International Journal of Computer Applications (0975 – 8887)
Volume 89 – No.2, March 2014

An Analysis of Task Scheduling in Cloud Computing using Evolutionary and Swarm-based Algorithms

Saurabh Bilgaiyan, Santwana Sagnika, Madhabananda Das
School of Computer Engineering
KIIT University, Bhubaneswar, Odisha, India, 751024

ABSTRACT
Cloud computing is a popular computing paradigm that performs processing of huge volumes of data using highly available geographically distributed resources that can be accessed by users on the basis of *Pay As per Use* policy. In the modern computing environment where the amount of data to be processed is increasing day by day, the costs involved in the transmission and execution of such amount of data is mounting significantly. So there is a requirement of appropriate scheduling of tasks which will help to manage the escalating costs of data intensive applications. This paper analyzes various evolutionary and swarm based task scheduling algorithms that address the above mentioned problem.

General Terms
Cloud Computing, Evolutionary Algorithms, Optimization, Soft Computing.

Keywords
Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), cloud computing, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Quality of Services (QoS), task scheduling.

1. INTRODUCTION
Cloud computing is an evolving concept which makes better use of multiple distributed resources that can be allocated to users as per requirements. This helps in cheaper and efficient utilization of available resources and easier handling of larger computational problems. Its advantages include transparency of resources, flexibility, location independence, reliability, affordability, and greater availability of services, etc. [1]. To provide these facilities, the tasks need to be scheduled properly on the resources so as to provide maximum performance in minimum time.

This resource allocation process is performed in two stages. In the first stage, the load balancer allocates resources to systems as requested by an application. The second stage takes place when incoming requests are assigned to an application in an effort to balance loads within the application as per Quality of Services (QoS) and minimum cost [2]. To minimize the total time taken, the scheduling principle should aim to reduce the amount of data transfer with minimum cost and ensure balanced distribution of tasks as per processing capability. Thus the proper task resources mapping allocation is an important issue for which various optimization techniques have been experimented with [3].

The total running cost of any task is the summation of communication cost and computation cost of that task. Data transfer cost depends on the size of data transferred. Ignoring locality of data when the data size is large leads to high communication cost. It requires a model where the tasks are equally distributed among resources and the cost is minimized as well [4]. In this paper, the authors summarily present a study of various swarm-based and evolutionary optimization techniques effectively used for task-resource mapping and scheduling on cloud computing systems.

The rest of the paper is organized as follows: Section 2 presents task scheduling description. In Section 3, a review of techniques is presented. In Section 4, a tabular format is used to present data summarily. Section 5 concludes the paper and discusses some future work.

2. TASK SCHEDULING ON RESOURCES
To realize service allotment in cloud computing systems, service scheduling and resource allocation have been the major issues. In a cloud environment, traditional scheduling methods are infeasible owing to its properties - dynamical, distributed, and sharable. The aim of resource allocation to tasks is for all services to meet their performance targets. Several jobs demand different resources while running simultaneously. It is important for efficient working of cloud to balance these jobs on appropriate resources for optimal performance, and various task parameters need to be considered for proper scheduling. The available resources should be used effectively without affecting the service parameters. Scheduling in the cloud environment system is an NP-complete problem. As the number of users increase, the tasks that need to be scheduled increase in proportion. Therefore, there is a need for better algorithms to schedule tasks on these systems. Algorithms required for scheduling are service-oriented and vary in different environments.

3. A REVIEW OF OPTIMIZATION TECHNIQUES FOR TASK SCHEDULING
The efficiency of task scheduling directly affects the performance of the system. Many optimization algorithms have been applied to solve this problem. Different researchers have proposed various algorithms for allocating and scheduling the resources efficiently in the cloud. Here the authors provide a comparative study of different evolutionary and swarm-based techniques that perform scheduling of tasks to resources, such as ant colony, genetic algorithm, simulated annealing, particle swarm, and bee colony, etc. Various modified scheduling algorithms like Improved Genetic Algorithm, Modified Ant Colony Optimization, Multi-objective Particle Swarm Optimization have also been analyzed [5].

3.1 Genetic Algorithm

Genetic Algorithm (GA) was the first evolutionary technique that is based on the principle of natural selection and Mendel's laws of inheritance. GA has various advantages over other techniques for computationally intensive problems, if provided with properly set operators and fitness functions. GA defines a set of solutions (chromosomes) that are collectively called a population. The method then performs crossover, mutation and selection operations iteratively till the stopping criteria is satisfied. The resultant set is the set of solutions. [6] Figure 1 represents the general steps of this algorithm.

