ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

# Job Shop Scheduling Using Modified Simulated Annealing Algorithm

**PV Senthiil*, Mirudhuneka, Aakash Shirrushti**
* Head, Mechanical Engineering, SPIHER, St.Peters University,Chennai-600054, India
SAP Consultant, IBM Ltd, Porur, Chennai, India
Department of Mech, SRM University, Chennai, India
drpvsenth@gmail.com

## Abstracts

Timely and cost factor is increasingly important in today's global competitive market. The key problem faced by today's industries are feasible allocation of various jobs to available resources i.e., machines (Scheduling) and optimal utilization of the available resources. Among the various problems in scheduling, the job shop scheduling is the most complicated and requires a large computational effort to solve it. A typical job shop scheduling problem has a set of jobs to be processed in a set of machines, with certain constraints and objective function to be achieved. The most commonly considered objectives are the minimization of make span, minimization of tardiness which leads to minimization of penalty cost, and to maximize machine utilization. Machine shop scheduling can be done using various techniques like standard dispatching rules, heuristic techniques like Simulated annealing, Tabu Search, Genetic algorithm, etc,.here a typical job shop shop scheduling problem is solved using simulated annealing(SA) technique, a heuristic search algorithm. SA is generic neighbourhood search algorithm used to locate optimal solution very nearer to global optimal solution. A software based program is developed in VB platform for a typical job shop problem and test instances were performed over it. Experimental results obtained were further tuned by varying parameters and optimal results were obtained

**Keywords**: Job Shop, simulated Annealing, Tardiness, Performance Measure.

## Introduction

Job scheduling deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives. The resources and tasks in an organization can take many different forms. The resources may be machines in a workshop, runways at an airport, crews at a construction site, processing units in a computing environment, and so on. The tasks may be operations in a production process, take-offs and landings at an airport, stages in a construction project, executions of computer programs, and so on. Each task may have a certain priority level, an earliest possible starting time and a due date. The objectives can also take many different forms. One objective may be the minimization of the completion time of the last task and another may be the minimization of the number of tasks completed after their respective due dates

Scheduling, as a decision-making process, plays an important role in most manufacturing and production systems as well as in most information processing environments. It is also important in transportation and distribution settings and in other types of service industries. The following examples illustrate the role of scheduling in a number of real world environments.

**The scheduling problem**

The difficulty inherent in scheduling activities arises from the complexity and uncertainty which are characteristic of flexible manufacturing environments. This is due to the fact that scheduling is not an isolated function separate from other manufacturing functions. Scheduling activities are dependent upon decisions made elsewhere. The presence of a limited number of machines performing different operations and having different features and capacities add further complexity.

Scheduling decisions which are made manually in the shop are not always satisfactory. The impact of a current choice upon future events is difficult, if not impossible, predict, programming errors, communication errors, hardware errors and manufacturing errors must be accounted for directly and acted upon in a way that causes minimum damage and downtime. Completely avoiding errors, or completely negating their effects, is rarely possible. However, the complexity imposed by a variety of constraints and preferences and the inherent uncertainty prevent human

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

schedulers from producing efficient schedules. This is evidenced by high work-in-process inventories, tardiness, and low machine utilization and high overhead costs. 50% of work instructions issued by the planning/scheduling function are subject to modification on the shop-floor

## Problem description

The most generic and complicated problem in scheduling field is a typical job shop with m parallel machines. A job shop comprises of set of n jobs to be processed in m machines with a pre-determined sequence. But in a job shop with identical parallel machines, sequences are not required. The problem is considered to be a flexible job shop scheduling problem (FJSP).

Scheduling of a FJSP requires a lot of computational efforts and large volume of data flow. FJSP models modern flexible manufacturing environment with work centers that comprise a number of machines that are identical or similar in functionality. An addition practical requirement is for a job to visit the same machine several times on its route through the system.

This feature of job recirculation further increases the computational complexity of the FJSP. Techniques to solve the FJSP include enumerative methods such as branch and bound [11] that can guarantee solution optimality but do not scale well for larger problems, while approximation and randomized algorithms sacrifice certainty and optimality for efficiency.

