International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-2, December 2019

An Improved Ant Colony Optimized Tabu Search Algorithm for Makespan Improvement in Job Shop

K.Sathya sundari, V, P, Eshwaramurthy

Abstract: ***In industries, the completion time of job problems is increased drastically in the production unit. In many existing kinds of research, the completion time i.e. makespan of the job is minimized using straight paths which is time-consuming. In this paper, we addressed this problem using an Improved Ant Colony Optimization and Tabu Search (ACOTS) algorithm by identifying the fault occurrence position exactly to rollback. Also, we used a short term memory-based rollback recovery technique to roll back to its own short term memory to reduce the completion time of the job. Short term memory is used to visit the recent movements in Tabu search. Our proposed ACOTS-C_{max} approach is efficient and consumed less completion time compared to the ACO algorithm.***

Keywords: Ant colony optimization, job shop scheduling, short-term memory, tabu search

I. INTRODUCTION

The job shop scheduling problem (JSSP) is defined as the jobs assigned to some resources at a particular time. Each job consists of some specific operations. Using that operation, jobs are scheduled for the resources. According to JSSP, the jobs are unable to complete on time. The industry should provide good service to consumers with on-time delivery of products. Good production scheduling is needed to make on-time delivery of products to the consumers. Jobs should be processed without interruption on each machine. Each job should be processed after

The preceding operation is over. This is the fair scheduling. In this work, a novel job shop scheduling technique is proposed. Tabu search is one of the heuristic search methods used for an improved solution. Tabu list consists of two types of memory they are Long-term memory and Short-term memory. Long-term memory is used to maintaining the history of all processes in tabu search. Short-term memory is used to keep the recently visited tabu movements. In this work short-term memory of the tabu list is used for better results.

Ant Colony Optimization Algorithm (ACO) is used to solve complex problems. It is used to identify the shortest path between the colony and food, also find the best solution for combinatorial optimization problems.

An ACO algorithm generally defined how the ants find out the shortest path to reach the destination. Ants couldn't see a food source, not having the knowledge to find out the correct path to reach the destination and come back. They are collecting the routes around their nest. Ants are producing the pheromone chemical which is used to communicate between the ants.

In this paper, we address the makespan problem which is defined as the maximum completion time of all jobs that are minimized in our paper. The objective of this work is to execute a job in the shortest possible time. In scheduling, completion time is the time spent between the first job and last job. At time zero, jobs are in ready state and during scheduling, the resources are continuously available.

Makespan is reduced in this work using short-term memory based rollback recovery techniques. When the jobs are flowed at the particular resources, that jobs are immediately stored in the short term memory. Using this memory, the rollback recovery technique is implemented.

Contribution of our work is using two heuristics algorithm i.e. ant colony optimization (ACO) and tabu search (TS) to solve the job shop scheduling problem and also minimizing the makespan with the help of short-term based rollback recovery. The goal of this work is to identify the nearest fault occurrence position to minimize the completion time of the job.

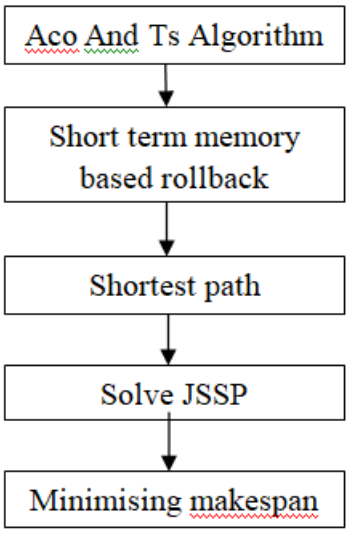

Fig 1. The flow of the proposed system

Revised Manuscript Received on December 30, 2019.
***** Correspondence Author
K.Sathya sundari*, Research Scholar Part time Category – B, Bharathiar University, Coimbatore. selvisathika@gmail.com
Dr V, P, Eshwaramurthy, Asst. Prof. of Computer Science, Komarapalayam Arts and Science College, Komarapalayam. eswar_murthy@yahoo.com

II. RELATED WORKS

Eswaramurthy and Tamilarasi [1] presented the tabu search of global optimization technique which also combines the ant colony optimization technique to solve the problem of job shop scheduling. The neighborhoods are selected based on the strategies in the ant colony algorithm and the dynamic tabu length strategies are selected in tabu search. The Pheromone trail was used for improving solutions in ant colony optimization.

