Contents lists available at GrowingScience

Journal of Project Management

homepage: www.GrowingScience.com

Effective heuristics for solving dynamic variant of single processor total tardiness problems

Saheed Akande[a*], Ayodeji Emmanuel Oluleye[b] and Elkanah Oyetunji[c]

[a]*Department of Mechanical and Mechatronics Engineering, Afe Babalola University, Ado-Ekiti, Nigeria*
[b]*Department of Industrial and Production Engineering, University of Ibadan, Nigeria*
[c]*Department of Mechanical Engineering, Lagos State University, Nigeria*

CHRONICLE

Article history:
Received: July 5, 2017
Received in revised format: October 10, 2017
Accepted: November 10, 2017
Available online:
November 14, 2017

Keywords:
Heuristics
Branch and Bound
Total Tardiness
Efficient solution
Effective

ABSTRACT

This paper considers the dynamic variant of single processor scheduling problem of minimizing total tardiness. In practice, it occurs when minimizing tardiness penalty. The problem is NP-hard; thus two heuristics were proposed. The utility of the proposed models was demonstrated through computational experiments and comparative analyses against existing solution methods and the Branch and Bound (BB) method. The results show that the proposed models yield efficient solutions and in most cases perform effectively better than the existing heuristics in the literature.

2018 Growing Science Ltd.

1. Introduction

Scheduling a set of jobs which are to be processed on a single processor to minimize the total tardiness is known as the Single Processor Total Tardiness Problems (SPTTP). When the release dates of all the jobs are effectively zero, the problem is static. Otherwise, it is called the dynamic variant (Pinedo, 2008). Effective scheduling of jobs to minimize the total tardiness is very important in manufacturing, production and servicing systems where penalty cost is proportional to total tardiness. Thus, to minimize the penalty cost, it is necessary to minimize the total tardiness. Though the single machine manufacturing systems are rare in practice, the results of single machine scheduling problems can be used for bottleneck machines in production lines as well as manufacturing cells (Süer et al., 2012). This work presents the solution methods for minimizing the total tardiness of jobs with release dates constraints. Using the three parameters notation $(\alpha|\beta|\gamma)$, the problem can be represented as: $1|r_j|\Sigma T_j$. (French, 1982).

* Corresponding author. Tel : +2348061231291
E-mail address: sakande1@abuad.edu.ng (S. Akande)

2018 Growing Science Ltd.
doi: 10.5267/j.jpm.2017.11.001

Due to the complexity of the SPTTP, Du and Leung (1990) proved that both the static and the dynamic variants of the problems are NP-hard with a given set of independent jobs. This implies that no heuristic has been found that yields optimal solutions. Thus, complete and implicit enumeration techniques are usually employed to solve the problems. However, computational time increases exponentially as the problem size grows. This limits the use of enumeration techniques. Nevertheless, Baker and Bertrand (1982) produced a near-optimal solution for the static variant of the problem. The solution is called the Modified Due Date (MDD) rule. The problem is thus, an NP-hard in the ordinary case. However, the dynamic variant of the problem is strongly NP-hard. Extensive literature review shows that researchers have explored either a metaheuristic like the genetic algorithm or an enumerative technique like the Branch and Bound (BB) method to solve the small-sized problems. For instance, Chu (1992) proposed a BB algorithm that can solve up to 20 jobs. Baptiste et al. (2004) presented a BB procedure with a time window constraint which represents the time interval within which each job can be scheduled. Chu and Portmann (1992) explored the Dynamic Modified Due Date (DMDD) rule to solve the problem. Naidu (2003), Bean and Hall (1985) also stated that the DMDD rule solves the problem satisfactorily. Moreover, Baker and Triesch (2013) listed authors who have explored heuristics to solve the problem. This is shown in Table 1.

Table 1
Some existing heuristics that solve SPTTP with non- zero release dates

Performance Measure	Best Rule (s)	Rules Compared	Author
Total tardiness	SPT, ODD	SCR	Kanet and Hayya (1982)
	SPT, EDD	MST, SCR	Baker and Bertrand (1981)
	MST,SPT,EDD	OST, S/OPN	Muhlemann et al. (1982)
	SPT/CR	COVERT	Anderson, and Nyirenda (1990)

Baker and Trietsch (2013) stated that none of the heuristics yielded the optimal solution. Thus, the problem remains open for further research. Therefore, proposing new heuristics that can produce better results or solutions not significantly different from the optimal even for small-sized problems will contribute to the state of knowledge on the subject matter. In addition, selecting among the existing heuristics based on performance for comparative analysis against the proposed heuristics would also contribute to the state of knowledge. This work will achieve this feat.

