International Journal of Advance Industrial Engineering
E-ISSN 2320 –5539
Available at http://inpressco.com/category/ijaie/

Research Article

Machine Utilization Technique for Job Shop Scheduling using Tabu Search Algorithm

Md. Hasan Ali*†, Md. Rashedul Haque‡ and Md. Fashiar Rahman§

†Department of Industrial and Production Engineering, Bangladesh Army University of Science and Technology, Saidpur, Bangladesh
‡Graduate School of Industrial Engineering & Management, Khulna University of Engineering & Technology, Khulna, Bangladesh
§Department of Industrial Engineering & Management, Khulna University of Engineering & Technology, Khulna, Bangladesh

Received 12 July 2018, Accepted 15 Sept 2018, Available online 20 Sept 2018, **Vol.6, No.3 (Sept 2018)**

Abstract

Efficiency in job shop scheduling plays an important role where a large number of jobs and machines are needed to be scheduled. As the number of jobs and machines increase, the job shop scheduling problems approach non-deterministic polynomial (NP) hard difficulty. It is one of the hardest combinatorial optimization problems for which it is extremely difficult to find optimal solutions. In the last two decades, many heuristic methods have been devised that produce solutions which are near optimal. Here, we present one such heuristic technique called Tabu search algorithm that can be applied to solve Job shop scheduling (JSS) problems. The purpose of this study is to generate different schedules with corresponding completion time using the Tabu search algorithm. The code for tabu search algorithm was done in python language. With the help of Gantt chart, it has found the percentage of machine utilization and idleness for all machines. At last, future directions have been provided for further improvement.

Keywords: *Job Shop Scheduling, Machine Utilization, Neighborhood, Tabu Search, Tabu List, Total Completion Time.*

1. Introduction

A schedule is a basic time-management tool that consists of a list of times at which possible tasks, events, or actions are required to take place, or of a sequence of events in the chronological order in which such things are required to take place. It relates to the use of equipment, facilities, and human activities. Manufacturers must schedule production, which means developing schedules for workers, equipment, purchases, maintenance, and so on. A job shop is a type of manufacturing process in which small amount of customized products are produced, require a variety set-up, and sequencing of process steps. In a shop floor, machines process jobs, and each job contains a certain number of operations. Each operation has its own processing time, and has to be processed on a specific machine. Each job has its own machine order, and no relation exists between machine orders of any two jobs. For each job, the machine order of processing operations is prescribed, and is known as technological production recipe or technological constraints, which are static to a problem instance. Minimum makespan feasible schedules are obtained by computing the processing order of operations on machines without violating the technological constraints. Glover (1993) recognized the availability of difficult optimization problems encountered in practical settings (e.g. telecommunications, logistics, financial planning, transportation and production), and stated it as the primary motivation of developing a proliferation of optimization techniques. The objectives of this problem is to find the optimal schedule by minimizing the makespan. In recent years, many heuristic techniques such as Genetic algorithm, Tabu search, Ant colony optimization, and Simulated Annealing were employed to solve this type of NP hard optimization problems. Tabu search is a general heuristic procedure for guiding search to obtain good solutions in complex solution spaces. One of the main components of the TS is its use of flexible memory which plays an essential role in the search process. Discoveries of more effective ways to exploit this memory, and of more effective ways to apply it to special problem settings, provide one of the key research thrusts of the discipline, and account for its growing success in treating hard problems. In most cases tabu search applications are designed specifically for their intended problems, with little or no consideration for design and codes reuse.

