ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Bi-objective Hybrid Particle Swarm Optimization & Ant Colony Optimization Workflow Scheduling Algorithm for Cloud

Ritu
Information Technology
UIET (Panjab University)
Chandigarh, India

Rajkumari
Information Technology
UIET (Panjab University)
Chandigarh, India

*Abstract:*Cloudcomputing is the latest distributed computing paradigm. The recent increased use of workflow management systems by large scientific collaborations presents the challenge of scheduling large-scale workflows onto distributed resources. As workflow scheduling belongs to the NP-complete problem, soto solve these problems meta-heuristic approaches are abetter option. But most of the existing studies try to optimize only one of the objectives. But the need of thehour is to focus on multiple-objective like time,cost, CPU utilization, Reliability and energy optimization etc.In this paper, our focus is on two objectives, makespan and cost, to be optimized simultaneously using two meta-heuristic search techniques PSO and ACO for scheduling workflow. To solve this bi-objective Time & Cost optimization workflow scheduling problem, wepresent, a hybrid of particle swarm optimization with cost function optimize using ant colony optimization. For the initialization of task to resources, we are using Pareto distribution (PD), a normal-likedistribution.Simulation result shows that hybridization of PSO and ACO performs better than the existing BPSO (HEFT+PSO) Technique.

Keywords: Cloud Computing, Scientific Workflow, Workflow Scheduling, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO)

I.

II. INTRODUCTION

Nowadays, Cloud Computing has become very popular in our day to day life.According to the NIST's definition, cloud computing is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort"[1]. The main promise that cloud computingfulfillsis that it provides computing on-demand, on the basis of pay-as-you-go paradigm. Cloud computing allows large-scale service sharing, which permits users to access technology-based services without any need of knowledge, expertise or control. The service-oriented nature of cloud computing makes it more interesting to general users. There are several services that provided by cloud like Xen [2], Amazon EC2 [3], IBM cloud [4] etc. Cloud Computing refers to leasing computing resources over the Internet. Advantages of using such a set up include minimized infrastructure cost, minimized overhead and pay only for the components used for the given amount of time. In providing Cloud Resources to the user, we need the benefit of the both CSP (Cloud Service Provider) and the CU (Cloud User). So the scheduling of the resources should be in an efficient way so that both CSP and CU can be benefitted. Scheduling of the task on the cloud depends on the organization of the task. If there is a dependency between the task, then a task only can be executed if all of its predecessors are already executed and we have their result in advance. On the other hand, if the task is independent of each other, then they can be executed in any order. The former is known as dependent scheduling & the later is known as independent scheduling. The dependent scheduling is also known as workflow scheduling. Workflow is represented as DAG(Direct Acyclic Graph) $G =< T, E >$ where T(nodes) denotes the number of task and E (edges) denotes the dependency between these tasks.Many metaheuristic scheduling algorithm is available to schedule the task on the available resources but very few people consider the large scale scientific applications i.e., often expressed as workflow.So our main goal of this paper is to efficiently allocate the task to the resources to get good performance on large scale scientific workflow. It is a well-known NP-Complete problem to schedule the workflow task on set of resources. Generally, this problem of dependent task scheduling can be represented as a Direct Acyclic Graph (DAG). In DAG, the nodes represent tasks and edges represent inter-task dependencies. The objective function is mapping the tasks onto the virtual machines (vms) such that task-precedence is satisfied and the overall time and cost can be optimized. Workflow scheduling in cloud requires that both time and cost constrained should satisfy [5].Our work is based on the two metaheuristic optimization technique, PSO and ACO. Initialization of the task is done using Parito Distribution [6].This Paper is organized as follows. Section II provides the Literature Review. Problem Statement is presented in Section III. Section IV provides the Algorithm and Flowchart of the hybridization of PSO-ACO respectively. Section V shows the Simulation and Results. Finally, Conclusion is drawn in Section VI. Section VII represents the References.

III. LITERATURE REVIEW

InReference [7],authors discussed briefly the CloudSim that is a simulation toolkit to test the performance of service delivery models without the real deployment of that model by saving large costs. It also provides the modeling and simulation of cloud environment, provisioning of VM allocation policy, Bandwidth allocation policy and many other services.

