International Journal of Academic Research and Development
ISSN: 2455-4197
Impact Factor: RJIF 5.22
www.academicsjournal.com
Volume 3; Issue 1; January 2018; Page No. 214-217

Dynamic job shop scheduling using ant colony optimization

[1] Dr. Anupam Bhatia, [2] Kirty
[1] Assistant Professor, Department of Computer Science & Application, Chaudhary Ranbir Singh University, Jind, Haryana, India
[2] M.Phil (C.S) Research Scholar, Department of Computer Science & Application, Chaudhary Ranbir Singh University, Jind, Haryana, India

Abstract

Scheduling refers to allocation of limited resources to perform set of task to fulfill of certain criteria. Job shop scheduling is one of the most popular modal of scheduling. Job shop scheduling is one of widely studied and difficult combinatorial optimization problem. There are variety of disruptions may occur unexpected e.g. arrival of new job, machine breakdown etc. This paper analyzes the limitation of static approach when rescheduling in dynamically insertion of new jobs and changing environments. There are three machine and four jobs. In this research we insert two job after some time and reschedule the process. We use ant colony optimization approach for minimum makespan and maximum utilization of resource.

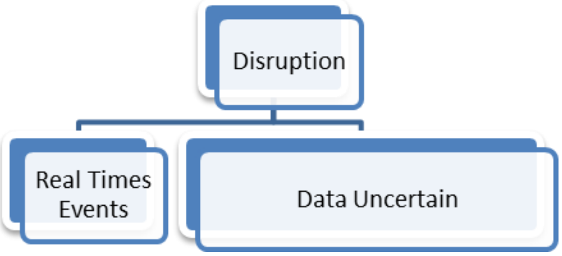
Keywords: job shop scheduling, dynamic scheduling, ant colony optimization, evolutionary algorithm

1. Introduction

In this section have description of the job shop scheduling problem. To formally define the problem using scheduling terminology, we have a set of n jobs to be processed on m different machines, where in each job has its own machining sequence [3]. The objective of this research is to generate best job schedules that minimize mean flow time and mean tardiness. The problem is denoted by n jobs (j1,j2,j3,………,jn) to be processed through m machines(m1,m2,m3,………,mm).

1.1 Dynamic Feature of Job Shop Scheduling Problem:

Environments of real word manufacturing are dynamic. There are arrive some disruption and divided it into two categories.



Data Uncertain

The information may be uncertain and change over time e.g. due date of a job may change dynamically since less important today may be more important tomorrow. Change in job priority or batch size also belong to information are uncertain.

Real times events

There are two type of real event occur. In the first way the non operation disruption assumption occur in job shop scheduling e.g. one machine break-down suddenly. In second way dynamic events occur unpredictably e.g. new job arrive.

Scheduling in job shop environment with occurring of uncertain and real time events is called dynamic job shop scheduling.

2. Literature Review

This paper discusses the literature in areas related to current research problem. The job shop scheduling problem is one of the widely studied problems in the scheduling literature and NP hard problem [5]. There are lot of research is done by researcher on job shop scheduling using many evolutionary algorithm like genetic algorithm, ant colony optimization, particle swarm optimization etc. Dynamic dispatching rule was most used scheduling algorithm for due date related job shop scheduling. Used Feasibility Function (FF) to schedule jobs in multi machine random shop where purposed to minimize inventory or tardiness costs. Reducing the difference between the maximum and the minimum lateness of jobs It had been observed that no single rule perform well for all important criteria related to flow time, job tardiness and other system performance measures [6]. In this Paper minimized three performance simultaneously mean flow time, mean tardiness and mean earliness. The system's performance when using dynamic scheduling could be improved by changing the corresponding dispatching rule at a correct frequency. The algorithm was found to be useful even the performance criteria used for comparison were made to be biased on purpose [7]. In this paper presented, They were able to take three decisional levels simultaneously into account and to provide a production scheduling aiming at minimizing the global makespan.GA had been compared with previous algorithm designed for the distributed scheduling problems and tested on a large set, providing satisfactory results [8]. The performance quality of a solution constructed by an artificial ant is improved by a job-index-based local search procedure incorporated with a threshold probability for choosing a job to insert into the other positions of the sequence. Two main formulation are used job shop problem and flexible job shop problem. Relationship between JSP, FJSP and EFJSP as JSP<=FJSP<=EFJSP [9]. Multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithm. The efficiency of two algorithms was compared based on the diversity, spacing, MMID, NPS and Time criteria. Statically hypothesis was used to evaluate the algorithm with high performance [10].

