GWOTS: Grey Wolf Optimization based Task Scheduling at the Green Cloud Data Center

Natesha B V, Neeraj Kumar Sharma, Shridhar Domanal, Ram Mohana Reddy Guddeti
Department of Information Technology, National Institute of Technology Karnataka
Surathkal, Mangalore, India
Email:{nateshbv18,neeraj16ks,shridhar.domanal, profgrmreddy}@gmail.com

***Abstract*—Task Scheduling is a key challenging issue of Infrastructure as a Service (IaaS) based cloud data center and it is well-known NP-complete problem. As the number of users' requests increases then the load on the cloud data center will also increase gradually. To manage the heavy load on the cloud data center, in this paper, we propose multi-objective Grey Wolf Optimization (GWO) technique for task scheduling. The main objective of our proposed GWO based scheduling algorithm is to achieve optimum utilization of cloud resources for reducing both the energy consumption of the data center and total makespan of the scheduler for the given list of tasks while providing the services as requested by the users. Our proposed scheduling algorithm is compared with non meta-heuristic algorithms (First-Come-First-Serve (FCFS) and Modified Throttle (MT)), and meta-heuristic algorithms (Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Cat Swarm Optimization (CSO)). Experimental results demonstrate that the proposed GWO based scheduler outperforms all algorithms considered for performance evaluation in terms of makespan for the list of tasks, resource utilization and energy consumption.**

***Keywords*-Cloud Computing, Task Scheduling, Resource Utilization, Grey Wolf Optimizer**

I. Introduction

Cloud computing is referred to as a computing infrastructure for providing the services to the user on-demand basis. IaaS is a pay-as-you-go model of cloud computing. Using virtualization techniques the virtual machines (VMs) are created and deployed over physical machines (PMs) through which the cloud services are provided to the users. The cloud data center consists of the computing resources such as the *RAM, CPU*, Storage and I/O devices. Further, the cloud data center hosts some tasks on PMs and thus provides the service to the end user. In the cloud aspect, the assignment of tasks into VMs in a cloud data center is decided by a scheduler. The resources in a cloud can be used efficiently by making use of better scheduling algorithm.

The challenges in the cloud computing environment are:

- The allocation of the resources for incoming tasks.
- Efficient utilization of the resources, and
- Providing the quality service (QoS) to the user without violating the service level agreement (SLA).

In the cloud data center, the allocation of incoming tasks to VMs is known as a NP-complete problem. As the number of requests for the service increases, the load on the data center also increases thus it requires more number of computing resources (VMs).

The main objective of the scheduling algorithm is to appropriately allocate the task to these VMs such that the key objectives such as makespan, resource utilization etc. are optimized. Hence, we need an efficient scheduler for allocating the tasks to the VMs such that the time taken to provide the service is minimized. This motivated us to design and develop an efficient task scheduling algorithm to allocate the tasks at the cloud data center to increase the resource utilization [1] and thus minimizing the makespan of the tasks allocated to the VMs [2].

When more number of PMs exist in the cloud data center, then the allocation of tasks among the existing VMs at the cloud data center should be done efficiently while utilizing the cloud resources efficiently. The core idea of our proposed GWO [3] based algorithm is to achieve efficient task scheduling such that the total makespan for the list of tasks can be minimized, while maximizing the resource utilization of VMs in a cloud data center. GWO is a meta heuristic algorithm which can be used for solving the NP-hard and NP-complete problems. The capability of exploring and exploiting the global optimal solution is better in GWO and thus it not only reduces the search time but also achieves the faster convergence rate for the optimal solution as compared to other swarm intelligence techniques such as CSO, GA, PSO etc. [3]. This motivated us to use GWO for solving the task scheduling problem in the cloud computing environment. GWO is based on the leadership and hunting procedure of the grey wolves in nature and more details of GWO are discussed in the following section.

The main contributions of our proposed work are:

- Efficient scheduling of tasks among VMs for providing the service to the user by reducing the total makespan for the tasks at the cloud data center.
- Energy efficient utilization of cloud resources (VMs) at the cloud data center.