InReference [8],authors briefly discussed the concept of task scheduling. Tasks Scheduling is considered to be based on different user's perspective like total execution time or task execution time. In this paper, they proposed a framework to satisfy these user requirements based upon the grouping of tasks. Greedy approach is followed to select the best available resources by satisfying the task constraints. Simulation results prove that the proposed work improves the task execution cost and execution time parameter over the sequential assignment of tasks. For the future work, they suggested grouping the cost based tasks before allocating them to resources to minimize the communication overhead.

InReference [9], as the task assignment in cloud computing, is considered to NP-hard problem so there many meta-heuristic techniques. This paper has done a brief survey on the various meta-heuristic techniques used in cloud computing like Ant Colony Optimization, Particle Swarm Optimization, Genetic Algorithm, and League Championship Algorithm (LCA). Comparative analysis is done on the basis of nature of tasks, optimization criteria, improving meta-heuristic technique etc.

InReference [10], a brief introduction to Cloud Computing is given. Tasks of Cloud Computing can be divided into two types of task, independent task and interrelated task. The objective of this paper is how to do load balancing on VMs (Virtual Machines) for the independent task so that we can minimize the execution time of the task and maximize the VMs resource utilization. To fulfill the purpose an improved PSO (Particle Swarm Optimization) based algorithm is proposed. A mutation operator and self-adaptation of inertia weight to the standard PSO algorithm is introduced to improve standard PSO.c1,c2 coefficients are set as 2.05.Here, inertia weight is variable based on some equation.

InReference[11], the benefit of availability of scientific workflow is briefly described.The paper provides detailed characterizations of five scientific workflows which include massively parallel workflows that process large amounts of data, pipelined application that split up input datasets and operate on different chunks in parallel. It describes basic workflows provide a detailed characterization of five scientific workflows that include workflows from gravitational physics, earthquake science, biology, and astronomy. Finally, it described workflow generator to create synthetic workflow similar to the workflow that is characterized in this paper.

In Reference[12], a taxonomy that characterizes and classifies various approaches for building and executing workflows on Grids is proposed. The taxonomy not only highlights the design and engineering similarities and differences of state-of-the-art in Grid workflow systems but also identifies the areas that need further research. The taxonomy focuses on workflow design, workflow scheduling, fault management and data movement. This paper thus helps to understand key workflow management approaches and identify possiblefuture enhancements.

InReference[13],authors briefly discussed grid computing and how to schedule workflow in a gridenvironment. The main target of the paper is the minimization of execution cost while meeting the deadline of the workflow application. It proposed Knowledge-Based Ant Colony Optimization (KBACO) algorithm for grid workflow to minimize the execution cost while meeting the deadline. KBACO model integrates the ACO model with knowledge model. It compares the result of KBACO with ACO to show the effectiveness of theproposed algorithm.

In Reference[14], the wide use of scientific workflow in different areas like astronomy, seismology, genomics etc. is described. Scheduling of scientific workflow is a challenging for large-scale workflow that has many jobs and data dependency. Among the solution of the problem workflow partitioning is an approach to divide the workflow into sub workflow and then submit this workflow into different execution sites. So the main aim of this paper is to partition the large-scale scientific workflows in conjunction with resource provisioning to reduce the workflow makespan and resource cost. To fulfill this purpose, authors first uses a heuristic to partition the workflow and results show that the partitioning into 2-3 sub workflow offers the best balance between communication and computation cost. The further genetic algorithm is used to address the problem of integration of workflow and provisioning of resources.

In Reference [15], a resource provisioning and scheduling strategy for scientific workflow on infrastructure as a service (IaaS)areproposed. Workflow is represented as DAG in which nodes represent the jobs and edges represent the dependency between jobs. Workflow tasks are executed in given specified order to accomplish functionality. An algorithm based on the meta-heuristic optimization technique, PSOwhich aims to minimize the overall workflow execution cost while meeting deadline constraints is presented.The performance of the algorithm is evaluated using an available scientific workflow like Montage,Cybershake,Sipht, and Ligo. Results show that PSO is better that the other state of art algorithms such as IC-PCP,SCS,and PSO-HOM.

In Reference [16], PSO based heuristic to schedule application to cloud resources that take into account both computation cost and data transmission cost is presented. The purpose of the paper is to optimize execution time and execution cost that arises from data transfer between resources.The experiment of a workflow application is done by varying its computation and communication costs. Results are compared with already existing BRS (Best Resource Selection).Results show that PSO is better than BRS and the reason for PSO's improvement over BRS is due to PSO's ability to find near-optimal solutions for mapping all tasks in the workflow to the given set of computing resources. The linear increase in PSO's cost also suggests that it takes both computation and communication cost into account. However, BRS simply maps a task to the resources that have minimum completion time (a resource with higher frequency, lower load and thus having higher cost).As the resource cost increase, the use of BRS leads to more costs due to the affinity towards better resource, irrespective of thesize of data. Whereas, PSO minimizes the maximum total cost of assigning all tasks to resources.