**Given industrial situation**

- AXLES INDIA PVT LTD .,CHENNAI.
- Typical Job Shop scheduling problem.
- Allocation of n jobs to m machines.
- Objective - Reduce Makespan, Tardiness and increase Machine utilization

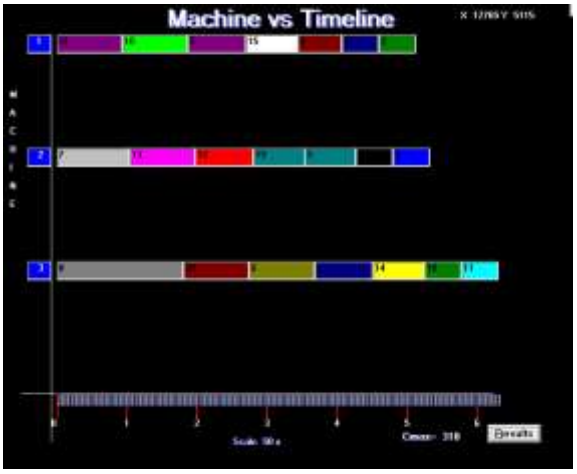


***Fig 1 Standard Dispatching Rule Gantt Chart***

This chart visually displays the schedule of the jobs marked in colour over the graph of machines as Y scale and time as X scale. The Gantt chart representation is most used representation as it easily helps to identify the bottleneck and idle ness of various jobs and machines.

Formula used for find out the results

- Completion time, cj: time at which the job is finished.
- Flow time, fj = cj – RJ: (1)
time spent by the job in the shop.
- Lateness, lj = cj – DJ (lj < 0 (2)
denotes earliness).
- Tardiness, tj = MAX{0, lj}. (3)
- Earliness, ej = MAX{0, - lj}. (4)
- Makespan, the maximum completion time of all jobs:

Cmax = MAX{cj}.

- Maximum lateness, lmax = MAX{lj}.
- Maximum tardiness, tmax = MAX{tj}.
- Total flowtime, □ fj.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

**Standard dispatching rule (SPT) results**
Cmax ,Tardiness and Earliness values are in mins

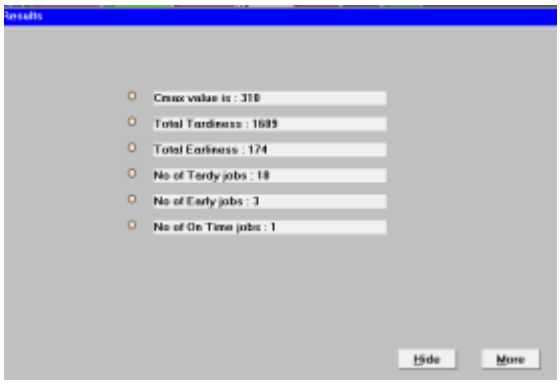


***Fig. 2 Standard Dispatching Rule Results***

## Simulated annealing algorithm

{ START
Step 1:
Set k =1 and select β1.
Select an initial sequence S1 using some heuristic. Select So = S1.

Step2:
Select a candidate schedule Sc from the neighborhood of Sk.

If G (So) < G (Sc) < G (Sk), set Sk+1 = Sc and go to step 3. If G (Sc) < G (So), So= Sk+1 = Sc and go to step 3.
If G (Sc) > G (Sk), generate a random number U from a uniform (0, 1) distribution; If Uk < P (Sk, Sc), set Sk+1 = Sc; otherwise, set Sk+1 =Sk and go to step 3.

Step 3:

Select βk+1 < βk.
Increment k by 1.
If k = N then STOP;
Otherwise, go to step 2.

}

**Algorithm description**
The simulated annealing procedure goes through a number of iterations. At iteration k of the procedure, there is a current schedule Sk as well as a best schedule found so far, So for a single-machine problem, these schedules are sequences (permutations) of the jobs. Let G (Sk) and G (So) denote the corresponding values of the objective function. Note that G (Sk) >G (So). The value of the best schedule obtained so far,G (So),is often termed the aspiration criterion. The algorithm, in its search for an optimal schedule, moves from one schedule to another. At iteration k a search for a new schedule is conducted within the neighbourhood of Sk. First, a so-called candidate schedule, say, Sc, is selected from the neighbourhood. This selection of a candidate schedule can be done at random or in an organized, possibly sequential, way. If G (Sc) < G (Sk), a move is made, setting Sk+1 = Sc. If G (Sc) < G (So), then S0 is set equal to Sc. However, if G (Sc) > G (Sk), a move is made to S with probability.

