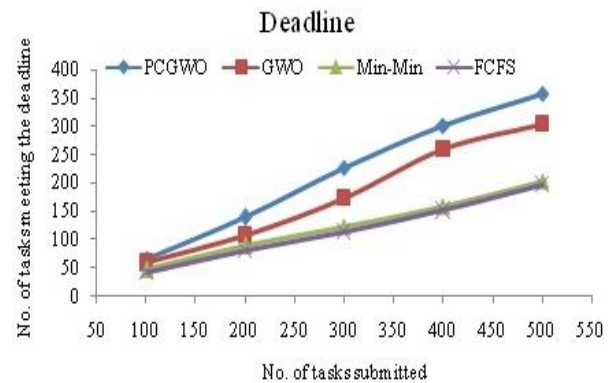


Figure 10. Comparison for no. of tasks meeting deadline, the arrival rate=75.

The average number of tasks meeting the deadline when scheduled by the standard GWO algorithm is 58% while Min-min completes 45% of its complete tasks whereas FCFS algorithm shows 43% completion. However, the PCGWO algorithm has a supreme completion percentage of 67% on an average. Thus it is proved that PCGWO algorithm is 3.5 times better than FCFS, 1.3 times better than standard GWO and is 3.2 times better than Min-min algorithm.

## 7. Conclusions

In this paper the multi-objective optimization is used in order to enhance the performance of the scheduling when compared to the single-objective function. The experimental outcome took based on an example by processing 100-500 tasks and proves that the proposed multi-objective based task scheduling is superior to the other existing approaches. The above study presents a Performance-Cost Grey Wolf Optimization algorithm to obtain nearest optimal schedule of tasks in terms of makespan, cost and the maximum number of task completion within the deadline. The simulation result shows that the PCGWO outperformed in terms of QoS (makespan, cost and the Maximum number of task completion within the deadline) when compared to other techniques. In future work, the hybridized PCGWO algorithm can be used and it is compared to other existing meta-heuristic techniques.

## References

- [1] Abdullahi M. and Ngadi M., "Hybrid Symbiotic Organisms Search Optimization Algorithm for Scheduling of Tasks on Cloud Computing Environment," *PloS one*, vol. 11, no. 6, pp. 1-29, 2016.
- [2] Abdullahi M., Ngadi M., and Abdulhamid S., "Symbiotic Organism Search Optimization Based Task Scheduling in Cloud Computing Environment," *Future Generation Computer Systems*, vol. 56, pp. 640-650, 2016.
- [3] Abramson D., Buyya R., and Giddy J., "A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker," *Future Generation Computer Systems*, vol. 18, no. 8, pp. 1061-1074, 2002.
- [4] Babaeizadeh S. and Ahmad R., "Enhanced Constrained Artificial Bee Colony Algorithm for Optimization Problems," *The International Arab Journal of Information Technology*, vol. 14, no. 2, pp. 246-253, 2017.
- [5] Buyya R., Yeo C., Venugopal S., Broberg J., and Brandic I., "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5<sup>th</sup> Utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599-616, 2009.
- [6] Carretero J., Xhafa F., and Abraham A., "Genetic Algorithm Based Schedulers for Grid Computing Systems," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 6, pp. 1-19, 2007.
- [7] Cheng M. and Prayogo D., "Symbiotic Organisms Search: A New Metaheuristic Optimization Algorithm," *Computers and Structures*, vol. 139, pp. 98-112, 2014.
- [8] Dorigo M., Birattari M., and Stutzle T., "Ant Colony Optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, 2006.
- [9] Ergu D., Kou G., Peng Y., Shi Y., and Shi Y., "The Analytic Hierarchy Process: Task Scheduling and Resource Allocation in Cloud Computing Environment," *The Journal of Supercomputing*, vol. 64, no. 3, pp. 835-848, 2013.
- [10] Grandinetti L., Pisacane O., and Sheikhalishahi M., "An Approximate  $\epsilon$ -Constraint Method for a Multi-Objective Job Scheduling in The Cloud," *Future Generation Computer Systems*, vol. 29, no. 8, pp. 1901-1908, 2013.
- [11] Jiao H., Zhang J., Li J., Shi J., and Li J., "Immune Optimization of Task Scheduling on Multidimensional Qos Constraints," *Cluster Computing*, vol. 18, no. 2, pp. 909-918, 2015.
- [12] Kalra M. and Singh S., "A Review of Metaheuristic Scheduling Techniques in Cloud Computing," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 275-295, 2015.
- [13] Liu Y., Zhang C., Li B., and Niu J., "DeMS: A Hybrid Scheme of Task Scheduling and Load Balancing in Computing Clusters," *Journal of Network and Computer Applications*, vol. 83, no. 1, pp. 213-220, 2017.
- [14] Mirjalili S., Saremi S., Mirjalili S., and Coelho L., "Multi-objective Grey Wolf Optimizer: A Novel Algorithm for Multi-Criterion Optimization," *Expert Systems with Applications*, vol. 47, pp. 106-119, 2016.
- [15] Natesan G. and Chokkalingam A., "Opposition Learning-Based Grey Wolf Optimizer Algorithm for Parallel Machine Scheduling in Cloud Environment," *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 1, pp. 186-195, 2017.
- [16] Pandey S., Wu L., Guru S., and Buyya R., "A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," in *Proceedings of 24<sup>th</sup> IEEE International Conference on Advanced Information Networking and Applications*, Perth, pp. 400-407, 2010.
- [17] Selvarani S. and Sadhasivam G., "An intelligent Water Drop Algorithm for Optimizing Task Scheduling in Grid Environment," *The International Arab Journal of Information Technology*, vol. 13, no. 6, pp. 627-634, 2016.
- [18] Somasundaram T. and Govindarajan K., "CLOUDRB: A Framework for Scheduling and Managing High-Performance Computing Applications in Science Cloud," *Future Generation Computer Systems*, vol. 34, pp. 47-65, 2014.
- [19] Song B., Hassan M., and Huh E., "A Novel Heuristic-Based Task Selection and Allocation Framework in Dynamic Collaborative Cloud Service Platform," in *Proceeding of 2<sup>nd</sup> IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, Indianapolis, pp. 360-367, 2010.
- [20] Tsai J., Fang J., and Chou J., "Optimized Task Scheduling and Resource Allocation on Cloud Computing Environment Using Improved Differential Evolution Algorithm," *Computers and Operations Research*, vol. 40, no. 12, pp. 3045-3055, 2013.
- [21] Verma A. and Kaushal S., "Deadline Constraint Heuristic-Based Genetic Algorithm for Workflow Scheduling in Cloud," *International Journal of Grid and Utility Computing*, vol. 5, no. 2, pp. 96-106, 2014.
- [22] Wei G., Vasilakos A., Zheng Y., and Xiong N., "A Game-Theoretic Method of Fair Resource Allocation for Cloud Computing Services," *The*