In Reference [17], Artificial Bee Colony (ABC) algorithm is discussed that simulates the foraging behavior of swarm. In this paper, this algorithm is used to optimize a large set of numerical test functions. Results are compared with particle swarm optimization algorithm and genetic algorithm to show the better result for ABC algorithm.

In Reference[18],a new algorithm named Proportional Deadline Constrained (PDC) is introduced that address workflow scheduling in the cloud. PDC's aim is to minimize costs while meeting deadline constrained. The PDC algorithm produces a deadline constrained schedule that minimizes the financial cost of execution and has a generally lower failure rate in constructing schedules for tighter deadline. Simulation results show that PDC achieves lower costs for a given deadline than state-of-the- art algorithm IC-PCP and GAIN.

In Reference [19], market-orientedbusinessmodel is described in which user can access the cloud services through internet and pay only for what they use. Large scale scientific application isoften expressed as workflow. The purpose of the paper is scheduling workflow application in such a way so that execution costas well as execution time incurred by using a set of homogeneous resources over cloud can be minimized. So to fulfill the requirement they propose bi-criteria priority based particle swarm optimization (BPSO) to schedule workflow task. The proposed algorithm is evaluated using simulation with four different real-world workflow application and results show that it performs better than existing PSO and BHEFT (Budget Constrained Heterogeneous Earliest Finish Time) algorithms.

III. PROBLEM STATEMENT

A workflow application is represented as a Directed Acyclic Graph (DAG) $G =< T, E >$. Let Tis the set of finite tasksT_i $(1 \leq i \leq n)$. Let E be the set of directed edges of the form (T_i ,T_j) where T_i is parent task of T_j , on the other hand T_j is the child task of T_i .We assume that a child task cannot be executed until all of its parent tasks are completed.Then, the workflow application can be described as DAG G=<*T*, *E*>. Additionally, each workflow task T_i has a task length **Tlen** given in Million Instructions (MI) and edge E represents the communication cost between these tasks. Suppose that we have a number of Virtual Machines (VMs).Runtime of the VMis in seconds.

Makespan, M, is the total elapsed time required to execute the entire workflow. The deadline D is considered as a constraint where the Makespan M should not be more than the deadline D i.e., $M < D$.The makespan of the workflow is computed as follows:

$$M = finish\_time - start\_time$$

Where $start\_time$ is the submission time of the workflow and $finish_time$is the end time of the exit node.$finish\_time$

$$finish\_time = \{ET_{t_i} | t_i \in \mathrm{T}\}$$

Here ET denotes the End Time of the exit node.$t_i$is the exit node.

Total Execution Cost, TEC, is the total cost of the workflow execution, which is the sum of the price for the VMs used to execute the workflow. Each VM type has a price associated with it, depending on its characteristics and types. The price of each VM is calculated based on its type and the duration of time it was provisioned. The duration of the time is calculated based on the number of hours a VM executes, from the time of its instantiation, until it is terminated or stopped. The time duration is always rounded to the next full hour (e.g. 5.1 hours is rounded to 6 hours).We aretakingcost on per hour basis. It is important to mention that multiple tasks can execute in a VM depending on the schedule. Moreover, to execute the entire workflow, multiple VMs can be used. Therefore, the total execution cost, C, is the sum price of all the VMs used in the workflow execution. Additionally, there is a budget B as a constraint, such that the total costs should be less than thebudget i.e.

$$COST < BUDGET$$

Where $COST$ $C$is calculated as

$$C = \sum_{j=1}^{vm} c_{vm_j} * et_{vm_j}$$

c_{vm_j} represents the cost to lease the resource vm_j for a unit of time and et_{vm_j} represents the execution time ofvm_j.