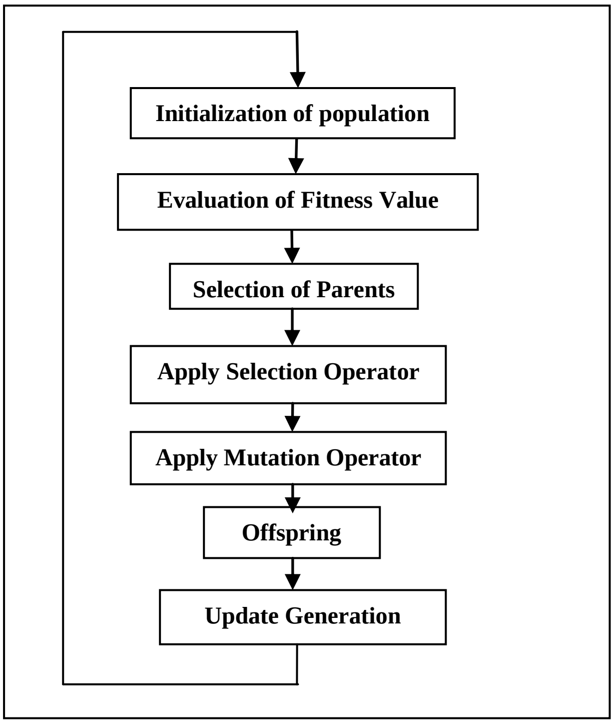

Figure 1: Steps of Genetic Algorithm

GA is implemented by the following steps:

Step 1: Create a population of a fixed number of random chromosomes.

Step 2: Calculate the fitness values for all chromosomes.

Step 3: Select two chromosomes having best fitness as parents (Selection step).

Step 4: Perform crossover among the parents to produce offspring using crossover ratio (Crossover step).

Step 5: Perform mutation if required at each position using mutation ratio (Mutation step).

Step 6: Add the offspring chromosomes to the original population.

Step 7: If the termination condition is satisfied, then stop and return best chromosome as a solution, else repeat from Step 2.

In this category, the different methods applied for task scheduling include Bi-level Multi-Objective task scheduling model, Energy-Aware GA, GA based scheduler, Genetic Simulated Annealing Algorithm, Multi-Agent GA, Improved GA, Multi-Objective GA (MOGA), Reputation Guided GA, GA-based task scheduling and GA-based independent tasks scheduling.

Zhao, Zhang, Liu, Xie and Hu (2009) [7] presented an GA-based optimized algorithm for scheduling divisible and autonomous tasks adapting to varying computational and storage needs in heterogeneous systems, where resources are of computational and communication heterogeneity. Ge and Wei (2010) [6] developed a new scheduler, which makes a scheduling decision by evaluating the entire group of tasks in the job queue and uses GA for optimization. Guo-ning, Ting-Lei and Shuai (2010) [8] designed an algorithm based on simulated annealing GA that considered the QoS requirements of different types of consumption tasks, corresponding to the characteristics of tasks in a cloud environment. Zhu, Song, Liu, Gao and Cheng (2011) [9] demonstrated the advantage of (Multi-Agent Genetic Algorithm) MAGA over traditional GA, by designing a load balancing model on the basis of virtualization resource management. Xiaoli and Yuping Wang (2012) [10] proposed an improved bi-level multi-objective evolutionary algorithm that used a modified operator and a local search scheme to speed up the convergence. The algorithm generated a number of scheduling schemes and also increased server efficiency through data layout policies and task scheduling strategies. Chang-Tian and Jiong (2012) [11] considered energy consumption, makespan criteria and users QoS as objectives, and using Dynamic Voltage Scaling (DVS) they proposed two algorithms - Energy consumption Time Unify Genetic Algorithm (ETU_GA) and Energy consumption Time Double Fitness Genetic Algorithm (ETDF_GA) - that balance makespan and energy. Jang, Kim, Kim and Lee (2012) [12] demonstrated a task scheduling model in which the scheduler invokes the GA scheduler function in each scheduling round, which generates a set of schedules that are evaluated as per the satisfaction of a user and availability of Virtual Machine (VM). Junwei and Yongsheng (2013) [13] presented an improved GA that considers the total task completion time, average task completion time and cost constraint. Liu, Luo, Zhang, Zhang and Li (2013) [14] proposed a Multi-Objective Genetic Algorithm (MOGA)-based scheduling algorithm, considering energy usage as well as earnings of the access providing agencies, that provided a dynamic means for selecting the appropriate scheduling mapping as per users' real-time needs. Pop, Cristea, Bessis and Sotiriadis (2013) [15] aimed to increase the profit and minimize the costs, so they used a reputation guided genetic scheduling algorithm for independent tasks.