The rest of this paper is organized as follows. Section II deals with Related Work; Section III describes the Proposed Methodology, Section IV discusses the Performance Evaluation and finally Section V concludes the paper with future direction.

978-1-7281-0441-6/18/$31.00 ©2018 IEEE
DOI 10.1109/SKG.2018.00034

Table I
SUMMARY OF KEY EXISTING WORKS

Author	Methodology	Remarks
Haitao Hu, Hongyan Wang [4]	Prediction based Ant Colony Optimization for Task Scheduling	Used for optimizing the resource utilization cost. Not considered the makespan of the schedule
Jin, H.Z. et al. [5]	Genetic Algorithm based Task Scheduling	Handles Scheduling Criteria but selection time can be huge and no uniqueness
Ma, L. et al. [6]	Greedy Task Scheduling	Good Energy Efficiency with SLA deadline but response time is more
Hongjia Li et al. [7]	Fast Energy Aware Resource Provisioning and Task Scheduling Algorithm	Used for optimizing the total energy cost by reducing the response time of VMs but resource utilization is not considered
Neeraj Kumar Sharma, and G. Ram Mohana Reddy [8]	HGAPSO based multi-objective VMs alloca -tion in cloud data center.	Designed HGAPSO based hybrid algorithm for VMs allocation on PMs to maximize the resource utilization by reducing the energy consumption in a cloud data center.
Zhu, X. et al. [9]	Framework for Task Scheduling	Good Energy and Cost Efficiency but with underutilized resources and more response time
Zhan, S. and Huo, H. [10]	Improved PSO based Task Scheduling	Did not Consider the utilization of resources
Ghribi, C. et al. [11]	Exact Allocation and Migration Algorithms	Efficient resource utilization but the response time is high due to migration process

II. RELATED WORK

Xiaodong Sheng and Qiang Li [12] proposed a template based Genetic Algorithm for the QoS aware task scheduling in the cloud environment. The template is expressed in terms of maximum size of the task to be allocated for the VM. Then Genetic Algorithm is used for allocating the task with multiple subset of these templates. But their approach takes more time to decide subset of template to be run by the VMs and further they did not consider the resource utilization.

Ma, L et al. [6] designed a greedy approach of task scheduling for reducing the energy consumption of data center. The key merits of this approach are: (i) it offers good energy efficiency and (ii) it takes care of the SLA between the cloud service provider and the user. But the main demerit of this approach is that its response time is compromised to a very high level. Zhan, S. and Huo H [10] designed an optimal task scheduling for a cloud data center by using an improved PSO technique. This approach considers the makespan of the VMs in a data center but the problem with this method is that the makespan of a task is compromised and the cloud resources are underutilized.

Tan, Y.M. et al. [13] developed a standard stochastic model which uses a type of queuing system for the tasks. The tasks in the queue are assigned to the resources (VMs) of a PM of cloud data center. The main advantage of this model is that it considers the power consumption along with the scheduling strategy. But the key disadvantages are: (i) there is no uniqueness of the task performed and (ii) the selection process of each task can take a huge amount of time. Said El Kafhali and Khaled Salah [14] proposed Queuing model for analyzing the QoS of the Cloud data center. Their model is also used for estimating the total energy consumption in the cloud data center using the Markov chain method

Ruonan Lin and Qiang Li [15] proposed a task scheduling algorithm based on pre-allocation strategy. In this approach the template size for each processor is calculated by dividing the tasks into subtasks, then scheduling of these sub-tasks is done using preallocation Ant Colony Optimization. But the problem with this approach is that it takes more time and further the resource utilization of data center is not considered.

Liu, N et al. [16] developed a novel task scheduling approach by considering the power of components in the cloud infrastructure. The main advantages of their approach are: it offers good energy efficiency and it is also very cost efficient. But the main disadvantages of this approach are that it does not consider the resource utilization and compromises with total response time for the completion of the tasks.

