International Journal of Advanced Computer Science and Technology.
ISSN 2249-3123 Volume 11, Number 1 (2021), pp. 1-12

http://www.ripublication.com

Hybrid CS+ACO algorithm for Job Scheduling

Sudhanshu Prakash Tiwari[1], Dr. Kapil Kumar Bansal[2]

[1]Research scholar, Mewar university.

[2]Associate professor, SRM university.

Introduction: Evolutionary computation is an optimization method based on these organic development processes. The calculations roused by evolutionary computation are known as evolutionary algorithms. These calculations can be used to solve complex problems. The job scheduling problem is one of the complex problems which are studied intensively in numerous domains. Various methods and techniques are connected to the job scheduling issue to solve the problem, though none of the methods or technique proves to get a solution to the problem in reasonable amount of time. Therefore various algorithms compete to get only better solutions than the already known. This also helps in checking the performance of the algorithms. Biological evolution consists of processes or mechanisms like selection, reproduction, migration and survival of the fittest, which makes the individuals in the population adapt to the changing environment.

Process planning is characterized as the readiness of an arrangement of directions to produce a section or to setup a get together, which will full-fill the design specifications and necessary functional. Process plan incorporates any or the majority of the accompanying data: processing time, setup subtle elements so on.

The goal of process planning [1, 5, 7, 8] action is to decide an ideal procedure plan and the required assembling resource whereas scheduling is worried about the allotment of tasks to assets over some stretch of time. There are so many decisions taken according to the constraints in the process of optimization problem. In optimization problem, the goal of the constraints is either maximize or minimize. The solution can be referred to as a function of certain design variables. The word optimum denotes the ‘maximum’ or ‘minimum’ value based on the given circumstances.

This paper presents the mixture of two algorithms. They are merging their advantages for resolving scheduling problems. Individual job has possible alternative operation sequence requiring alternative machining resources.

The notations are:

i, j: index of jobs (i, j = 1, 2, ..., n)

k: index of machine (k = 1, 2,......, m)

For solving the IPPS problem various assumptions are followed. They are explaining in following:

1. All jobs and machines are independent of each other.
2. All jobs have same priority.
3. At a time each machine can process only one operation.
4. Multiple operations of a job cannot be processed simultaneously.

Various nature inspired algorithms are used to resolve the IPPS problem. Our objective is to apply hybrid ACO to resolve the IPPS problem. ACO is a an distinctive algorithm for some of the reasons like the optimum solution is built not only by a single thing but various things, which traverse the length and size of the scheduling network and then these separately build upon the solution and thus it is used for planning and scheduling.

Cuckoo search algorithm:

Recently settled developmental optimization instrument which is absolutely reliant upon the rearing conduct of the Cuckoo is cuckoo search technique. Like GA and PSO [2], CS [3] is likewise occupant focused calculation is presented in 2009 by Yang and Deb. This calculation is really endless supply of a few new breed that connects with some flying birds. To find nourishment resource for younger ones cuckoos flies in search of better locations .Storage of resources in own home is done by these cuckoos before searching resource of sustenance. Kids of cuckoos will be kept in own homes before searching nourishment resource. For security purpose cuckoo puts their eggs in area homes. If owned of home returned back and found that youngster is not her own then she toss those eggs out.

It can be indicated by the accompanying two ideal thoughts:

a) An egg is anchored by each cuckoo and stores youngster in arbitrary chosen one.
b) Prevalent nature related to eggs with best homes as well as will continue for coming populace of ages;