Fattahi et al [2] described the job shop scheduling problem. There are two solutions to this problem that have both strengths and weaknesses in the solution space of searching to reach an acceptable solution.

Ali et al [3] presented a job shop scheduling problem that is solved by the tabu search algorithm. This study is used to generate the corresponding completion time with different schedules.

Florez et al [4] described the elitist ant system (EAS) commonly known as an implementation of an act model algorithm which is applied to job shop scheduling problem. It reduces the delays so that the operations are immediately available.

Habibeh Nazif [5] outlined the ACO for job shop scheduling which provides the best solutions through the directed search near the neighborhood. A good solution has generated in negligible computation time. The dynamic changes are made in a new way due to the limited trails of upper and lower bounds.

Sun et al [6] determined the multi-objective optimization algorithm for solving the flow shop scheduling problem. They first described the MFSP and complexity of MFSP then they introduced the taxonomy and publications of multi-objective optimizations.

Idris et al [7] proposed the Ant Colony Optimization Algorithm (ACO) for the successful execution of jobs even during the failure of the resource. They also determined the checkpoint-based rollback recovery strategy technique. The amount of work has been reduced by checkpointing during the system failure through the immediate saving of system state.

Hosseinabadi et al [8] proposed the extended genetic algorithm for open shop scheduling (EGA-OS) which provides the effects of mutation operator and crossover selection for solving open-shop scheduling problems in genetic algorithms (GA).

Wang et al [9] developed an improved ant colony optimization (ACO) algorithm and the makespan optimization is used for FJSP (Flexible JSP). This algorithm utilized the updated pheromone's mechanism, change pheromone's guiding mechanism, initialize uniform distributed mechanism and select node method for the aspects of ants.

Manoj Kumar et al [10] determined the checkpointing and rollback recovery techniques. Synchronous and asynchronous are the two different fundamental approaches of those two techniques.

III. METHODOLOGY

A checkpointing based rollback recovery technique is used for avoiding the restarting process of whole application in the conventional system due to the occurrence of failure. Generally, the jobs get rescheduled and execution has started from beginning on another grid resource when a failure occurs in a conventional system. The execution state of the application is periodically saved in stable storage through the checkpointing mechanism. From the last successful state of the job, the job gets resumed when a volunteer computation has interrupted by failure to increase the QoS requirement of the user [7].

A process will never be considered a part of a globally consistent state when it takes a useless checkpoint. Multiple checkpoints have maintained for every process and the garbage collection algorithm has periodically invoked to reclaim the checkpoints which are not used for a long time period. The global coordination is required in the computation of the required line of application, so this method is not suitable for frequent output commits' applications. Maximum overhead has involved in the consistent global checkpoint of large systems and the process has restarted from the beginning due to the absence of consistent global checkpoint except for the initial state [10].

In this proposed work, a rollback recovery technique is used to avoid restarting the application from the beginning which is based on short-term memory. The short term memory is used to the recently visited job movements. In this work circle denotes the job i.e. (j1, j2, j3), rectangle denote the machine i.e. (m1, m2, m3), square denote the short-term memory, cloud symbol denote the crash, arrows which are connecting the resources.st denote the starting time of the job, ct denote the completion time of the job.

A. Without short-term memory based rollback recovery technique

- Job1 flows at machine 1 for 3 minutes and processed for 7 minutes and completed after processing.
- Job3 flows at machine 3 for 8 minutes and processed for 8 minutes and completed after processing.
- Job2 flows at machine 2 for 5 minutes and processed for 3 minutes and get crashed. When crashed, again that job starts from the beginning repeat the step without crashing.
-

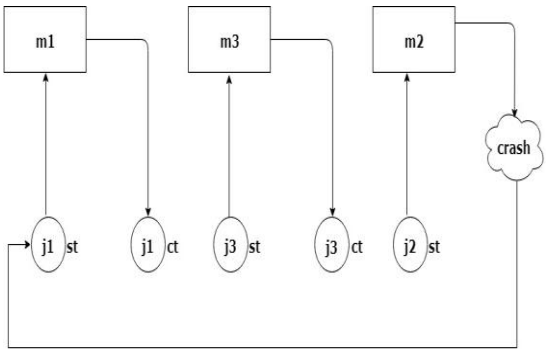

Fig2: Without short- term memory

Fig 2 shows the ant colony optimization algorithm without a short-term memory based rollback recovery technique.

