International Journal of Advanced Mechanical Engineering.
ISSN 2250-3234 Volume 4, Number 2 (2014), pp. 169-174

http://www.ripublication.com/ijame.htm

Implementation of Simulated Annealing Technique for Optimizing Job Shop Scheduling Problem

Nakandhrakumar R.S[1] and Balachandar M[2]

[1,2,3,4]Department of Mechanical Engineering, Hindustan University, Rajiv Gandhi Road (OMR), Chennai, India.
E-mail: [1]rsn_kumar@rediffmail.com, [2]mbalachandar@hindustanuniv.ac.in

Abstract

The Job Shop Scheduling(JSS) problem consists of 'n' jobs and 'm' operations on each of the jobs and it is hardest combinatorial optimisation problems for which it is very complex to find optimal solutions. In recent year, much attention has been given to general heuristics technique such as Genetic algorithm, Ant Colony Optimisation, Tabu Search and Simulated Annealing for solving throughoptimization of this type of combinatorial optimisation problems. In this paper we present how the adaptive search algorithms namely Simulated Annealing Search technique is applied to solve Job shop scheduling (JSS) problem. The method uses dispatching rules to obtain an initial solution and searches for new solutions in a neighbourhood based on the critical paths of the jobs. Several benchmark problems are tested using this algorithm for the best makespan and the obtained results are giving encouraging results which compared with benchmark values.

Keywords: Job Shop Scheduling, Optimization, Simulated Annealing.

1. The Salient Features of Job Shop Scheduling Problem

Scheduling is an important phase in Production Planning. It concern the allocation of limited resources to tasks over time and focuses on how best to use the existing components, takes into account technical production constraints. The objective of this problem is to find optimal schedule by minimizing the makespan.In recent years, many heuristic techniques were made such as Genetic algorithm, and Simulated Annealing etc., to solve this type of combinatorial optimization problems. In this paper, an

attempt has been made to solve job shop scheduling problem by Simulated Annealing (SA) for minimizing the makespan, which satisfy minimum processing time constraint.

The standard model of the n-job, m-machine job shop problem is denoted by: n / m / G/ C^*_{max}.

The parameter G indicates that jobs are connected with technological production rules, describing their processing order of machines. This order is specified in the technological matrix. An example for Technological matrix (G) could be

$$\text{Technological matrix, } G = \begin{matrix} M1 \\ M2 \\ M3 \end{matrix}\begin{pmatrix} M2 & M3 & M4 \\ M3 & M4 & \\ M2 & M1 & \end{pmatrix} \qquad \text{Processing time, } T = \begin{pmatrix} t(O_{11}) & \ldots\ldots\ldots & t(O_{1m}) \\ t(O_{21}) & \ldots\ldots\ldots & t(O_{2m}) \\ t(O_{n1}) & \ldots\ldots\ldots & t(O_{nm}) \end{pmatrix}$$

The parameter $\hat{C}^*_{max}$ stands for the minimum makespan of the job shop and indicates the performance measure used to minimize the production time to finish all the jobs, taking into account the imposed restrictions of machine occupation.

1.1 Representation Models

A schedule is defined by a complete and feasible ordering of operations to be processed on each machine and in job shop there are two main ways of graphically representing such an ordering

- Gantt chart
- Disjunctive Graph

*1.1.1 Gantt chart.*The simplest and most widely used representation method is the Gantt chart. This chart (Table 1) is a matrix illustrating the relationship of jobs and machines when idle times and starting and /or completion times are inserted.

Table 1: Processing times and operation orders for a 4X3 instance.