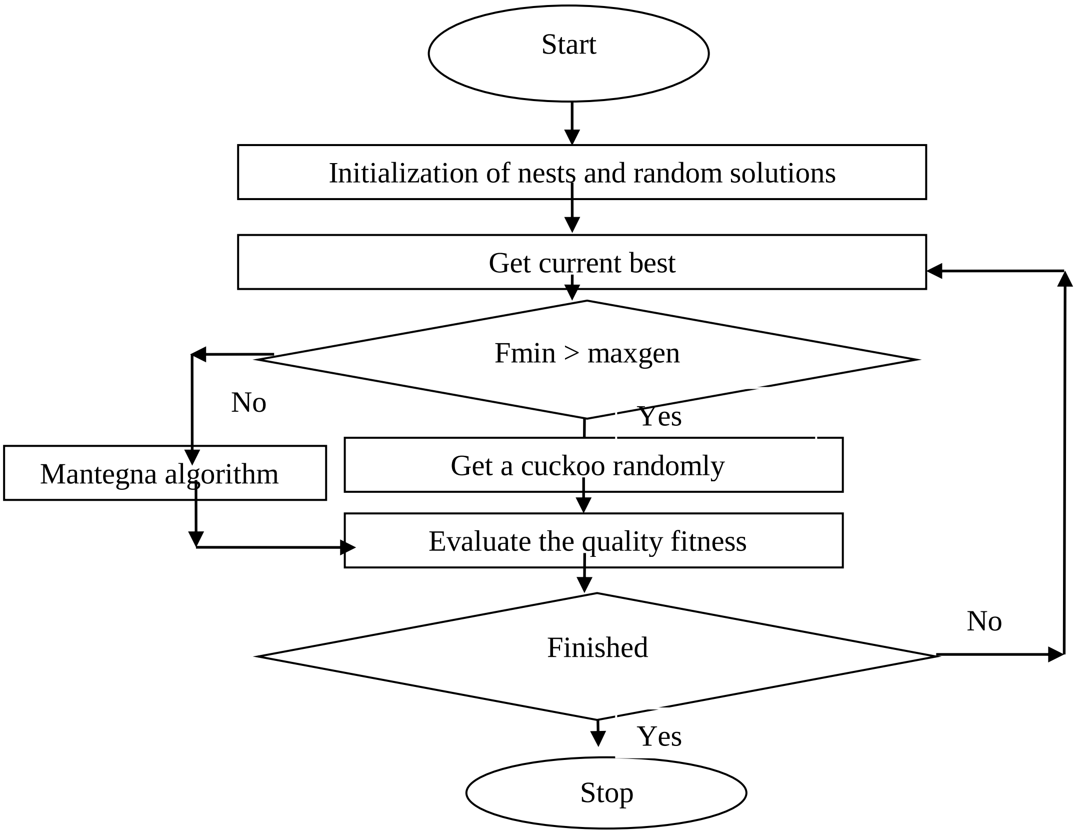

Figure 1. Cuckoo search flowchart

Cuckoo search [3] and PSO [2] has been utilized in various field of processing and building knowledge as far as trying situation. CS and PSO have particularly striking execution over numerous different calculations. That is the reason we are working upon this sort of calculation.

Truth to be told, experts suggest that PSO can't demonstrate that the assembly of global_search issue and consequently there is no any component in-worked in PSO to discover universally.

Adaptation and Evolution: There are some female cuckoos having certain capabilities to specialize in mimicry in colors of another birds and also pattern of the egg of a few chosen egg host birds' species.

Consequence of CS: Some host bird can direct engage conflicts with the interrupting cuckoos. For instance, if a host flying creatures finds the eggs are not their own particular, it will either discard these outsider eggs or just relinquish its home and fabricate another home anyplace else.

Representation of CS: In a nest, each and every present or available egg provides the solution and cuckoo's egg provides a new solution. This technique is used to get better with worst solution that is presented in nests. In simple words, we can say that each and every nest has on egg. The algorithm become more complicated when each and every nest has more than one eggs represent the set of solutions.

Levi flight: In simple words, we can say that Levi flight is a random selected path or walk that is used by cuckoos to find the resources.

Ant colony optimization: Ant Colony Optimization [4] is propelled by the scrounging or nourishment looking conduct of genuine insect provinces. The primary concern that works in ACO is mimicked pheromones that pull in ants to best trail in the chart since they pick the way in view of the pheromone thickness. Ants are normally visually impaired, hard of hearing and stupid so when they move, leave pheromone (substance material) in its way. Based on this pheromone thickness different ants take different way. ACO [6, 9] has mathematical parameters (α),(β), (ρ) and (Q) which are used for influence of pheromone on direction, influence of adjacent node distance, pheromone evaporation coefficient and pheromone depositing factor . This algorithm updates the trail on the basis of current pheromone density.

At the point when a fractional result is build by an ant stopover in city I, the likelihood proceeded to the following city j neighboring on city I is given by

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \in allowed_k} [\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}} & if\ k \in allowed_k \\ 0 & otherwise \end{cases} \tag{1}$$

Where $\sum_{i}^{m} x_{i,j} = 1, \forall j \in T$ τ_{ij} is the acuteness of pathways between arrow (i.j) and η_{ij} is the heuristic evident of arrow (i, j), and $\eta_{ij}=1/d_{ij}$. Let α as pheromone influence factor, β is local node influence, and remaining cities to be stayed represented by $J_k(i)$. After each ant completes its tour, the following equality represents pheromone quantity on every path:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{2}$$

where ρ is pheromone evaporation coefficient and

$$\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}(t) \tag{3}$$

$$\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau^{k}{}_{ij}(t) \tag{4}$$

These are required for calculation of ACO pheromone thickness and other related factors. (1- ρ*)* is the pheromone decay parameter (0<ρ<1) where it represents the trail evaporation when the ant chooses a city and decide to move. L_k is the length of the tour performed by ant *k* and *m* is the number of ants.

