Journal of Fundamental and Applied Sciences

ISSN 1112-9867

Available online at http://www.jfas.info

Research Article

Special Issue

DISJUNCTIVE PROGRAMMING-TABU SEARCH (DP-TS) APPROACH FOR JOB SHOP SCHEDULING PROBLEM

K. L. Wong* and S. Z. Nordin

Universiti Teknologi Malaysia, Skudai, Johor, Malaysia

Published online: 7 March 2018

ABSTRACT

Scheduling, in short, is the allocation of various resources with the aim to arrange and control to find an optimization in the work process. Appearance of job shop scheduling problem brings a big impact to the manufacturing sector since this method affect the production process and profit of the industry. To improve the efficiency of the manufacturing process as well as minimizing the cost, it is vital to apply job shop scheduling problem in this sector. One of the metaheuristic methods, tabu search has been selected to apply to the job shop scheduling problem. The initial solution is gained by disjunctive programming using Lingo software and the neighborhood structure is based on the swapping of the jobs in critical paths. The final result with the best makespan will be obtained in this paper.

Keywords: job shop scheduling; tabu search; disjunctive graph.

Author Correspondence, e-mail: wkl0718@hotmail.com

doi: http://dx.doi.org/10.4314/jfas.v10i6s.29

1. INTRODUCTION

At the present time, due to the vast development in the manufacturing sector, the competitiveness within each industry becomes tense. To ensure the large volume of production, the industries will have many machines to process the jobs. However, each job may take a long time to complete without proper scheduling on these machines. If this situation remains unchanged, the problems such as long idle and waiting time and long process route may occur and these will directly affect the profits of the industries. Long idle and waiting time will affect the productivity too as more product could be produced if there is a short idle time. Different job may have different flow pattern throughout the machine which means that the production process and time may be long or short. To be able to compete with others, the industries keep on searching for a good solution to solve the scheduling problem.

Appearance of job shop scheduling problem brings a big impact to the manufacturing sector since this method affect the production process and profit of the industry. To improve the efficiency of the manufacturing process as well as minimizing the cost, it is vital to apply job shop scheduling problem in this sector.

Research has been done on job shop scheduling a few decades ago. This type of scheduling defines the sequence of jobs and operation to optimize the production process. Exact methods are the earliest method that used to solve JSSP. Researchers found this method was good in solving the small size problem and obtained the optimal solution. However, it is difficult to use this method to solve a larger problem and it takes time as well. Branch and Bound algorithm is the common exact method used in solving JSSP. However, in [1] claimed that for the problem size larger than 10 x 10, effort still need to be taken to obtain a better result. Later, in [2] again mentioned that this method cannot be used to solve large size problem. He believed that approximation method especially tabu search was good in solving JSSP.

Many researchers had turn to approximation method, which involves heuristics and metaheuristics methods in solving JSSP after they realized that it was impossible to solve this kind of NP-hard problem in polynomial time. Dispatchng rules are common heuristic method used for scheduling in the industries. It consists of a number of priority rules which allocate the jobs order to be completed at each work centre. Most of the previous researches claimed that SPT performed better in sequencing. However, experiment done by [3] denied this statement. They discovered that there were no single dispatching rules that can give optimal results so they measured the performance of the dispatching rules using ranking list. They found that SPT did perform well in some circumstances but overall MWTR gave a better performance in most of the criterions. It even performed better than the hybrid rules.

Shifting bottleneck heuristics are another heuristics method which focuses on the bottleneck machine with maximum lateness. The path on the bottleneck machine will be arranged first and a new bottleneck machine will be determined in each iteration. In [4] came out with the idea that the subproblems of the job shop scheduling were independent of each other. Based on this idea, they discovered a parallel implementation method of shifting bottleneck heuristic. Their results showed that the parallel implementation was better than sequencial implementation of shifting botlleneck heuristic. In [5] solved job shop scheduling problem that using parallel machine by applying shifting bottleneck heuristic. The heuristic was later tested with some priority rules. It was proven that the shifting bottleneck heuristic was able to solve large scale of problem, that was, more than fifty jobs.