2. Problem definition

Given a single processor scheduling problem where a set of n jobs have to be sequenced on a processor in order to minimize the total tardiness. Assuming that only one job can be processed at a time and that the problem is deterministic with a known parameter. The problem parameters are represented by the notations outlined as follows;

JobSet A : The set of jobs to be scheduled, A= $\{J_1, J_2, \ldots, J_n\}$
P_i : Processing time of job Ji, i = 1, 2, . . . , n.
d_i : Due date of job Ji, i = 1, 2, . . . , n.
C_i : Completion time of job Ji, i = 1, 2, . . . , n.
r_i Release date of job Ji, i = 1, 2, . . . , n.
T_i : Job tardiness of job
S_i: The time the processing of a job, i starts on the processor
S_T: The slack time
p_j: The processing time of job j,
$a_j(t)$: The allowance of the job j.
t : The sequencing time
Π_i The priority index

$$S_i \geq r_i \quad (1)$$

$$C_i = S_i + p_i \quad (2)$$

$$S_i = C_{i-1} \text{ or } r_i \quad (3)$$

For i = 1, $S_i = r_i$, $C_i = p_i + r_i$ (4)

For job in position i+1, $C_{i+1} = C_i + p_{i+1}$, $S_{i+1} = C_i$, If $C_i \geq r_{i+1}$ (5)

Otherwise, $C_{i+1} = r_{i+1} + P_{i+1}$, $S_{i+1} = r_i$ (6)

The tardiness is given by: $T_i = max\ \{0\ ,(C_i - d_i)\}$ (7)

The total tardiness is (T_{tot}): $\sum_{i=1}^{n} T_i = \sum_{i=1}^{n} max\ \{0\ ,(C_i - d_i)\}$ (8)

According to Oyetunji (2009), a job is said to be late or tardy if it is completed after its due date.

3. Materials and methods

The methods adopted in this study involve proposing and implementing two new heuristics to solve the problem. Two heuristics were also selected (based on their performance) from the literature to solve the problem. In addition, the BB procedure was also implemented for small sized problems. Comparative analyses was then carry out on all the implemented solution methods. This methodology was also explored by Erenay et al. (2010).

3.1 Selected Solution Methods from the Literature

Among the solution methods found in the literature, the following heuristics were selected based on their reported performance for implementation. The solution methods are now outlined;

i. The Minimum Slack Time (MST) Rule: The MST rule schedules jobs in the order of increasing slack time. The slack time (S_T) of a job j is given by:

$$S_T = a_j(t) - P_j \quad (9)$$

$$a_j(t) = d_j - t \quad (10)$$

ii. The Modified Due Date (MDD) Rule: The MDD rule schedules the next job from the set of unscheduled jobs 'U' with the smallest priority index (Π_i). The priority index is given by:

$$\Pi_i = \{max\{t + p_i, d_i\}\} \quad (11)$$

If there are only two jobs j and k to be scheduled at a time t, job j will precedes job k if $\{max\{t + p_j,\ d_j\}\} \leq \{max\{t + p_k,\ d_k\}\}$. However, MDD rule does not considers two jobs at a time when there are more than two unscheduled jobs. It considers all the available jobs, computes their priority indices (Π_i) and chooses the job with the least priority index.

iii. The Shortest Processing Time (SPT) Rule: The SPT rule schedules jobs in the order of non-decreasing processing time.

iv. The Early Due Date (EDD) Rule: Jobs are scheduled in order of non-decreasing due date.

3.2 Proposed Solution Methods

Two heuristics (named the AA5 and the AA6) are proposed for this problem. The proposed solution methods are now described.

i. AA5 heuristic: In order to minimize the total tardiness on a single processor with non-zero release dates jobs, three parameters are involved; the processing times, the release dates and the due dates. In any schedule, the effects of all the parameters are significant at the beginning while towards the tail end the effect of both the processing times and the due dates are dominant. However, as the length of the schedule increases, the processing time is the most dominant parameter. In this regard, the AA5 algorithm combines the EDD with the SPT to obtain a schedule. The algorithm is as follows:

Initialization

JobSet A = [$J_1, J_2, J_3, \ldots J_n$], set of given jobs,   JobSet B = [0], set of scheduled job

JobSet C, JobSet D, JobSet E, JobSet U = [J_1’, J_2’, J_3’, . . . , J_n’], set of unscheduled jobs.

The steps now follow;

STEP 1: Arrange JobSet A in the order of increasing processing time and put same in JobSet C. If there is a tie, break the tie arbitrarily.