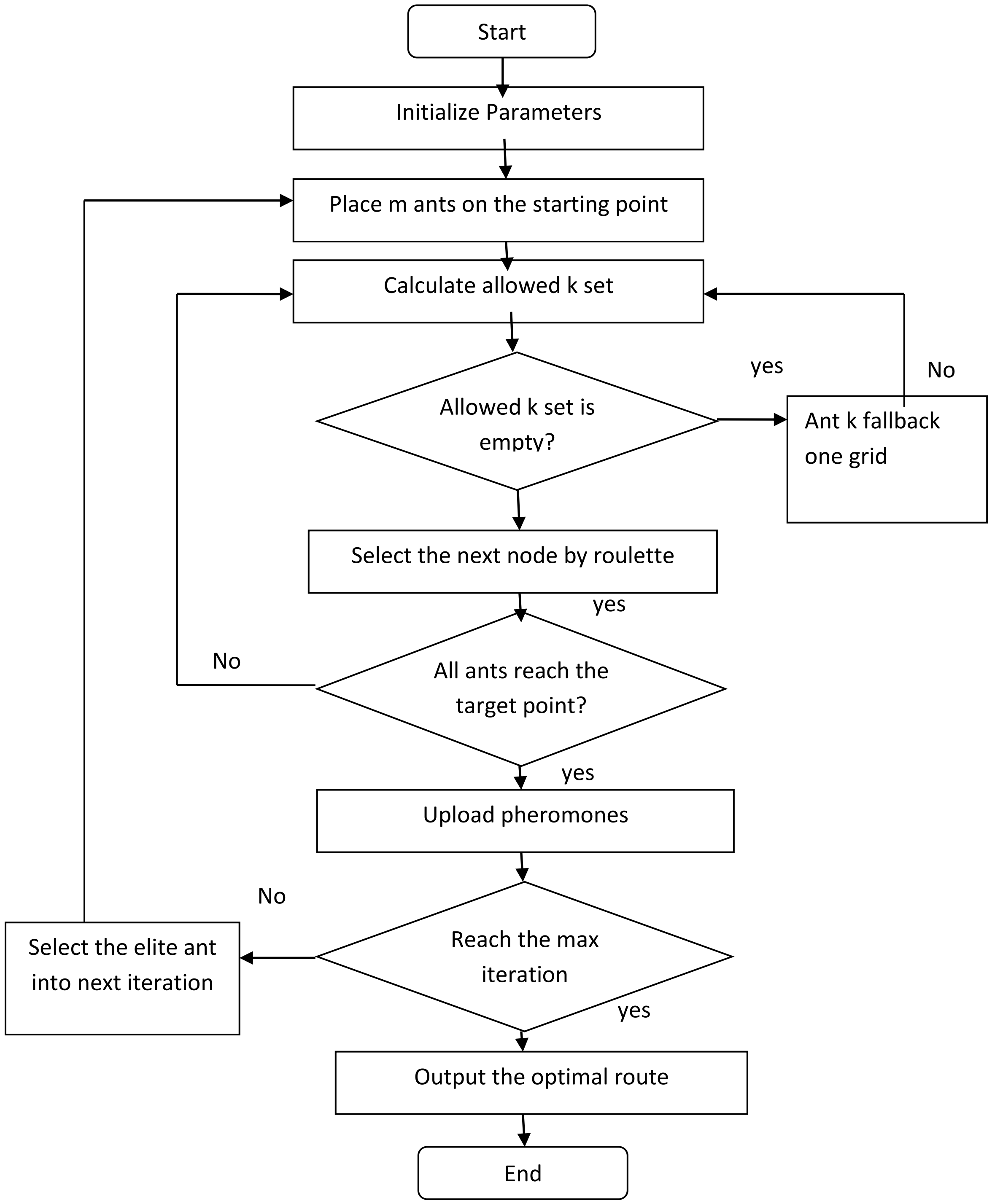

Figure 2 Flow chart of Ant Colony Optimization

- It has search space large.
- It provides slow global best solution.

Proposed Algorithm: Hybrid algorithm is combination of Ant Colony Optimization (ACO) and cuckoo search (CS).