3. Ant Colony Optimization

Ant Colony Optimization approach was proposed in 1992 by Marco Dorigo *et al.* to solve several discrete optimization problems. ACO system also deal with artificial systems that are inspired from the foraging behavior of real ants for solve discrete optimization problems. There are indirect communication between the ants by means of chemical pheromone trials to find short paths between their nest and food [4]. There are main three ideas which transfer from the natural ant colony to the artificial ant colony:

1. The preference for paths with a high pheromone level.
2. The higher rate of growth in the amount of pheromones on shorter paths.
3. The information exchanged among ants.

The basic idea of real ant system is represented in Figure 3.1(a).The ants move in a straight line from nest to food source (Figure 3.1 (a)) for searching the food. At the next stage, we assume that there is an obstacle (Figure 3.1 (b)). In this case, to avoid the obstacle initially each ant chooses to left or right at random (Figure 3.1 (c)). We see that ants move at the same speed depositing pheromone in the trail uniformly. By chance which ants select to turn left will reach the food source sooner, whereas the ants that turn right will follow a longer path. The quantity of pheromone deposited on shorter path is more than the other path. So ants will select to move on the shorter path (Figure 3.1 (d). The quantity of deposited pheromone is one of the most important factors for ants to find the shortest path (figure 3.1).

Fig 3.1(a): Ants Straight Line

Fig 3.1(b): Obstacle

Fig 3.1(c): randomly choose path

Fig 3.1(d): Choose Shortest path

1. Ants following a straight path between their nest and food source.
2. Assume obstacle.
3. Selecting randomly path of ants.
4. Finding shortest path of ants.

ACO find the shortest path between a food source and their nest. The ants communicate with each other by means of pheromone trails. Pheromone is the chemical material which deposited by the ants as critical communication medium among ants and guiding the determination of the next movement. Ants traces the higher pheromone path and depositing their own pheromone. Ants find the shortest path based on quantity of pheromone deposited on different paths. This characteristic of ants is accepted on ant colony optimization algorithms to solve real problems. ACO techniques have been successfully employed in many fields, such as job shop scheduling problem, traveling salesman problem, scheduling, vehicle routing etc. The general ACO algorithm structure can be described as follows:

1. Initialize the pheromone trail
2. While not terminate
   a. Select a feasible solution by each ant
   b. Local updates the pheromone trail.
   c. Local Search
   d. Global update of pheromone trail.
   e. End loop
3. Select the best solution.

4. Problem and Objective

The general description of the problem environment is given below:

- There are M machines and N jobs in the system.
- The machine order matrix. MJ (i, j) M means the machine No. of job i in its jth process.
- The arrangement of the machine No. of job i in order of priority is J (i,*) M.
- Jobs ordered array JM. M (i, j) J means the job No. of machine i in its jth process.
- The arrangement of the jobs No. followed by processing order on machine i M (i,*) J.
- Processing time matrix Tij. T means the processing time of job i processed on machine j.
- Complete time of job i on the machine Cij.

4. Implementation & Result

Our work is based on the multi objective dynamic job shop scheduling problem using ant colony optimization technique. ACO is inspired by foraging behavior of real ant's colony. In this research, we use an ACO-based approach to find best solutions to the job shop scheduling problem. It is capable of finding the shortest path from food source to their nest.

This section is going to implementing of job shop scheduling using ant colony optimization. In this section, there are 3 machine and 4 jobs. Jobs sequence is predefined and 2 jobs are inserting at the beginning; but 2 jobs are inserting after some time. There are many cases using different data and different time.

Job Sequence

Job1	Machine1	Machine2	Machine3
Job2	Machine3	Machine2	Machine1
Job3	Machine2	Machine1	Machine3
Job4	Machine3	Machine1	Machine2

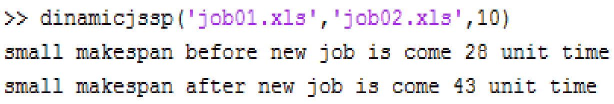
Case 1

Table 1

Machines Jobs	Machine 1	Machine 2	Machine 3
Job 1	11	7	10
Job2	6	9	11
Job 3	9	7	9
Job 4	6	9	11

Case 1 (a)

Case 1 (a) has used table1 data to find out the makespan. First we find out makespan of two jobs and if two another jobs inserted after 10 then rescheduling takes place. Makespan of 2 jobs before inserting new job is 28 and inserting 2 jobs after 15 units of time later from starting point then makespan is 43. If job3 and job4 is inserted after the completing of job1 and job2 then makespan is 59. Total time saved in 16 units of time. Figure1 and Figure 2 has shown the output.



```
>> dinamicjssp('job01.xls','job02.xls',10)
small makespan before new job is come 28 unit time
small makespan after new job is come 43 unit time
```