## Developed algorithm for job shop scheduling using simulated annealing technique

{ START STEP 1:

Input No. of parallel machines (mi): i = 1….. to m
No. of jobs, (nj): j = 1….. to n
Processing Time, (Pij) Due Time, (dj) Weightage, (Wj)

STEP 2:
Allocate the jobs to machines using Standard Dispatching Rule.
The obtained solution (Schedule) is considered as initial solution (So). Let Go be the corresponding objective function of So .
STEP 3:

Get N – maximum iteration count (Termination condition).

Set k =1 and select β1.

Set So = Sk

STEP 3a:

Check for idle times in each machine m.

If any idle time value, ID> Pij, Processing Time of any job then

Check for

Is_Same_Job_Running_On_Another_Machine_At_Same_Time

If above condition is true then check for other equivalent jobs which can fit in the idle time gap.

If possible to shift then transfer the job to new position, and the solution (schedule) obtained is candidate solution Sc.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

Calculate G (Sc)

STEP 4:

If G (So) < G (Sc) < G (Sk),
[< is used as for Cmax objective function minimization condition is used] Set Sk+1 = Sc and go to step 5.

If G (Sc) < G (So) < G (Sk)
Set So= Sk+1 = Sc and go to step 5. If G (Sc) > G (Sk), then
P (Sk. Sc) = exp [(G (Sk) - G (Sc))/ βk]

If Uk < P (Sk, Sc) where Uk is a randomly generated from (0,1) distribution.

Set Sk+1 = Sc;

Otherwise, set Sk+1 =Sk and go to step 5.

STEP 5:

Select βk+1 < βk.

Increment k by 1.

If k = N then STOP; otherwise, go to step 3a.

END

}

## Algorithm description

The simulated annealing algorithm is modified for job shop scheduling problem. The modified algorithm is used to develop a VB based program which is used to perform the calculations and to develop a schedule.

The first step of the algorithm gets the various inputs like no. of machines, (m), no. of jobs (n), processing times (Pij), due time (Dj) and store it in a database.

In the second step the jobs are allocated to machines using the standard dispatching rule. The obtained schedule (solution) is considered as initial solution (S0). Let G0 be the corresponding objective function of S0.

The no. of maximum iteration count N is fixed and k is set as k=1. Also the cooling parameters β1are selected. The value βk is calculated from βk= ak where a is selected between 1 and 0.

In the step 3a the idle times of each machine are founded out. The idle time gap is analyzed so that any of the processing time, (Pij) of jth job ith machine can be reallocated to that gap. This process involves in two steps.

First the gap should be enough to fix a job of (Pij) in it. Secondly, the same job should not run in another machine. To find this a function called is is_Same_Job_Running_On_Another_Machine_At_Same_Time developed and checked for its condition.

If the above condition is true then check for other jobs with equivalent processing times to the available gap. If the condition becomes false then the job can be shifted to newer position and the solution (schedule) obtained is termed as candidate solution (Sc).

The corresponding objective function values G (Sc) are calculated.

In the step 4 the objective function values of initial solution, G (So), candidate solution, G (Sc) current solution, G (Sk) are compared.

If G (So) < G (Sc) < G (Sk) then set Sk+1 = Sc and go to step 5. i.e., the candidate solution is made as next current solution.

If G (Sc) < G (So) < G (Sk) then set So = Sk+1 = Sc and go to step 5. i.e., the candidate solution is set as next current solution and so far found best solution.

If G (Sc) > G (Sk), then find the probability function

P (Sk. Sc) = exp [(G (Sk) - G (Sc))/ βk]

If Uk < P (Sk, Sc) where Uk is a randomly generated from (0,1) distribution.

