International Journal For Technological Research In Engineering
Volume 7, Issue 9, May-2020 *ISSN (Online): 2347 - 4718*

A REVIEW ON FLEXIBLE JOB SHOP SCHEDULING PROBLEM AND RESCHEDULING STRATEGY

Mr.S.V.Kamble[1], Mr.V.C.Sakhare[2], Dr.P.D.Gangawane[3], Mr.B.G.Warvante[4]
1,2,3,4Assistant Professor
[1,2,3]DKTE'S Textile & Engineering Institute, Ichalkaranji, [4]TKIET, Warananagar

Abstract: Many heuristics methods have been proposed and applied for solving the Job Shop Scheduling Problems (JSSP). Several researches have been proposed methods to solve JSSP with objectives to minimize simultaneously: makespan, machine idle time, total workload, and total tardiness. This paper is discuss flexible job shop scheduling, rescheduling strategy for FJJSP and different heuristics method used for solving FJSSP. Rescheduling in FJJSP updates an existing production schedule in response to disruptions or other changes and it affects the performance of a manufacturing system.
Keywords: Job shop scheduling problem (JSSP), Rescheduling, Flexible job shop scheduling (FJSSP).

I. INTRODUCTION

Job shop scheduling or the job-shop problem (JSP) is an optimization problem in operational research which various manufacturing jobs are assigned to the machines. Job shop scheduling problem (JSSP) consists of the most challenging one which has been the subject of many research studies that have been interested in solving production planning in JSSP and few methods have been proposed and implemented and applied to solve the job shop scheduling problem in the production system during the recent decades [1].
The job shop scheduling is NP-hard in a strong sense and it is one of the hardest combinational optimization problems .The job shop scheduling problem (JSSP) which is consists in Scheduling a set of jobs on a set of machines for the objective of minimizing the make-span, machine ideal time, total workload. The maximum of completion time needed for finishing all the jobs. Any scheduling is subject to the constrains that each job has a fixed processing order through the machines and each machine can process at most one job at a time [7].
Where as in The Flexible Job-shop Scheduling Problem (FJSSP) is an extension of the classical JSSP, Job Shop Scheduling (JSSP) is one of the most complex scheduling problems related to manufacturing industries. However, FJSSP is also an NP-hard problem as the number of jobs increases; it becomes more difficult to obtain the optimal schedule in short time. The problem of scheduling jobs in FJSSP could be decomposed into
two sub-problems: one consist of Assignment Problem which is to select a machine from several available machines for each the operation. And second sub problem, Sequence Problem, is to identify a sequence of all operations on each machine [9].
The rest of the paper is organized as follows: Scheduling model classification explained in section 2, Problem description and formulation is presented in section 3, section 4 explained rescheduling strategies, section 5 describes the Algorithm and section 6 conclusions and feature scope for rescheduling in FJSSP.

II. SCHEDULING MODEL CLASSIFICATIONS

Scheduling involves determining the allocation of plant Resources based in this strategies multiple scheduling model get classified as follows:

A] Basic type (JSP) although the basic type of job shop scheduling model (JSP) is the simplest model, The basic JSPs are still NP-hard problems. In this kind of scheduling model, a specified like operation to be processed on a specified machine tool and no more machines can be chosen. Optimization algorithms or multi-objective optimization algorithms are employed to make the makespan or cost or both of them minimum and achieve a final optimal operation sequence [5].

B] Multi-resources FJSP model it is considered as multi-resources are used in Flexible job shop scheduling (FJSO). The purpose of such type of scheduling is to generate operation sequences of jobs on each machine to achieve the optimization of certain parameters under the constraint conditions. Usually, multi-objective optimization algorithms are applied to achieve the objectives of FJSP with minimum resource transition times [5].

C] Flexible JSP Flexible job shop scheduling model (FJSP) is advanced and complex JSP as machines can be selected for some or all operations [9].

III. PROBLEM DEFINITION

Let J = {1, 2, …, n} be a set of jobs, M = {1, 2, …, m} be a set of machines and O = {0, 1, 2, …, N, #} be a set of operations. Each job consists of a sequence of operations each of which has to be processed on a given machine for a given time. Here, 0 and # represent the dummy start and finish operations, respectively.
Assumptions:

- All job released at time 0.
- All machines available at time 0.
- Only one job carried out on one machine at one time
- Once an operation starts, it cannot be terminated before it finishes.

Objectives:
The objectives are to find an assignment and a schedule to

 Copyright 2020.All rights reserved.

minimize objective simultaneously like makespan, machine workload, total workload, job tardiness etc.

IV. RESCHEDULING STRATEGY

Rescheduling:

In rescheduling strategy will be implemented in Flexible Job Shop Scheduling in concern of once particular machine is breakdown randomly at that time it shuffle the entire workload on another available machines to keep objectives of FJSSP will be minimized simultaneously [9].

Rescheduling strategy majorly consist of three primary types of studies: one, methods for repairing a schedule that has been disrupted; two, methods for creating a schedule that is robust with respect to disruptions; and three, studies of how rescheduling policies affect the performance of the dynamic manufacturing system [4].