IV. ALGORITHM DESIGNING FOR CLOUD COMPUTING WORKFLOW SCHEDULING

A) Particle Swarm Optimization(PSO)

PSO was first introduced by Kennedy and Ebehart in 1995, inspired by social behavior of bird flocking or fish schooling. In this algorithm, a number of particles flow through the swarm space and each particle represents a candidate solution to the optimization problem. At any point of time, every particle has some velocity and position in the search space. Initially, positions and velocities of particles are randomly assigned. After each iteration, velocities and positions of particles are updated using equations (1) and (2) respectively. Every particle in PSO has its local best position i.e., $pbest$ and the population has a global best position i.e, $gbest$. Global best position of the population is the position of the particle which is more close to the optimal value. Every particle will move towards the best position of the swarm i.e., global best position, as it is close to the optimal value. Global best position of the population will be refreshed if some other particle's position becomes more near to the optimal value. Now, this particle's position will be the global best position of the population. Now, every particle will move towards this refreshed global best position. We repeat this process until our terminate criteria met. In this way, at some point of the time, all the particles will converge at one point and this point will give the optimal value of the objective function.

$$V_i^{t+1} = w.v_i^t + c_1 r_1 (pbest - x_i^t) + c_2 r_2 (gbest - x_i^t) \ldots . (1)$$

$$X_i^{t+1} = x_i^t + v_i^t \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (2)$$

Where

$$w = inertia$$
$$c_i = acceleration\ coefficient, i = 1{,}2$$
$$r_i = random\ number\ , i = 1{,}2\ \ 0 < r_1 < 1$$
$$x_i = best\ position\ of\ particle\ i$$

$\mathrm{V}_\mathrm{i}^{t+1}$ And $\mathrm{X}_\mathrm{i}^{\mathrm{t+1}}$ is the velocity and position of particle i at iteration t+1 respectively. On the other hand, Particle Swarm Optimization (PSO) has become popular because of its simplicity and its effectiveness in a broad range of application. Some of the applications that have used PSO to solve NP-Hard problems like Scheduling problem and the task allocation problem. In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods [20].

B) ANT COLONY OPTIMIZATION

Ant colony optimization is a technique for optimization that was introduced in the early 1990's. The inspiring source of ant colony optimization is the foraging behavior of real ant colonies. Ant Colony Optimization (ACO) is an optimization technique that is used by ants to find the shortest path between their food and nest. Ants communicate with each other through some kind of secretion called Pheromones. While moving, every ant secretes its pheromones on the path. These pheromones evaporate after some time. Every ant finds the solution of the problem iteratively. At every iteration, each ant moves from one position to another position to complete the partial solution. For an ant, the probability of moving from position x to position y depends upon two factors:-

(1) The attractiveness of the edge: It is the prior desire of the move and is calculated by some heuristic. Normally, it is the reciprocal of the distance between $x$and y.

(2) Pheromone density on the edge: It is the amount of the pheromones on the edge of$x$and y.

An ant moves from $x$to y with probability as follows:-

$$P=\frac{(Pm^{\alpha}_{xy})\,(\lambda^{\beta}_{xy})}{\sum_{z\in allowed\ \ x}(pm^{\alpha}_{xz})(\lambda^{\beta}_{xz})} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

Where Pm^{α}_{xy} is the pheromone amount deposited on the move from x to y , λ^{β}_{xy} is the attractiveness for move from x to y(normally $(1/D)$ where D is the distance between $x$and$y$), $\alpha\geq 0$ is the parameter that controls the pheromone amount and $\beta\geq 1$ is the parameter that controls the attractiveness [21].