3.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) has recently emerged as a prominent heuristic approach, applicable to various large and complex problems, like task scheduling problem, knowledge extraction in data mining, electric power systems, etc. PSO follows the principle of random searching in entire solution space using a large population, depending upon the problem domain. Recent work shows that PSO gives better performance over other existing techniques in efficient optimization. Figure 2 represents the general steps of this algorithm.

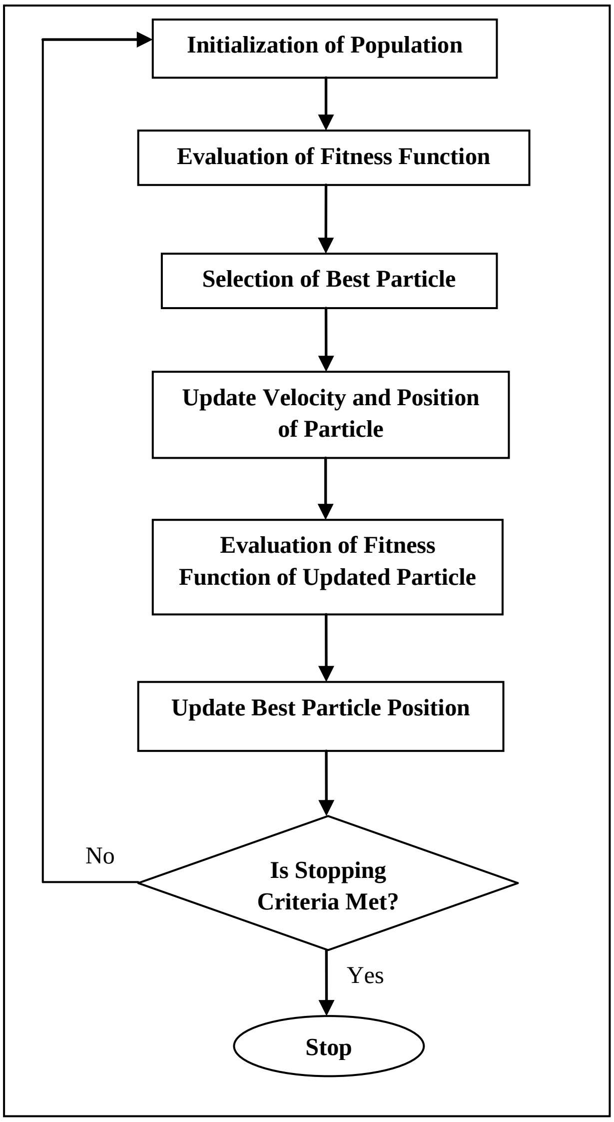

Figure 2: Steps of Particle Swarm Optimization

PSO is implemented by the following steps:

Step 1: Assign random velocities and positions to all particles across d dimensions.

Step 2: Assess the value of fitness function for all particles in every dimension.

Step 3: Match the pbest of every particle with the present value of fitness function and, if better, then replaces pbest with new value, and best location with new location x_i.

Step 4: Assign the index of particle with the best success to p_g.

Step 5: Update the particles' positions and velocities of the as per equations (1) and (2).

Step 6: Repeat from Step 2 till the stopping criteria is reached or maximum number of iterations is completed.

$$v_{id}^{(t+1)}) = wv_{id}^{t} + c_1 r_1 (p_{id}^{t} - x_{id}^{t}) + c_2 r_2 (p_{gd}^{t} - x_{id}^{t}) \qquad (1)$$

$$x_{id}^{(t+1)} = x_{id}^{t} + v_{id}^{(t+1)} \qquad (2)$$

where v_{id} is the velocity for i^{th} particle that lies in the d^{th} dimension, x^{id} is the corresponding answer of the i^{th} particle present in the d^{th} dimension. Also, i is the particle index, t is the present state, t+1 gives the subsequent state, while w represents the multiplier factor. c_1, c_2 are defined as accelerating constants; r_1, r_2 are two random values, such that $0 \leq r_1, r_2 \leq 1$. x_{id} gives the present location for the i^{th} particle, p_{id} is the best local solution for i^{th} particle, and p_{gd} is the best global solution for all the particles [16].