Rescheduling is the process of updating an existing production schedule in response to disruptions or other changes. This includes the arrival of new jobs, machine break down, and machine repairs.

Schedule Generation		Schedule Repair		
Nominal Schedule	Robust Schedules	Right shift Rescheduling	Partial Rescheduling	Complete regeneration

Fig.1 Rescheduling Method [4]

V. THE ALGORITHM

5.1 Simulated Annealing for JSSP:-

Simulated Annealing proposed by Kirkpatrick et al, SA mimics the physical evolution of a solid to thermal equilibrium, slowly cooling until it reaches its lower energy state. The process of a heated solid's annealing from its maximum energy state to minimum state gradually by controlling parameter [5][6].

The SA algorithm generally starts from a high temperature, and then the temperature is gradually lowered. At each temperature, a search is carried out for a certain number of iterations, called the epoch length. When the termination condition is satisfied, the algorithm will stop [9].

It starts from an initial basic solution x_0 and randomly generates a neighbor solution x' from its neighborhood. x' is accepted as a new basic solution with probability P, which calculated as

$$P=\{\ 1,\ \text{if}\ \Delta Lmax < 0;\quad \exp\{-\Delta Lmax/T\},\ \text{otherwise}\ \}$$

Where $\Delta Lmax = Lmax(x') - Lmax(x_0)$ is the difference in objective value between the two solutions, and T is the temperature which decreases with iterations.

Step 1: Produce an initial solution x_0. Set $x^* = x_0$.

Step 2: For iter = 1, 2, . . . , Imax

(2.1) Randomly generate a neighbour solution x' from the neighbourhood of x_0.

(2.2) If Lmax(x') < Lmax(x), set x* = x'.*

(2.3) Set x_0 = x' with probability P = $\min\{1, \exp\{(Lmax(x_0) - Lmax(x'))/T\}\}$.

(2.4) Reduce the temperature by setting $T := T \times \lambda$, where λ (0, 1) is the cooling rate.

Step 3: Return x-.

Fig.2 SA Algorithm [11]

5.2 Tabu Search for JSSP:-

Tabu Search was presented by Glover (1986), It is simple search procedure, can prohibit the moves which is the same as or similar to the previously achieved solutions according to the search information stored in the memory, and then avoid local optimum solutions.

```
JSSP is an instance of the Job Shop Scheduling problem
sol ←InitialSolution (JSSP)
bestCost ←cost(sol)
bestSolution ←sol
tabuList ←∅ ;
while keepSearching ()
do N Valid (sol) ←{s € N(sol) I Move[sol; s] € tabuList
if Nvalid (sol) =∅
then sol' ← x € Nvalid(sol)j alid(sol) cost(x) ▯ cost(y)
updateTabuList(sol0)
if cost(Move[sol; sol0]) < bestCost
then bestSolution   sol0
bestCost   cost(sol0)
sol   sol0
```

Fig.3 Tabu Search Algorithm [10]

5.3 Genetic Algorithm for JSSP: -

Holland developed the idea of GAs based on Darwin's evolution theory. The earliest application of GAs to JSSPs was proposed by Davis (1985) [8]. Genetic algorithm (GA) is one of popular meta-heuristics which is based on the genetic evolution mechanism of biology. GAs' main characteristics are to directly operate on the problem structure without derivation and function Continuity limitation. Davis formed a preferred sequence of operations

 Copyright 2020.All rights reserved.

for every machine in which GA is an indirect method. After that various efforts have been made to adapt genetic algorithms to solve different JSPs and have been improving the performance of genetic search by integrating other heuristic methods [5].

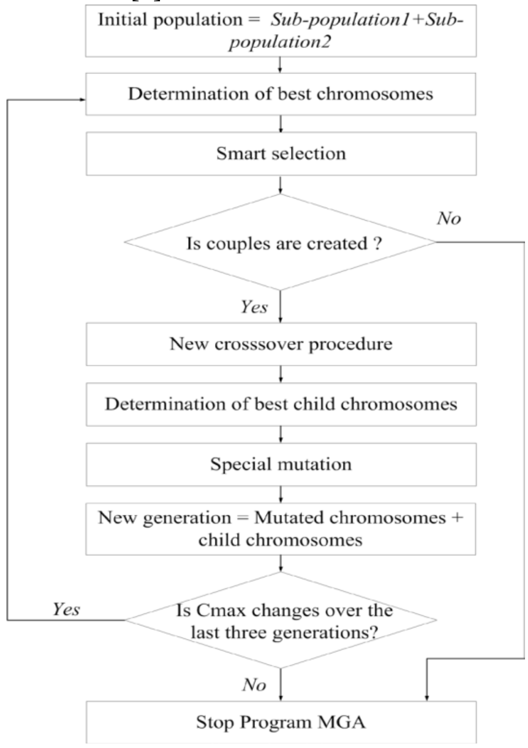

Fig.4 Flowchart of the running mechanism of the GA [1]

5.4 Ant colony Optimization for JSSP:-

Ant colony optimization (ACO) which imitates the process of ant colony foraging is another meta-heuristic presented by Colorni et al. (1991),

The main idea in ant colony optimization algorithms is to consist to be trails used by real ants

Searching for feed as a medium for communication and feedback. In the ACA, a rather good solution is firstly generated in negligible computation time, and then the pheromone trails are initialized depending on this solution [12].