2. Literature review

First local search algorithms were tailored for the job shop problem in late 1980's, many different

*Corresponding author's ORCID ID: 0000-0003-3300-1977
DOI: https://doi.org/10.14741/ijaie/v.6.3.1

approaches have been developed. Laarhoven *et al.,* (1992) introduced the first simulated annealing algorithm for the job shop problem. From that, there has been considerable improvement, and an algorithm of computational analysis of JSSP using tabu search by Aarts *et al.*, (1994) is now the standard bearer of the area in terms of mean percentage error from the optimal. Buscher and Shen (2009) worked on lot streaming problem in job shop environment. In this lot streaming problem, they achieve sustainable makespan reduction. Solimanpur and Elmi (2013) investigated an approach for tabu search in cellular manufacturing system. They proposed a model for the attempted cell scheduling means bottleneck machines to solve the problem heuristically. Zhang *et al.,* (2007) proposed a tabu search algorithm with new neighborhood structures and move evaluation strategies. They have found a large number of better upper bounds among the unsolved instances with this new neighborhood tabu search algorithm. Their computational results show that for a rectangular problem new tabu search provide the solution that is improved both in solution quality and performance. Christoph (2006) presented a fast tabu search algorithm for the no wait job shop scheduling.

The fast tabu search algorithm was extensively tested on no wait job shop scheduling and compares extremely better solutions than other approaches. Ponnambalam *et al.,* (2000) considered tabu search algorithm for job shop scheduling and this consideration was performance measure in makespan time. The results were provided by this tabu search algorithm compared with simulated annealing and genetic algorithms. The minimization of makespan time is comparable with simulated annealing and genetic algorithms. Udaiyakumar and Chandrasekaran (2013) defiance the job shop scheduling problem with total makespan minimization using tabu search algorithm. They have developed tabu search not only for single objective but also for multi objective and multi constrained JSSP problem. Peng *et al.,* (2015) incorporated a tabu search algorithm with path relinking to generate solutions for JSSP. They have tested that tabu search with path relinking is able to improve the upper bounds for 49 out of the 205 tested instances and it solves a challenging instance that has remained unsolved for over 20 years. González *et al.,* (2013) investigated lateness minimization using tabu search algorithm with sequence dependent set up times for job shop scheduling problem. They performed an experiment conventional benchmarks with 960 and 17 instances respectively. The results demonstrate that the proposed tabu search algorithm is superior to the state-of-the-art methods both in quality and stability. Nakandhrakumar *et al.,* (2014) minimize the makespan of job shop scheduling problem. 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. Vilcot and Billaut (2008) analysis job shop scheduling problem with multiple constriants for printing and boarding industry. Their main objective is to minimization of makespan and maximum lateness. The methods have been tested on instances derived from the literature and the results show the interest of introducing the solutions of the tabu search algorithm into the initial population of the genetic algorithm, both in terms of solutions quality and of computation time. Raza *et al.,* (2006) examined tabu search algorithm for solving economic lot scheduling problem. The computational study revealed that the performance of the tabu search algorithm is superior to the best known Dobson's heuristics and Genetic algorithm. Bajeh and Abolarinwa (2011) made a comparative study between genetic algorithm and tabu search algorithm. In this study they have found that with the stated objectives, both algorithms have effectively demonstrated the ability to solve complex optimization problems of which examination timetabling is part of. Tabu search however, produced better results than Genetic Algorithm with respect to quality and speed of generating the timetables. Wang *et al.,* (2016) proposed a neighborhood expansion tabu search algorithm based on genetic factors (NETS) to solve traveling salesman problem.

3. Tabu search algorithm

Tabu search is a local search algorithm that requires a starting solution. After obtained a feasible starting solution neighborhood structure required to be generated according to the objective function. The procedure begins with an initial solution and stored the initial solution as the current seed, and the best solution. The neighbors are produced of the current seed by a neighbor structure. The neighbors are produced of current seed are called candidate solutions. They are produced for an objective function and a neighbors of candidate solutions which is not tabu selected as a new seed solution. This selection is called move, and add to the tabu list in order to create a memory. The new seed solution is compared with the current seed solution, if the new seed solution is better than current seed solution stored it as the new best solution. According to this process iterations are continued until termination criteria is satisfied. Essential elements of tabu search algorithms are as follows:

3.1 Initial solution

A feasible initial solution is obtained by selecting from any one priority dispatching solutions. Initial solution affects the scheduling solution quality of the final output of the job shop scheduling problem.