Genetic Algorithm (GA) is one of the common metaheuristics method that based on the ideas of natural selection and genetics that mimics the biological evolution. In [6] first proposed this method before it was been widely used by the researchers. In [7] applied an improved genetic algorithm that based on the opposition-based learning in flexible job shop scheduling problem. They even made comparison of their algorithm with other genetic algorithm proposed by previous researchers and they found their results are almost the same, which means that their algorithm was effective in solving the scheduling problem.

Based on [8], annealing is a thermal process by which the solid in a heat bath is heated up and then cooled down to reach thermal equilibrium while simulated annealing is a method that imitates the annealing of solids to solve large combinatorial optimization problems. Both simulated annealing and tabu search methods are local search techniques that solve the problems with the aid of neighborhood finding in the provided solution. However, simulated annealing is possibly to search back to the old solutions causing an oscillation in local optimum surrounding which is also time consuming. Hence, tabu search with tabu list will prevent this kind of problems.

Tabu search is used in this paper and the literature review of this method is shown in the next section.

2. JOB SHOP SCHEDULING

Job shop scheduling belongs to the class of NP-hard problem and even the one of the ‘harder’ problem in this class, claimed by [9]. NP-hard problem stands for decision problem that are included in NP-complete problems' class. For example, you are asked to find the shortest make span of the fabrication process in automobile manufacturing, it is considered NP-hard since it is not easy to find a way to determine if the method used can obtain the shortest make span.

One of the earliest researches that can be found on job shop scheduling was done by [10] who applied a simple decision rule on both two-stage and three-stage production schedule. In two-steps process, if the lowest operation time appeared in step one, the job was allocated at the first position in the sequence. If it was appeared at step two, the job was assigned at the last position in the sequence. He also found that the optimal ordering was different for each stage for the stages greater than three.

In [11] defined job shop as a shop comprised machines that performed jobs in which each job associated with a routing in an ordered sequence of operations. In [12] also has a similar description to job shop. He described job shop as a shop that involved small batch production and the scheduling was determined by the routing sheet.

A standard job shop scheduling problem consists of a set of n jobs, $J = (J1,\ldots,Jn)$ and a set of m machines, $M = (M1,\ldots,Mm)$. Each of the job comprises a set of mj operations $O1j,\ldots, Omjj$ and the operations are associated with a duration of processing time Pij. The processing of two jobs simultaneously on a machine at any time is strictly prohibited. This helps to find the operating schedule that minimize particular objective function. In this paper, the objective function is to minimize makespan.

There are two ways to represent the schedule graphically. The first one is disjunctive graph. A disjunctive graph helps to decide the suitable order of operations in a job shop scheduling so that optimal results are achieved. The conjunctive arc between each node represents the same job that undergoes the consecutive operations, whereas the disjunctive arc represents the connection between the tasks that share the same machine. After the order for the operations have been decided, the undirected disjunctive arc will be turned into the directed arc. Meanwhile, Gantt chart is used to demonstrate the connection between the job operations which is useful to analyse the make span and total completion time of the manufacturing process. In [13] mentioned that Gantt chart was originated by Henry L. Gantt. He also stated that Gantt chart allowed a smooth flow of the job orders throughout the manufacturing process and instead of algorithmic solution, it concentrated on systematic. An illustrative example is given for both disjunctive graph and Gantt chart in Fig. 1 and Fig. 2.