In this category, the different methods applied for task scheduling include PSO-based heuristic for scheduling, Cost Optimization using PSO, Multi-Objective Task Assignment using PSO, PSO based on small position value rule.

Pandey, Wu, Guru and Buyya (2010) [17] have used PSO-based heuristic for workflow scheduling in cloud environment, which considers not only execution cost but also the cost for transmitting dependent data. Netjinda, Sirinaovakul and Achalakul (2012) [18] used PSO technique for cost optimization by converting real data in the particles into integral representation of result, showing a potential performance in both the viewpoint of the total cost and convergence and also yielding various alternatives in procuring on changes in usage behavior. Guo, Shao and Zhao (2012) [19] formulated a model for the multi-objective task assignment and described a PSO algorithm in cloud that optimized the time and cost. Guo, Zhao, Shen and Jiang (2012) [20] devised a model using a PSO algorithm which was based on the small position value rule that ran faster and also saved processing time.

3.3 Ant Colony Optimization

ACO is a meta-heuristic method inspired from models of cooperative food search in ants. A set of agents is used to implement the behavior of real ant colonies where ants cooperate and communicate through pheromone trails. An ant solves a problem iteratively by using a construction graph where edges represent the possible partial solution that the ant can take according to a probabilistic state transition rule. After selecting a partial or a complete solution, a rule of pheromone updating starts. This rule gives a feedback mechanism to speed up convergence, and also prevents premature solution stagnation. Due to the elaborate characteristics of ACO, various algorithms based on the ACO meta-heuristic have been applied to many difficult optimization problems [21].

The method for ACO is described:

Step 1: Perform initialization of the pheromone

Step 2: while (stopping criteria not reached) do

Step 3: Set locations of all ants in an initial Virtual Machine

Step 4: while (all ants have not reached at a solution) do

Step 5: for each ant do

Step 6: Choose VM for the subsequent task using the intensity of pheromone trail

Step 7: end for

Step 8: end while

Step 9: Update the pheromone

Step 10: end while

Step 11: end

Figure 3 represents the general steps of this algorithm.

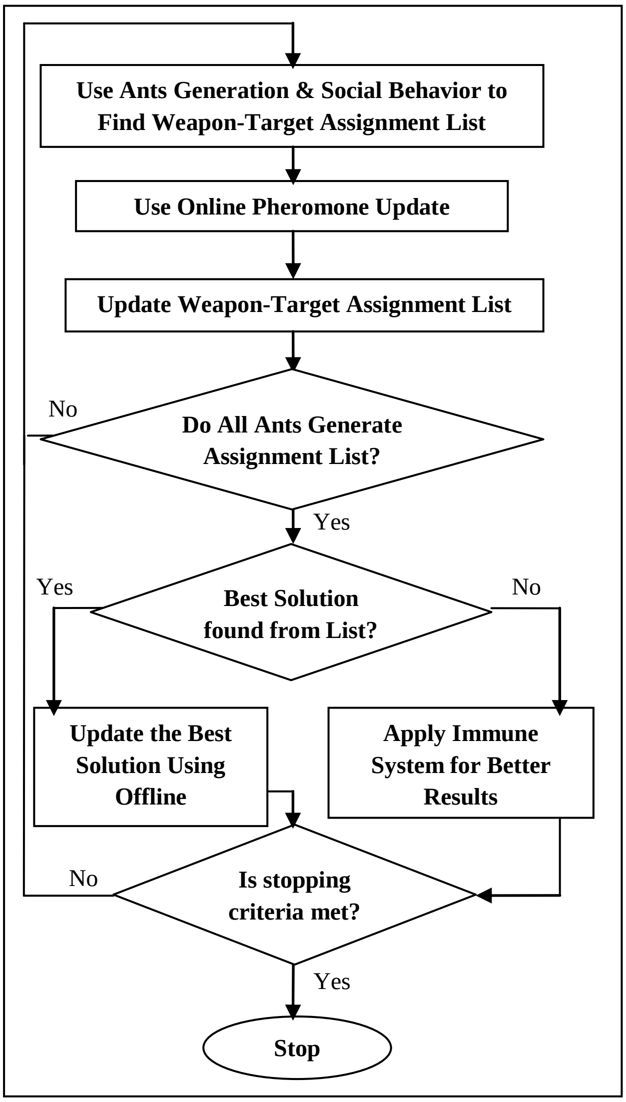

Figure 3: Steps of Ant Colony Optimization

In this category, the different methods applied for task scheduling include ACO for task matching and Load Balancing ACO.