Fig 1

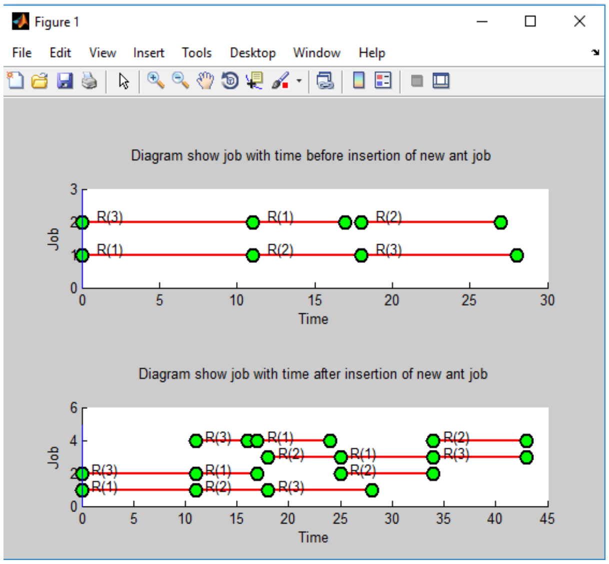
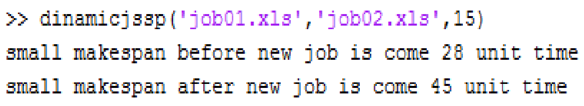


Fig 2

Case 1(b)

Case 1 (b) has used table1 data to find out the makespan. First we find out makespan of two jobs and if two another jobs inserted after 15 units of time then rescheduling. Makespan of 2 jobs before inserting new job is 28 and inserting 2 jobs after 15 units of time later from starting point then makespan is 45. If job3 and job4 is inserted after the completing of job1 and job2 then makespan is 59. Total time saved in 13 units of time.Figure3 and Figure 4 has shown the output.



```
>> dinamicjssp('job01.xls','job02.xls',15)
small makespan before new job is come 28 unit time
small makespan after new job is come 45 unit time
```

Fig 3

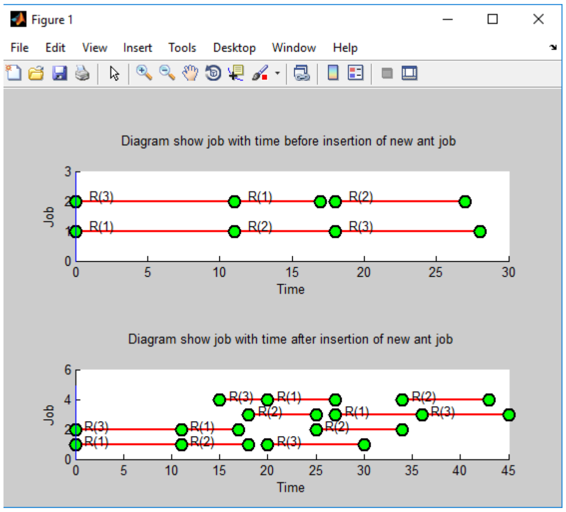


Fig 4

In case 1 (a) and case 1 (b) has used same data but times of jobs inserted were different.

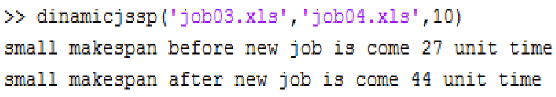
Case 2

Table 2

Jobs	Machine 1	Machine 2	Machine 3
Job1	9	7	8
Job2	6	10	11
Job3	12	8	6
Job4	5	8	10

Case 2 (a)

Case 2 (a) has used table2 data to find out the makespan. First we find out makespan of two jobs and if two another jobs inserted after 10 units of time then rescheduling. Makespan of 2 jobs before inserting new job is 27 and inserting 2 jobs after 10 unit of time later from starting point then makespan is 44. If job3 and job4 is inserted after the completing of job1 and job2 then makespan is 60. Total time saved in 16 units of time. Figure 5 and Figure 6 has shown the output.



```
>> dinamicjssp('job03.xls','job04.xls',10)
small makespan before new job is come 27 unit time
small makespan after new job is come 44 unit time
```

Fig 5

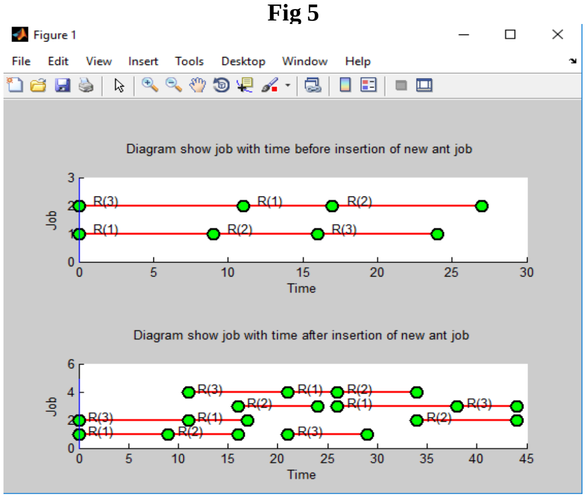
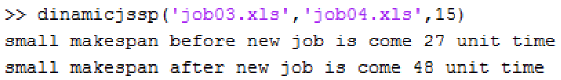