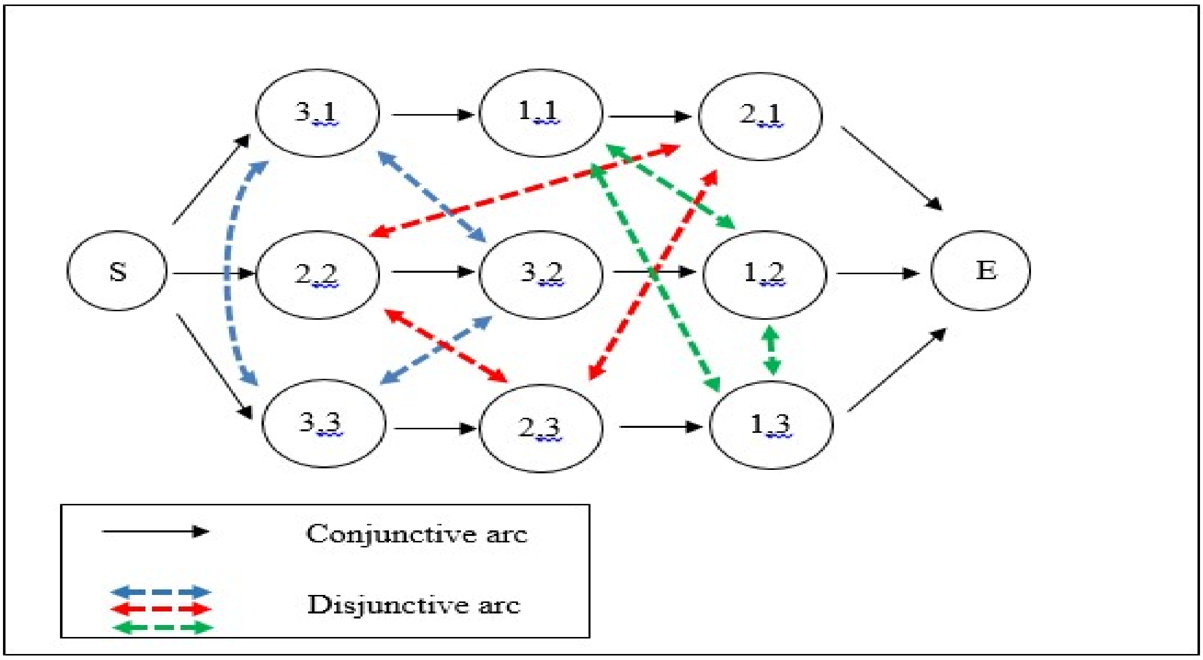

Fig.1. Disjunctive graph

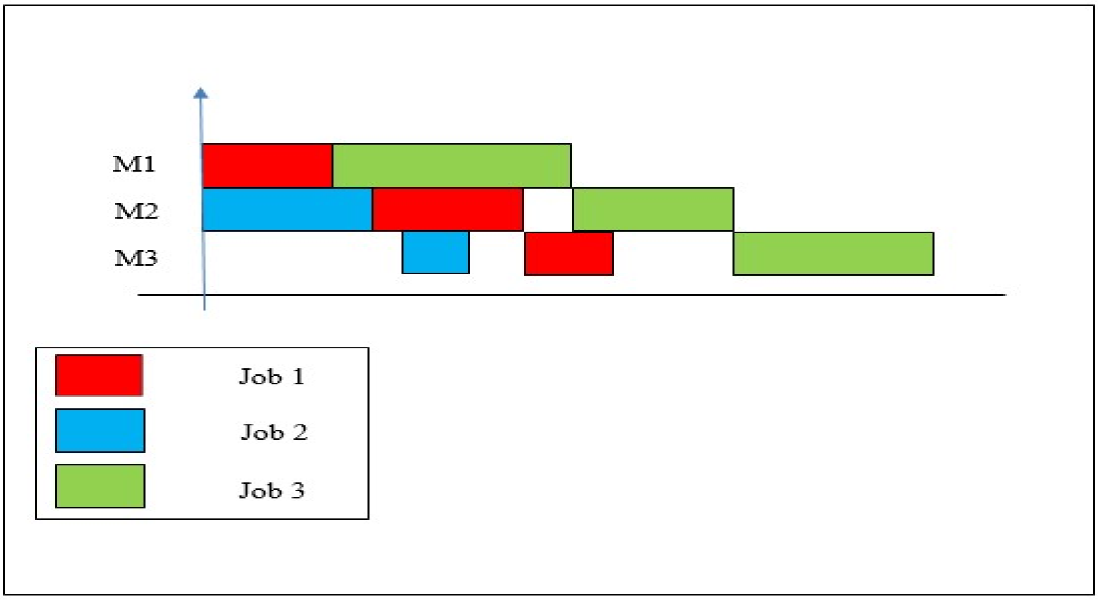

Fig.2. Gantt chart

3. TABU SEARCH

Tabu search is one of the methods that use local search technique to solve the combinatorial optimization problem. However, there is always a possibility to revisit the previous solution and trap in the local optimum. The term tabu gives a meaning of prohibitions or bans. The tabu list generates by tabu search will definitely help to reduce this kind of situation. If a potential solution is visited, it will be stored in the list and it is forbidden to get back to them. This method was discovered by [14]. Since then, numerous researches have been done on this method and most of them show a remarkably result towards the problems.

In [15] obtained initial solution by comparing 14 sets of dispatching rules and the smallest makespan was chosen to be proceeded for tabu search. The tabu search method was briefly described in this paper and an analysis was made by comparing the computational results with some common benchmark problem.

In [16] focused on the job shop scheduling problem in flexible manufacturing system. They came out with a problem, which considered the system with ready tools and they named the problem as job shop scheduling problem with multi-purpose machines. Two types of neighbourhood structure, N1 and N2, were used in this paper and the results of tabu search gave excellent results for solving most of the benchmark problems.

In [17] chose makespan minimization as an objective function and ran the tabu search for 100 iterations. They generated neighbourhood by using adjacent pairwise interchange method. 25 problems were used for the performance evaluation on tabu search method, genetic algorithm as well as simulated annealing. Among all, tabu search methods were reported that had a better performance in 6 problems. For the rest of the problems, tabu search also had comparable results with those of GA and SA.

In [18] clearly outlined the strategies and parameters in tabu search. They gave explanation for these elements which comprised initial solution, neighbourhood structure, move, tabu list, aspiration criterion and termination criterion. The initial solution was generated using Gliffer and Thompson algorithm and SPT as its tie-break rule. Tabu search with 6 different neighbourhood structures 14 different legth of tabu list were tested in this paper.