Chiang, Lee, Lee and Chou (2006) [21] proposed an algorithm ACO-TMS that adopted a new state transition rule that reduced the time required when finding the satisfactory scheduling results, along with a local search procedure to help improve the scheduling results. Li, Xu, Zhao, Dong and Wang (2011) [22] proposed a cloud task scheduling policy based on Load Balancing Ant Colony Optimization (LBACO) algorithm that aimed at stabilizing the whole workload, besides optimizing the makespan of the tasks set. It also adapted to the dynamic cloud system and attained better load balancing in the system.

3.4 Bee Colony Optimization

The BCO algorithm is based on the activities of bees while searching for nectar, and sharing the information with other bees. There are three types of agents - the employed bee, the onlooker bee, and the scout. The employed bee stays on a food source and provides its surroundings in memory; the onlooker acquires this data from the employed bees and selects one of the food sources to forage; and the scout performs the task of finding new nectar sources [23].

The procedure for BCO is as follows:

Step 1. Initialization: Distribution of the populations into the solution space randomly, and evaluation of their fitness values, in the ratio ne that represents the percentage of employed bees in the total population.

Step 2. Movement of Onlookers: Calculation of the probability of selection of a food source by the equation (3), selection of a target food source to move to for onlooker bees and determination of the nectar amounts, following the equation (4).

Step 3. Movement of Scouts: If the fitness values are not improved by continuous iterations, those food sources are abandoned, and these employed bees convert into scouts and are moved by the equation (5).

Step 4. Updating of the Best Food Source: Memorization of the best fitness value and its position.

Step 5. Termination Checking: Checking of the termination condition, if satisfied, termination of the program and output of the results; otherwise repetition of Step 2.

$$p_i = \frac{fit_i}{\sum_{i=0}^{N} fit_i} \tag{3}$$

where fit_i is the fitness value of the solution i evaluated by the employed bee, N represents the number of employed bees, and p_i is the probability of selecting the ith employed bee.

$$v_{ij}^{(t+1)} = x_{ij}^{\ t} + \phi_{ij}^{\ t}(x_{ij}^{\ t} - x_{kj}^{\ t}) \tag{4}$$

where i ϵ 1, 2,.., N and j ϵ 1, 2,..,d are randomly chosen indexes in k ($\neq$ i,j) dimensions and $\phi(\cdot)$ generates a random series within the range [−1, 1] generated at time t, which controls the production of a neighbour solution around x_{ij}.

$$x_{ij} = x_{ijmin} + r \cdot (x_{ijmax} - x_{ijmin}) \tag{5}$$

where r is a random number and r ϵ [0, 1].

Figure 4 represents the general steps of this algorithm.