STEP 2: Arrange JobSet A in the order of increasing due date and put same in JobSet D. If there is a tie, break the tie arbitrarily.

STEP 3: Compute the tardiness of each of the jobs in the JobSet C and the JobSet D.

STEP 4: Combine the two schedules by scheduling the job in the same level and with the lower tardiness. If there is a tie break the tie with due date. The resultant schedule is called JobSet E.

STEP 5: Check if any job exists more than once in the JobSet E. If yes, remove the repeated job from the back position. Compute the length of the resultant schedule. The resultant schedule is called the JobSet U. Otherwise, JobSet E is renamed JobSet U.

STEP 6: If the length of JobSet U is equal to length of JobSet A. Go to step 10. Else go to step 7.

STEP 7: Subtract JobSet U from JobSet A to obtain the jobs that have not been scheduled. The jobs constitute JobSet H.

STEP 8: Arrange JobSet H at the back of JobSet U in the order of the due date.

STEP 9: Compute the total tardiness of the JobSet U. If the total tardiness of JobSet U is less than that of JobSet C, JobSet C is the JobSet B, otherwise JobSet U is the JobSet B.

STEP 10: Compute the total tardiness of the required schedule.

STEP 10: Stop.

ii. **AA6 heuristic:** This is a modification of the AA5 heuristic. The two heuristics are based on the same principle. However, the step 1 in the AA5 heuristic is replaced by arranging the JobSet A in the order of the sum of the processing time and the release date.

3.3 Application of the BB to Solve the Problem

The frontier search method was explored to branch while the DMDD heuristic was used to bound the branching tree. The bounding procedure calculates a lower bound by applying the dynamic MDD heuristic at the outset and compares the value to the objective function obtained from all the branches in all the nodes at each level.

4. Model implementation

The proposed as well as the selected solution methods were implemented to solve some randomly generated single processor scheduling problems. Matlab R2010 programing language was explored to generate the problem parameters which include the number of jobs, processing time, release date and due date. Süer et al. (2012) relation defined as follows was adopted

$$Pi = U(1, 10), \tag{12}$$

$$Ri = U(0, 40), \tag{13}$$

$$Di = R_i + kP_i. \tag{14}$$

where *k* is uniformly distributed between U(1,4). Furthermore, the performance of a given solution method can be influenced by the problem size. In other words, some solution methods perform better under small-sized problems while some perform under large-sized problems. Thus, in evaluating the performance of a solution method to scheduling problems, the method must be tested with a wide ranges of problem sizes. To achieve this, ten different problem sizes ranging from 5 to 100 jobs and with fifty problem instances under each problem size were generated and explore. A sample size (n) that is not less than 30 (n ≥ 30) is referred to as large sample size (Oyawale, 2006). Using this fact, the considered problem sizes were classified into the small-sized (5≤n≤ 25) and large-sized (30≤n≤ 100) problems. The adopted ranges of problem sizes (5-100 jobs) is to show that the heuristics proposed would be suitable for the small scale, medium scale, large scale firms. For instance, a small scale firm like small auto mechanic shops that can have up to 5 jobs waiting for processor. Also, a medium scale firm can have up 30 jobs on their production line. Furthermore, a very large and global firm (like Amazon shipping company, Maersk shipping Group) can have up to 100 jobs on the line. The adopted fifty instances is to ensure that the results to be obtained are not due to chance but a true reflection of the performance of the heuristics. The generated problems were solved by the solution methods. The coding of the solution on Matlab programing was executed with the single instance file which solves a single instance of the problem, execution file where the 50 instances were loaded and solved to obtained the mean of the objective function, and the run problem file where the single instant file and the execution file were loaded and run to give the output (Hahn & Valentine, 2016).

5. Results and discussion

The results obtained in solving the simulated problems by the implemented solution methods are discussed in this section. It involve the effectiveness and efficiency as well as the result of comparative analyses. Table 2 shows the mean of the total tardiness obtained.