In [19] constructed a tabu search algorithm with sophisticated neighborhood structure to solve job shop scheduling problem related to the sequence dependent setup time. They compared their computational results with the results obtained by using simulated annealing. The final results showed that tabu search performed better than simulated annealing.

A general tabu search algorithm is listed below as referred to [18]:

Step 1. Generate initial solution. Store the current makespan and current schedule as best makespan and best schedule.

Step 2. Select neighbour and move.

- Choose the neighbour from the critical path that is not tabu and move it as current new seed solution.
- Update tabu list.
- Store the selected neighbour as the new best solution provided if it gives a better solution.

Step 3. Repeat step 2 until a stopping criterion is achieved.

A brief description for the elements involved in tabu search also tabulated in Table 1.

Table 1. Elements of tabu search

Elements	Brief Description
Initial solution	The initial solution can be gained by various methods ranging from dispatching rules, shifting bottleneck heuristic and random method. A good initial solution will generate a quality solution.
Neighborhood structure	It is obtained by the pair wise exchange of the operations. The feasible schedule will remain feasible since the neighbor from a feasible solution will always give feasible solution.
Move	Move refers to the swapping of operations in neighborhood. The move that gives a better solution and does not include in the tabu list is selected for each iterations.
Tabu list	Tabu list helps to avoid the local optima. It stores the pair that is prohibited to be exchanged.
Aspiration criterion	Given a condition that if the solution is in the tabu list but performed better than the current best-known solution, aspiration criterion allows the move which is tabu to be performed.
Termination criterion	Termination criterion can be the number of iterations or tabu list length. Once the termination criterion is met, the search is stopped.

