Comparative Analysis of ACO and PSO over Task Scheduling

Somya Arya[1]*, Kavita Khanna[2*] & Reshma Thomas[3]

[1]Department of CSE (M.Tech), The Northcap University, Gurgaon, India
*Corresponding Author E-mail: *somyaarya28@gmail.com*

[2]Associate Professor, CSE & IT, The Northcap University, Gurgaon, India
E-mail: *kavitakhanna@ncuindia.edu*

[3]Department of CSE (M.Tech), The Northcap University, Gurgaon, India
E-mail: *reshma15csp013@ncuindia.edu*

***Abstract**:* With the ever-growing population over the internet, and more and more people depending upon the cloud service to store, handle and schedule the data, task scheduling has become one of the most discussed topic. The key problems in distributed heterogeneous computing systems is scheduling, to benefit from colossal computing capacity of these systems and this problem is NP-complete. This paper will discuss about the comparative analysis of Ant Colony Optimization and Particle Swarm Optimization algorithms for Task Scheduling. For this purpose, the algorithms were run on CloudSim which is an artificial cloud environment. The comparison of the algorithms is based on makespan, cost and deadline. To see the effectiveness of algorithms, the size of the resource matrix was increased gradually.

Keywords: Ant Colony Optimization, Particle Swarm Optimization, CloudSim

I. INTRODUCTION

Task scheduling [1] is an important aspect when talking about cloud computing. Management of resources like cost and time is very essential for any service, and when talking about cloud computing, scheduling of tasks in an efficient way is a major issue. From decades, many algorithms are being suggested for task scheduling. Every algorithm aims at providing a reduced time and space complexity, even if by a slight margin. In the current period, many metaheuristic algorithms are used for scheduling purpose. In this paper, we will discuss about two of them: Ant Colony Optimization [2] and Particle Swarm Optimization [3]. The test results being taken in this paper were conducted on CloudSim [4], which is a cloud simulator. The algorithms were compared on the criteria of makespan, cost and deadlines when the resource matrix was increased gradually over the test cases. Ant Colony Optimization algorithm is established on the natural conduct of ants and how they search for food covering the least possible distance. Whereas Particle Swarm Optimization has a swarm of particles and all of them individually search for their best solutions. The best solution found by an individual particle is called the personal best solution (p_best). After each iteration, every particle communicates with its neighbors and maintains and updates a global best solution (g_best). PSO is best suited where a minimum or maximum is to be found. In this paper, we will be making a comparative analysis of ACO and PSO [5 & 6] on the ground of their makespan [7], cost and deadline [7 & 8]. Makespan is generally the measure of how fast an algorithm works when certain parameters are increased or decreased (like number of jobs submitted). Cost consideration is important as normally the cloud services are paid services, hence cost cutting is also a measure of efficiency here. Deadlines are also used as a measure of comparison as not only the completion of a task is important but completing it on or before the deadlines plays a vital role in cloud services. This paper is structured as follows: Section II covers Ant Colony Optimization Algorithm. Section III explains the Particle Swarm Optimization Algorithm. Section IV talks of the analysis of ACO and PSO. Section V discusses the Comparison between ACO and PSO. Section VI concludes the paper.

II. ANT COLONY OPTIMIZATION

The ant colony optimization algorithm [9] has been inspired by the natural behavior of group of ants. The ants are social insects. The ants when go searching for the food source, at first go in random directions. The ants leave pheromone trails among the path they choose. Now among all the ants who found a food source, the ant who found the food source nearest to the nest, will accumulate the pheromone faster. Ants sense the pheromone amount among different paths and follow the path which has maximum deposit of pheromone. In this way, after a while almost all of the ants follow the same trail as the smallest path will accumulate maximum amount of pheromone in the same time. Also we need to keep in mind that the pheromone has a certain evaporation rate. Various experiments were conducted to test the accuracy of this theory. In the very first experiment conducted to test this theory, two paths of different lengths were created from the nest to the food source. At start, ants randomly choose one of the paths. But after a certain period, all the ants took the shortest path. This verified the theory observed and was later used to develop ant colony optimization algorithm. In Ant Colony Optimization, the artificial ants develop many solutions to an optimization problem and trade the information about the quality of solutions by virtue of a communication scheme that recall of the one adopted by the real ants. Fig. 1 shows a classic example of ant behavior.