Job	**Machine, (Processing Time)**		
	Operation 1	**Operation 2**	**Operation 3**
1	1 (4)	2(3)	3(2)
2	2(1)	1(4)	3(4)
3	3(3)	2(2)	1(3)
4	2(3)	3(3)	1(1)

1.1.2 Disjunctive graph model. A job-shop scheduling problem is often represented by a disjunctive graph.In a disjunctive arc, vertices, drawn as circles, represent tasks. Conjunctive arcs, which are directed lines, represent precedence constraints among the tasks of the same job. Disjunctive arcs, which are pairs of opposite directed lines, represent possible precedence constraints among tasks belonging to different jobs, these tasks being performed on the same machine. A graph representation for the problem stated in Table 1 is shown in Fig. 1

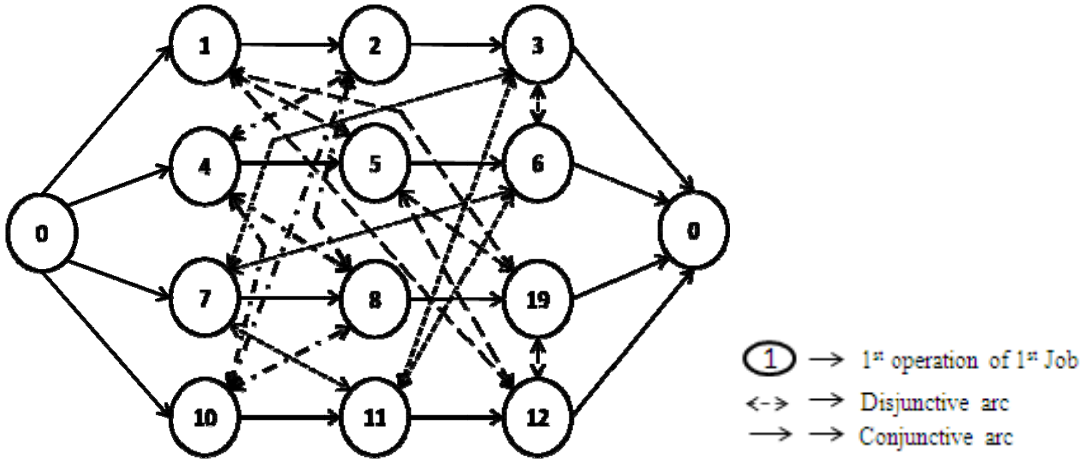

Fig. 1: Disjunctive Graph Representations for 4 X 3 Problems in Table 1.

2. Literature Review

Based on the Annealing concept of thermal process, Wiley has founded a new paradigm of evolutionary computation that has come to known as Simulated Annealing (SA) Atabak et al., [2011]. Since its development, researchers have applied SA to a number of combinatorial optimization problems. One area particularly important to manufacturing where there has been a lot of work is scheduling.

Min Dai et al. [2013] presented, the SA Optimization technique used to solve the problem of Job Shop Scheduling. This paper outlines the algorithm's implementation and performance when applied to job shop scheduling. In this paper some statistic analysis for parameter tuning is presented and the obtained solutions are compared with benchmark problems in job shop scheduling. Yuan Yuanet al. [2013] considered the SA technique for solving job shop scheduling problems. The performance measure considered is makespan time. The adjacent pairwise interchange method is used to generate neighborhoods. The results of search technique are compared with genetic algorithms. It was concluded that the performance of SA search is comparable to that of genetic algorithm.

3. Simulated Annealing

Simulated Annealing of Yuan Yuanet al., [2013] is a local search algorithm that requires a starting solution and a neighborhood structure. The procedure begins with an initial solution and stored it as the current seed and the best solution. The neighbours of the current seed are then produced by a neighbourhood structure. These are candidate solutions. They are evaluated for an objective function and a candidates of which is not a SA or satisfies the choose criterion is selected as a new seed solution. This selection is called move and added to SA list in order to create memory. The new seed solution is compared with the current seed solution. If better, it is stored as new best solution. Iterations are repeated until a stop criterion is satisfied.

3.1 Elements of SA Algorithm

The elements of the SA algorithm (John B. Chambers et al., 1995)[4] in connection with job shop problem are defined as follows.

3.1.1 Initial solution. A feasible initial solution is obtained by selecting from any one priority dispatching solutions. Initial solution affects the scheduling solution quality. In this paper, the shortest processing time (SPT) rule is used as an initial solutions method.

*3.1.2 Neighborhood Structure.*A neighborhood structure is a mechanism, which contain new set of neighbour solutions by applying a simple modifications to given solutions. Each neighbour solution is reached from a given solution by move. A sequencing move is defined by the exchange of certain adjacent critical operation pairs and then considered the exchange of every adjacent critical operation pair on every machine.