Fatima AlQayedi et al. [17] proposed adaptive cloud resource allocation scheme to minimize the response time. Their scheme makes use of the queuing theory for predicting the number of VMs required to satisfy the service level objective and further minimize the response time based on the current workload. Fayyaz, A. et al. [18] developed an energy efficient dynamic resource scheduling through VMs consolidation in cloud computing environment. The resources are utilized efficiently but this approach consumes more time for consolidating the VMs in the data center hence the total service time is more. Howraa M. Mohammad Ali et al. [19] proposed energy efficient resource provisioning and VMs migration for disaggregated servers. Their approach was based on considering the heuristic methods for resources provisioning and migrating the under loaded VMs to the energy efficient VMs. But they did not consider the scheduling of tasks. Neeraj Kumar Sharma, and G. Ram Mohana Reddy [8] proposed a multi-objective energy efficient virtual machine allocation in both homogeneous and heterogeneous cloud data center using hybrid approach referred to as HGAPSO (combining GA with PSO). Their approach was based on energy efficient resource allocation and VMs migration in a cloud data center.

Yang Jun et al. [20] proposed a deadline constrained energy-aware task scheduling in the cloud environment. In this approach the tasks were allocated by defining the urgency level to satisfy the SLA of the cloud end user. Based on the urgency level, scheduling was done to reduce the energy consumption of the devices. But the problem with this approach is that it is time consuming and the resources are underutilized.

Some key existing works are summarized in Table I.

III. Proposed Methodology

A. Task Scheduling in the Cloud Data Center

The scheduling indicates the procedure of assigning the tasks to the VMs which are hosted on PMs of a cloud data center. The scheduler which accepts the incoming tasks and allocates the available resources to the incoming cloudlets using GWO based scheduling algorithm. This scheduling procedure is used to allocate the tasks to the VMs such that the resources of a cloud data center can be used efficiently by reducing the total makespan time and also reduces the total energy consumption by optimal utilization of the resources in the cloud data center. The power consumption P_d at the cloud data center is defined as follows.

$$P_d = u * P_{max} + (1 - u) * P_{max} * m \tag{1}$$

Where, P_{max} is the maximum power consumed and CPU utilization is a function of time and is represented as m(t) and *u* is the fraction of power consumed by the devices in idle state, and *(1-u)* represents the power consumed by the active devices.

$$E_c = \int_{t_0}^{t_1} P_d(m(t))dt \tag{2}$$

The energy consumption of PMs *(E_c)* in the data center for the time duration t_0 to t_1 on variable workload length is given by Eq. (2), and the overall energy consumption (T_{ec}) in cloud data center is given by Eq. (3).

$$T_{ec} = min \sum_{i=1}^{n} E_c, \quad for \quad i = 1, 2, ...n\ PMs \tag{3}$$

The resources in the data center should be allocated and utilized efficiently to reduce the total energy consumption. The allocation of tasks to the VMs should satisfy the following available resource constraints,

$$\sum_{i=1}^{n} t_{CPU}^{req} * A_{ij} <= T_{CPU}^{VM} \tag{4}$$

$$\sum_{i=1}^{n} t_{RAM}^{req} * A_{ij} <= T_{RAM}^{VM} \tag{5}$$

$$\sum_{i=1}^{n} t_{BW}^{req} * A_{ij} <= T_{BW}^{VM} \tag{6}$$

The t^{req} is the resources requirement for each assigned *n* tasks and it should not exceed the total available resources *T* {*CPU, RAM, BW*} of VMs in a cloud data center. A_{ij} is equal to 1 if the $task_i$ allocated to the VM_j else it is 0 as defined by Eq. (7). Efficient allocation of these available resources will reduce both total makespan and the energy consumed in the cloud data center.

$$A_{ij} = \begin{cases} 1, & \text{if task } t_i \text{ is placed in } VM_j \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

B. Basics of Grey-Wolf Optimizer

Grey-wolf optimizer is a meta-heuristic algorithm based on the life cycle of the grey wolves in nature [3]. They belong to the Canidae family where they will search for the prey in nature and attacks the prey for the food. Here the pack of grey wolves is grouped into the different social hierarchy and among them, one group will act as the leader of the pack where rest of the pack will follow the group leader along with the social hierarchy of grey wolves. Group hunting is also a behaviour of the wolves.