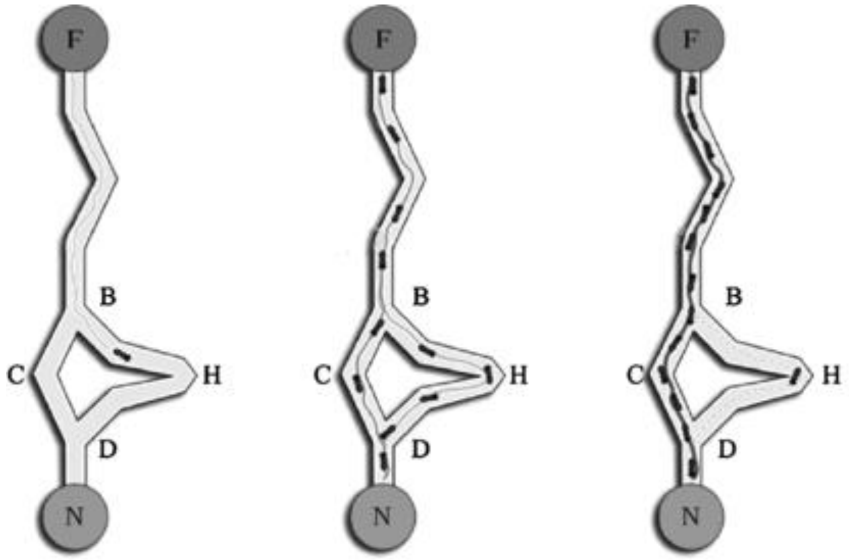

Fig. 1: Ant Behavior towards different routes from source to destination.

When using ant colony optimization for task scheduling [10], the programmers thrive for getting the minimum makespan and cost even when the job size increases. Also, completion of the jobs in a certain deadline is one of the major issues.

The pseudo algorithm for ACO is:

1. Initialization of first pheromone.

2. As an entry state, initialize the location of the ant.

3. Select the next state.

4. Check if next state is the final state, if it is not then go to step 3, else go to step 5.

5. Update the value of pheromone (deposit, daemon and evaporation of pheromone).

6. If satisfying the stopping criteria, stop the execution, else repeat from step 2.

III. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization deals in two major methodologies. Conceivably more obvious are its relativity to artificial life (A-life) in general, and particularly in fish schooling, bird flocking and swarm theory. However, it is also analogous to evolutionary computation, and is related to both evolutionary programming and genetic algorithms. Particle swarm optimization [11] is based on the observations of swarms of particles (bees, insects etc.). Particle swarm optimization (PSO) is a swarm intelligence modeled algorithm, that finds the solutions to an optimization problem in a model, or search space and predict the social behavior in the presence of objectives to achieve. The PSO algorithm is a population-based, stochastic computer algorithm that is shaped on swarm intelligence. Swarm intelligence is primarily established on social-psychological propositions and gives an understanding into social conduct. Particle swarm optimization algorithm is basically a maximum minimum algorithm. PSO finds the global best and personal best solutions. Personal best solution is the one that the particle updates when it gets a result better than the previous one. Also, the particle maintains the best solutions of its neighbors, and if it considers the whole swarm as its neighbors, it is called Global best solution. Particle Swarm Optimization is a metaheuristic algorithm and hence it makes few or no assumptions on the problem being optimized. In the PSO algorithm, at first a population of candidates, also called swarm is put in the search space at random locations. After that the particles search their individual solutions and update the personal and global best with each iteration. The algorithm is run over a desired number of iterations or until a fulfilling condition is met. As the personal and global best solutions improve with each iteration, they become a guiding medium for the swarm. Particle Swarm Optimization is the base for many other Swarm Intelligence algorithms.

