MODEL ANALYSIS ON JOB SHOP SCHEDULING IN AUTOMOBILE INDUSTRY USING ANT COLONY OPTIMIZATION AND PARTICLE SWARM OPTIMIZATION

Paridhi Singh, Prapti Goyal, Varun S. Gajendragadkar, Sayantan Pal , Vignesh S., Supreeta Balasubramanian, Sanjiban Sekhar Roy

School of Computer Science and Engineering, VIT University, Vellore

Corresponding author: Sanjiban Sekhar Roy, sanjibanroy09@gmail.com

ABSTRACT: Job Shop Scheduling is an optimization problem and is considered to be one of the most daunting combinatorial problems. It can be used to maximize the productivity in many industries, particularly in the automobile industry. There are two finite sets involved in this problem, one for the number of machines and the other for the number of jobs which each machine has to do. The real challenge is to find out the most efficient way to complete these tasks. This problem remains one of the most discussed problems, with researchers from all over the world discovering new and different methods to solve it. A plethora of methods and algorithms, including different types of queuing algorithms and even some genetic algorithms have been used to solve this problem. The practicality of the problem further makes it interesting and the computer science community is motivated to make the solution even more efficient. In this paper, we have used Ant Colony Optimization and Particle Swarm Optimization, techniques which are probabilistic and iterative respectively to solve the problem. The tool used for this purpose is MATLAB. After tabulating and visualizing the results, it is found that the Particle Swarm Optimization is much more efficient than the Ant Colony Optimization method. The processing time of the Ant Colony Optimization is approximately four times more than that of the Particle Swarm Optimization.

KEYWORDS: Job Shop Scheduling, Automobile Industry, Ant Colony Optimization, Particle Swarm Optimization, MATLAB.

1. INTRODUCTION

Job Shop is an organization that has various work stations, meant to perform operations on objects, i.e., they accept a contract of producing an object by putting them through a series of operations in return for a fee. To profit from this business, it is required for them to be able to conduct these operations at a cost lesser than the contracted amount they receive. Whereas, scheduling is the day to day regulation of these operations in order to obtain maximum net profit. According to Grefenstette [Gre14], the scheduling factor becomes more difficult by the fact that the factors taken into account in one's schedule may change, for instance, machines breakdown, workforce diminish unexpectedly, delay in supplies and so on. The Job Shop Scheduling system must be able to generate schedules to fill job shop contracts by keeping the revenue profit to its maximum. This task is the most formidable and onerous for the scheduler, who must be able to react quickly to these changes. This makes the research to reduce the complexity of the task very essential and of utmost importance. The first competitive solution to this problem was submitted by Graham [Gra66]. Different studies in the past which were carried out in this field have used natural processing to obtain the solutions. A genetic algorithm based solution was put forth by Yamada & Nakano [YN97]. Through this study, we find out the best suited algorithm for the problem data presented to us by comparing each of the two models. Ant Colony Optimization (ACO) is a probabilistic algorithm which is used to find the optimum path in a system. Dorigo [CDM92] states that in his study this method has been inspired from the real life behavior of ants. The basic premise of the algorithm is as follows: Ants, when in their natural habitat, always wander randomly until they find food. After managing to procure the foodstuff, they return to their colony, leaving behind a chemical trail. This chemical is called pheromone. The ants which hunt for food just after the pioneering ants follow this pheromone trail. This goes on for a certain time, until the pheromones evaporate. It is known that every ant which treads any path leaves behind a pheromone trail. Hence, the path which has been used the most will take the highest time to evaporate. Ants are most likely to choose the shortest path in order to keep their effort minimal. Hence the path in which the pheromone trail takes the highest time to evaporate has to be the shortest path. This eventually leads all the ants to follow the same path, which also is the smallest one. Ant Colony Optimization is an optimizarion algorithm, and can be considered as a sub-part of machine learning. As explained by Roy et al. [R+13], Optimization is important in machine learning.