The social hierarchy of these wolves is classified into four groups namely: alpha, beta, delta and omega. Alpha wolf will be the leader and the decision maker in the group for searching the prey and attacking the prey. Beta wolf group is the advisor to the alpha group and responsible for maintaining the discipline in the group. Delta is the care taker and watches the territory of the wolves. Omega wolves in the group act as a supporter to the alpha for finding and attacking the prey and they also obey the order of other groups. The social hierarchy of Grey Wolves is shown in Fig. 1. The important feature of the Grey-Wolves is that they maintain the social hierarchy and stability very strictly in a group and mutually coordinate with each other while searching and hunting the prey.

The following methods are involved in the design of Grey Wolf algorithm for task scheduling.

Search for a Prey: Let us consider the value ***a***, which indicates the convergence and divergence of the wolves with respect to the position of the prey while searching for the

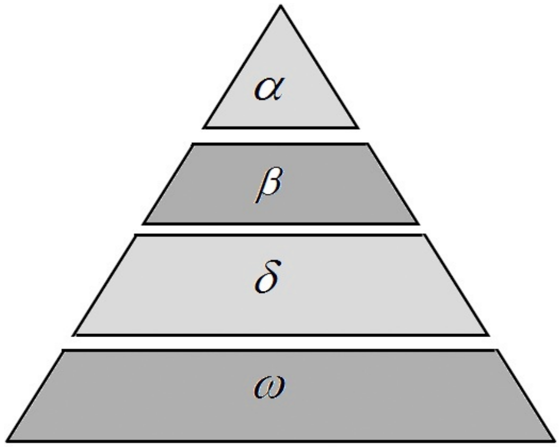

Figure 1. Social hierarchy of Grey wolves

prey. All the wolves diverge themselves and search for the prey independently. Based on the value of ***a***, the wolves will keep updating the positions for each iteration. During the first half of the iterations, wolves will diverge from the prey when the value of ***a*** is greater than 1 and during the next half of the iterations, wolves will converge towards the prey when the value of ***a*** is less than 1.

Encircle and Hunt the Prey: In each move, the wolves will move towards the prey and encircle the prey such that it cannot move further. Grey wolves will be able to identify the location of the prey and encircle them, once the prey is located in a particular position Grey-wolves can attack the prey lead by the group leader (alpha). The social hierarchy of the wolves will help to store the previous optimal solution. Using this social hierarchy in each move, the position of these wolves can be updated and at the end of the procedure the alpha value will be considered for the scheduling process. The social hierarchy of these grey wolves will save the best solutions in each iteration and helps to find the exact locations of the prey. Hence the capabilities of the exploration and exploitation for the global optimum solution are superior in GWO as compared to the other meta-heuristic algorithms [3].

C. Fitness Function

The size of the incoming cloudlet can be a deciding factor. Cloudlet which is larger in size require more MIPS, more CPU execution time, and therefore more energy is consumed by the data center during the execution. Hence we conclude that the execution time is directly proportional to the length of the cloudlet. Let f be the fitness function value of VMs then the fitness function designed for the task scheduling is defined by the Eq. (8).