Table 2
Mean of the total tardiness by solution methods and problem sizes

S/N	Sizes	MST	DMDD	EDD	SPT	AA5	AA6	BB
1	5 × 1	48.64	2.82	2.82	38.42	4.32	0.96	0.25
2	10 × 1	236.82	44.75	45.82	175.25	33.9	29.18	21.6
3	15×1	498.16	169.32	171.72	409.18	150.46	146.2	84.72
4	20 × 1	961.94	407.5	413.16	728	407.74	394.25	285.5
5	25×1	1563	770.28	778.94	1175.25	768.25	757	491.68
6	30×1	2202	1230	1241.75	1705.15	1115.48	1245.1	
7	40×1	3855.25	2450.2	2428	2946	2410	2550.04	
8	60×1	8803.35	6206	6268.2	6521	5816.89	6462	
9	80 ×1	15848.2	11708.2	11833	11534	11040.5	12162.1	
10	100 ×1	24571.1	18742.5	18970	17981.7	16960.7	19522	

Results based on Table 2, shows that among the solution method from the literature, SPT and DMDD yielded the best result. Thus, the two heuristics are explored for comparative analysis against the proposed solution methods. However, for small-sized problems; 5 ≤ n ≤ 25, the AA6 heuristic yielded the best results besides the optimal solution method (BB). Similarly, for larger-sized problems; 30 ≤ n ≤ 100, the AA5 heuristic produced the best results. In order to compare the effectiveness of the solution for proper ranking, the following analytical tests were carried out.

5.1 The Percentage Deviation (P.D) test

The ranges of deviation of the solution methods from the optimal were computed. Table 3 shows the percentage deviations of all the solution methods for the problem ranges; 5 ≤ n ≤ 30.

Table 3
The percentage deviation of the solution methods for problem ranges; 5 ≤ n ≤ 25

Heuristics	% Min. deviation (Best case)	% Max. deviation (Worst case)	% Range
AA5	$\frac{407.74-285.5}{285} \times 100 = 43$	$\frac{4.32-0.25}{0.25} \times 100 = 1628$	43% - 1628%
AA6	$\frac{29.18-21.6}{21.6} \times 100 = 35\%$	$\frac{0.96-0.25}{0.25} \times 100 = 284$	35% - 284%
DMDD	$\frac{407.5-285.5}{285} \times 100 = = 43$	$\frac{2.82-0.25}{0.25} \times 100 = 1028$	43% - 1028%
SPT	$\frac{1175-491.68}{491.68} \times 100 = 138$	$\frac{38.42-0.25}{0.25} \times 100 = 15268$	138% - 15268%

Thus, based on the P.D test, the heuristics can be ranked in the order; AA6, DMDD, AA5, SPT.

5.2 Approximation Ratio (A.R) Test

This test shows the closeness of heuristics to the optimal or the standard solution method. In other words, it gives the overall measure of how many times the optimal or the standard solution performed better than other implemented solutions. The BB solution results (the optimal) were used as the benchmark in the problem ranges; 5 ≤ n ≤ 25 while the AA5 heuristic results were used in the problem ranges; 30 ≤ n ≤ 100. Fig. 1 shows the plots of approximation ratios for the solution methods in the problem ranges; 5≤n ≤25. (See APPENDIX A, Table A.1 for the approximation ratio Table for small sized problems corresponding to Fig. 1).

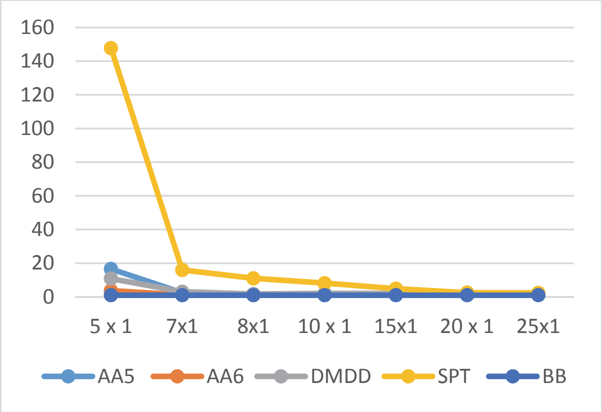

Fig. 1. Plots of approximation ratio of the solution methods for 5 ≤n ≤ 25

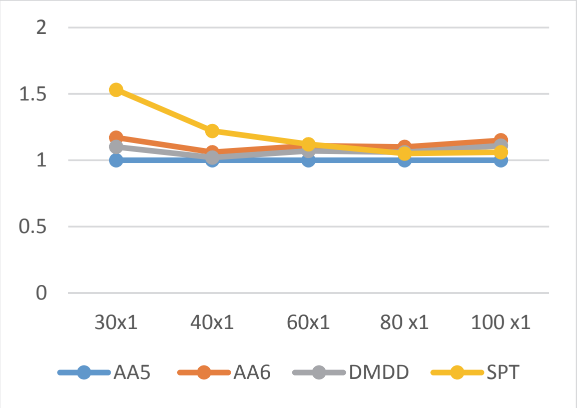

Fig. 2. Plots of approximation ratio of all the solution methods for 40 ≤n ≤ 100

The figure shows that the heuristics are closer to the BB plot in the ranking order; AA6, DMDD, AA5, SPT. The figure also shows that as the problem size increases, the differences between the heuristics decreases. Furthermore, Table 4 shows the overall means of approximation ratio of the solution methods for problem ranges; 5≤n ≤ 25.