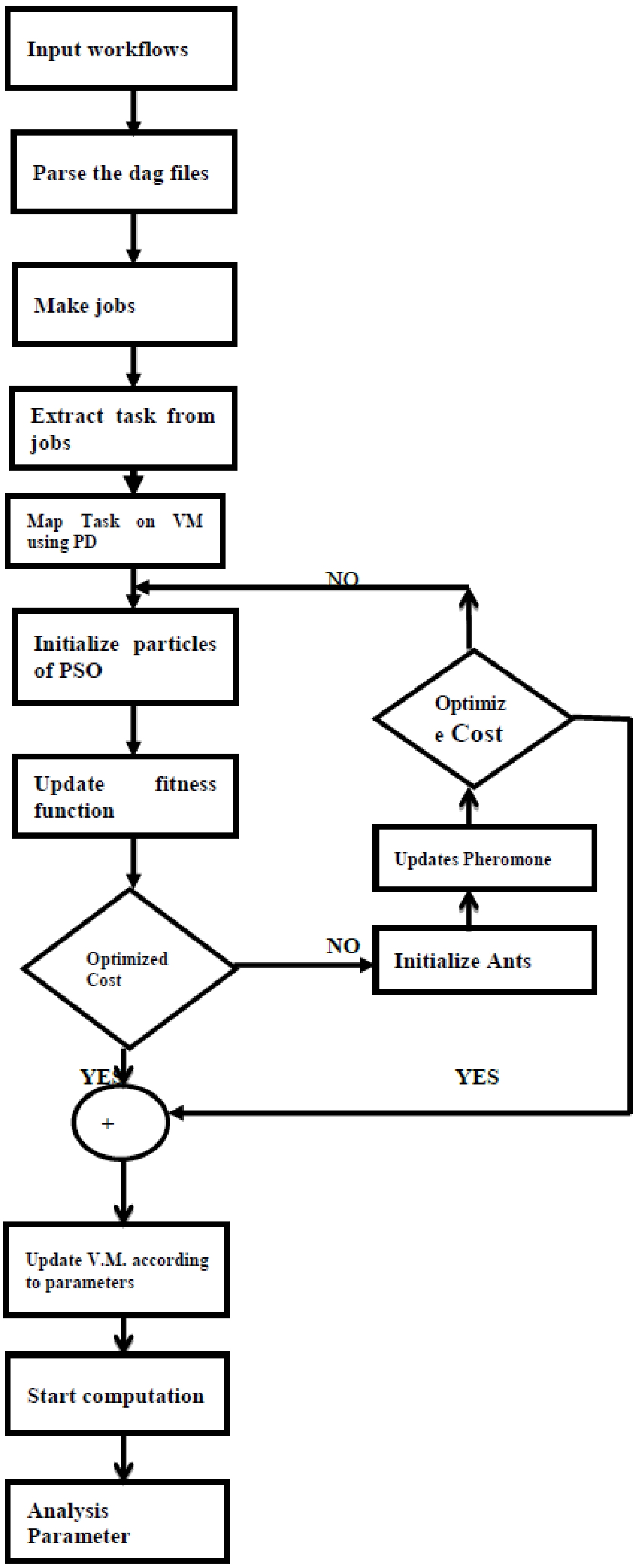
On finding food, ants take food and return back to the nest through the same path. An ant that will reach the nest first has chosen the shortest path as it comes back in the shortest period of time. The Pheromone density on this path will be higher than that on the other paths as it is shortest and ant has deposited the pheromones while going and returning. On the longer paths, when the ants will return, the previous pheromones will get evaporated and the pheromone density on these paths will be lower than the shortest path. Now, the next ants will choose the shortest path as the pheromone density on this path is more than the longer paths. These new ants will further increase the density of the pheromones on the shortest path. Pheromones on the paths are updated by using the equation (4) [12]

$$Pm_{xy} = (1-\Phi)Pm_{xy}+\sum_{k} \Delta Pm^{k}_{xy} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

where Pm_{xy} is the pheromone amount on the transition from x to y, Φ is the evaporation coefficient of pheromones, ΔPm^{k}_{xy} is the amount of pheromones that kth ant deposited on the transition from x to y.At one point of time, pheromone density on the longer paths will become zero and all ants will go through the shortest path.

Time and Cost of workflow scheduling can be optimized using Hybridization of PSO and ACO. Particle Swarm Optimization (PSO) is used to optimize the decision but it has still high-cost convergence problem which is reduced by Ant colony optimization (ACO) which optimizes the cost of PSO by fast convergence.

IV. FLOW CHART



V. ALGORITHM

Input:A DAG G with job wise deadline d and total Budget ξ
Output: A Cost Optimized Schedule

1. Divide the workflow into number of jobs. Assign each job to his own deadline. All taskinsame job has same deadline.
2. for (i=0; i<number_of_jobs; i++)
3. for (j=0; j<number_of_task(i); j++)
   vm←a[i][j] using Pareto Distribution
   end

Call PSO (vm)
end.

Fun PSO (vm)

1. Initialize PSO particles and PSO parameter
2. Update velocity & position using equation (1) and (2) respectively.
3. Evaluate fitness using cost function.
4. If($cost_p < \xi$)
5. { Analyze the parameters}
   Else
   {Call ACO (vm)}
   While: termination criteria are not met.