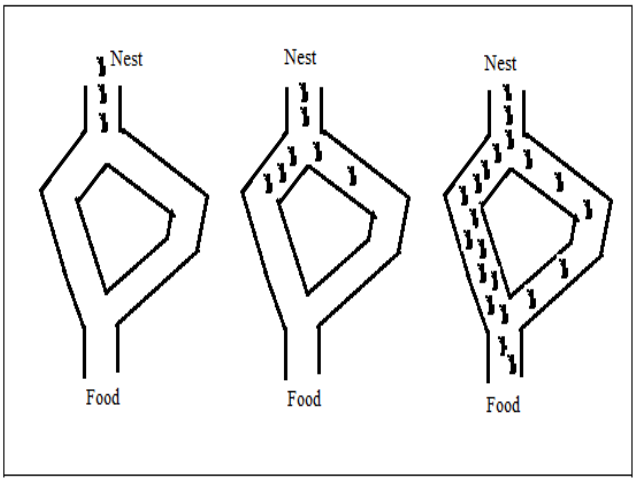

Fig. 1. Sketch Map of the Ant Theory

Particle Swarm Optimization is an optimization algorithm which uses iterations to improve the solution. This algorithm solved the problem by moving the candidate solutions by applying common mathematical formulae over the candidate solution's velocity and position. The candidate solutions are called particles, hence the name Particle Swarm Optimization. The movement of every particle depends on its local best known position and has its position constantly updated as and when better positions are found by other particles. This influence of the local best known position plus the dependency on other particles helps the swarm of particles move towards the best possible solution of the system. PSO was originally proposed by Kennedy, Eberhart and Shi. [KES01] They compared the particles to birds and the swarm to a flock of birds. The flock of birds arrives near a part in the ocean which is stocked with food. The bird closest to the food chirps the loudest, influencing the other birds to change their direction. If any of the other birds, while turning, comes closer to the food source, that bird starts chirping louder than the previous one, making the other birds turn in his direction as mentioned in the study of Kennedy & Eberhart [KE95]. This continues until the swarm reaches the food source, thus achieving its objective. They say that the PSO model is based on swarm intelligence and the swarm will always take the shortest path required. PSO is another optimization technique now widely used in machine learning. Optimization in machine learning has been mentioned in the study of Roy et al. [RVK16]. A similar use of optimization has been discussed in classification of spam by Roy & Vishwanatham [RV16].

Structure of the rest of the paper is described ahead. Section 2 deals with the different methodologies used in solving the problem, thus introducing the framework of the project. Section 3 talks about the data set used and the tool used to showcase the algorithms. Section 4 compares the different methods used in this paper, along with the analysis of the data source referred and the outcomes of the experimental methods.

2. METHODOLOGY

The two methods used, as described earlier for solving the problem, are Ant Colony Optimization and Particle Swarm Optimization. Both are optimization algorithms, working on probabilistic and iterative systems respectively.

2.1. Ant Colony Optimization

To solve the Job Shop Scheduling Problem using Ant Colony Optimization, the process is similar to that of finding the distance between two nodes in a weighted tree graph. In the initial state, all ants start at level 0 of the tree, indicating that no job has been scheduled yet. Every route is given an initial pheromone value of one, signifying that the paths have not yet been used. The pheromone values reduce by 40% after every iteration. The next task in the algorithm is to find the local search criteria. This depends on two factors, namely the pheromone levels and the cost of moving from one level to the next one. This cost is calculated by knowing the extra amount of time required for including the next job in a certain machine. This cost can be zero. The most important question faced in this algorithm is whether to use previous experience or to go for the next scheduling, as these are the only two options. Like in any optimization algorithm, it is needless to say that the path which will take the least time needs to be chosen. There are three steps in solving this problem.

1. A rand value is generated to compare with r_0
2. If this rand value > r_0, then the local search will go for the route that has the highest amount of pheromone deposits
3. Else, the local search does a roulette wheel selection based on (pheromone/ (route - cost + 1)). Roulette wheel selection (also called Fitness proportionate selection) is a genetic operator used for potentially useful solutions in recombination.

Pheromone deposition is another important aspect in this algorithm, as this actively determines the path the algorithm will take. Pheromone is only deposited by the best ants in each round. The amount of pheromone deposited is (1/ (best ant total cost)).

2.2. Particle Swarm Optimization

The usage of Particle Swarm Optimization in solving the Job Shop Scheduling Problem is starkly similar to the system of Ring Topology, a type of network topology. Each particle is connected to four of its adjacent neighbors. Every particle calculates its own speed depending on the best solution as suggested by

its neighbors and also the one by itself in each iteration. Speed and location is defined for all n dimensions. In the initial state, all particles are given the speed 0 at all n directions, indicating that nothing has been initiated at that point of time. Every particle is assigned a random position between 1 and m in all the dimensions. The local best solution is the same as the particle's location in this case. Neighbor best solution is the best solution among the four neighbors of the particle, based on the neighbor index. The local search criteria is the next logical step. To calculate the speed of each particle, its speed and the best speed of its neighbors is considered.

$$Vt+1i = w \times vti + c1r1i(pbestti-xti) + c2r2i(Nbestti-xti) \quad (1)$$

The previous location and its new speed are then added to get the new solution. The local best and neighbor best are updated once the new cost and speed are lower than the previous best cost and speed.

$$xt+1i = xti+vt+1i \quad (2)$$

To reduce the runtime load requirement, an asynchronous update method is used. Here, the neighbor best is updated when all particles finish their calculations for the present iteration. The termination of the algorithm occurs when a set number of iterations are performed by a set number of particles. The amount of exploration is defined by the number of particles and the amount of exploitation is defined by the number of iterations.