4. IMPLEMENTATION OF TABU SEARCH ALGORITHM

The initial solution is first obtained before apply tabu search method. In this paper, the initial solution is gained by disjunctive programming using Lingo software. The disjunctive programming formulation of JSSP is used based on [20]. Set N represents the set of all operations (i, j), O_{ij}, while set A is the set of constraints that the job j is necessary to processed on machine i before it proceeds to machine k. In addition, y_{ij} is the variable that represents start time of O_{ij} and p_{ij} is the processing time of the operation.

$$\text{Min } C_{max}$$

subject to

$$y_{kj} - y_{ij} \geq p_{ij} \qquad \forall (i, j) \rightarrow (k, j) \in A$$

$$C_{max} - y_{ij} \geq p_{ij} \qquad \forall (i, j) \in N$$

$$y_{ij} - y_{il} \geq p_{il} \quad \text{or } y_{il} - y_{ij} \geq p_{ij} \qquad \forall(i, l), \forall(i, j), \quad i = 1,\ldots, m$$

$$y_{ij} \geq 0 \qquad \forall(i, j) \in N$$

In this model, the objective function is to minimize makespan. Constraint 1 implies the precedence relations of the operations with the same job. Constraint 2 defines C_{max} as total time duration for all jobs. Constraint 3 will help in choosing the disjunctive arc and the last constraint denotes a non-negative start time.

The sample data used is presented in Table 2 below such that $M1$, $M2$,…refer to the machines and P_{11}, P_{21} refer to the processing time.

Table 2. A 4x3 JSP data

Job	**Operating Sequences**				**Processing Time**
1	M1	M2	M3		$P_{11} = 10$, $P_{21} = 8$ $P_{31} = 4$
2	M2	M1	M4	M3	$P_{22} = 8$, $P_{12} = 3$, $P_{42} = 5$, $P_{33} = 6$
3	M1	M2	M4		$P_{13} = 4$, $P_{23} = 7$, $P_{43} = 3$

The initial solution obtained from Lingo and its resulting schedule is also shown below.

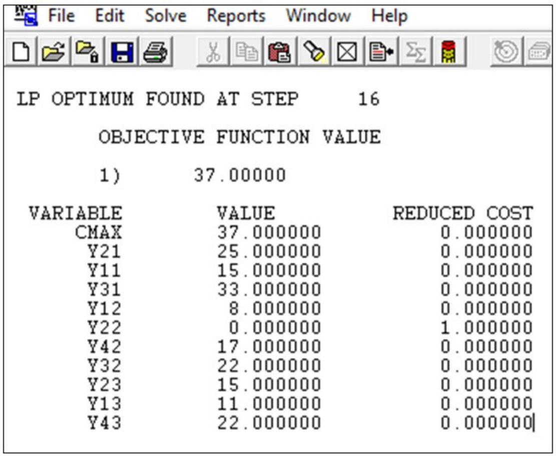

Fig.3. Initial solution from Lingo

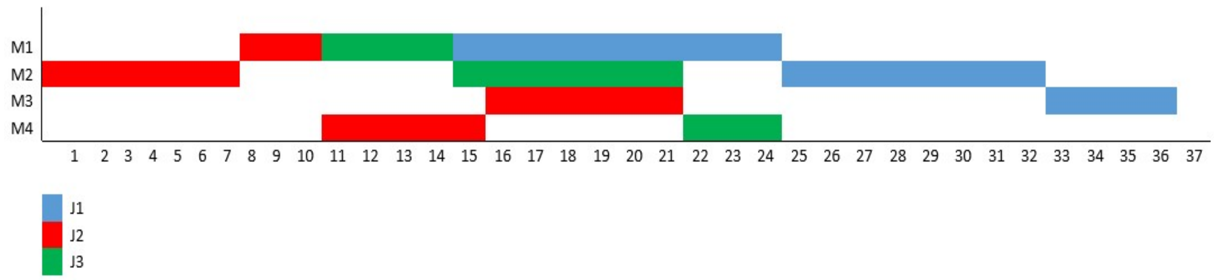

Fig.4. Gantt chart of initial solution

After that, determine the earliest start time and latest start time for each operation and calculate the slack time. Operation with slack time of zero is in the critical path. The critical path gained from the above result is $\{O_{22} \rightarrow O_{12} \rightarrow O_{13} \rightarrow O_{11} \rightarrow O_{21} \rightarrow O_{31}\}$. Neighbor is selected from the critical path and the neighbor with a better solution will be chosen as move. From initial solution,