Set Sk+1 = Sc;

Otherwise, set Sk+1 =Sk and go to step 5.

i.e., if the objective function value of candidate solution is greater than that of current solution then probabilistic criterion is used to select the solution.

In the next step the new βk+1 values are selected which should be lesser than the older βk value. The iteration counter k is incremented by one and checked for termination condition. If no. of iteration exceeds the

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

preset count then the algorithm stops, else it loops again from step
3a.

**Machine job input form**
In this form we give no. of machines and no. of jobs as input


*Fig 3 Machine and Job Input Form*

DATA INPUT FORM

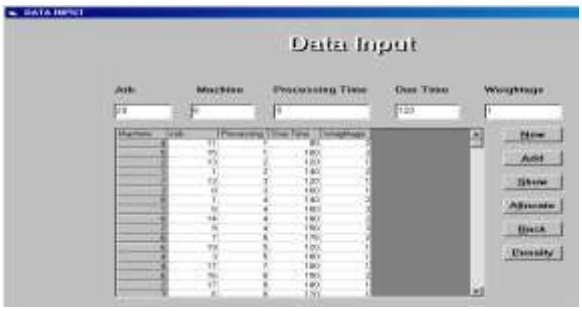


*Fig. 4 Data Input Form*

4.2 SIMULATION TABLE

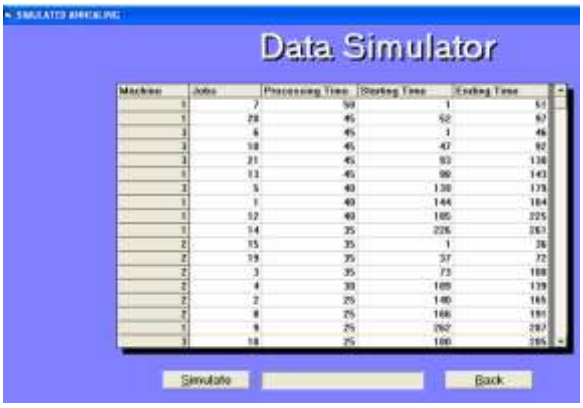


*Fig 5 Simulation Table Form*

This form shows the process of the simulation. The result (schedule) from standard dispatching rule is taken as initial solution and is iterated using simulation annealing algorithm and final optimal result (schedule) is obtained in this form

## Final result gantt chart

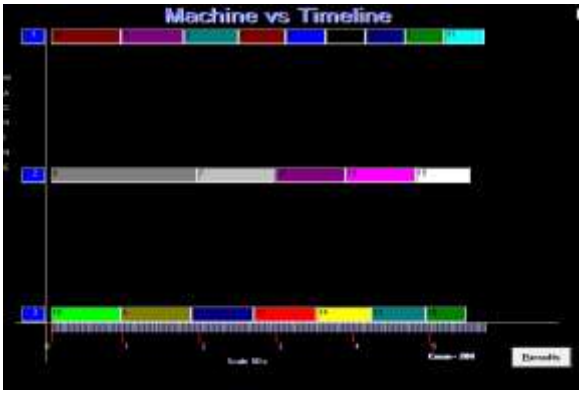


*Fig 6 Gantt chart of Final Schedule*

**Developed simanneal program results**
Cmax ,Tardiness and Earliness values are in mins

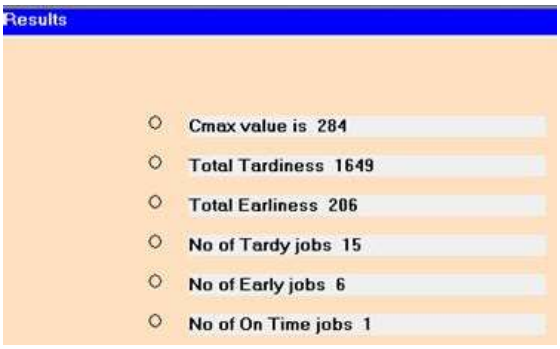


*Fig.7 Results of developed SIMANNEAL program*

Results

| Machine | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Machine Running Tim | 255 | 235 | 250 | 0 |
| Machine Idle Time | 6 | 26 | 11 | 261 |
| Machine Utilization % | 97.70 | 90.03 | 95.78 | 0 |