Table 1: calculate the average completion time

SEQUENCE	PROCESSING TIME (minutes)	FLOW TIME (minutes)
J1	3	7
J3	8	8
J2	5	3
total	16	18

Average Completion time of total jobs = $\frac{total\ flow\ time}{no\ of\ jobs} = \frac{18}{3}$ = 6 minutes

Because job 2 is crashed at particular times and again the job starts from the beginning, in this work makespan is take more time and the product is not delivered on time.

B. With short-term memory based rollback recovery technique

Fig 3 shows the ant colony optimization algorithm with a short-term memory based rollback recovery technique.

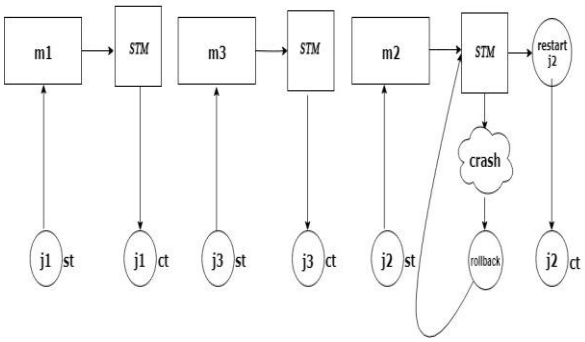

Fig3: with short-term memory

- When the job1 flow at machine 1 with 2 minutes and STM visited the recent job movements i.e. j1 that same job processing the 3minutes, j1 completed after processing the job.
- When the job3 flow at machine3 with 3 minutes and STM visited the recent job movements i.e. j3 that same job processing the 4minutes, j3 completed after processing the job.
- When the job2 flow at machine2 with 2 minutes that same job processing the 3 minutes with crashed, apply the rollback recovery technique which is used for avoiding the restarting from the beginning based on short-term memory, it is used for recently visited the movements. Repeat the technique until the crash will not occur at the processing time.

SEQUENCE	PROCESSING TIME (min)	FLOW TIME (min)
J1	3	2
J3	4	3
J2	3	2
total	10	7

Table2: Calculate the completion time

Completion time of total jobs = $\frac{total\ flow\ time}{no\ of\ jobs} = \frac{7}{3}$ =2.5 minutes

As job 2 is crashed at the particular time the short-term memory based rollback recovery technique is applied to avoid restart from the beginning. The nearest fault occurrence position is detected and rollback is made up to that position to minimize the completion time of a job. Thus the makespan is reduced and the product is delivered on time with the help of our proposed techniques.

MAKESPAN ARCHITECTURE

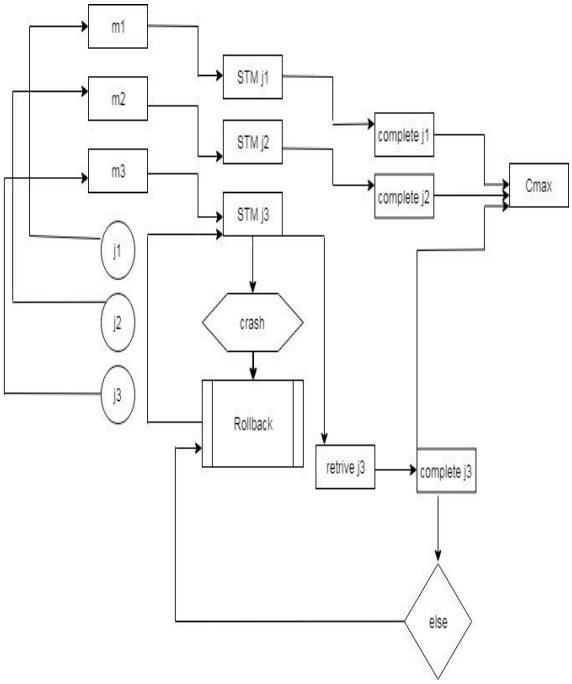

Fig4: architecture of makespan

Fig 4: proposed makespan architecture. The makespan suggests minimizing the completion time by avoiding the longest pathway technique in this work which makes the shortest way to complete the work. In an existing system, the completion time is higher and the number of resources is more but in the proposed work minimal resources are used to handle the job shop scheduling problem. In each process, the STM is visited to trace the recent movements and execute job completion. If the crash occurs, the rollback technique will be applied to retrieve the job.