3.2 Neighborhood Structure

A neighborhood structure is a mechanism, which contain new set of neighbor solutions by applying a simple modification to give initial solutions. Each neighbor solution is reached from a given solution by the 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.

3.3 Move

The best neighbor, which is not tabu is selected as a new seed solution. The best neighbors are one whose objective function C_{max} is minimum, means whose total completion time is minimized. If the entire neighborhood is tabu then the oldest neighbor entering the tabu list at first, is selected as a new seed solution.

3.4 Tabu list

The purpose of the tabu list is to block the search from degenerating by starting to cycle between the same solutions. In a tabu list, the elements added on the list are attributive. The main purpose of using an attributive representation is to save computer memory. The attribute that represents an element in sequencing is two operations that have interchanged in a recent move. It is the arc (j,i), which is obtained by swapping a pair of processing operations (i,j). To identify the tabu status of the neighbor (j,i) it is looked whether or not it is on the tabu list; if so it is named as "tabu". Tabu list is updated after each move in so far as the strategic forgetting occurs. So in that way best solution can be found without taking the same solution taking into account.

3.5 Aspiration criterion

The aim of the aspiration criterion, when it is necessary, is to override the tabu status of a neighbor. The aspiration criterion used as follows, if the move yields a solution better than the best obtained so far, then the move is performed even as it is tabu.

3.6 Termination criterion

The search in the job shop scheduling problem could go on forever, unless the optimal value of the problem at hand is known beforehand. In practice, obviously, the search has to be stopped at some point. The most commonly used stopping criteria in TS are given as follows: (i). A fixed number of iterations, (ii). After some number of iterations without an improvement in the objective function value, (iii). When the objective function reaches a pre-specified threshold value. That is, termination procedures are generally used to stop the iteration process, if the quality of the procedures has crossed a certain value.

4. Methodology

The tabu search algorithm was adapted to solve the JSSP problem and the algorithm is depicted in figure 1. The Python version 3.5.1 was used to implement the algorithm. The detailed description of the flowchart is as follows:

4.1 Generating initial solution

There has been a great deal of research to find good, efficient heuristics for the job shop scheduling problem. Notably among these are called dispatching rule. In the manufacturing world scheduling problems are extensively implementing dispatching rules. These are constructive heuristics which examine a subset of operations and schedule these operations one at a time. While there are no guarantees on their quality, these algorithms have the advantage of running in sub-quadratic time, and producing reasonable results with any of a number of good priority rules. Priority dispatching rules were first developed in the mid 1950's, and until about 1988 were the only known techniques for solving arbitrary large (≥ 100 element) instances. While List Scheduling algorithms are no longer considered to be the state of the art for solving large job shop instances, they can still produce good initial solutions for local search algorithms. In our job shop scheduling problem, dispatching rule is required for tabu search to generate initial solution.

