

Application of Simulated Annealing in Flow Shop Scheduling

Arjun.K.R Dr. M. S Jayamohan

Assistant Professor, Dept of Mechanical Engineering, Amal Jyothi College of Engg, Kanjirappally, Kerala, India

Professor Dept. Of Mechanical Engineering, Rajiv Gandhi Institute of Technology, Kottayam, Kerala, India

Abstract

Scheduling processes deal with the rank ordering and allocation of jobs at different resources and/or locations available in a manufacturing shop. The conventional researchers associated with shop scheduling gave thrust on the optimization of one or many of the performance measures available in literature. The complexity of the problem depends on the number of jobs and machines involved in each case. The recent advances in the arena reveal application of meta-heuristics in scheduling to be relatively more efficient in many cases than regular conventional techniques.

This study intends to discuss the application of Simulated annealing as applied to flow shop scheduling optimization. The influence of the algorithm parameters and operators are initially surveyed in detail. The flow shop instances considered for the study are open static flow shops to cover a larger number of candidate schedules in the search-space, due to the inherent levels complexity and variety with which machines process the jobs.

Keywords: Scheduling, Meta heuristics, Simulated Annealing

I. INTRODUCTION

Scheduling is the allocation of resources over time to perform a collection of tasks. This definition conveys two meanings. First, Scheduling is a decision making function i.e., process of determining a schedule. Second, scheduling is a body of theory i.e., it is a collection of principles, models, techniques and logical conclusions that provide insight into the scheduling function. Sequencing is the order of processing a set of tasks (or jobs) over resources (or machines). Scheduling involves sequencing as well as the determining of time of process commencement and completion, i.e., time tabling .

Among the typical goals of scheduling problems, maximizing machine utilization is not only a measure of academic interest, but also useful and important in practice. As a decision making function, it tries to attain a local or global optimization of one or more objectives. The resources described may be men, machines, runways at airport etc. and task may be operation in a production process, and take-off or

landing in an airport etc. Each task may have a different priority level, earliest possible starting time or date and a due date. The objectives may be one or many, such as minimizing the completion times of the individual tasks, or minimizing the completion time of the last task or even minimizing the number of tasks for which the completion time exceeds the pre defined due date.

In its general form, problem of scheduling exists in most manufacturing and production systems, information processing environments, transportation distribution systems and in many service situations. Production scheduling is a function to determine an actual (optimal or feasible) implementation plan with respect to the time schedule for all the jobs to be executed. This is done after product items and production quantities have been decided by production planning and production processes have been decided by process planning. In any organization or system, the scheduling function interfaces with many other functions. In a manufacturing system, orders are released and are to be translated to jobs with associated due dates. The jobs are to be processed by machines in a work center in a specific order or sequence. Jobs may have to be waiting for processing on machines that are busy, and in actual practice preemption may occur when higher priority jobs arrive at the center.

In all these cases, scheduling of tasks effectively helps in maintaining an efficient and effective control of operations on the shop floor. The origin of the terminology of scheduling is from the manufacturing systems. Manufacturing systems can be classified into jobshops, flowshops, automatic transfer lines, flexible transfer lines, cellular manufacturing systems, flexible manufacturing systems, assembly shops, flexible assembly systems, multiple cell systems, just-in-time manufacturing systems etc.

The application of conventional optimization techniques has many inherent limitations. They include complex mathematical computations, development of generalized software solutions, procurement and use of problem specific information and computational complexity. The study intends to analyze the scenario with an attempt to optimize different performance parameters for flow shop scheduling. The use of Simulated Annealing helps to reduce the computational complexity. This also helps to compare the efficiency of optimization tools in solving scheduling problems.