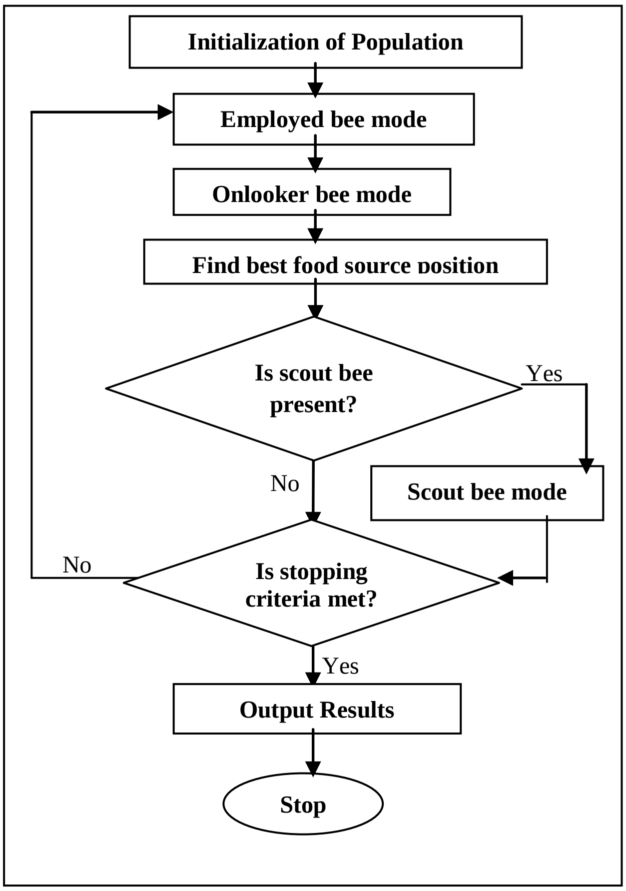

Figure 4: Steps of Bee Colony Optimization

In this category, the different methods applied for task scheduling include Bees Life Algorithm and Modified Bees Life Algorithm.

Bitam (2012) [24] proposed an efficient algorithm called Bees Life Algorithm (BLA) for optimal job scheduling by assigning end-user tasks to the relevant datacenters in an optimal way. Mizan, Masud and Latip (2012) [25] modified this algorithm along with greedy method to gain optimistic value of service in hybrid cloud and get a positive reply at the users' end and deployment of resources in a temporary mode.

This section gave an introduction to the different types of evolutionary and swarm-based algorithms and outlined the different approaches in each category that have been applied to task scheduling in cloud systems, as found in research history. These techniques are further represented in a tabular format alongwith their characteristics and advantages in the following section, where each table represents the methods available under a particular class of algorithms.

4. SUMMARY OF VARIOUS TECHNIQUES

Table 1. GA based techniques

Authors	Algorithm	Objectives	Key Principle	Advantages
Zhao, Liu, Zhang, Xie and Hu (2009)[7]	GA based independent task scheduling	Independent task scheduling in heterogeneous systems	Conflict measurement and competitive mechanism to find best fit solution	Scheduling on computation and communication heterogeneity
Ge and Wei (2010) [6]	GA based scheduler	Scheduler to evaluate total job queue	Centralized scheduler that makes a choice by referring to a global view of the total system	Shorter makespan for jobs, better balanced load across nodes
Guo-ning, Ting-lei and Shuai (2010)[8]	Simulated annealing GA	QoS requirements for different task types	Complete evaluation by dimensionless dealing with parameters	Efficient resource search and allocation
Zhu, Song, Liu, Gao and Cheng (2011)[9]	Multi agent GA	Load balancing	Collaboration and self-learning to attain global optimization	Improved virtualization resource management
Xiaoli and Yuping Wang (2012)[10]	Bi-level multi-objective task scheduling	Faster convergence	A reducing difference between resource utilization and optimal points by local search operator	Generation of scheduling schemes and increased server efficiency
Chang-Tian and Jiong (2012)[11]	Energy aware GA	Energy consumption, makespan, QoS	Use of unify and double fitness method for selections	Balanced makespan and energy consumption by

			and fitness determination	Dynamic Voltage Scheduling(DVS)
Jang, Kim, Kim and Lee (2012)[12]	GA based task scheduling	Quality with virtual machine accessibility and user contentment	GA scheduling procedure that uses information of tasks and machines to make task schedules	Creating a set of task schedules with quality evaluation
Junwei and Yongsheng (2013)[13]	Improved GA	Total task completion time, average task completion time, cost constraints	Conserves high potential genes with growth of generations	Balancing of multiple objectives
Liu, Luo, Zheng, Zheng, Li (2013) [14]	MOGA	Energy consumption, profits of service providers	Recognizes decision components to analyze the application	Dynamic selection mechanism of scheduling schemes
Pop, Cristea, Bessis and Sotiriadis (2013) [15]	Reputation based GA	Increased profit, minimal cost	Considers ranking as per reputation of resources for selection and ensuring of QoS	Reputation-guided QoS for independent tasks