*Table 1 Machine Utilization Result Form*

## Results and discussion

- Cmax obtained using standard dispatching rule 310 mins
- Cmax obtained using SIMANNEAL algorithm 284 mins.
- % of improvement in Cmax 8%
- Tardiness obtained using standard dispatching rule 1689 mins
- Tardiness obtained SIMANNEAL algorithm 1649 mins
- % of improvement in Tardiness 3%
- Machine utilization obtained using standard dispatching rule 86.34%
- Machine utilization SIMANNEAL algorithm 95.06%

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

- % of improvement in Machine utilization 10%

Compared to standard despatch rules Cmax is minimum, tardiness is minimum , machine utilisation is improved and number of Tardiness job is reduced and number of Earliness jobs is increased in Simulated Annealing process. So simulated Annealing Technique is suitable for all type of industrial job shop parallel machine scheduling problems.

**Makespan comparison**

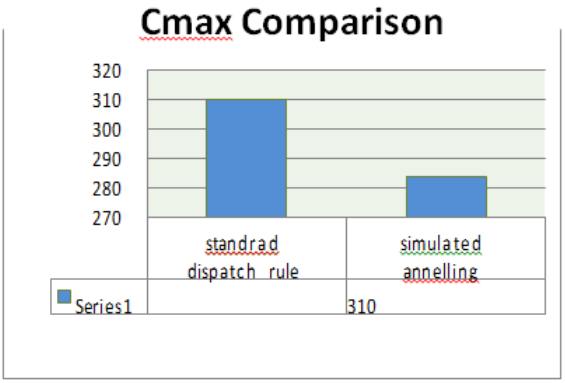


Fig 9 Makespan comparison chart

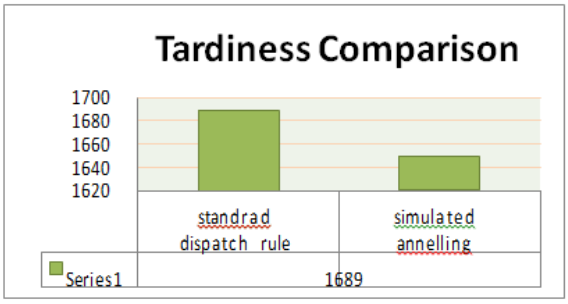


**Fig 10 Tardiness comparison**

**Machine utilization comparison**

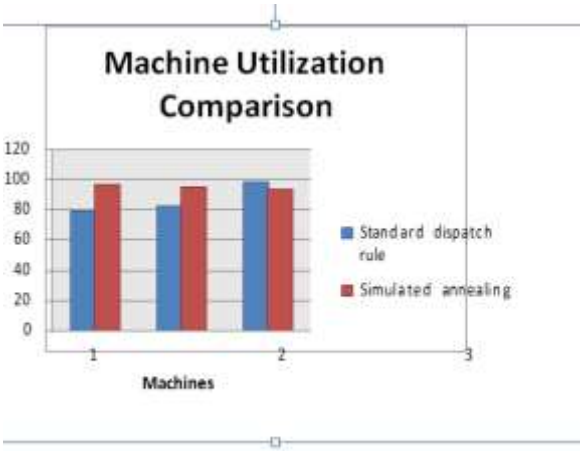


**Fig 11 Machine Utilization Comparison Chart**

## Conclusion

Thus it is concluded from the results obtained that SA proves to be a better heuristic search algorithm to solve complex job shop scheduling problem. It had been proved through different case analysis that Simulated Annealing is one of the near best algorithms for solving combinatorial optimization problems. SA produces quality results with a low degree of performance variance. Carried out experimental results show that the developed software program obtains very promising results for solving small sized benchmark problems and some large randomly generated problems. The results show that SA can be used to resolve industrial scheduling problems, considering variation in input data from time to time. With the idea that simulated annealing has a uniform probability distribution over global optima and its better usage for multi-objective problems, it can be used to resolve Multi- objective shop floor problems.

## Scope for further work

Due to the fact that there is no general SA algorithm that works well for all problems out there, there is still much to be done in terms of research about SA. The algorithm must be modified for further different classes of scheduling problems.