The work done include (i) Analyzing the scenario with an attempt to optimize makespan for flow shop scheduling (ii) Use of Simulated Annealing to reduce computational complexity (iii) To apply the algorithms on benchmark problems

2. LITERATURE SURVEY

The flow-shop scheduling problem has been an area of key research for over last three decades. Johnson 1954 proposed a 2 stage scheduling problem with the objective of minimizing the make span. The scheduling problem was taken as a conventional mathematical programming problem by Bowman (1959), Wagner (1959), Manne (1960) and Balas (1969) and a linear programming formulation with integer constraints in its solution was suggested by them. The problem of scheduling arises from the fact that the undirected arcs in the network (being represented as disjunctive arcs (Balas (1969)) do not allow a straightforward determination of the minimal maximal path in the network which reflects the minimal maximal flow time for scheduling problem. The problem is to solve this semi-directed network problem. Balas (1969) proposed a general method of solving scheduling problem. Charlton and Death (1970) also proposed a general method for solving scheduling problem but stated that applicability of the proposed method was limited to problems of very moderate size only.

The most successful solution procedure for flowshop scheduling problems is the branch and bound algorithm (Baker (1974), and French (1982)). Ignall and Schrage (1965) applied the basic branch and bound technique of Little, Murthy, Sweeney and Karel (1963) to the flowshop problem having three stages with makespan objective. In a parallel but independent effort, Lominicki (1965) also proposed a

similar algorithm. McMohan and Burton (1967) proposed job-based bounds complimentary to the machine-based bound that was determined from unsatisfied requirement for unscheduled jobs

Ashour and Quraishi (1969), however, obtained a result contradicting with that of McMahan and Burton(1967), in that the machine-based bound was noticeably faster than the combination of machine-based and job-based bounds. Ashour(1970) came up with a new bound called noninterference bound. Baker (1975) carried out a comprehensive and detailed investigation of the machine-based, job-based and non interference bounds, and elimination strategies in a comparative study along with an improved solution algorithm. Rajendran and Chaudhuri (1993) developed a branch and bound algorithm for scheduling to minimize makespan in the flowshop with parallel processors. They found that some parallel-processor flowshop problems could be solved easier than the single-processor flowshop problems. Abadi and

Sriskandarajah (1995) described the blocking flow shop problem as follows. The flowshop has no intermediate buffer therefore a job cannot leave a machine until the next machine downstream is free. If that is not the case, the job (and the machine as well) is said to be blocked. Aldowaisan and Allahverdi (1998) described the case in which once a job begins its processing on machine 1 of the production line, that job must continue without delay to be processed on each of the m machines in line. Not only are there no integer stage buffers to hold delay jobs, but also no job may wait on one machine until the subsequent machine in line is free to begin processing on that job. Aldowaisan and Allahverdi (1998) refer to this as the no-wait flow shop problem. More recently (2003), they proposed two heuristics based on Simulated Annealing and Simulated Annealing for the no-wait flow shop problem to minimize makespan. Hybrid flowshop problems have attracted much research during the last few years. Zhixing *et al.* (2002) describe the no-wait hybrid flowshop problem. Oyetunji (2009) has studied the common performance measures in scheduling problems. It discusses twenty nine different scheduling criteria and the mathematical expressions for each of the cases.

Some of the noteworthy dispatching rules that have been recently proposed and extensively studied are due to Baker and Kanet (1983), Vepsalainen and Morton (1987), Anderson and Nyirendra (1990), Raghu and Rajendran (1993), and Holthaus and Rajendran (1997). Jayamohan and Rajendran (1999) suggested new dispatching rules incorporating weights relating to flow time and tardiness of jobs

Osman and Potts (1989) proposed a simple Simulated Annealing algorithm using a shift neighborhood and a random neighborhood search. The SPIRIT algorithm proposed by Widmer and Hertz was a two phase heuristic. Taillard (1990) also presented a similar procedure to that of Widmer and Hertz. A Tabu search technique is applied to a schedule generated by an improved NE heuristic. Ogbu and Smith (1990) proposed a Simulated Annealing approach to the Permutation Flow Shop Problem which involved an initialization with the Palmer and Dannenberg heuristics