$M_1 = \{O_{12} \rightarrow O_{13} \rightarrow O_{11}\}$

$M_2 = \{O_{22} \rightarrow O_{23} \rightarrow O_{21}\}$

$M_3 = \{O_{32} \rightarrow O_{31}\}$

$M_4 = \{O_{42} \rightarrow O_{43}\}$

$C_{max} = 37$

Critical path = $\{O_{22} \rightarrow O_{12} \rightarrow O_{13} \rightarrow O_{11} \rightarrow O_{21} \rightarrow O_{31}\}$

From the critical path of the initial solution, the neighbors are $N_1 = (O_{13} \rightarrow O_{12})$ and $N_2 = (O_{11} \rightarrow O_{13})$. A new neighbourhood will be obtained from the move of both neighbours.

For N_1, the new operation sequences in each machine are:

$M_1 = \{O_{13} \rightarrow O_{12} \rightarrow O_{11}\}$

$M_2 = \{O_{22} \rightarrow O_{23} \rightarrow O_{21}\}$

$M_3 = \{O_{32} \rightarrow O_{31}\}$

$M_4 = \{O_{42} \rightarrow O_{43}\}$

$C_{max} = 33$

Critical path = $\{O_{22} \rightarrow O_{12} \rightarrow O_{11} \rightarrow O_{21} \rightarrow O_{31}\}$

For N_2, the new operation sequences in each machine are:

$M_1 = \{O_{12} \rightarrow O_{11} \rightarrow O_{13}\}$

$M_2 = \{O_{22} \rightarrow O_{21} \rightarrow O_{23}\}$

$M_3 = \{O_{32} \rightarrow O_{31}\}$

$M_4 = \{O_{42} \rightarrow O_{43}\}$

$C_{max} = 39$

Critical path = $\{O_{22} \rightarrow O_{12} \rightarrow O_{11} \rightarrow O_{21} \rightarrow O_{23} \rightarrow O_{43}\}$

N_1 gives a better C_{max}. Thus, N_1 will be the selected move as the objective function is to minimize makespan and $(O_{13} \rightarrow O_{12})$ is added into the tabu list as a new seed solution. The search is continued upon meeting a termination criterion.

5. RESULTS AND DISCUSSION

As shown in Table 3, the minimum makespan gained from the 5th iteration is 28. It is the current best solution at this stage. The search may continue for more iteration, however, there is no more improvement for further search. Thus, the iteration is stopped at the 5th iteration and the minimum makespan obtained is 28. The final result is also presented in disjunctive graph and Gantt chart in Fig. 4 and Fig. 5.

Table 3. A 4x3 JSP data

No. of Iteration	Neighbors	C_{max}	Move	Tabu List
1	$(O_{13}\rightarrow O_{12})$	33*	$(O_{13}\rightarrow O_{12})$	$(O_{12}\rightarrow O_{13})$
	$(O_{11}\rightarrow O_{13})$	39		
2	$(O_{11}\rightarrow O_{12})$	33*	$(O_{11}\rightarrow O_{12})$	$(O_{12}\rightarrow O_{11})$, $(O_{12}\rightarrow O_{13})$
3	$(O_{23}\rightarrow O_{22})$	37	$(O_{42}\rightarrow O_{43})$	$(O_{43}\rightarrow O_{42})$, $(O_{12}\rightarrow O_{11})$, $(O_{12}\rightarrow O_{13})$
	$(O_{42}\rightarrow O_{43})$	32*		
	$(O_{31}\rightarrow O_{32})$	33		
4	$(O_{11}\rightarrow O_{13})$	28*	$(O_{11}\rightarrow O_{13})$	$(O_{13}\rightarrow O_{11})$, $(O_{43}\rightarrow O_{42})$, $(O_{12}\rightarrow O_{11})$,
	$(O_{12}\rightarrow O_{11})$	33^{+}		$(O_{12}\rightarrow O_{13})$
	$(O_{31}\rightarrow O_{32})$	33		
5	$(O_{13}\rightarrow O_{11})$	33^{+}	$(O_{12}\rightarrow O_{13})$	$(O_{13}\rightarrow O_{12})$, $(O_{13}\rightarrow O_{11})$, $(O_{43}\rightarrow O_{42})$,
	$(O_{12}\rightarrow O_{13})$	28^{+}*		$(O_{12}\rightarrow O_{11})$, $(O_{12}\rightarrow O_{13})$