Further we can develop the algorithm for dynamic scheduling problems and problems of following nature,

- Scheduling problems with precedence constraints and job weight
- Preemptive scheduling problems.
- Industrial problems with machine break down conditions.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 2.114

- Comparison with other Standard rules and other heuristic techniques.
- Bunch marking with high number of job cases.

## References


1. Joseph Y-T. Leung. (2000), „Handbook of scheduling. Algorithms, Models And Performance Analysis," CRC Press Company, part.( I-II).
2. Peter Brucker, (2006), „Scheduling Algorithms," Springer-Verlag Berlin. Part 5 (107-155).
3. Dr. Jacek Blazewicz, Dr. Gunter Schmidt, Dr. Klaus H. Ecker, Dr. Erwin Pesch., (2007) Handbook on Scheduling From Theory to Applications,' Springer Berlin Heidelberg New York.part 10.
4. Vincent Tkindt Jean-Charles Billaut. (2002), 'Multicriteria Scheduling Theory Models and Algorithms,' Springer Berlin Heidelberg New York.part (8,9)
5. Kenneth R. Baker, Dan Trietsch, (2009), „principles of sequencing and scheduling,' A John Wiley & Sons, Inc. Publication
6. Pierre Lopez, Francois Roubellat., (2008), „Production scheduling," A John Wiley & Sons, Inc. Publication.part(9,10)
7. Michael L. Pinedo., (2002), „scheduling Theory, Algorithms, and Systems ," Original edition published by Prentice Hall. part(II).
8. Maciej Drozdowski., (1983), „Scheduling for Parallel Processing, Springer Dordrecht Heidelberg London . part(5) http://citeseer.ist.psu.edu/kirkpatrick83optimization.html.
9. S.French,(1984), „sequencing and scheduling For Job-Shop ,' International eills horwood series . chapter 4.
10. Stanisław Gawiejnowicz. Wilfried Brauer, Dr. Arto Salomaa (2008), „time-dependent scheduling," Springer-Verlag Berlin Heidelberg, chapter 7.
11. Kirkpatrick S., Gelatt C. D., and Vecchi M. P., (1983), „Optimization by Simulated Annealing," Science, Vol. 220, No 4598, pp. 671-680
12. Graham, R.E., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. (1979), 'Optimization and approximation in deterministic sequencing and scheduling: a survey,' Ann.Discrete Math. Vol. 4, pp.287-326..
13. Kalai.A. T., Santhosh Vempala, (2006), „Simulated Annealing for Convex Optimization," Math of operations research, vol.12, pp. 656-673.
14. Marc Sevaux., (2011), „Heuristics and metaheuristics for a parallel machine scheduling problem: a computational evaluation," University of Valenciennes, France.
15. Jun-Gyu Kim, (2006),"Common Due-Date Assignment and Scheduling on ParallelMachines with Sequence-Dependent Setup Times,' 9th Asia Pasific Industrial Engineering & Management Systems Conference,korea.
16. Norman M. Sadeh and Yoichiro Nakakuk, (1994), „Focused Simulated Annealing Search: An Application To Job-Shop Scheduling", Camegie Mellon University
17. T.C.E. Cheng., (1999), Due-date assignment and parallel-machine scheduling with deteriorating jobs, 2Department of Mathematics, Shanghai University, Shanghai,.
18. Alexandre S. Mendes. (1995), Comparing meta-heuristic approaches for parallel machine scheduling problems," Universidade Federal de Santa Maria Journal of Operational Research..
19. Metropolis N., Rosenbluth A. W.Rosenbloth M. N., Teller A. H., and. Teller E, (1953), Equation of State Calculation by Fast Computing Machines,' Journal of Chemical Physics, vol.21, pp.1087-1092
20. Dong-Won Kima(2002), "Unrelated parallel machine scheduling with setup times using simulated annealing" Robotics and Computer Integrated Manufacturing PP 223-231
21. Kanate Ploydanai (2010), "Algorithm for Solving Job Shop Scheduling Problem Based on machine availability constraint"(IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 05, 2010, PP 1919-1925.