In Permutation Flow Shop scheduling research Chen *et al.* (1995) developed a simple Simulated Annealing. Ishibuchi *et al.* (1995) presented Simulated Annealing algorithms characterized by having robust performance with respect to the temperature cooling schedule. The results attained from the algorithm were comparable with the SA of Osman and Potts. Younes *et al.* (1998) discusses a Simulated Annealing approach to scheduling problems involving multiple processors. This deals with the minimization of the makespan in a flowshop. The algorithm used is Simulated Annealing which is initial candidate solution formed using heuristic based search. Wodecki and Bo_zejko (2002) proposed a SA algorithm specifically meant to run in parallel computing environment. The algorithm was compared with NEH heuristic and yielded better results. The authors (2004, 2006) also studied on the application of genetic algorithm in parallel computing environments. The former of the studies deal with the application of minimizing total weighted completion time where as the latter uses a inter-island genetic operator for optimization problems with block properties. Rajendran & Ziegler (2004) have proposed two very

effective ant-colony optimization (ACO) algorithms and Grabowski & Wodecki (2004) a very fast TS approach. Wang *et al.* (2008) proposed a two-stage GA to find the optimal solution. Sayadi *et al.* (2010) discusses a discrete fire fly algorithm for combinatorial optimization. This deals with minimizing the makespan and flow time in permutation flowshops.

3. FLOW SHOP SCHEDULING USING SIMULATED ANNEALING

Shop scheduling problems belong to the class of multi-stage scheduling problems, where each job consists of a set of operations. For describing these problems, we use the standard 3-parameter classification $\alpha|\beta|\gamma$. The parameter α indicates the machine environment, the parameter β describes job characteristics, and the parameter γ gives the optimization criterion.

Flow-shop scheduling problem is a set of jobs that flow through multiple stages in the same order. In a flow shop problem ($\alpha=F$), each job has exactly m operations, and the technological route (or machine order) in which the job passes through the machines is the same for any job. Without loss of generality, we assume that the technological route for any job is given by $M1 \rightarrow M2 \rightarrow \dots \rightarrow Mm$

The problem of scheduling in permutation flow shops has been extensively investigated by many researchers. Exact and heuristic algorithms have been proposed over the years for solving the static permutation flow shop scheduling problems with the objectives of minimizing the makespan or total flow time of jobs. The makespan is the time required to complete the processing of all the jobs in the shop. The minimization of the makespan results in an efficient utilization of resources. Total flow time is the time spent by a job in the system. The minimization of mean flow time leads to the reduction of the average number of jobs waiting in queues. If the jobs spend less time in the system, the in-process inventory is reduced.

The general flowshop scheduling problem is a production problem where a set of n jobs have to be processed with identical flow pattern on m machines. When the sequence of job processing on all machines is the same we have the permutation flowshop sequencing production environment. Since there is no job passing, the number of possible schedules for n jobs is $n!$. Usually, the schedule performance measure is related to an efficient resource utilization looking for a job sequence that minimizes the makespan; that is the total time to complete the schedule and the problem is then denoted as $n/m/P/f_{max}$ or as $F/prmu/C_{max}$ (Pinedo, 2002).

A flow shop scheduling problem is generally modeled on the following assumptions:

- i) The operation processing times on the machines are known, fixed and some of them may be zero if some job is not processed on a machine;
- ii) Set-up times and breakdown times are included in the processing times and they are independent of the job position in the sequence of jobs;
- iii) At a time, every job is processed on only one machine, and every machine processes only one job;
- iv) The job operations on the machines may not be preempted
- v) The machines are continuously available

The simulated annealing method is analogous to the cooling process of molten metal through annealing. At very high temperatures the atoms in the molten metal can move freely with respect to one another. The temperature when gets reduced, the movement of the atoms get restricted. The atoms at reduced temperatures start to get ordered and finally form crystals having minimum possible energy. The formation depends on the cooling rate. If the temperature gets reduced at a fast rate the crystalline state may not be achieved and may be replaced by a poly crystalline state. The energy associated with the state is relatively higher. So as to achieve a minimum energy level the temperature needs to be reduced at a

slow rate. This process of slow cooling in metallurgical parlance is called as annealing.

The simulated annealing approach is based on the ideas from statistical mechanics and motivated by an analogy to the behaviour of physical systems in the presence of a heat bath. The non-physicist, however, can view it simply as an enhanced version of familiar technique of local optimization or iterative improvement, in which an initial solution is repeatedly improved by making small local alteration until no such alteration yields a better solution. Simulated annealing randomizes this procedure in a way to attempt to reduce the probability of becoming stuck in a poor, but locally optimal solution. As with local search, simulated annealing can be adapted readily to new problems (even in the absence of deep insight into the problems themselves) and, because of its ability to avoid poor local optima, it offers hope of obtaining significantly better solutions (Johnson *et al.* 1989).