Step 1. Set parameters; generate a seed solution and initialize the pheromone trails.
Step 2. While the termination condition is not met, do the following:
2.1. For each ant in the colony do:
By repeatedly applying the transition rule, construct a solution;
Improve the solution quality by the local search;
In case of an improved solution, update the best solution generated so far.
2.2. Modify the pheromone trails according to the global updating rule.
2.3. Update the minimum and maximum trail bounds, and limit the pheromone trails.

Fig.5 General structure of the ACA[12]

5.5 Particle Swarm Optimization for JSSP:

PSO is an evolutionary computation technique, A PSO algorithm consists of behavior of flying birds and it means that they exchange of information to solve optimization problem. Which was originally implemented by Kennedy and Eberhart (1995). Xia and Wu (2005) implement hybridizations techniques with PSO and SA which assign operation on machine and schedule on every machine to solve FJSP hierarchically.

PSO is similar to the evolutionary algorithm in that the system is initialized with a population ("Swarm") of random solutions. Each particle remembers the best position that it has found so far during the search process personal best (pbest), and knows the best position of the swarm global best (gbest). In PSO techniques each particle interacts with other and every particle in the swarm tries to gradually move toward the promising areas of the search space and in this way an optimum solution is found [9].

Rescheduling Algorithm using hybrid PSO and SA [9]
Begin
Step I Initialization
1. PSO
a. Initialize swarm, including swarm size, particle position and velocity.
b. calculate the three objects (Ms, Mw, Tw, Ti, Tr), Ranking all particles with Pareto Set concept
2. SA
c. Determine the initial temperature level. T, Tend, B
Step II Computation
PSO
While (T> Tend) for each particle do
a. Particle displacement
b. Calculate the three objects (Ms, Mw, Tw, Ti, Tr), Ranking all particles with Pareto Set concept.
Calculate the Crowding distance Cd according to Eq. (3) on each Pareto Rank. If necessary, update gbest.
c. evaluate each particle according Eq. (4).
2. SA begin
Set k=1;
While (k<=Km) for each particle do
a. Randomly choose a neighbor
b. Accepting neighbor if find an optimal solution with respect to the current one, update it
k=k+1;
End while
Update T=T Kb;
SA end
Step III:
Output optimization result
End While

 Copyright 2020.All rights reserved.

VI. CONCLUSION

This article provides review of job shop scheduling problem and rescheduling strategy. it focus on different heuristic techniques used for solving JSSP and FJJSP. Rescheduling is a process of updating an existing production schedule in response to disruptions or other changes in manufacturing area.

REFERENCES

[1] N. Fnaiech C. Fitouri, "A New Heuristic Method for Solving Joint Job Shop Scheduling of Production and Maintenance", FAC-PapersOnLine 48-3 (2015) 1802–180
[2] Sadia Azem, Riad Aggoune,"Heuristics for Job Shop Scheduling With Limited Machine Availability",INCOM 2012, May 23-25, 2012
[3] Susana Fernandes, Helena R. Lourenço,"A SIMPLE OPTIMISED SEARCH HEURISTIC FOR THE JOB-SHOP SCHEDULING PROBLEM", DOI: 10.1007/978-3-540-70807-0_13 · Source: OAI
[4] Guilherme E. Vieira,Jeffrey W. Herrmann , " Rescheduling manufacturing systems: a framework of strategies, policies, and methods",
[5] Jian Zhang,Jian Zhang,"Review of job shop scheduling research and its new perspectives under Industry 4.0",J Intell Manuf,DOI 10.1007/s10845-017-1350-2
[6] Pankaj Sharma,Ajai Jain,"A review on job shop scheduling with setup times", J Engineering Manufacture 1–17 @ IMechE 2015,DOI: 10.1177/0954405414560617
[7] Ai-Hua Yin,"A Heuristic Algorithm for the Job Shop Scheduling Problem",H. Jin et al. (Eds.): NPC 2004, LNCS 3222, pp. 118-128, 2004.
[8] Holland JH., "Adaptation in natural and artificial systems", Detroit, MI: University of Michigan Press, 1975.
[9] Kamble S.V, "Hybrid Multi-Objective Particle Swarm Optimization for Flexible Job Shop Scheduling Problem", IJISA,2015,04,54-61, DOI: 10.5815/ijisa.2015.04.08
[10] Keith Schmidt,"Using Tabu Search to Solve the Job Shop Scheduling Problem with Sequence Dependent Setup Times"
[11] Rui Zhang, “"A simulated -Annealing based heuristic algorithm for job shop scheduling to minimize lateness", DOI: 10.5772/55956
[12] Habibeh Nazif, "SOLVING JOB SHOP SCHEDULING PROBLEM USING AN ANT COLONY ALGORITHM", Journal of Asian Scientific Research, 2015, 5(5):261-268

 Copyright 2020.All rights reserved.