* refer to selected move

$^{+}$ refer to the move that is tabu

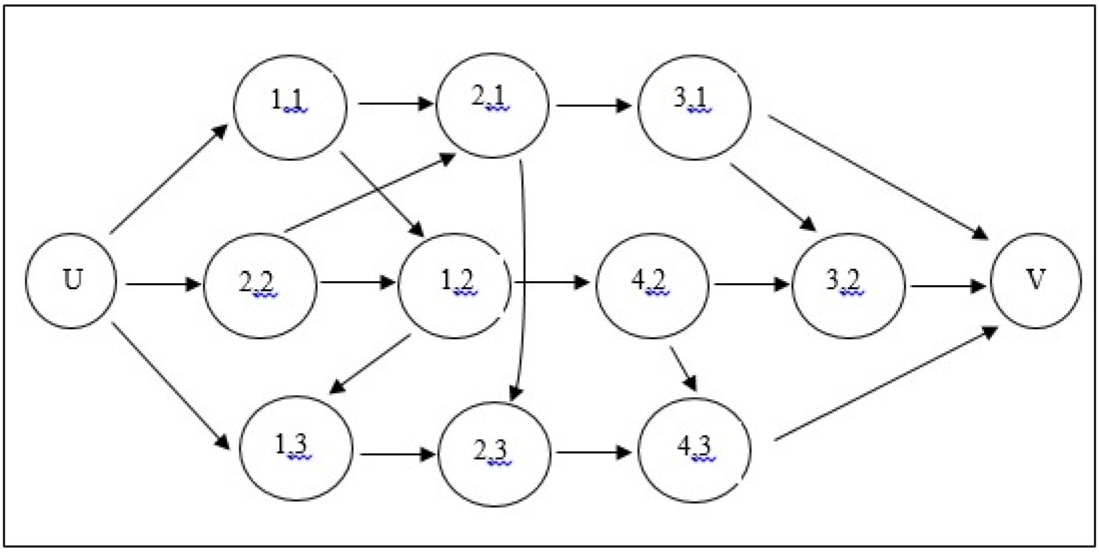

Fig.4. Disjuntive graph of optimal solution

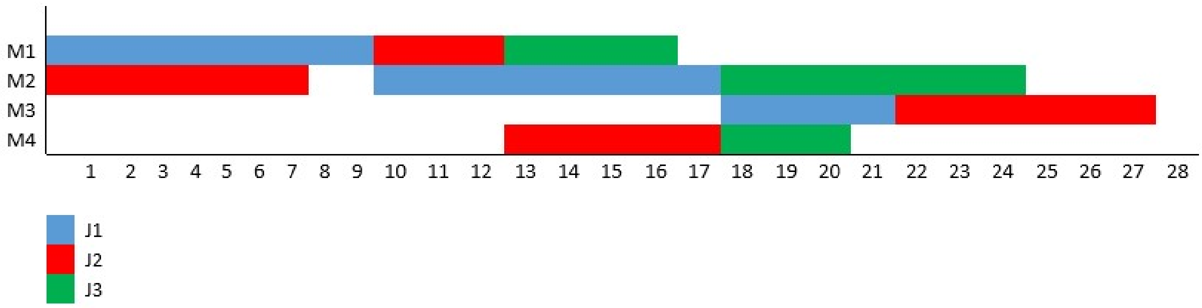

Fig.5. Gantt chart of optimal solution

6. CONCLUSION

Tabu search method has been proposed and implemented in this paper. There are some important elements include in the tabu search such as initial solution, neighbourhood structure, tabu list, move, aspiration criterion and termination criterion. N1 neighbourhood is applied which involved swapping of operation in the critical path. The result shows that tabu search is efficient in finding the optimal solution.