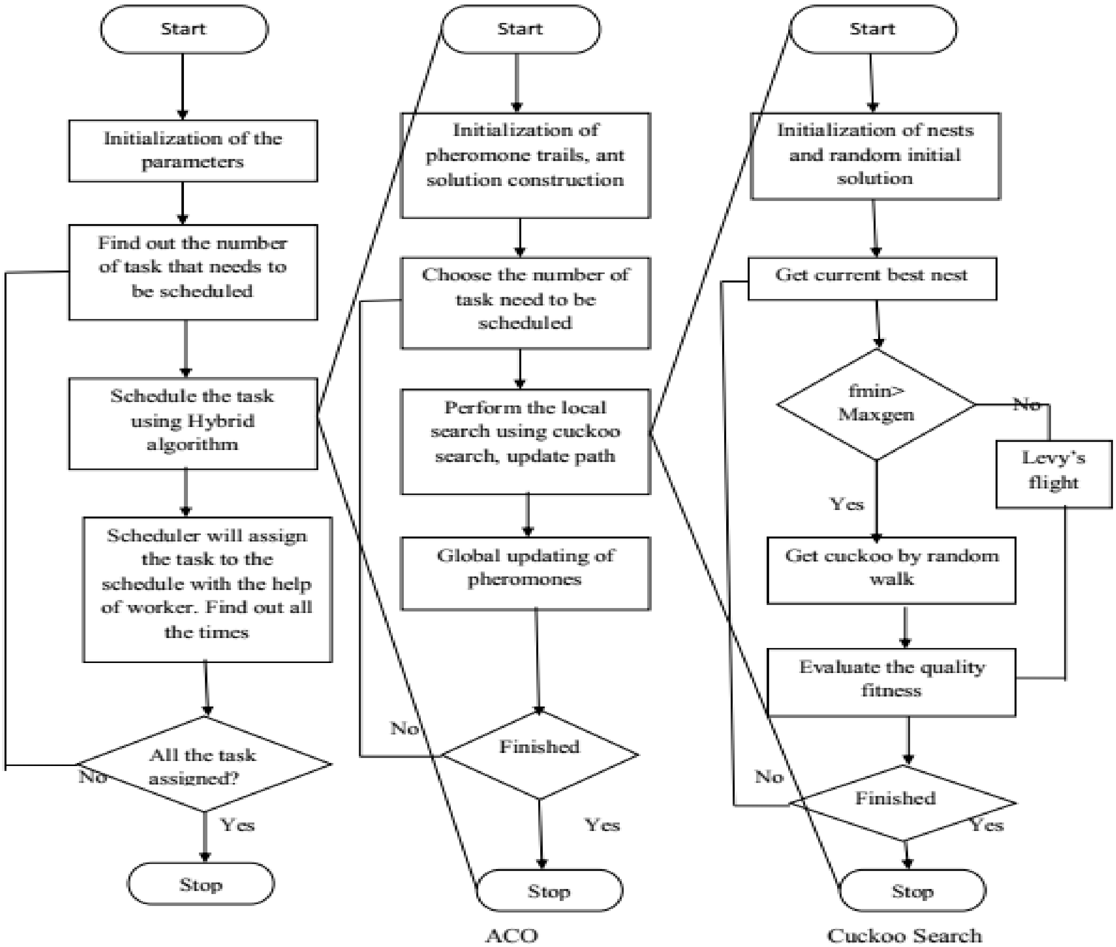

Figure 3 Flowchart of Hybrid algorithm

Algorithm has benefit of both algorithms. In scheduling search has to perform faster for solving the combinational optimization problem, but in pheromone will be deposited. It is a big disadvantage of ACO. It becomes a luring component for the artificial ants. Hence local search will be performing at the faster rate than in the ACO. CS is used for overcoming the disadvantage of ant colony optimization algorithm.

Experimental Setup: For high quality outcome various computations have few control parameters. So suggested algorithm also have few control parameters. After tremendous literature survey related parameters are found and applied. These values are benchmark values and applicable for trials. These are: pc (crossover probability), pm (mutation probability), and in cuckoo search value of pa is taken from 0 to 1. For

Mantegna search the value of alpha is taken from 0.3 to 1.99. Another parameter is maximum number of population that is from 10 to 100.

Results: Comparative results are shown by considering 20 jobs on 15 machines, 30 jobs on 15 machines, 30 jobs on 20 machines, and 50 jobs on 15 machines. These benchmark problems are adopted from Taillard benchmark [211]. In which every instances can be divided into two categories: number of jobs and number of machines. The makespan time taken by each job is measured many times and the average time is taken in this work. The following figures are showing the result of 20*15 taillard benchmark problem. In which 20 jobs are executed on 15 machines. And the result of hybrid ACO is, the makespan time and cost is lesser than ACO. The algorithm is giving the accurate result of taillard benchmark problem. And same apply the different taillard benchmark problem such as 20*15, 30*15, 30*20, and 50*15. First we generated random problem instances for 30 jobs on 20 machines, 50 jobs on 15 machines and 100 jobs on 20 machines.

Table 1: Comparative results of ACO and Hybrid ACO

Taillard benchmark problem	ACO		Hybrid ACO	
	Makespan Time	Cost	Makespan Time	Cost
20*15	37.1137	185.5687	35.8269	179.1352
30*15	1.2095e+03	6.0474e+03	165.464951	827.3252
30*20	1.2095e+03	6.047e+03	413.692287	2.0685+03
50*15	1.2095e+03	6.047e+03	1115.393931	5.5770e+3

The makespan time taken by each job is measured many times and the average time is taken in this work. The result shows in table 1 the hybrid ACO give the better result than ACO i.e. Makespan time and cost.