The temperature T is a parameter which acts like an iteration (or time counter) for the algorithm. The temperature is successively reduced by a reduction factor, r . When the temperature reaches a pre-specified level, called the freezing temperature, the procedure is said to be 'frozen', and is terminated. The main reason for which simulated annealing is supposed to give a good solution is due to the fact that it also considers inferior solutions (of course, with a certain acceptance criterion) in search of a good solution.

(i) *Simulated Annealing procedure*

- a) Initialise a random schedule S .

Set Initial temperature, Final temperature and Temperature decrement factor. Initiate FR_CNT ,

$TOTAL$ and $ACCEPT$ and set its upper limit

- b) Find the objective function value for the initial sequence

Assign $BEST = S$

- c) Use Perturbation Operator and Generate schedule S' .

Increment $TOTAL$ and $ACCEPT$. Repeat till counter exceeds upper limit

- d) Find out the needed performance measure value for each S' and pick the best value
e) Compare the performance measure values of S' and $BEST$ and assign the better value to $BEST$
f) Compare the performance measure values of S and S' and assign the better value to S .
g) If performance measure value of S' is worse than S apply Boltzmann Probability function to accept the schedule S' to replace S
h) Decrement Temperature and Repeat the process
i) Terminate when the termination condition is reached

4. PROBLEM FORMULATION

The area of the study is flowshop scheduling. The flowshop scheduling problem involves many parameters of performance. The consideration of any one of the parameters as the objective of the study does not depict the actual scenario. This demands for the need of considering more than one of the objectives. The study analyses the performance criteria of Makespan

i. *Parameters:*

- n: Number of jobs
- m: Number of machines
- i: Machine index, ($i=1, \dots, m$)
- j: Job index, ($j=1, \dots, n$)
- k: Order index, ($k=1, \dots, n$)
- t_{ij} : Processing time of job j on machine i

ii. *Decision variables:*

- q_{ijk} : Completion time of job j on machine i in k th order
- x_{jk} : Binary variable taking value 1 if job j is processed in k th order and 0, otherwise.

iii. *Objective Function*

The mathematical model for minimizing the makespan is as follow:

$$\text{Min } \sum q_{\min} x_j \quad (1)$$

iv. *Constraints*

$$\sum_{i=1 \dots m-1} q_{(i+1)jk} - t_{(i+1)j} x_{jk} > \sum q_{ijk} x_{jk} \quad (2)$$

$$\sum q_{ij(k+1)} - t_{ij} x_{j(k+1)} > \sum q_{ijk} x_{jk} \quad (3)$$

$$\sum x_{jk} = 1 \quad \forall i, k \quad (4)$$

$$\sum x_{jk} = 1 \quad \forall i, j, k \quad (5)$$

$$q_{ij} \leq M x_{jk} \quad \forall i, j, k \quad (6)$$

$$q_{ijk} \geq 0, \quad \forall i, j, k \quad (7)$$

$$x_{ijk} = \{0, 1\} \quad \forall i, j, k \quad (8)$$

The objective function (1) represents the minimization of the makespan. Constraints (2) and (3) certify that a job does not start on a machine until it finishes processing on the previous machine and its predecessor has completed processing on that machine. Constraint (4) insures that each sequence position is filled with only one job and constraint (5) insures that each job is assigned to only one position in the job sequence. The constraint set (6) is a relationship between the binary variables and the completion time variables. When each binary variable becomes zero, then its completion time variable will be also equal to zero. Finally (7) and (8) are logical constraints

5.BENCHMARK PROBLEMS

The benchmark problems are similar to real time problems in industrial scenario. The types of problems

vary over shop configurations viz flow shop, job shop and open shop. There are n jobs that have to be performed on m unrelated machines; in our case, every job consists of m non pre-emptible operations. Every operation of a job uses a different machine during a given time and may wait before being processed. They form a common base for the comparison of the scheduling problems.