$$f_j = \sum_{i=1}^{m} \frac{Length_i * PE_i}{MIPS_j * PE_j}, \quad for \quad j = 1, 2, ...n\ VMs \tag{8}$$

$$F = Min(f_j) \tag{9}$$

Where, Length_i is the length of cloudlet *i*; PE_i is the number of Processing Elements (PE) requested by cloudlet *i*; MIPS_j (millions of instruction per second) is the MIPS allocated to VM_j; PE_i is the number of PE in VM_j and *n* is the number of VMs in a data center. Here, the value of fitness function will be greater when the execution time of the data center is more. Hence, we need to choose the solution schedule such that the *F* value is minimized using Eq. (9) and thus reducing the total time for providing the services to the users. If the length of the incoming cloudlet is more, then the time taken for completing this task will be high. In VMs, if MIPS is less, then it consumes more time for completing the task. Hence the number of PE and MIPS of VMs are considered such that the time for that task can be reduced. Here the value of the *F* will indicate the total minimum time consumed by all the VMs for completing the allocated task in a cloud data center. So by reducing the value of *F* for all the VMs in a data center we can minimize the make span for a given set of tasks.

D. Algorithm and Flowchart

Algorithm 1 describes the efficient task scheduling using GWO technique. The list of cloudlets and the list of VMs are given as the input and this scheduling algorithm provides the best schedule as output, where the time taken for completing all the tasks is the lowest. Steps 1 to 4 describe the required parameters for the initialization of scheduling algorithm. Initially schedule all the incoming cloudlets (tasks) to VMs and based on the fitness value the list of VMs are grouped into alpha, beta and delta. Steps 5 to 11 describe iterations over the value of a. To find the fitness of each schedule and in each iteration the values of alpha, beta, delta are updated. Once the value of ***a*** becomes zero, the iteration will stop and Step 13 returns the value of alpha as the best schedule.

The flowchart for the GWO based task scheduling algorithm is depicted in Fig. 2. Initially the number of cloudlets and the list of VMs are given as an input for the algorithm. In Step 1, the parameters such as ***a*** and t_{max} are initialized. In Step 2, the initial schedule is generated. Using this schedule, we calculated the fitness function of the VMs using the Eq. (8). In Step 3, VMs are grouped into alpha, beta, gamma group of wolves, where alpha will be the least *F* value, and beta is greater than alpha and less than other wolves and delta is greater than beta and less than other wolves and rest will be considered as omega wolves. Then the values of ***a*** and t_{max} are varied in each loop to identify the better schedule. By updating the values of wolves group in each

Algorithm 1 GWO based Task Scheduling

Input: Set of virtual machines $VM = v_1, v_2, . . , v_n$
Set of tasks (cloudlets) C= $c_1, c_2, . . . , c_m$

Output: Schedule for the list of incoming tasks.

Initialize a $\leftarrow$ 2 $t_{max} \leftarrow numberofiterations$
Find the initial schedule for the given cloudlets
Find the fitness for all the allocated virtual machines
Group virtual machines in to α, β, and δ
while a > 0 **do**
  for t=0 to t_{max} **do**
    Compute the fitness for all the schedule using Eq. (8).
    Update the values of α, β and δ
    t $\leftarrow t + 1$
  end for
  update **a** using Eq.(10).
end while
return α

iteration, the value of ***a*** is continuously decreasing from 2 to 0.

$$\boldsymbol{a} = 2 - \frac{2 * t}{t_{max}} \tag{10}$$

When the value of ***a*** becomes zero then value of alpha will be returned as the best value and it will be considered as the schedule for allocating tasks to the VMs.

E. Time Complexity

The time complexity for the GWO based scheduling algorithm depends on the number of VMs *(n)* and the number of incoming cloudlets *(m)*, the grouping of VMs takes *O(n)* and scheduling of cloudlets to VM is *O(mn)*. Hence, the overall time complexity of our proposed GWO based task scheduling algorithm is *O(mn)+O(n) = O(mn)*.

IV. Performance Evaluation

In this section, we described the experimental setup used for evaluating the performance of the proposed task scheduling algorithm and accordingly the results are analyzed. Here we considered independent VMs for all incoming cloudlets.