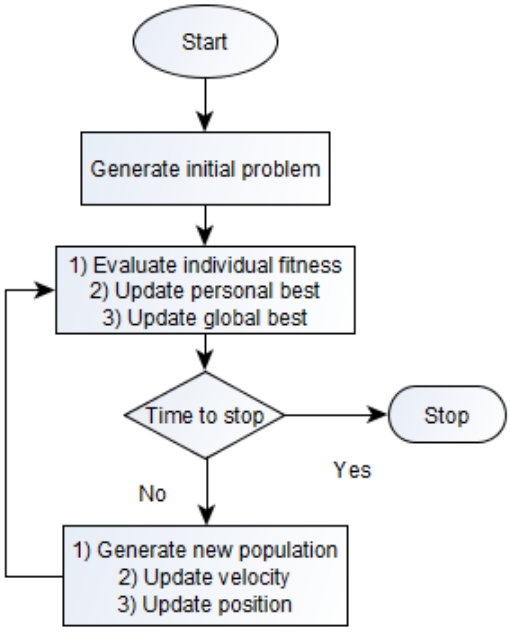

Fig. 2. Flow Chart of Particle Swarm Optimization

3. DATA SET AND TOOLS USED

The Data Set used is of four, five and six number of jobs. The results have been tabulated and visualized in the form of graphs. We have used MATLAB to implement these algorithms and get the results.

4. EXPERIMENTAL RESULT AND DISCUSSION

We have used the algorithms of Ant Colony Optimization and Particle Swarm Optimization to solve the Job Shop Problem. We have constructed the algorithms in MATLAB and have used the tool extensively to visualize the results in the form of graphs and get the necessary outputs.

Table 1. Ant Colony Optimization

Number of Jobs	Time (seconds)
4	88.93
5	89.44
6	90.47

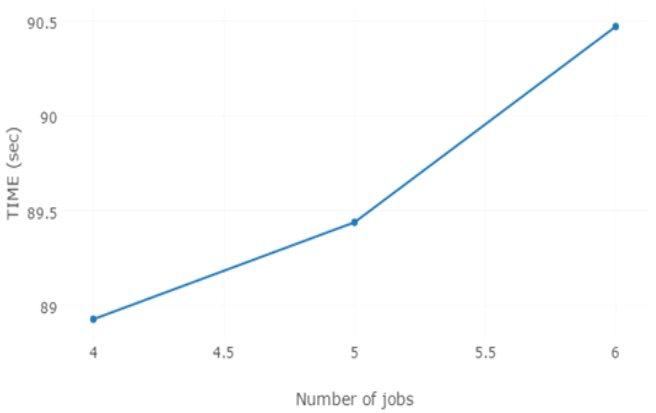

Fig. 3. Graph for Ant Colony Optimization

As explained earlier, we have used the dataset of four, five and six jobs for both the algorithms. By using the ACO, we have found out that as the amount of jobs increases, so does the time taken to solve the problem. For completing four jobs, the algorithm required approximately 88.93 seconds. For the fulfillment of five jobs, the algorithm took approximately 89.44 seconds and for six jobs, it required approximately 90.47 seconds.

Table 2. Particle Swarm Optimization

Number of Jobs	Time (seconds)
4	21.61
5	21.73
6	21.88

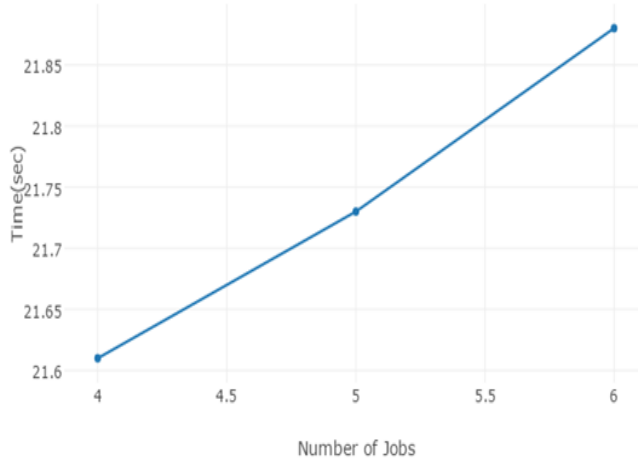

Fig. 4. Particle Swarm Optimization

The algorithm for Particle Swarm Optimization was also done using MATLAB. The same dataset, as that of ACO, was fed to this algorithm. The execution time for PSO model was very less compared to the ACO model. When given four jobs, the algorithm required approximately 21.61 seconds to complete all of them. On giving five jobs, the algorithm took approximately 21.73 seconds to complete the task and for six jobs, the algorithm took 21.88 seconds.