Fig 6

Case 2(b)

Case 2 (b) has used table2 data to find out the makespan. First we find out makespan of two jobs and if two another jobs inserted after 15 units of time then rescheduling. Makespan of 2 jobs before inserting new job is 27 and inserting 2 jobs after 15 units of time later from starting point then makespan is 48. If job3 and job4 is inserted after the completing of job1 and job2 then makespan is 60. Total time saved in 12 units of time. Figure7 and Figure8 has shown the output.



```
>> dinamicjssp('job03.xls','job04.xls',15)
small makespan before new job is come 27 unit time
small makespan after new job is come 48 unit time
```

Fig 7

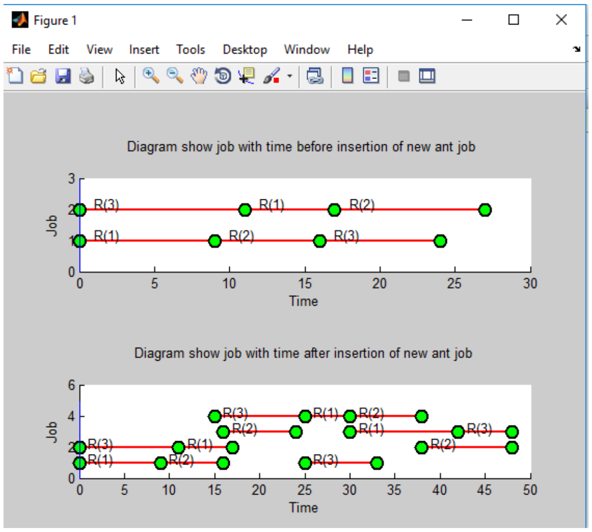


Fig 8

In case 2 (a) and case 2 (b) has used same data but times of jobs inserted were different.

5. Conclusion & Future Work

This study evaluates a methodology of optimizing both makespan and utilization of machine for dealing with new job insertion in FJSP. In literature, most of the research in scheduling problems is dealing with a single objective or a single solution for a linear combination of objectives known as multi objective methodology. To solve the DJSSP more effectively, an ant colony algorithm is developed with the makespan.

The algorithm is implemented on Matlab. 2 different cases with two sets of job were studied. Each set is studied with job insertion after 10 units & 15 units of time.

In first case, if new job is inserted after 10 units of time, then total makespan is 43, and if new job is inserted after 15 units of time, then total makespan are 45.

In second case, if new job is inserted after 10 units of time, then total makespan is 44, and if new job is inserted after 15 units of time, then total makespan are 48.

It clearly indicates that Ant Colony Optimization (ACO) algorithm facilitate to insert the job as soon as it arrive and reschedule the process. The cost of rescheduling is lesser than the cost of machine idle time. Hence by using, ACO algorithm machine idle time can be reduced as much as possible and makespan of jobs not coming simultaneously can be reduced.

In future studies, other uncertainties such as job cancellation and machine breakdown can be examined.

6. References

1. Nazif H. Solving Job Shop Scheduling Problem Using an Ant Colony Algorithm, J. Asian Sci. Res. 2015; 5(5):261-268.
2. A Study of Research Issues in Dynamic Job Shop Scheduling. 2017; 8(1):2015-2017.
3. Arisha A, Young P, El Baradie M. Job Shop Scheduling Problem : an Overview. 2001.
4. Blum C. Ant colony optimization : Introduction and recent trends. 2005; 2:353-373.
5. Moslehi G, Mahnam M. A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search, Int. J. Prod. Econ. 2011; 129(1):14-22.
6. Lengyel A, Hatono I, Ueda K. Scheduling for on-time completion in job shops using feasibility function, Comput. Ind. Eng. 2003; 45(1):215-229.
7. Chan FTS, Chan HK, Lau HCW, Ip RWL. Analysis of dynamic dispatching rules for a flexible manufacturing system, J. Mater. Process. Technol. 2003; 138(1-3):325-331.
8. Huang M, Wu T, Liang X. GA-ACO in job-shop schedule problem research, Commun. Comput. Inf. Sci. CCIS. 2010; 107:226-233.
9. Genova, Krasimira, Leoneed Kirilov, Vassil Guliashki. A Survey of Solving Approaches for Multiple Objective Flexible Job Shop Scheduling Problems. Web. 2015; 15(2):3-22.
10. Ahmadi E, Zandieh M, Farrokh M, Emami SM. A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms, Comput. Oper. Res. 2016; 73:56-66.