Fun ACO (vm)

1. Initialize the pheromone trails, Ant solution construction.
2. For each ant compute fitness value and update pheromone using equation (3).
3. Optimizeα,β
4. If ($cost_p < \xi$)
   {Analyze the parameters}
   Else
   {Call PSO (vim)}

VI. V.SIMULATION RESULT

To prove the effectiveness of our algorithm we are doing simulation in cloudsim. We apply the algorithm hybridization of ACO and PSO on four standard workflows. These are GENOME,CYBERSHAKE,MONTAGE AND LIGO. This workflow scheduling technique is implemented using CloudSim 3.0.3, a JAVA based platform, to simulate a largescaleworkflow. To get the results of PSOACO algorithm we used Intel(R) Core™ i5-6200U CPU @2.30 GHz CPU and 8GB RAM. The parameters of PSOACO which are used in this paper are given in Table I.

Table I

PSO and ACO parameter		
P_0	No. of Particles	10
It_{max}	Maximum number of Iteration	50
w	Inertia Weight	0.99
c_1, c_2	Acceleration Coefficient	1.5
A	Alpha	0.6
B	Beta	2.5
ϕ	Pheromone Evaporation Coefficient	1

Our Simulation results show that as the number of task increases in workflow,our PSOACO Perform better than BPSO. The characteristics of four workflows on which we are doing simulation are represented here.

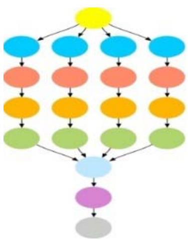
GENOMICS:-This workflow is being used by Epigenome Center in the processing of production DNA methylation and histone modification data.

CYBERSHAKE:-TheCybershake workflow is used by the Southern California Earthquake Center (SCEC) to characterize earthquake hazards using the Probabilistic Seismic hazards analysis (PSHA) technique.

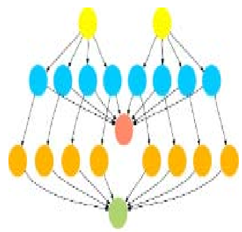
MONTAGE:-Montage is used to generate the custom mosaics of the sky using input images in the Flexible Image Transport System (FITS) format.

LIGO:-The Laser Interferometer Gravitational WaveObservatory (LIGO) attempts to detect gravitational waves produced by various events in the universe as per the Einstein's Theory of general relativity.

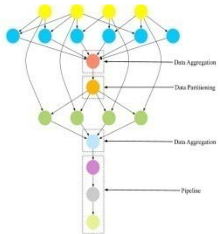
The workflows structures are shown below.