Taillard in his different research works in scheduling discusses 120 benchmark instances for the flow shop scheduling problems for different 'm' machine and 'n' job combination. The benchmark instances specify the processing time which can be applied for simulating the system and iterating to attain performance measures. Taillard on solving the scheduling problems discusses the upper bound and lower bound of the objective function when applied by any of the heuristics. For the study in flowshop scheduling Taillard's instances of two different dimensions are taken. They are problems with 20 jobs x 10 machines and 20 jobs x 20 machines respectively. Three variations of each case has been taken for the study.

6.RESULTS AND DISCUSSSIONS

Simulated Annealing for the optimization of makespan and tardiness for both the flowshop and jobshop are coded in MATLAB and ran on Intel Core2 Duo PC with 2GB memory. A detailed study of the different parameters and their influence in the scheduling optimization were carried out. This was done to fix the best parameters for the benchmark problems considered.

The flowshop scheduling scenario deals with the following instances proposed by Taillard.

- 1) Ta (20 × 10) – 3 cases
- 2) Ta (20 × 20) – 3 cases

TABLE 1MAKESPAN OF 20 X 10 FLOWSHOP INSTANCES

Instances	Makespan
Ta_20 × 5 case a	4910
	5065
	4916
Ta_20×5 case b	4727
	4713
	4449

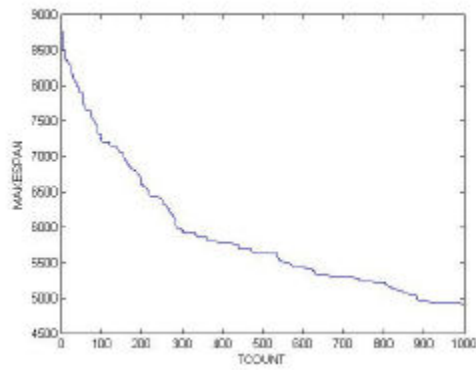


Fig 1. Makespan optimization using SA in $T_a20 \times 10$ case a

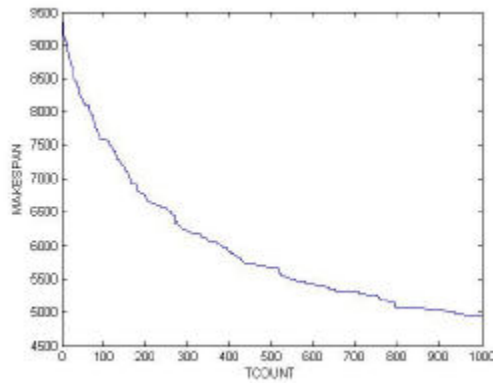


Fig 2. Makespan optimization using SA in $T_a20 \times 10$ case b

TABLE 2 MAKESPAN OF 20 X 20 FLOWSHOP INSTANCES

Instances	Makespan
Ta ₂₀ × 20 case a	9497
	9779
	9744
Ta ₂₀ × 20 case b	9060
	8991
	8773

Fig 3. Makespan optimization using SA in T_a 20×20 case a

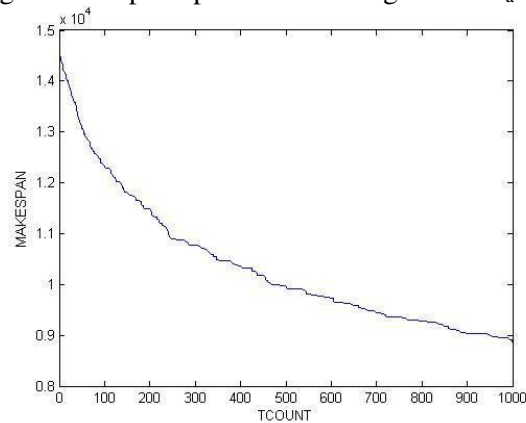
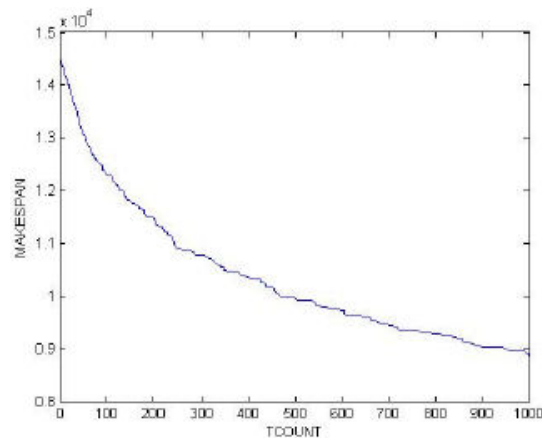