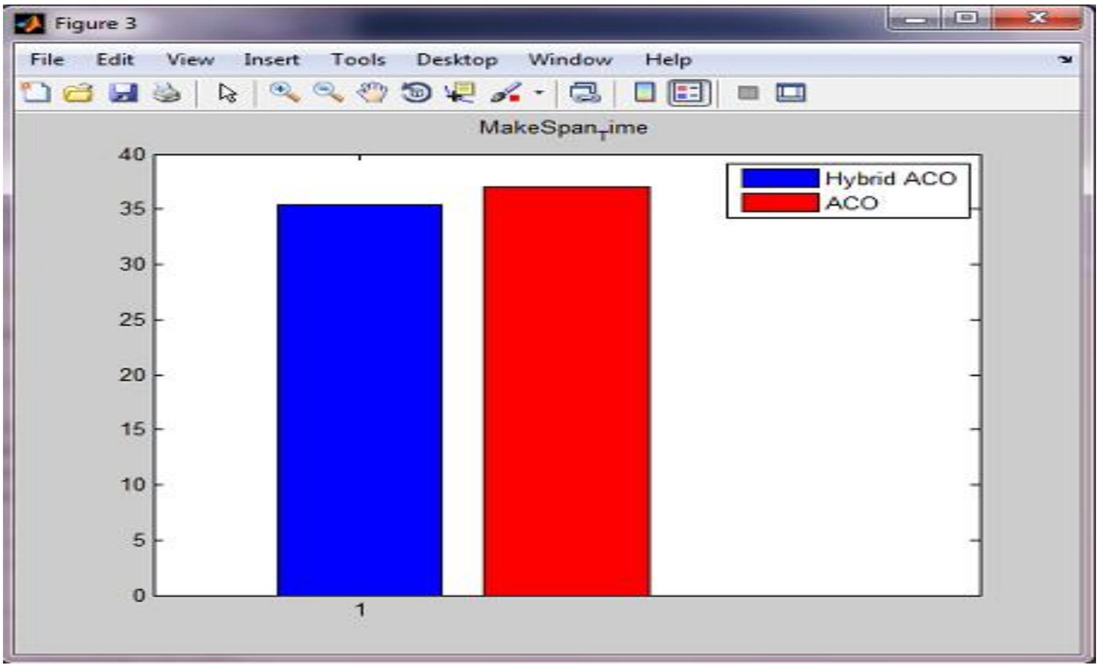

Figure 4 Makespan time of 20*15 taillard benchmark problem

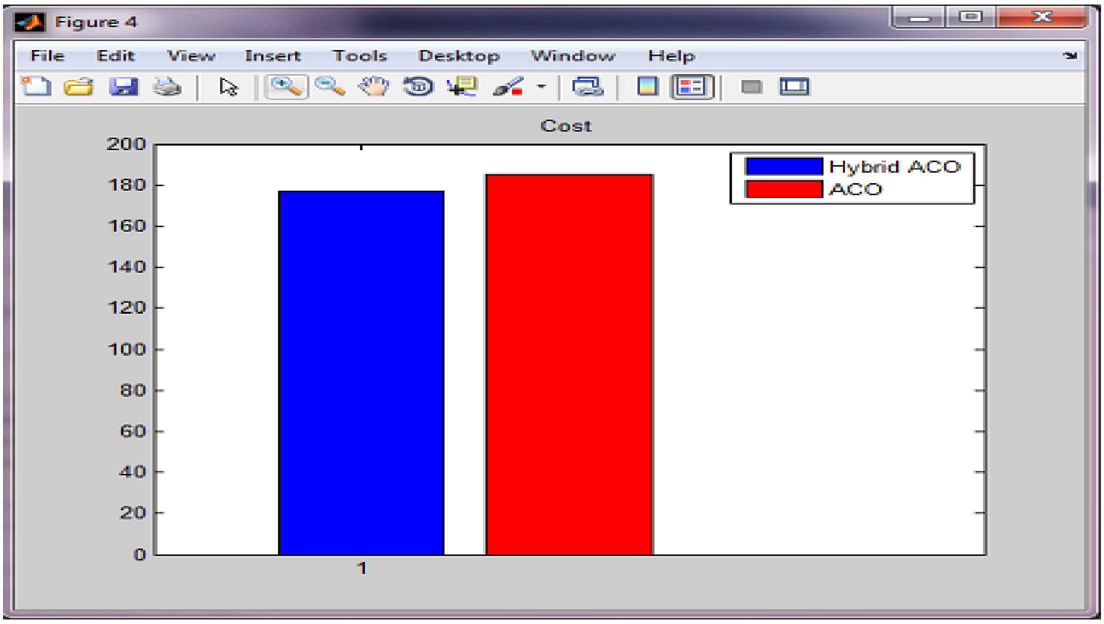

Figure 5 Cost of 20*15 taillard benchmark problem

Figure 6 Makespan time of 30*15 taillard benchmark problem

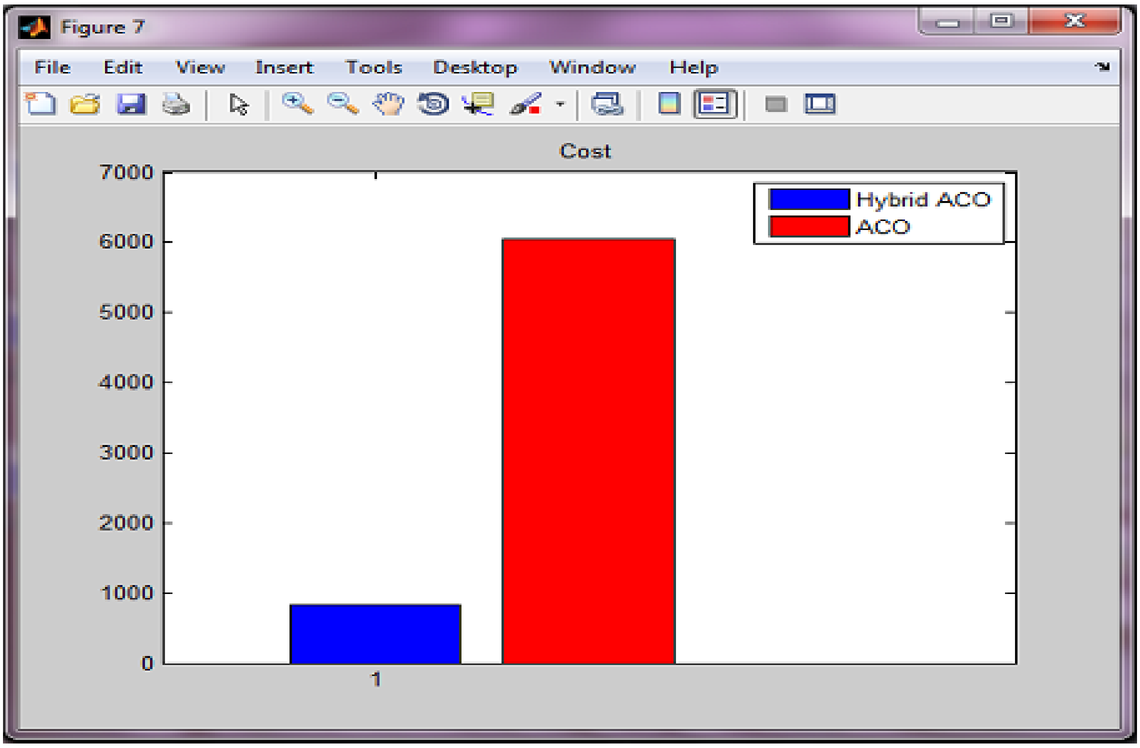

Figure 7 Cost of 30*15 taillard benchmark problem

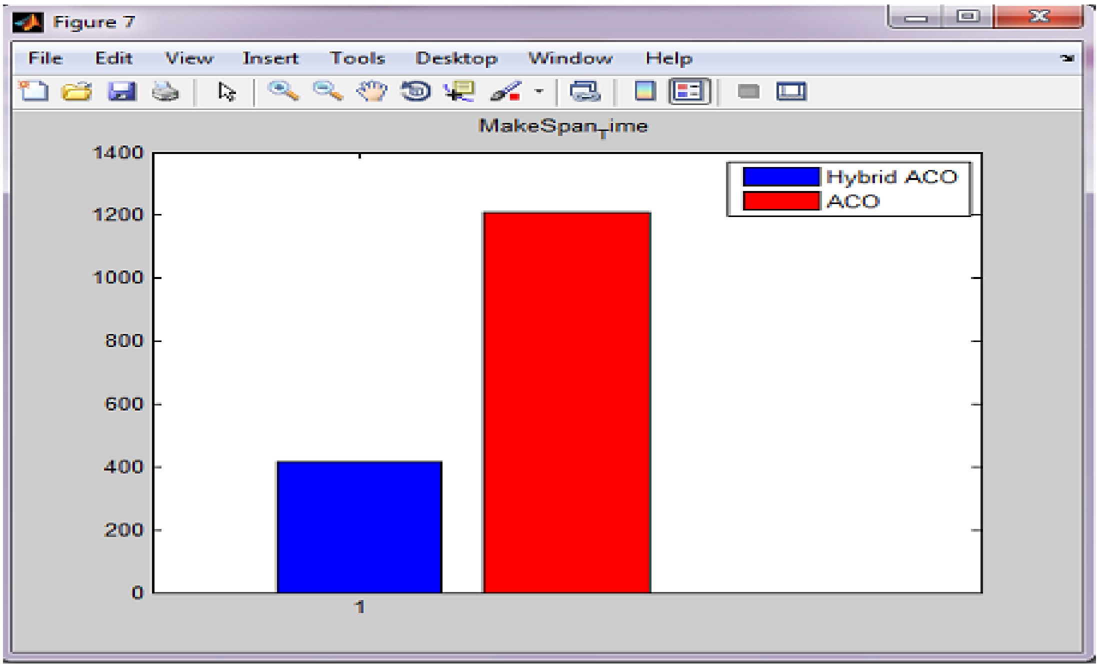

Figure 8 Makespan time of 30*20 taillard benchmark problem

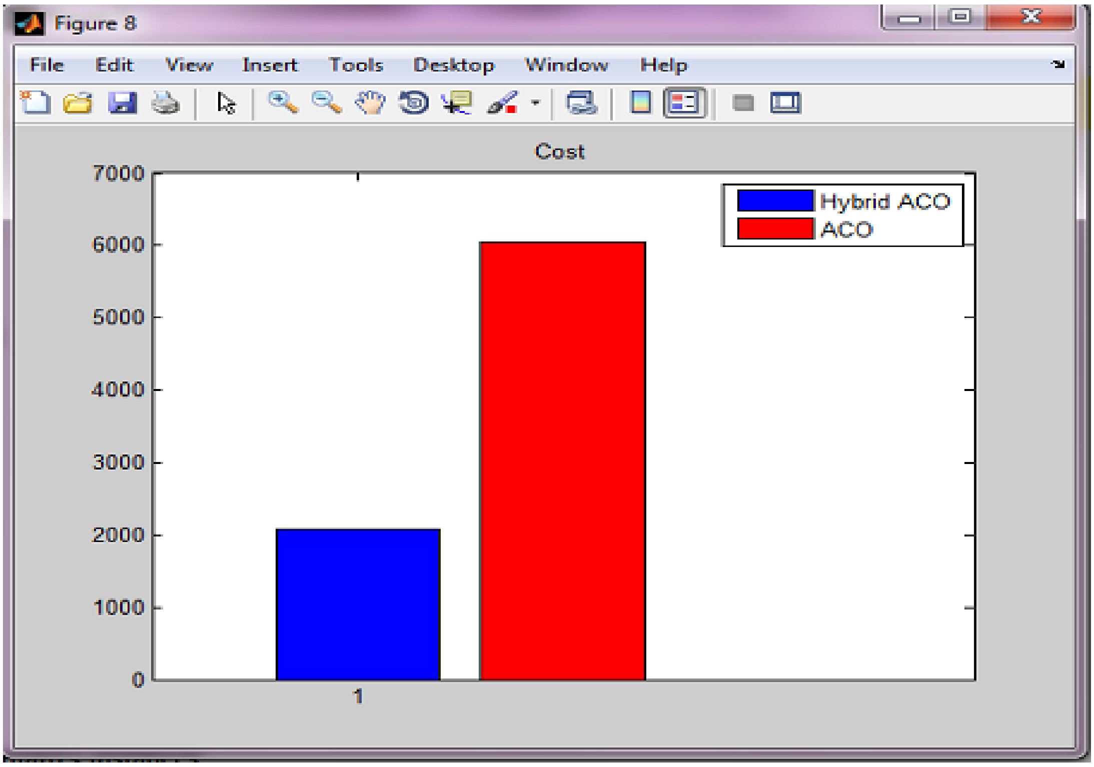

Figure 9 Cost of 30*20 taillard benchmark problems

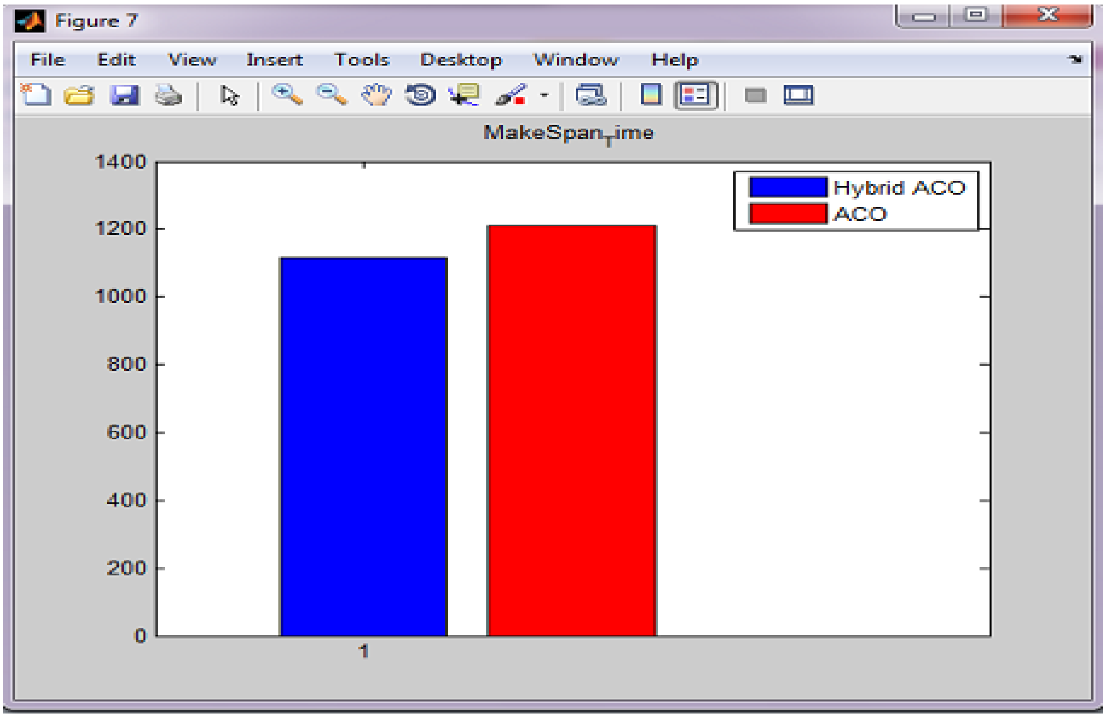

Figure 10 Makespan time of 50*15 taillard benchmark problem

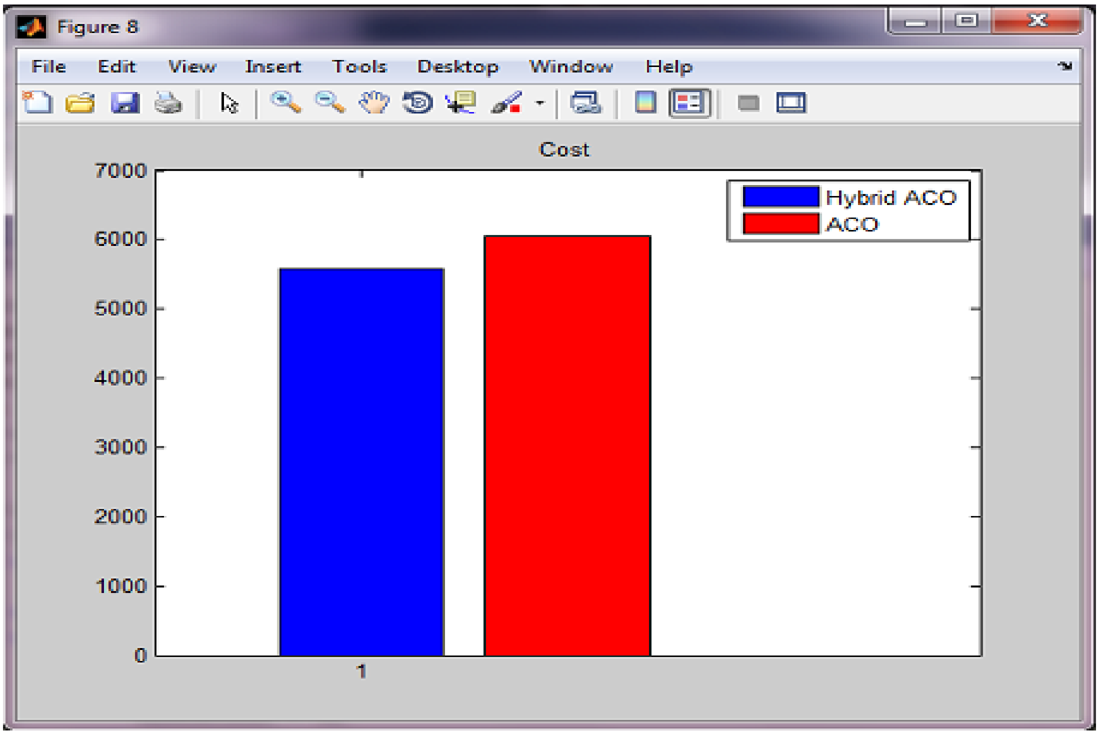