Table II
DATA CENTER PARAMETERS FOR SIMULATION

Number of PMs	1-10
Number of VMs	10-150
VMs MIPS	500-2000
VMs RAM in MB	512-2048
Number of PEs	1-4

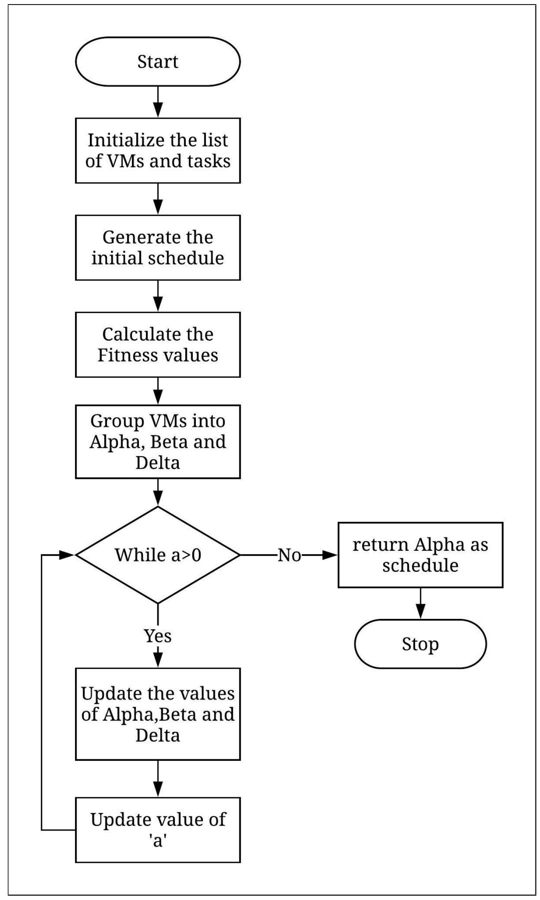

Figure 2. Flowchart for GWO based Scheduling Algorithm

A. Experimental Setup

We implemented all these algorithms in a Python Simulation environment. The cloud data center resource parameters are given in Table II. For implementing our proposed task scheduling algorithm, we considered heterogeneous VMs which are hosted in PMs and are capable of running the allocated tasks and provide the service.

B. Results and Analysis

For performance evaluation, we considered heterogeneous VMs which are hosted by PMs of cloud data center and we calculated the makespan for different types of cloud task requests which requires the variable CPU (MIPS) and other resources. These tasks are allocated to the VMs which satisfy the resource constraints for the tasks which are specified by Eqs. (4)-(6). Then total time taken for completing the task is recorded. We calculate the total makespan time for our proposed GWO based scheduling algorithm and compared the results with other state-of-the-art algorithms. Results confirm that GWO scheduler is more efficient than other algorithms in terms of makespan for completing the tasks and resources (VMs) utilization in the cloud data center.