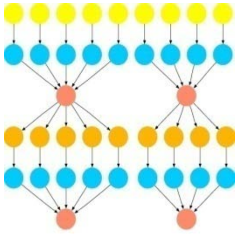


GENOME



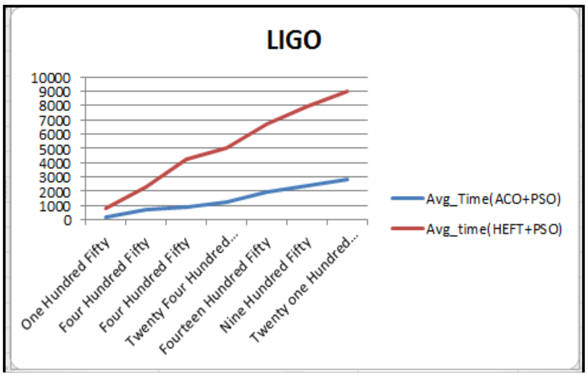
CYBERSHAKE

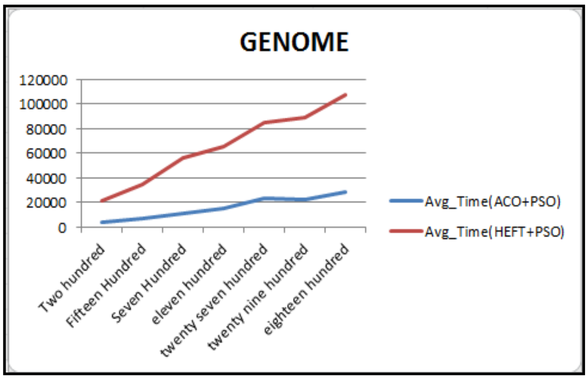


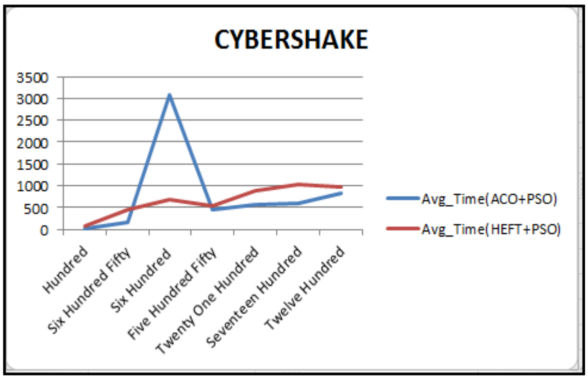


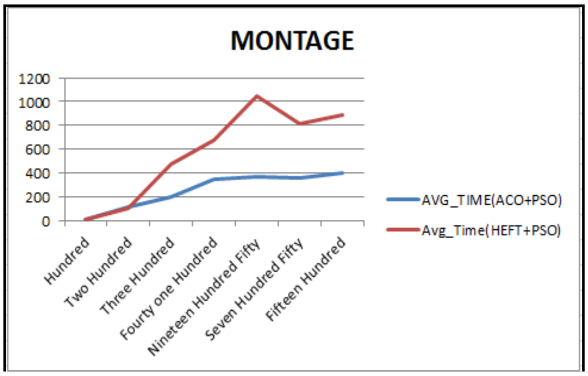
MONTAGE **LIGO**

A) *EXPERIMENTAL RESULT OF AVERAGE TIME:*

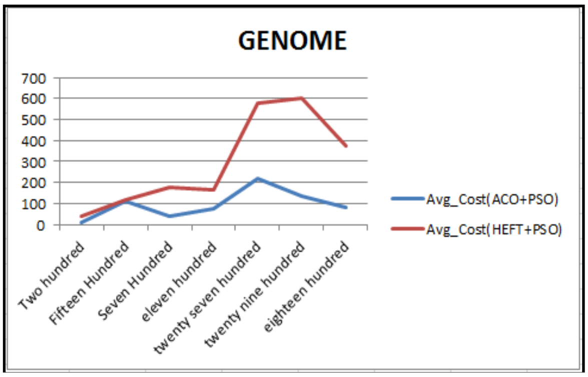


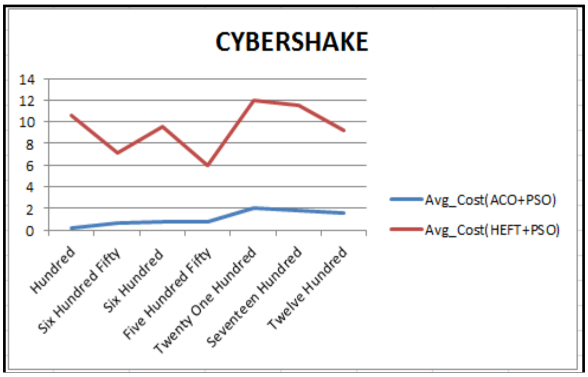


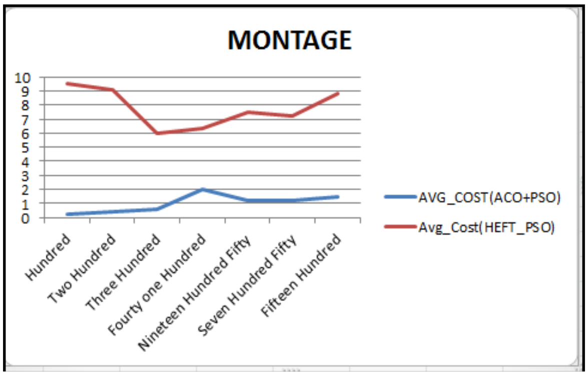


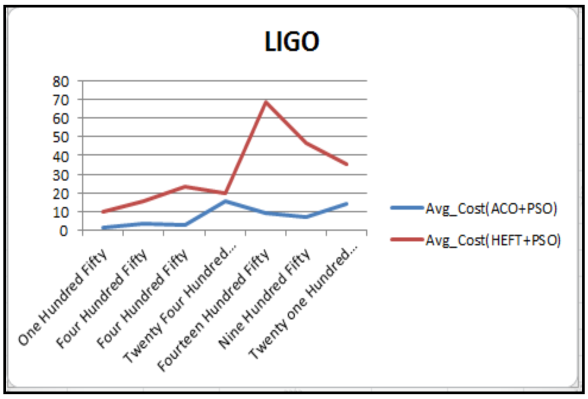


B) *EXPERIMENTAL RESULT OF AVERAGE COST:*









VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented hybrid PSO with cost function optimized using ACO to schedule workflow applications to cloud resources that minimizes the execution cost while meeting the deadline. We have divided the workflow task into number of jobs and the tasks are scheduled using parito distribution. The proposed algorithm is evaluated with synthetic workflows that are based on realistic workflows with different structures and different sizes. The Experimental work shows comparison of proposed algorithm is done with BPSO. The simulation results show that our proposed algorithm has a better performance as compared to BPSO because PSO converges at highcost. So, we are using ACO to minimize the cost function. In future work, we will try to use hybrid PSO and ACO to fulfill multiple objectives of minimizing the makespan and cost along with increasing the reliability of the workflow scheduling.