3.1.3 Move. The best neighbour which is not Simulated Annealing or satisfies a given aspiration criterion is selected as a new seed solution. “The best” neighbours is one whose objective function C_{max} is minimum and perturbation of the placement through a defined move.

3.1.4 Choose. Depending on the change in score, accept or reject the move. The probability of acceptance depending on the current “temperature”.

*3.1.5 Update and repeat.*The aim of the updating is the temperature value by lowering the temperature. If the temperature yields a solution better than the best obtained so far then it will be stored in the list otherwise go back to step 2..

3.1.6 Termination criterion. The process is done until “Freezing point” is reached.

3.2SA Algorithm

Initialization

- Represent the problem in using a disjunctive graph. Text
- set Cycle = 1 // initialize the cycle counter
- Set $SA^{cycle} = 0$ //Initialize the SA list

Neighborhood structure

- Finding the Critical path for the job order
- Finding the neighbours for above critical path
- Store the neighbor, which has minimum makespan as a best.
- Store the job sequence in each machine as a current job order.

Move

- Check whether best neighbor is in a SA list, if not so, go to step 2 or go to step 4
- Find the neighbors with minimum cycle in to visit’s list and deletes the neighbor in to visit list.
- Store the above neighbor as a best.
- Restore the job order in to visit’s list as current job order and go to step 4.
- Choose
- Store best neighbor, its makespan and its job order in the SA list.

- Store neighbors with its job order in neighbors list which is not in SA list as in to visit's list.

Update and repeat

- Cycle = cycle + 1; and go to step move
- Termination
- Finding the minimum makespan in SA list
- Store its makespan and job order as a best.

4. Implementation

Consider the technological matrix has been given above. It should be defined into a graph while using the SA system for Job shop scheduling. For this, General Initial temperature level is found to all the operations. The temperature travels in different paths and record the path it travels and the time of operation to complete all the jobs. When all the levels complete a tour, a cycle is completed. Then the temperature which is having the minimum completion time is chosen and is updated to this annealing path. After completing the total number of cycles, the final path is recorded. Then the makespan for the path is found to the machine order. Then using the makespan, its compared with other known makespan of benchmark values.

5. Results and Discussion

The developed software for the proposed ACO and TS algorithm have been tested for different bench mark problems proposed by Muth and Thompson (1963) (MT06), E. Taillard (1993) (TA00–TA05), Fisher and Thompson (FT10), LA 05, ORB (10x10)(ORB 01–ORB05). The obtained computational results are as follows,

Table 2: Results.

Test Instances	Makespan Known	Obtained Makespan SA
MT 06	62	63
TA01	60	60
TA02	66	67
TA03	58	58
TA05	68	68
LA05	593	593
FT10	1307	1311
ORB01	1328	1329
ORB02	1261	1262
ORB03	1551	1555

From the table it is found that, the obtained makespan value is more or less equal to the benchmark value for the small case problems. But the deviation exists when the problems goes higher.

6. Conclusion

Job shop scheduling can be solved by many techniques like Genetic algorithm, Simulated annealing, Tabu Search, Ant Colony Optimisation Shifting Bottleneck Procedure, etc. In this paper, SAis applied to JSS problem to obtain best makespan. The approach seems to be a promising one because of its generality in nature and its effectiveness in finding very good solutions to the difficult problems.

References

[1] AtabakElmi, MaghsudSolimanpur(2011), A Simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts, Computers & Industrial Engineering, 61, pp. 171-178.

[2] Min Dai, Dunbing Tang and Adriana Giret (2013), Energy–efficient scheduling for a flexible flow shop using an improved genetic–simulated annealing algorithm, Robotics and Computer–Integrated Manufacturing, 29, pp. 418–429.

[3] Yuan Yuan, HuaXuand Jiadong Yang (2013), A hybrid harmony search algorithm for flexible job shop scheduling problem, Applied Soft Computing, 13, pp. 3259–3272.

[4] Min Liu, Zhi-Jiang Sun, Jing-Song Kang and Jun-Wei Yan (2011), An adaptive annealing genetic algorithm for the job-shop planning and scheduling problem, Expert Systems with Applications, 38, pp. 9248-9255