The pseudo algorithm for PSO is:

1. Initialize the population with particles of random velocities and position.

2. Compute the fitness function for every single particle.

3. Compare the current fitness value of the particles with fitness value of other particles and find the value of P_b.

4. Compare the fitness value and overall previous best value of the population to obtain the value of G_b.

5. By using the equations of PSO, update particle's position and velocity.

6. If the maximum is reached for the number of equations, stop else repeat from step 2.

IV. ANALYSIS OF ACO AND PSO

The ACO algorithm is incited through the behavior of the ant colonies. The indirect transmission between the ants helps them to find shortest course between their nest and the food sources. This attribute of real ant colonies is exploited in ACO algorithm to crack, discrete optimization problems. The PSO technique is shaped on the social behaviors that are observed in animals or insects. Over the time PSO has procured increasing popularity amid researches and practitioners as an efficient and robust technique for solving the difficult optimization problems based population. Both the ACO algorithm [12] and the PSO algorithm [13] are data clustering algorithms by enforcing swarm behavior. ACO is generally more applicable in the areas for problems where source and destination are predefined. Whereas PSO is a clustering algorithm in the areas of multi-objective, dynamic optimization and constraint handling. The ACO algorithm is more suitable for problems that requires crisp outcomes and PSO is useful for problems that are fuzzy is nature.

V. COMPARISON BETWEEN ACO AND PSO

In this paper both the optimization algorithms are compared on the grounds of makespan, cost and deadline. The average makespan of the PSO and ACO algorithms [7] is shown in Fig. 2. It can be seen that, with the expanding quantity of task, PSO takes less time than ACO algorithm. Although ACO performance is very close to PSO, PSO outperformed ACO in all the instances of tasks. The reduction given by PSO is approximately 20% to 34% for the makespan [8]. This indicates that the PSO algorithm take less time to execute in all job sizes than the other method.

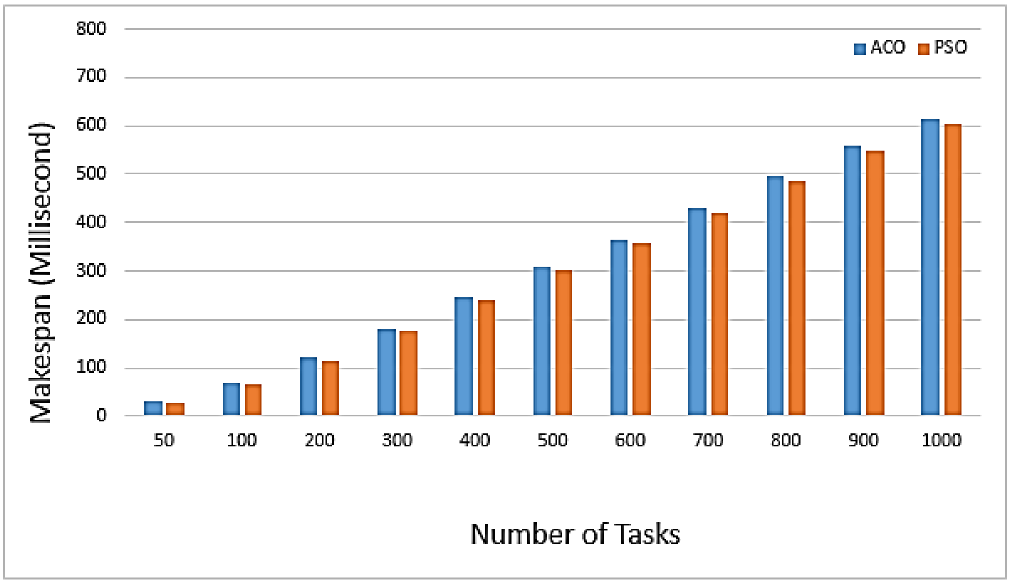

Fig. 2: Comparison of ACO and PSO on the basis of Makespan.