Table 2. PSO based techniques

Authors	Algorithm	Objectives	Key Principle	Advantages
Pandey, Wu, Guru, Buyya (2010) [17]	PSO based heuristic for scheduling	Minimal computation and transmission cost	Considers communication cost and dependencies to calculate costs for all tasks at a time	Minimization of cost and load balancing
Netjinda, Sirinaovakul and Achalakul (2012) [18]	PSO based cost optimization	Total cost, convergence	Uses decoding system to translate particle positions from real-valued data into distinct integer data	Faster convergence and different purchasing alternatives as per usage behavior
Guo, Shao and Zhao (2012) [19]	Multi-objective PSO for task assignment	Optimal time and cost	Increases scalability by having less number of parameters to adjust	Feasible minimization of both time and cost
Guo, Zhao, Shen, Jiang (2012) [20]	PSO based on small position value rule	Faster convergence	Uses SPV to transform a continuous position vector to a dispersed value permutation vector	Faster execution and low processing time

Table 3. ACO based techniques

Authors	Algorithm	Objectives	Key Principle	Advantages
Chiang, Lee, Lee, Chou (2006) [21]	ACO for task matching	Reduction of time for finding results	Uses state transition rule to reduce time and applies Taguchi Method for high efficiency	Reduced time and improved results by local search
Li, Xu, Zhao, Dong, Wang (2011) [22]	Load balancing ACO	Balancing system load and minimizing makespan	Loads each Virtual Machine and defines load balancing factor to improve balancing ability	Adaptation to dynamic systems with balancing of load

Table 4. BCO based techniques

Authors	Algorithm	Objectives	Key Principle	Advantages
Bitam (2012) [24]	Bees life algorithm	Optimal and reliable job scheduling, minimum makespan	Global optimization using crossover to guarantee solution diversity and escape local optima	Efficiency in execution time, diversity of solutions
Mizan, Masud and Latip (2012) [25]	Modified Bees life algorithm	Optimistic value of service, proper utilization of resources	Uses greedy mechanism as a local searching procedure to attain best individual solution in the neighborhood thus enhancing each step	Minimum makespan, affirmative response at end-users

5. CONCLUSION

In this paper the authors analyze the various evolutionary and swarm-based algorithms that have been used for task scheduling on resources in cloud computing environments. Scheduling is an important activity in multi-tasking systems to efficiently manage resources, minimize idle time and increase performance of systems. Hence there is extreme need of proper scheduling in cloud computing systems as well, because real-time execution and higher throughput are essential requirements for multiple users. To implement scheduling, any of the discussed methods can be utilized, depending on the required objectives that need to be optimized. The benefits of using these methods include searching of large and complex problem spaces and reaching an optimal solution in less time. Multiple objectives can be simultaneously optimized and alternative solutions can be explored. Future work in this field can include application of newer optimization methods (e.g. Cat Swarm Optimization) and other variants of the above-mentioned methods (e.g. discrete PSO, modified ACO, etc.), which may be able to provide better and faster results. Further research can include optimization of more number of objectives, like budget, service availability, energy efficiency, users' comprehensive QoS, etc.

With the development and application of cloud computing technology, cloud computing has brought about a dramatic makeover of conventional software, business and enterprise handling. This makes it more advantageous. It can provide services as per users' specifications while CIS (cloud information server) takes care of the organization and execution of tasks and maintaining of resources' information. Thus research work in the field of cloud computing and its related areas holds high potential for advancement. Efficient scheduling of tasks on resources can significantly improve the viability and applicability of cloud computing systems in a wide range of fields.

6. REFERENCES

[1] Rimal, B.P., Choi, E., and Lumb, Ian. 2009. A taxonomy and survey of cloud computing systems. In Proceedings of the 5th IEEE International joint conference of INC, IMS and IDC, 44-51.

[2] Patel, R., and Patel, S., "Survey on Resource Allocation Strategies in Cloud Computing", International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 2, February- 2013, 1-5.

[3] Guo, L., Zhao, S., Shen, S., and Jiang, C., "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm", Journal Of Networks, Vol. 7, No. 3, March 2012, 547-553.

[4] Lakhani, J., and Bheda, H. 2012. Scheduling Technique of Data Intensive Application Workflows in Cloud Computing. In Proceedings of the Nirma University International Conference On Engineering, 1-5.

[5] Lin, C.T., "Comparative Based Analysis of Scheduling Algorithms for Resource Management in Cloud Computing Environment", JCSE International Journal of Computer Science and Engineering, Vol.-1, Issue-1, July 2013, 17-23.

[6] Ge, Y., and Wei, G. 2010. GA-Based Task Scheduler for the Cloud Computing Systems. In Proceedings of theIEEE International Conference on Web Information Systems and Mining, 181-186.