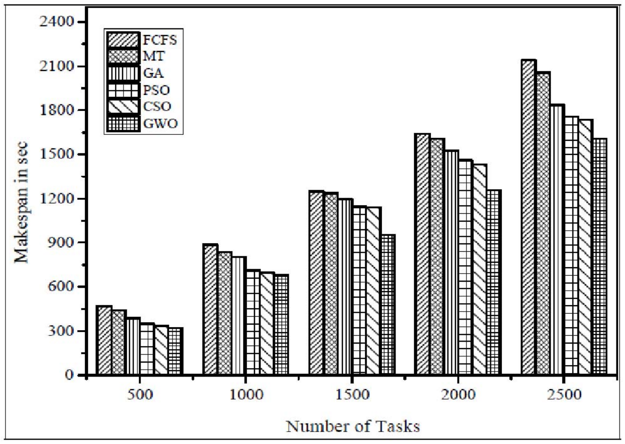

Figure 3. Makespan for Different Algorithms

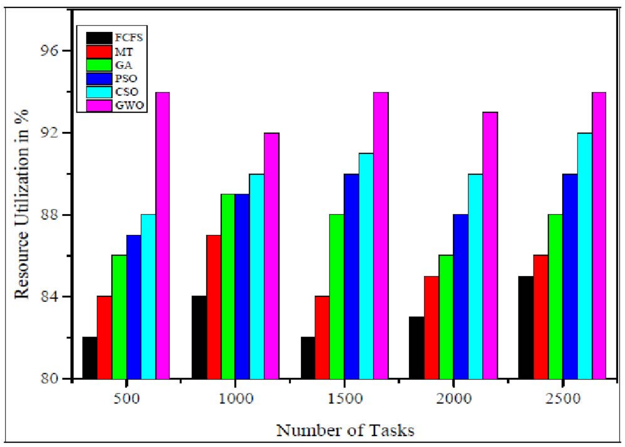

Figure 4. Utilization of Resources for Different Algorithms

The proposed GWO based scheduling algorithm is compared with the meta-heuristic algorithms (GA, PSO and CSO) and non meta-heuristic algorithms (FCFS and MT) for the different set of real time tasks and the makespan for the completion of these tasks is calculated for each algorithm considered for performance evaluation. The results depicted in Fig. 3 show that the proposed GWO task scheduler outperforms all other algorithms considered in terms of makespan. The GWO based scheduling is also considered for performance evaluation in terms of percentage $(\%)$ utilization of resources (VMs) at the cloud data center and it is observed from Fig. 4 that the GWO based scheduling algorithm is more efficient than all the other algorithms considered for performance evaluation. The percentage utilization of the resources (VMs) in the cloud data center of all algorithms considered for performance evaluation is shown in Fig. 4. Better utilization of resources will help to reduce the number of active devices in the data center and switching-off the idle devices will also reduce the total energy consumption at the

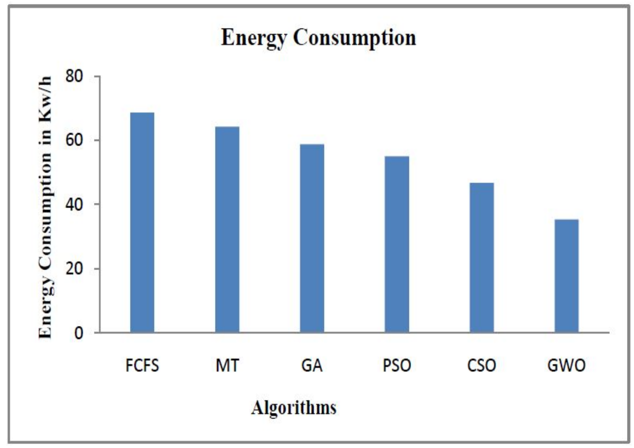

Figure 5. Energy Consumption of the Cloud Data Center

cloud data center. Fig. 5 shows that our proposed algorithm consumes less energy as compared to other algorithms considered for the performance evaluation.

V. Conclusion

The problem of task scheduling is addressed using the Grey-Wolf based task scheduling algorithm which is implemented in Python. Proposed scheduling algorithm not only reduces the time taken by the data center to complete the tasks but also improves the percentage utilization of cloud resources (VMs) (by 6 to 12%) as compared to both non meta-heuristic and meta-heuristic algorithms considered for performance evaluation. Thus GWOTS can be used for static as well as dynamic scheduling. In future GWOTS can be combined with other meta-heuristic algorithms to solve the problem of resource allocation/scheduling and also for energy optimization problems in a cloud data center environment.

Acknowledgment

This work has been supported by the Visvesvaraya Ph.D. Scheme for Electronics and IT (Media Lab Asia), the department of MeitY, Government of India. This work carried out at the Department of Information Technology, NITK Surathkal, Mangalore, India.

References

[1] M. Masdari, S. S. Nabavi, and V. Ahmadi, "An overview of virtual machine placement schemes in cloud computing," *Journal of Network and Computer Applications*, vol. 66, pp. 106–127, 2016.

[2] R. M. Guddeti, R. Buyya *et al.*, "A hybrid bio-inspired algorithm for scheduling and resource management in cloud environment," *IEEE Transactions on Services Computing*, 2017.