Algorithm 1 explains about proposed ACO with our techniques. First, we assign the job and machines second algorithm 1 consists of three case sequences.
Case 1 denotes the flow time of job1 and the processing time of job1**.**
Case 2 denotes the flow time of job3 and process time of job3.
Case 3 denotes the flow time of job2 and process time of job2 and crash occur at the processing period so apply our technique to call the short-term memory which is used to recently visit the

ALGORITHM 1: proposed ACO with TECHNIQUES
START
1. Assign the jobs j= (j1, j2, j3)
2. Assign the machines m= (m1, m2, m3) for each job
Case: 1
3. Job1 flow at machine1 with 2minutes
4. Process the job1 with 3minutes
5. After the STM visited the recent jobs
6. End Process of j1
7. Complete the j1
Case: 2
8. Job3 flow at machine3 with 3minitues
9. Process the job3 with 4minitues
10. After the STM visited the recent jobs
11. End Process of j3
12. Complete the j3
Case: 3
13. Job2 flow at machine2 with 2minitues
14. Process the job2 with 3minitues
15. After the STM visited the recent jobs
16. Crash occur at the process period
17. Apply the rollback recovery technique
18. To avoid the job start from the beginning stage
19. using the rollback recovery, restart the job at current stage
20. End process of j2
21. Complete the j2
REPEAT the step 17
UNTIL the job process without crashed
Finally MAKESPAN is reduced at certain times.

movements so using the rollback recovery technique which is based on short-term memory to restart the current state of the job.

IV. RESULT AND DISCUSSION

This section compared the proposed ACOTS-C_{max} with ACO with Fault Tolerance (ACOwFT). This section consists of three parameters to show how the existing and proposed work implemented through the parameters.

Table 3: Average makespan time for jobs

No of jobs	Average makespan (min)	
	ACOwFT	*ACOC$_{max}$*
5	4	3
10	8	6
15	11	9
20	15	12
25	17	15
30	20	18

Table 3 explains the average makespan values in minutes. The first column contains the number of jobs performed at a certain time.

The second column is the completion time of jobs in the existing system and the third column contains the proposed completion time of jobs.

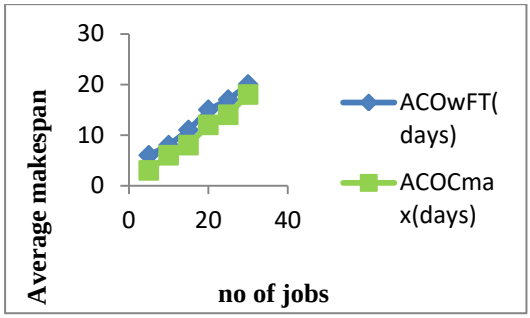

Fig 5: Average makespan

Fig 5 shows the average makespan comparison of proposed ACOC$_{max}$ with ACOwFT. In the proposed ACOC$_{max}$ technique, the makespan is less than the existing system. The minimum amount of additional resources has been used by using effective ACO to get the minimum makespan.

Table 4 explains the average throughput values in minutes. It is observed that the average throughput of ACOC$_{max}$ is better than ACOwFT.

Table 4: Average throughput for jobs

No of resources	Average throughput (minutes)	
	ACOwFT	*ACOC$_{max}$*
500	0.4	0.2
1000	0.2	0.1
1500	0.3	0.2
2000	0.2	0.1
2500	0.3	0.2
3000	0.5	0.3

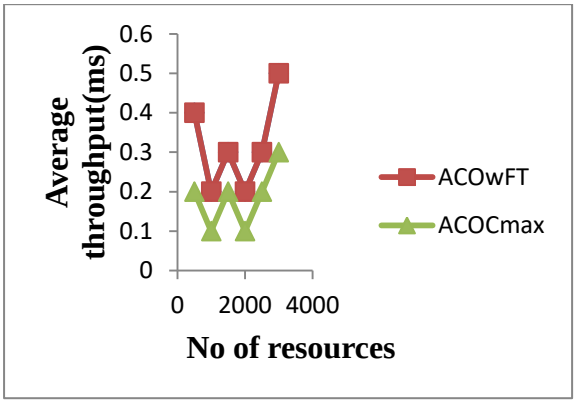

Fig 6: Average throughput

Fig. 6 shows the average throughput comparison of ACOC$_{max}$ with ACOwFT. Here throughput is the number of information processed by a system within a given time period.