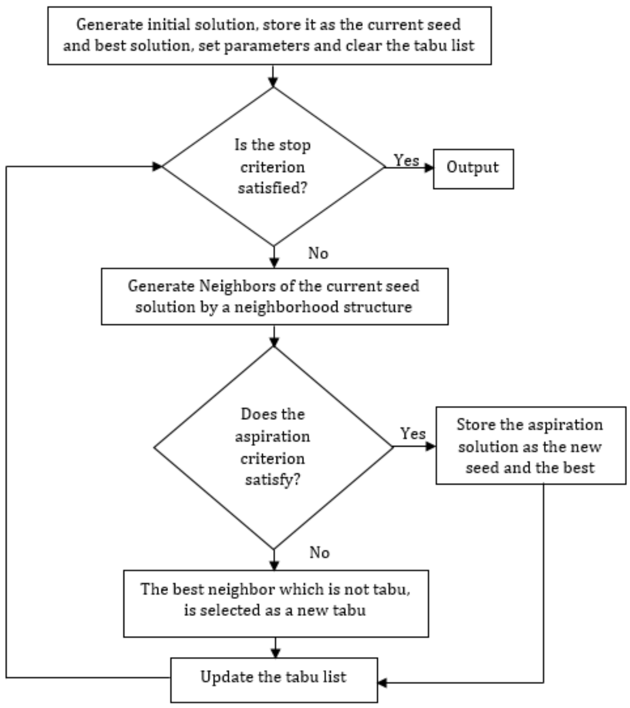

Fig.1 Flowchart representing the tabu search algorithm for solving JSSP

4.2 Exploring the neighborhood

The performance of a local search algorithm, both in terms of the quality of the solutions, and in the time required to reach them is heavily dependent on the neighborhood structure. Typically for a given solution 's' a neighborhood is a set N (s) of candidate solutions which are adjacent to us. It illustrates that if we are currently exemplifying solutions the next solution will be some s' ∈ N (s). In our JSSP problem for creating neighborhood structure from initial solution we only change the machine sequence as variable. In an iteration machine sequence are changed among a generation span, according to arcs reversed. The two main goals for designing neighborhoods for the job shop scheduling problem are feasibility and connectivity. A neighborhood with the former property ensures that, if provided a feasible solution, all neighboring solutions will be feasible as well. The latter ensures that there exists some finite sequence of moves between any feasible solution and a globally minimal solution.

4.3 Stop Criterion

As it has stated earlier the motivation for using tabu search algorithm to solve JSSP is that it can provide near optimal solution for NP hard problems within much less time that required by other similar algorithms. But the downside of tabu search algorithm is that it can enter into cycle that only improves the required solutions after certain number of iterations. This characteristic can be explored by plotting the minimum completion time in each generation against the corresponding generation number. This certain amount of iterations serves as the stop criterion for obtaining the best solution within reasonable amount of time.

4.4 Storing the best solution

In tabu search algorithm initial solution is fed to the framework to start the iterations. This initial solution such both as the best solution and the current as the start first iteration. With each iteration through exploring the neighborhood the completion time is calculated and stored in the memory. When the stop criterion is made the final best solution is found and the minimum completion time and the corresponding sequence of jobs, operations and machines are obtained. Each unique combination found in each iteration is inserted in the tabu list, and thereby updates the tabu list. This list is used to find matches between the combination currently being explored and the members of the tabu list. If there is a match, then the combination is not considered, and the next combination is explored. Thus, it ensures that the repetition is avoided, and each iteration is efficient. Tabu's are sometimes too powerful that they may prohibit attractive moves, even when there is no danger of cycling, or they may lead to an overall stagnation of the searching process. To overcome this problem, aspiration criteria are defined. The intention of the aspiration criterion is to avoid bypassing moves, which lead to substantially better solutions. The aspiration mechanism, in some cases, is used to protect the search from the possibility that in a given state, all moves are tabu. In such cases, the aspiration function is set in such a way that at least one move fulfills its criterion, and its tabu status is removed.

4.5 Clearing the tabu list

In order to ensure that the maximum solution is explored by searching for an optimal solution. After a certain number of iterations, the tabu list is cleared. As a result, as the beginning of each generation there is an empty tabu list and the intensity of a number of combinations explored is greatly increased.

5. Problem definition

The job shop scheduling problem consists of a set of jobs J = {1 . . n}, a set of machines M = {1 . . m}, where, J_i denotes ith job ($1 \le i \le n$) and M_j denotes jth machine ($1 \le j \le m$). On the machines M_1, M_2 … M_m, the jobs J_1, J_2 … J_n are to be scheduled. Let V be the set of all operations in all jobs. Each job J_i has a set of operations oi_1, oi_2…oi_k, where, k is the total number of operations in the job J_i. Operation's precedence constraints are associated with each job, and ensure that the operation will be processed only after the processing of operation according to the shortest processing time and machine availability. Any number of jobs and machines, combination can be solved by proposed Tabu search algorithm. Here, a combination of 8 jobs, and 7 machines is taken as an instance as shown in Table 1.