Fig 4 Makespan optimization using SA in T_a 20×20 case b



In the case of instances involving 20 jobs and 10 machines, Simulated Annealing algorithm uses the two random-site shift perturbation operator. The increase in the order and complexity of the problem due to the increase in the number of machines limits the operator in its efficiency. The dimensions of the search space involving candidate solutions increase and the operator needs more number of iterations to effectively search for better random candidate solutions in the space. The iterations were stopped at 1000 which was one of the termination conditions. The solutions showed better convergence after 800 iterations.

In the case of instances involving 20 jobs and 20 machines, the higher degree of complexity of the instances owing to the increase in the dimensions of the problem limits the efficiency of the two random-site shift perturbation operator. The search space of candidate solutions is large in instances of larger dimensions.

7.SCOPE FOR FURTHER STUDY

- i. The study can be extended to the use of other heuristics viz. Genetic Algorithm, Ant colony optimization, Taboo Search and Particle Swarm Optimization etc and the results can be compared
- ii. More performance measures relating to tardiness and flow time can be incorporated to supplement more detailed scheduling

- iii. The scheduling study can be extended to job shops.
- iv. The same methodology adopted can be improved by the application of hybrid algorithms.
- v. Problem of scheduling can be addressed using dispatching rules in cases of increased complexity and varied and higher utilization levels

VIII. CONCLUSION

The proposed Simulated Annealing model is found to be effective in addressing scheduling problem in a flow shop. Same problem was done using Genetic Algorithm (GA) and SA model was found more effective and shows better convergence as the complexity of the problem was small.

REFERENCES

- [1] Sanjay Radhakrishnan & Jose A. Ventura (2000), Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence dependent set-up times, *International Journal of Production Research*, 38, 2233- 2252
- [2] Li-Ning Xing ,Ying-Wu Chen ,Ke-Wei Yang (2009), An efficient search method for multi-objective flexible job shop scheduling problems , *Journal of Intell Manufacturing* ,20, 283–293
- [3] Taillard, E , "Some Efficient Heuristic Methods for the Flow Shop Sequencing Problem," *European J. of Operational Research*, 47, 65-74, 1990
- [4] Widmer M , Hertz. A , "A new heuristic for the flowshop sequencing problem", *European Journal of Operational Research* 41 (2), 186-193, 1989
- [5] Taillard.E , "Benchmarks for Basic Scheduling Problems." *European Journal of Operational Research* 64, 278-285, 1993
- [6] Osman I H, Potts C N(1989) , "Simulated annealing for permutation flow-shop scheduling." *Omega*, Volume 17, 551-557
- [7] Younes N. *et al.*(1998), "A Simulated Annealing Approach to Scheduling in A Flow Shop with Multiple Processors", *Industrial Engineering Research Conference Proceedings*, Banff, Canada
- [8] Ponnambalam.S.G., Aravindan.P, , Chandrasekaran.S , "Constructive and improvement flow shop scheduling heuristics: An extensive evaluation", *Production Planning and Control*, 12 (4) , 335-344, 2001
- [9] Ruiz.R, Maroto. C.A , Comprehensive review and evaluation of permutation flowshop heuristics, *European Journal of Operational Research*, 165, 479-494, 2005
- [10] Jayamohan.M.S, C. Rajendran , "A comparative analysis of two different approaches to scheduling in flexible flow shops", *Production planning & control*, 11(6), pp. 572-580, 2000
- [11] Pinedo, M, *Scheduling: Theory, Algorithms and Systems*, 2nd ed, Prentice-Hall Englewood Cliffs, NJ, 2002
- [12] Mahadevan B "Operations Management : Theory and Practice " Pearson Education Ltd., Fifth Impression, 2009
- [13] Deb.K, "Multiobjective Optimization using Evolutionary Algorithms." John Wiley and Sons ,USA ,2002