VII. REFERENCES

[1]. P. Mell and T. Grance, “The NIST definition of cloud computing,” Communications of the ACM, vol.53, no.6, pp. 50-50, Jun. 2011.

[2]. Santos, Nuno, Krishna P. Gummadi, and Rodrigo Rodrigues. "Towards Trusted Cloud Computing." HotCloud 9.9(2009):3.

[3]. http://aws.amazon.com/ec2/

[4]. http://www.ibm.com/ibm/cloud

[5]. Pandey S., “Scheduling and management of data intensive application workflows in grid and cloud computing environment”, Ph.D. Thesis, University of Melbourne, Australia, 2011.

[6]. QUANDT, R. E.: Old and new methods of estimation and the Pareto Distribution, Metrika,**10**, 1966, p. 55–82.

[7]. Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and Experience 41, no. 1: 23-50 (2011).

[8]. Choudhary Monika, and Sateesh Kumar Peddoju. “A dynamic optimization algorithm for task scheduling in cloudenvironment." International Journal of Engineering Research and Applications (IJERA) 2, no. 3: 2564-2568(2012).

[9]. Shishira S. R., A. Kandasamy, and K. Chandrasekaran. "Survey on meta heuristic optimization techniques in cloud computing." In Advances in Computing, Communications, and Informatics (ICACCI), 2016 International Conference on, pp. 1434-1440.IEEE, 2016.

[10]. Z. Liu, X. Wang**A**PSO-based algorithm for load balancing in virtual machines of cloud computing environmentLect Notes ComputSci (including SubserLect Notes ArtifIntellLect Notes Bioinformatics), 7331 LNCS (2012), pp. 142–147.

[11]. Bharathi, S., Lanitchi, A., Deelman, E., Mehta, G., Su, M.H., and Vahi, K., “Characterization of scientific workflows”, In Workshop on Workflows in Support of Large ScaleScience,CA, USA, pp:1-10, Nov.17,2008.

[12]. Yu, J. and Buyya, R., “Taxonomy of workflow management systems for grid computing”, Journal of Grid Computing, vol.3, no.1-2 ,pp: 171–200, Jan 2005.

[13]. Hu Y, Xing L, Zhang W, Xiao W, Tang D.A knowledge-Based Ant Colony Optimization for a Grid Workflow Scheduling Problem. In: Adv. swarm intel notes compute sci. Springer. 2010. p. 241-8.

[14]. Chen, W., Deelman, E.: Integration of workflow partitioning and resource provisioning. In: IEEE/ACM Int. Symposium on Cluster Computing and the Grid (CCGRID), pp. 764–768 (2012).

[15]. Rodriguez M.,BuyyaR.: Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds. In IEEE transactions on cloud computing 2014; 2:222-35.

[16]. Pandey S, Wu L, Mayura S., BuyyaR:A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments.

[17]. "A comparative study of artificial bee colony algorithm." Applied mathematics and computation 214, no. 1: 108-132 (2009).

[18]. Arabnejad V, Bubendorfer K, “Cost Effective and Deadline Constrained Scientific Workflow Scheduling for Commercial Cloud”,in Proc. Of 2015 IEEE 14th International Symposium on Network Computing and Applications.

[19]. A. Verma, S. Kaushal, “Bi-criteria priority based particle swarm optimization workflow scheduling algorithm for cloud,” in Proc. of 2014 Recent Advances in Engineering and Computational Sciences, pp. 1-6, 2014.

[20]. J. Kennedy, R. Eberhart., “Particle swarm optimization”, In Proceedings International Conference on Neural Networks, Perth, WA, pp: 1942 – 1948, Nov.27- Dec.01, 1995.

[21]. Dorigo M, Blum C (2005) Ant colony optimization theory: a survey. TheorComputSci 344(2-3): 243–278.