Another measure of comparison used is the cost induced when running the two algorithms on different sizes of resource matrices. A reduction of 20% to 30% for cost was observed in PSO from ACO when comparing on different sizes of resources matrices [8]. The comparison is shown in Fig. 3.

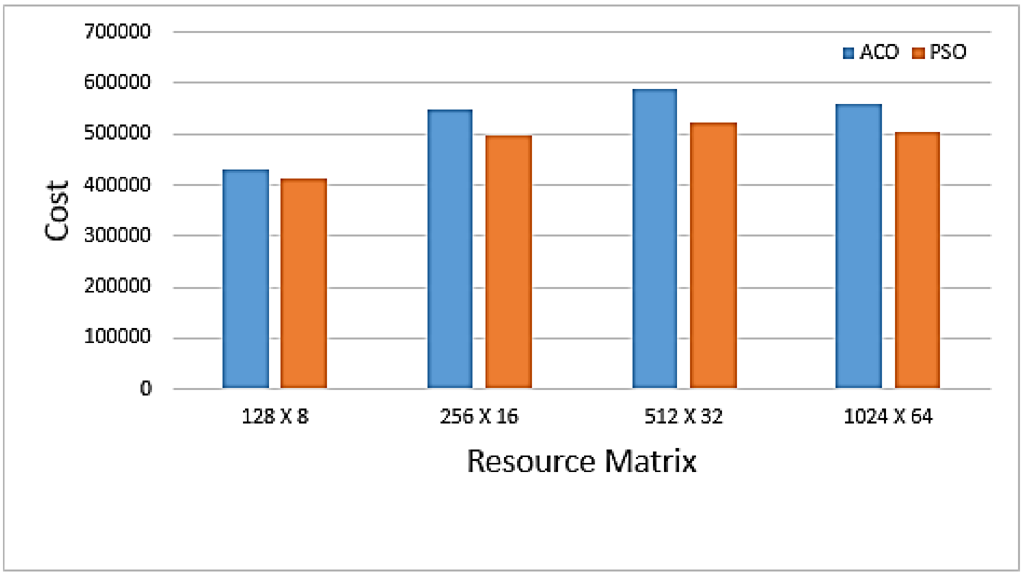

Fig. 3: Comparison of ACO and PSO on the basis of Cost.

The main goal of scheduling is to increase the deadline percentage, which means more tasks are completed before the deadline. The PSO algorithm outruns the ACO algorithm by showing approximately 10% to 35% increase in the Deadline [8]. Fig. 4 shows the comparison on the basis of Deadline.