[7] Zhao, C., Zhang, S., Liu, Q., Xie, J., and Hu, J. 2009. Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing. In Proceedings of 5th IEEE International Conference on Wireless Communications, Networking and Mobile Computing, 1-4.

[8] Guo-ning, G., Ting-Iei, H., and Shuai, G. 2010. Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment. In Proceedings of IEEE International Conference on Intelligent Computing and Integrated Systems, 60-63.

[9] Zhu, K., Song, H., Liu, L., Gao, J., and Cheng, G. 2011. Hybrid Genetic Algorithm for Cloud Computing Applications. In Proceedings of IEEE Asia-Pacific Services Computing Conference, 182-187.

[10] Wang, X., and Wang, Y. 2012. An Energy and Data Locality Aware Bi-level Multiobjective Task Scheduling Model Based on MapReduce for Cloud Computing. In Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 648-655.

[11] Chang-Tian, Y., and Jiong, Y. 2012. Energy-aware Genetic Algorithms for Task Scheduling in Cloud Computing. In Proceedings of Seventh IEEE ChinaGrid Annual Conference, 43-48.

[12] Jang, S.H., Kim, T.Y., Kim, J.K., and Lee, J.S., "The Study of Genetic Algorithm-based Task Scheduling for

Cloud Computing", International Journal of Control and Automation Vol. 5, No. 4, December, 2012, 157-162.

[13] Junwei, G., and Yongsheng, Y. 2013. Research of cloud computing task scheduling algorithm based on improved genetic algorithm. In Proceedings of 2nd International Conference on Computer Science and Electronics Engineering, 2134-2137.

[14] Liu, J., Luo, X.G., Zhang, X.M., Zhang, F., and Li, B.N., "Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013, 134-139.

[15] Pop, F., Cristea, V., Bessis, N., and Sotiriadis, S. 2013. Reputation guided Genetic Scheduling Algorithm for Independent Tasks in Inter-Clouds Environments. In Proceedings of 27th IEEE International Conference on Advanced Information Networking and Applications Workshops, 772-776.

[16] Qing, W., and Han-Chao, Z. 2011. Optimization of Task Allocation And Knowledge Workers Scheduling Based-on Particle Swarm Optimization. In Proceedings of IEEE International Conference on Electric Information and Control Engineering, 574-578.

[17] Pandey, S., Wu, L., Guru, S.M., and Buyya, R. 2010. A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments. In Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications, 400-407.

[18] Netjinda, N., Sirinaovakul, B., and Achalakul, T. 2012. Cost Optimization in Cloud Provisioning using Particle Swarm Optimization. In Proceedings of 9th IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 1-4.

[19] Guo, L., Shao, G., and Zhao, S. 2012. Multi-objective Task Assignment in Cloud Computing by Particle Swarm Optimization. In Proceedings of 8th International Conference on Wireless Communications, Networking and Mobile Computing, 1-4.

[20] Guo, L., Zhao, S., Shen, S., and Jiang, C., "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm", Journal Of Networks, Vol. 7, No. 3, March 2012, 547-553.

[21] Chiang, C.W., Lee, Y.C., Lee, C.N., and Chou, T.Y., "Ant colony optimization for task matching and scheduling", IEE Proceedings - Computers and Digital Techniques, Vol. 153, No. 6, November 2006, 373-380.

[22] Li, K., Xu, G., Zhao, G., Dong, Y., and Wang, D. 2011. Cloud Task scheduling based on Load Balancing Ant Colony Optimization. In Proceedings of Sixth IEEE Annual ChinaGrid Conference, 3-9.

[23] TSai, P.W., Pan, J.S., Liao, B.Y., and Chu, S.C., "Enhanced Artificial Bee Colony Optimization", International Journal of Innovative Computing, Information and Control, Volume 5, Number 12, December 2009, 1-12.

[24] Bitam, S. 2012. Bees Life Algorithm for Job Scheduling in Cloud Computing. In Proceedings of The Third International Conference on Communications and Information Technology, 186-191.

[25] Mizan, T., Masud, S.M.R.A., Latip, R., "Modified Bees Life Algorithm for Job Scheduling in Hybrid Cloud", International Journal of Engineering and Technology Volume 2 No. 6, June, 2012, 974-979.