- Journal of Supercomputing*, vol. 54, no. 2, pp. 252-269, 2010.
- [23] Xu B., Zhao C., Hu E., and Hu B., "Job Scheduling Algorithm Based on Berger Model in Cloud Environment," *Advances in Engineering Software*, vol. 42, no. 7, pp. 419-425, 2011.
- [24] Xu J., Lam A., and Li V., "Chemical Reaction Optimization for Task Scheduling in Grid Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1624-1631, 2011.
- [25] Yao G., Ding Y., Jin Y., and Hao K., "Endocrine-Based Coevolutionary Multi-Swarm for Multi-Objective Workflow Scheduling in A Cloud System," *Soft Computing*, vol. 21, no. 15, pp. 4309-4322, 2017.
- [26] Yu J. and Buyya R., "Scheduling Scientific Workflow Applications with Deadline and Budget Constraints Using Genetic Algorithms," *Scientific Programming*, vol. 14, no. 3-4, pp. 217-230, 2006.
- [27] Zhang Q., Cheng L., and Boutaba R., "Cloud Computing: State-Of-The-Art and Research Challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7-18, 2010.
- [28] Zhong H., Tao K., and Zhang X., "An Approach To Optimized Resource Scheduling Algorithm for Open-Source Cloud Systems," in *Proceedings of 5<sup>th</sup> Annual IEEE ChinaGrid Conference*, Guangzhou, pp. 124-129, 2010.
- [29] Zuo L., Shu L., Dong S., Zhu C., and Hara T., "A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing," *IEEE Access*, vol. 3, pp. 2687-2699, 2015.
- [30] Zuo X., Zhang G., and Tan W., "Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 564-573, 2014.



**Gobalakrishnan Natesan** pursued his Bachelor's degree in Information Technology at Anna University, Tamilnadu, India in 2005. Then he obtained his Master's degree in Software Engineering from Bharathidhasan University, Tamilnadu, India in 2008. Currently, he is a research scholar in Sathyabama University Chennai, India and he is an Assistant Professor in the Department of Information Technology, St.Joseph's College of Engineering, Tamilnadu, India. His current research interests are Cloud computing, Virtualization and Big Data.



**Arun Chokkalingam** received his Ph.D. from Anna University, Chennai, India. Currently, he is working as a Professor in the Department of Electronics and Communication Engineering, R.M.K College of Engineering and Technology, Tamilnadu, India. He has more than 15 years of teaching experience in Engineering College. He has published many papers in International Conferences, National & International Journal in areas such as cloud computing, VLSI, image processing and wireless sensor. His current research interests are Cloud Computing, VLSI and Optimization Techniques. He is a life time member of ISTE.