Table 3. Comparison in execution time

Number of Jobs	Time ACO (seconds)	Time PSO (seconds)
4	88.93	21.61
5	89.44	21.73
6	90.47	21.88

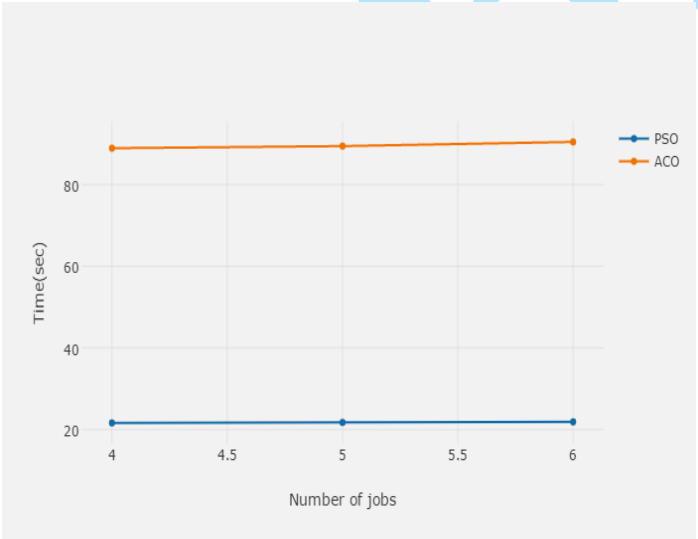

Fig. 5. Graph for the comparison in execution time for ACO and PSO

The time taken by PSO is less than 1/4th of the time taken by ACO. This indicates that the PSO method is more efficient than the ACO method in solving the Job Shop problem. This can be attributed to the fact that in the ACO method, it requires considerable time for a strong pheromone trail to be available for the remaining mass of the solution. As a result, in order to complete the problem, initial few solutions will have to go in a non-optimum way, resulting in a considerable loss of time. In the case of PSO, this problem doesn't exist. As the algorithm is completely based on swarm behavior, every individual solution turns in the way of optimum path. Thus, at every point of time, the swarm moves closer to the optimum solution and hence gets there faster. As a result of which, the PSO reaches its destination much earlier than the ACO.

5. CONCLUSION

In this paper, we have applied an original heurisitic model, i.e., Ant Colony Optimization and a Particle Swarm Optimization model to the same data of job shop scheduling problem. Both, the Ant Colony Optimization and the Particle Swarm Optimization algorithm are found to be the excellent ways for solving the Job Shop Scheduling problem but as efficiency of any model depends on its execution time as well, Particle Swarm Optimization model seems to have won the battle. Kennedy & Ebehart [KE95] stated in their study that Particle swarm optimization is an extremely simple algorithm that seems to be effective for optimizing a wide range of functions.The algorithms have been coded in MATLAB and the results have been tabulated and visualized using its graph library. The graphs then have been compared along with the tables to find out the more optimum algorithm.

REFERENCES

[CDM92] **Alberto Colorni, Marco Dorigo, Vittorio Maniezzo** – *An investigation of some properties of an "Ant algorithm"* in Proceedings of parallel problem solving from nature conference, 1992.

[Gra66] **R. L. Graham** - *Bounds for Certain Multiprocessing Anomalies*, In Bell Labs Technical Journal, 1966.

[Gre14] **J. J. Grefenstette** - *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, 2014.

[KE95] **James Kennedy, Russell Eberhart** – *A New Optimizer Using Particle Swarm Theory*, in IEEE Xplore, 1997.

[KES01] **J. Kennedy, R. Eberhart, Y. Shi** – Swarm Intelligence, 2001.

[RV16] **S. S. Roy, V. M. Viswanatham** - *Classifying Spam Emails Using Artificial Intelligent Techniques*. International Journal of Engineering Research in Africa, 22, 2016.

[RVK16] **S. S. Roy, V. M. Viswanatham, P. V. Krishna** - *Spam detection using hybrid model of rough set and decorate ensemble*. International Journal of Computational Systems Engineering, 2(3), 139-147, 2016.

[R+13] **S. S. Roy, V. M. Viswanatham, P. V. Krishna, N. Saraf, A. Gupta, R. Mishra** - *Applicability of rough set technique for data investigation and optimization of intrusion detection system*. In International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (pp. 479-484). Springer Berlin Heidelberg.

[YN97] **Takeshi Yamada, Ryohei Nakano** – *Genetic algorithms in engineering systems*, Chapter 7: Job-shop scheduling (pp. 134–160). The Institution of Electrical Engineers, 1997.