Table 1 Problem Instance of 8 jobs and 7 machines

Jobs	Operation	Machine	Time (minutes)
1	1	1	5
		3	3
	2	2	2
		5	1
		6	4
2	1	4	7
	2	7	3
3	1	2	9
		4	4
	2	5	8
		6	10
4	1	1	2
		4	6
		5	3
5	1	1	8
		4	7
	2	5	5
		7	4
6	1	1	7
		2	2
		4	3
7	1	1	5
		3	1
	2	1	4
	3	3	5
		4	9
8	1	5	9
		6	7
	2	1	5
		7	8

6. Results

The tabu search algorithm is coded in the python language version 3.5.1. The computing system used is HP Core i3, 4GB RAM. The aim of this study is to explore the sequence of jobs, operations and machines and finally find the optimum solution. To evaluate the functionality of the tabu search algorithm a numerical analysis has been performed in this chapter. The total completion and minimum time among generation is required for this optimum sequence is 63 minutes. For the instance of 8 jobs and 7 machines the optimum sequence of job, machine, and operation are:

113 125 214 227 314 325 411 514 527 612 713 721 733 816 821

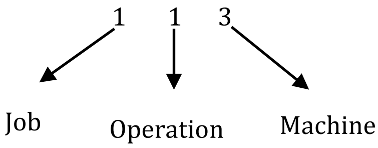

Fig.2 Examples of job, operation and machine sequence

The corresponding gantt chart is shown in figure 3. Gantt chart visualizes makespan among the machines. Each machine has its own corresponding chart to identify the machine utilization and machine idle time. From the figure 4 it is concluded that among all of the machines M2, M3, M5, M6 and M7 has idler time than machine utilization time.

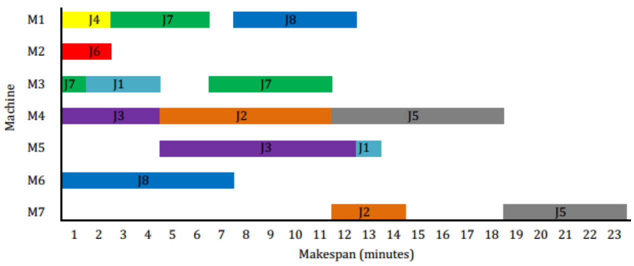

Fig.3 Gantt Chart

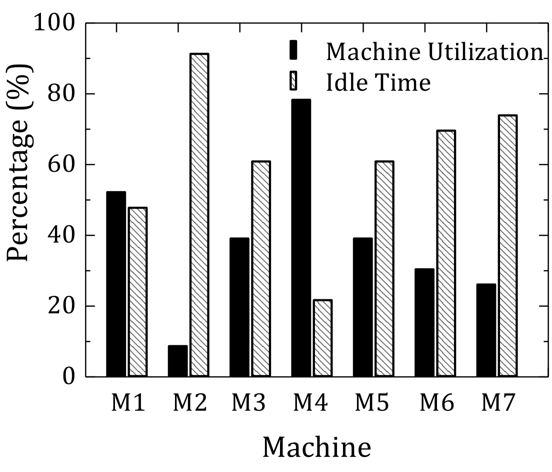

Fig.4 Comparison of machine utilization and idle time

Figure 5 represents the performance of tabu search algorithm by plotting completion time against each generation. From this graph it is seen that which generation required less time, and it is generation 14 in figure 5. Among all of the generation 14th generation time is the lowest time which is 63 minutes. All other generation has larger time than that generation.