Table 5: Average turnaround time for jobs

No of jobs	Average Turnaround time (ms)	
	ACOwFT	*ACOC$_{max}$*
500	3	2
1000	2	1
1500	4	2
2000	3	1
2500	5	2
3000	2	3

Table 5 explains the average turnaround time. Turnaround time commonly refers to the amount of time measured from the submission of a request to the conclusion and delivery to the requester. It is clearly observed that the turnaround time of ACOC$_{max}$ is better than ACOwFT.

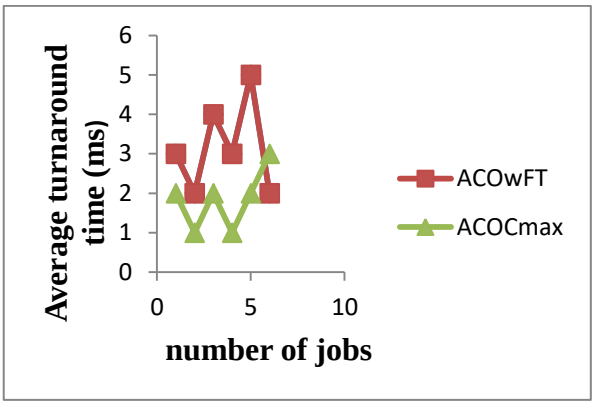

Fig 7: Average turnaround time

Fig. 7 shows the average turnaround time comparison proposed ACOC$_{max}$ with ACOwFT. Here turnaround time is the required time period for finishing a certain task or process.

V. CONCLUSION

This paper determines the minimizing makespan in job shop scheduling with the help of short-term based rollback recovery techniques in the Ant Colony Algorithm (ACO). Use this technique to find the makespan in the shortest way. Makespan, throughput time, turnaround time factors are reduced in this technique. Finally (ACO-C$_{max}$) explores efficient and effective compare than (ACOwFT).

REFERENCES

1. V. P. Eswaramurthy and A. Tamilarasi, "Hybridizing tabu search with ant colony optimization for solving job shop scheduling problems", Int J Adv Manuf Technol, 2009.
2. Parviz Fattahi, Masume Messi Bidgoli and Parvaneh Samouei, "An Improved Tabu Search Algorithm for Job Shop Scheduling Problem Trough Hybrid Solution Representations", Journal of Quality Engineering and Production Optimization, 2018.
3. Md. Hasan Ali, Md. Rashedul Haque and Md. Fashiar Rahman, "Machine Utilization Technique for Job Shop Scheduling using Tabu Search Algorithm", International Journal of Advance Industrial Engineering, 2018.
4. Edson Flórez, Wilfredo Gómez and MSc. Lola Bautista, "An ant colony optimization algorithm for job shop scheduling problem", Research gate, 2014.
5. Habibeh Nazif, "solving job shop scheduling problem using an ant colony algorithm", Journal of Asian Scientific Research, 2015.
6. Yi Sun , Chaoyong Zhang, Liang Gao & Xiaojuan Wang, "Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects", Int J Adv Manuf Technol, 2011.
7. Hajara Idris, Absalom E. Ezugwu, Sahalu B. Junaidu , Aderemi O. Adewumi, "An improved ant colony optimization algorithm with fault tolerance for job scheduling in grid computing systems", research article, 2017.
8. comma Ali Asghar Rahmani Hosseinabadi, Javad Vahidi, Behzad Saemi, Arun Kumar Sangaiah, Mohamed Elhoseny, "Extended Genetic Algorithm for solving open-shop scheduling problem", Springer, 2018.
9. Lei Wang, Jingcao Cai, Ming Li, and Zhihu Liu, "Flexible Job Shop Scheduling Problem Using an Improved Ant Colony Optimization", Scientific Programming, 2017.
10. Manoj Kumar, Abhishek Choudhary and Vikas Kumar, "A Comparison between Different Checkpoint Schemes with Advantages and Disadvantages", International Journal of Computer Applications, 2014.