Figure 11 Cost of 50*15 taillard benchmark problem

Conclusion: Paper proposed about hybrid algorithm composed of ACO+CS hybrid algorithm and compared with ACO. Parameters selected for both algorithms have played vital role in execution. Varied number of jobs and machines are taken and outcome is compared. New hybrid metaheuristic methodology specifically Enhanced PSO calculation is created and actualized for job shop scheduling.

References

1. Gabi, D., Ismail, A. S., Zainal, A., Zakaria, Z., & Al-Khasawneh, A. (2018). Hybrid Cat swarm optimization and simulated annealing for Dynamic task scheduling on Cloud computing environment. *Journal of ICT, 17*(3), 435-467.
2. Bhatt, M. C. (2017, October). Hybrid PSO for job scheduling to minimize makespan in heterogeneous grids. In *2017 2nd International Conference on Communication and Electronics Systems (ICCES)* (pp. 586-591). IEEE.
3. Chhabra, A., Singh, G., & Kahlon, K. S. (2020). Performance-aware energy-efficient parallel job scheduling in HPC grid using nature-inspired hybrid meta-heuristics. *Journal of Ambient Intelligence and Humanized Computing*, 1-35.
4. Thiruvady, D., Blum, C., & Ernst, A. T. (2020). Solution Merging in Matheuristics for Resource Constrained Job Scheduling. Algorithms, 13(10), 256.
5. Guddeti, R. M., & Buyya, R. (2017). A hybrid bio-inspired algorithm for scheduling and resource management in cloud environment. *IEEE Transactions on Services Computing*.
6. Guo, Q. (2017, April). Task scheduling based on ant colony optimization in cloud environment. In *AIP Conference Proceedings* (Vol. 1834, No. 1, p. 040039). AIP Publishing.
7. Chhabra, A. (2018). Hybrid PSACGA Algorithm for Job Scheduling to Minimize Makespan in Heterogeneous Grids. In *Industry Interactive Innovations in Science, Engineering and Technology* (pp. 107-120). Springer, Singapore.
8. Chhabra, A., Singh, G., & Kahlon, K. S. (2020). Multi-criteria HPC task scheduling on IaaS cloud infrastructures using meta-heuristics. Cluster Computing, 1-34.
9. Wei, X. (2020). Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing. Journal of Ambient Intelligence and Humanized Computing, 1-12.