[3] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[4] H. Hu and H. Wang, "A prediction-based aco algorithm to dynamic tasks scheduling in cloud environment," in *Computer and Communications (ICCC), 2016 2nd IEEE International Conference on*. IEEE, 2016, pp. 2727–2732.

[5] H. Z. Jin, L. Yang, and O. Hao, "Scheduling strategy based on genetic algorithm for cloud computer energy optimization," in *Communication Problem-Solving (ICCP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 516–519.

[6] L. Ma, Y. Lu, F. Zhang, and S. Sun, "Dynamic task scheduling in cloud computing based on greedy strategy," in *International Conference on Trustworthy Computing and Services*. Springer, 2012, pp. 156–162.

[7] H. Li, J. Li, W. Yao, S. Nazarian, X. Lin, and Y. Wang, "Fast and energy-aware resource provisioning and task scheduling for cloud systems," in *Quality Electronic Design (ISQED), 2017 18th International Symposium on*. IEEE, 2017, pp. 174–179.

[8] N. Sharma and R. M. Guddeti, "Multi-objective energy efficient virtual machines allocation at the cloud data center," *IEEE Transactions on Services Computing*, 2016.

[9] X. Zhu, Y. Zha, L. Liu, and P. Jiao, "General framework for task scheduling and resource provisioning in cloud computing systems," in *Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual*, vol. 1. IEEE, 2016, pp. 664–673.

[10] S. Zhan and H. Huo, "Improved pso-based task scheduling algorithm in cloud computing," *Journal of Information & Computational Science*, vol. 9, no. 13, pp. 3821–3829, 2012.

[11] C. Ghribi, M. Hadji, and D. Zeghlache, "Energy efficient vm scheduling for cloud data centers: Exact allocation and migration algorithms," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*. IEEE, 2013, pp. 671–678.

[12] X. Sheng and Q. Li, "Template-based genetic algorithm for qos-aware task scheduling in cloud computing," in *Advanced Cloud and Big Data (CBD), 2016 International Conference on*. IEEE, 2016, pp. 25–30.

[13] Y.-m. Tan, G.-S. Zeng, and W. Wang, "Policy of energy optimal management for cloud computing platform with stochastic tasks," *Ruanjian Xuebao/Journal of Software*, vol. 23, no. 2, pp. 266–278, 2012.

[14] S. El Kafhali and K. Salah, "Modeling and analysis of performance and energy consumption in cloud data centers," *Arabian Journal for Science and Engineering*, pp. 1–14, 2018.

[15] R. Lin and Q. Li, "Task scheduling algorithm based on pre-allocation strategy in cloud computing," in *Cloud Computing and Big Data Analysis (ICCCBDA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 227–232.

[16] N. Liu, Z. Dong, and R. Rojas-Cessa, "Task scheduling and server provisioning for energy-efficient cloud-computing data centers," in *Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on*. IEEE, 2013, pp. 226–231.

[17] F. AlQayedi, K. Salah, and M. J. Zemerly, "Adaptive cloud resource allocation scheme to minimize slo response time violation," in *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of*. IEEE, 2016, pp. 1–5.

[18] A. Fayyaz, M. U. Khan, and S. U. Khan, "Energy efficient resource scheduling through vm consolidation in cloud computing," in *Frontiers of Information Technology (FIT), 2015 13th International Conference on*. IEEE, 2015, pp. 65–70.

[19] H. M. M. Ali, A. M. Al-Salim, A. Q. Lawey, T. El-Gorashi, and J. M. Elmirghani, "Energy efficient resource provisioning with vm migration heuristic for disaggregated server design," in *Transparent Optical Networks (ICTON), 2016 18th International Conference on*. IEEE, 2016, pp. 1–5.

[20] Y. Jun, M. Qingqiang, W. Song, L. Duanchao, H. Taigui, and D. Wanchun, "Energy-aware tasks scheduling with deadline-constrained in clouds," in *Advanced Cloud and Big Data (CBD), 2016 International Conference on*. IEEE, 2016, pp. 116–121.