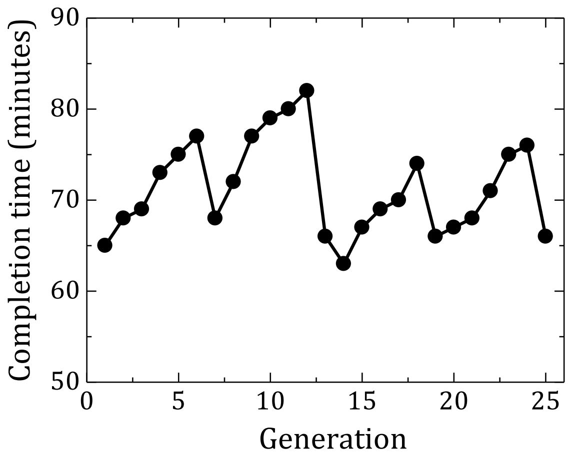

Fig.5 Completion time vs. generation for 8 jobs, and 7 machines

7. Discussion

This study concludes the scheduling process view where many jobs, machines, and operations are involved to determine the optimum sequence of scheduling. In this study, we have taken a combination of 8 jobs and 7 machines. But our coding in python language, it can be used for any combinations of Jobs and machines. From this sequence machine utilization, and scheduling generation time is easily determined. For the future scope of work from the gantt chart, it is visualized that some machine has idler time. If makespan is minimized from the gantt chart, it will be more effective scheduling for any job shop with less machine idleness.

Acknowledgement

We are grateful to Mr. Suhail Mahmud, Software Engineer of Enosis Solutions, Bangladesh for his sincere efforts in this study.

References

Glover, F., (1993), A user's guide to tabu search, *Annuals of Operations Research,* 41, 3-28.

Laarhoven, P.J.M.V., Aarts, E.H.L., Lenstra, J.K., (1992), Job Shop Scheduling by Simulated Annealing, *Operations Research*, 40, 113-125.

Aarts, E.H.L., Laarhoven, P.J.M.V., Lenstra, J.K., and Ulder, N.L.J. (1994), A computational study of local search algorithms for job shop scheduling. *ORSA Journal on Computing*, 6, 118-125.

Buscher, U., Shen, L., (2009), An integrated tabu search algorithm for the lot streaming problem in job shops, *European Journal of Operational Research*, 199, 385-399.

Solimanpur, M., Elmi, A., (2013), A tabu search approach for cell scheduling problem with makespan criterion, *Int. J. Production Economics*, 141, 639-645.

Zhang, C.Y., Li, P., Guan, Z., Rao, Y., (2007), A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem, *Computers & Operations Research*, 34, 3229-3242.

Schuster, C.J., (2006), No-wait job shop scheduling: tabu search and complexity of sub-problems, *Math. Meth. Oper. Res.*, 63, 473-491.

Ponnambalam, S.G., Aravindan, P., Rajesh, S.V., (2000), A Tabu Search Algorithm for Job Shop Scheduling, *Int J Adv Manuf Technol*, 16, 765-771.

Udaiyakumar, K.C., Chandrasekaran, M., (2013), Tabu Search Algorithm For Solving of Job Shop Scheduling With Single Objective – Makespan Minimization, *Indian Journal of Applied Research*, 3, 365-367.

Gonzalez, M.A., Vela, C.R., Rodriguez, I.G., Varela, R., (2013), Lateness Minimization with Tabu Search for Job Shop Scheduling Problem with Sequence Dependent Setup Times, *Journal of Intelligent Manufacturing*, 24, 741-754.

Nakandhrakumar, R.S., Seralathan, S., Azarudeen, A., Narendran, V., (2014), Optimization of Job Shop Scheduling Problem using Tabu Search Optimization Technique, *International Journal of Innovative Research in Science, Engineering and Technology*, 3, 1241-1244.

Peng, B., Lu, Z., Cheng, T.C.E., (2015), A tabu search/path relinking algorithm to solve the job shop scheduling problem, *Computers & Operations Research*, 53, 154-164.

Vilcot, G., Billaut, J.C., (2008), A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem, *European Journal of Operational Research*, 190, 398-411.

Bajeh, A.O., Abolarinwa, K.O., (2011), Optimization: A Comparative Study of Genetic and Tabu Search Algorithms, *International Journal of Computer Applications*, 31, 43-48.

Raza, S.A., Akgunduz, A., Chen, M.Y., (2006), A tabu search algorithm for solving economic lot scheduling problem*, J Heuristics*, 12, 413-426.

Wang, D., Xiong, H., Fang, D., (2016), A neighborhood Expansion Tabu Search Algorithm Based On Genetic Factors, *Open Journal of Social Sciences*, 4, 303-308.