7. ACKNOWLEDGEMENTS

The authors would like to thank especially to MOHE, UTM and FRGS (vote number 4F778) for funding the project.

8. REFERENCES

[1] Brucker P, Jurisch B, Sievers B. A branch and bound algorithm for the job-shop scheduling problem. Discrete Applied Mathematics, 1994, 49(1-3):107-127

[2] Brucker P., Brucker P. Scheduling algorithms. Berlin: Springer, 2007

[3] Kaban A K, Othman Z, Rohmah D S. Comparison of dispatching rules in job-shop scheduling problem using simulation: A case study. International Journal of Simulation Modelling, 2012, 11(3):129-140

[4] Drießel R, Hönig U, Mönch L, Schiffmann W. A parallel shifting bottleneck heuristic for scheduling complex job shops: Architecture and performance assessment. In IEEE Conference on Automation Science and Engineering, 2010, pp. 81-86

[5] Ong J O, Maulana I. Model development of shifting bottleneck heuristic to solve jobshop

scheduling problem with parallel machines. In International Conference on Engineering of Tarumanagara, 2013, pp. 1-6

[6] Holland J. H. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Massachusetts: MIT Press, 1992

[7] Huang M, Mingxu W, Xu L. An improved genetic algorithm using opposition-based learning for flexible job-shop scheduling problem. In IEEE 2nd International Conference on Cloud Computing and Internet of Things, 2016, pp. 8-15

[8] Van Laarhoven P J, Aarts E H, Lenstra J K. Job shop scheduling by simulated annealing. Operations Research, 1992, 40(1):113-125

[9] Ivers B, Yen G G. Job shop optimization through multiple independent particle swarms. In IEEE Congress on Evolutionary Computation, 2007, pp. 3361-3368

[10] Johnson S M. Optimal two-and three-stage production schedules with setup times included. Naval Research Logistics, 1954, 1(1):61-68

[11] Conway R W. An experimental investigation of priority assignment in a job shop. California: RAND Corporation, 1964

[12] Yang Y B. Methods and techniques used for job shop scheduling. Master thesis, Orlando: Florida Technological University, 1997

[13] Wilson J M. Gantt charts: A centenary appreciation. European Journal of Operational Research, 2003, 149(2):430-437

[14] Glover F. Future paths for integer programming and links to artificial intelligence. Computers and Operations Research, 1986, 13(5):533-549

[15] Barnes J W, Chambers J B. Solving the job shop scheduling problem with tabu search. IIE Transactions, 1995, 27(2):257-263

[16] Hurink J, Jurisch B, Thole M. Tabu search for the job-shop scheduling problem with multi-purpose machines. Operations-Research-Spektrum, 1994, 15(4):205-215

[17] Ponnambalam S G, Aravindan P, Rajesh S V. A tabu search algorithm for job shop scheduling. International Journal of Advanced Manufacturing Technology, 2000, 16(10):765-771

[18] Geyik F, Cedimoglu I H. The strategies and parameters of tabu search for job-shop scheduling. Journal of Intelligent Manufacturing, 2004, 15(4):439-448

[19] Shen L. A tabu search algorithm for the job shop problem with sequence dependent setup times. Computers and Industrial Engineering, 2014, 78:95-106

[20] Pinedo M. L. Scheduling. New York: Springer, 2008

How to cite this article:
Wong K L, Nordin S Z. Disjunctive programming-tabu search (dp-ts) approach for job shop scheduling problem. J. Fundam. Appl. Sci., 2018, *10(6S), 737-751*.