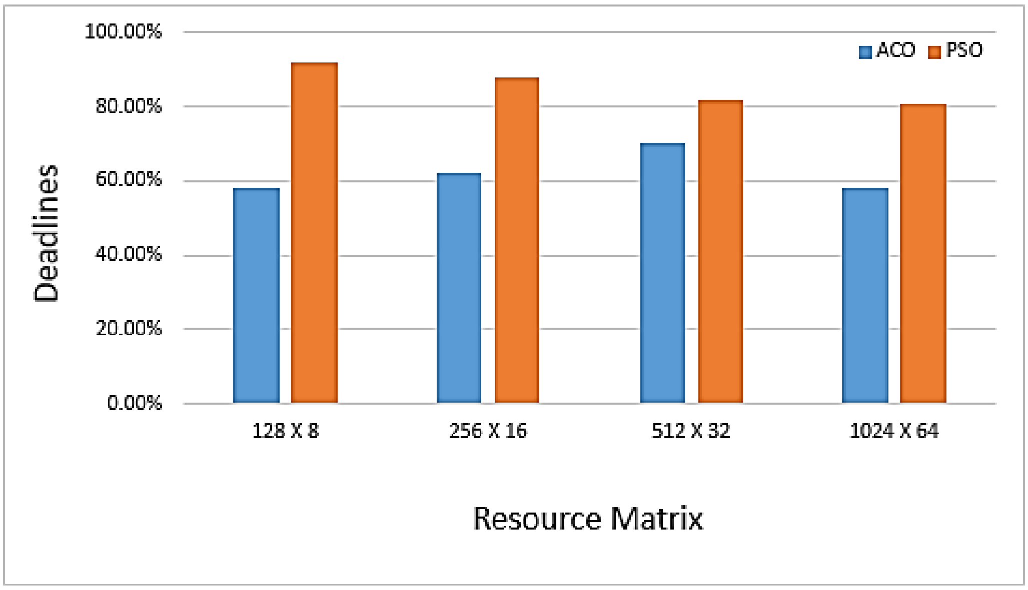

Fig. 4: Comparison of ACO and PSO on the basis of Deadlines.

VI. CONCLUSION

The main goal of task scheduling algorithms is to lower the cost, makespan and escalate the deadline. Outcomes show that PSO is superior to ACO on the grounds of makespan, cost and Deadline because it intelligently integrates different concepts for exploring the search space. It makes use of parallelism technique and cooperative strategies that are used to structure information so as to efficiently provide the near-optimal solutions. The ACO and PSO can be examined for future improvement such that new researches could be focused on to develop better solution by reducing the limitations and improving the effectiveness. Future work can focus on other methods of comparison like reliability and throughput of algorithms.

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical statement: The authors declare that they have followed ethical responsibilities

REFERENCES

[1] Chauhan, Pankaj Kumar, and Poonam Dabas (2016). A Review of Prioritize Task Scheduling in Cloud Computing. *International Journal of Advanced Engineering Research and Applications (IJA-ERA) Volume-2, Issue-1*.

[2] Maruthanayagam, D., & Prakasam, T. A. (2014). Job Scheduling in Cloud Computing using Ant Colony Optimization. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume*, *3*.

[3] Bhatt, K., & Bundele, M. (2013). Study and impact of CloudSim on the run of PSO in cloud environment. *International Journal of Innovations in Engineering and Technology (IJIET)*, *2*(4), 254-262.

[4] Buyya, R., Ranjan, R., & Calheiros, R. N. (2009, June). Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In *High Performance Computing & Simulation, 2009. HPCS'09. International Conference on* (pp. 1-11). IEEE.

[5] Selvi, V., & Umarani, D. R. (2010). Comparative analysis of ant colony and particle swarm optimization techniques. *International Journal of Computer Applications (0975–8887)*, *5*(4).

[6] Shakerian, R., Kamali, S. H., Hedayati, M., & Alipour, M. (2011). Comparative study of ant colony optimization and particle swarm optimization for grid scheduling. *The Journal of Mathematics and Computer Science*, *2*(3), 469-474.

[7] Booba, B., & Gopal, T. (2013). Comparison of Ant Colony Optimization & Particle Swarm Optimization in Grid Environment. *International Journal of Advanced Research in Computer Science and Applications Vol.1, Issue 5*.

[8] Kalpana, C., Kumar, U. K., & Gogulan, R. (2012). A Randomized load balancing algorithm in grid using Max Min PSO Algorithm. *International Journal of Research in Computer Science*, *2*(3), 17.

[9] Tawfeek, M. A., El-Sisi, A., Keshk, A. E., & Torkey, F. A. (2013, November). Cloud task scheduling based on ant colony optimization. In *Computer Engineering & Systems (ICCES), 2013 8th International Conference on* (pp. 64-69). IEEE.

[10] Bilgaiyan, S., Sagnika, S., Mishra, S., & Das, M. (2015). Study of task scheduling in cloud computing environment using soft computing algorithms. *International Journal of Modern Education and Computer Science*, *7*(3), 32.

[11] Al-maamari, A., & Omara, F. A. (2015). Task scheduling using PSO algorithm in cloud computing environments. *International Journal of Grid and Distributed Computing*, *8*(5), 245-256.

[12] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, *1*(4), 28-39.

[13] Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning* (pp. 760-766). Springer US.