Table 4
The overall means of approximation ratio for problem ranges; 5≤n ≤ 25

Solution methods	Overall means of approximation ratio	Solution methods	Overall means of approximation ratio
AA5	3.85	DMDD	3.24
AA6	1.72	SPT	27.52

The overall means of A.R computed implies that the heuristics; AA6, DMDD, AA5, and SPT are 1.72, 3.24, 3.85, and 27.52 times the optimal respectively on the average.

Fig. 2 shows the plots of approximation ratio for the large-sized problems; 30≤n ≤ 100. (See APPENDIX A, Table A.2 for the approximation ratio Table corresponding to Fig. 2). The figure shows that as the problem size increases, the SPT method converges towards the standard solution (AA5) while other solution methods diverge away. Furthermore, Table 5 shows the overall means of approximation ratio of the solution methods.

Table 5
The overall mean of approximation ratio for problem ranges; 30 ≤ n ≤ 100

Solution methods	Overall means of approximation ratio	Solution methods	Overall means of approximation ratio
AA6	1.12	DMDD	1.07
SPT	1.20		

5.3 The t-test

The statistical t-test of paired two samples for means was carried out using Spreadsheet 2013 platform to determine the significance of the observed difference. Tables 6 and 7 show the results of the t-tests for the problem ranges; $5 \leq n \leq 25$ and $30 \leq n \leq 1000$ respectively.

Table 6
t-test for mean of total tardiness for $5 \leq n \leq 25$ problems

Solution methods	AA5	AA6	DMDD	SPT	BB
AA5	--------	>0.05	>0.05	<0.05*	<0.05*
AA6	>0.05	-------	>0.05	<0.05*	>0.05
DMDD	>0.05	>0.05	-------	<0.05*	<0.05*
SPT	<0.05*	<0.05*	<0.05*	----------	<0.05*
BB	<0.05	>0.05	<0.05	<0.05*	--------

Table 7
Test of means of mean total tardiness for $30 \leq n \leq 100$ problems

Solution methods	AA5	AA6	DMDD	SPT
AA5	-------	>0.05	>0.05	<0.05*
AA6	>0.05	-------	>0.05	>0.05
DMDD	>0.05	>0.05	--------	>0.05
SPT	<0.05*	>0.05	>0.05	------

Note: *indicates significant result; Sample size = 50 ; -----indicates not necessary

Table 6 shows that for the problem sizes; $5 \leq n \leq 25$, the effectiveness of all the heuristics are significantly different ($p < 0.05$) from that of the optimal except for the AA6 heuristic. The mean values of the total tardiness (effectiveness) of the AA6 heuristic are not significantly different ($p > 0.05$) from the optimal. The table also shows that the SPT solution method produced the worst results. The results of the SPT are significantly different from all other heuristics. Table 7 shows that for the problem ranges; $30 \leq n \leq 100$, the differences in the performance of the implemented solution methods are not statistically different except for the SPT and AA5 heuristics. The AA5 heuristic performed significantly better than the SPT ($p < 0.05$).

5.4 Results Based on the Efficiency with Respect to the Total Tardiness

The performance of the implemented solution methods were also measured based on the time required to solve an instance of the problem. Table 8 shows the mean execution time obtained for all the solution methods by problem sizes. Expectedly, the BB method has the highest execution time. The time complexity associated with this method limits its application for the real life problems. The table also shows that the SPT algorithm yielded the minimum execution time. Furthermore, the t-test and approximation ratio test were carried out on the efficiency of the solution methods.

Table 8
Mean of the execution time by solution methods and problem sizes.

S/No	sizes	AA5	AA6	DMDD	SPT	BB
1	5 × 1	0.00061	0.00063	0.00067	0.0002	288
2	10 × 1	0.00064	0.00066	0.00069	0.0002	2979
3	15×1	0.00069	0.0007	0.00078	0.0002	4535
4	20 × 1	0.00071	7.2E-05	0.00086	0.00023	8727
5	25×1	0.00072	0.00074	0.00092	0.00026	17128
6	30×1	0.00075	0.00077	0.00099	0.00028	29367
7	40×1	0.00079	0.00079	0.00148	0.00029	
8	60×1	0.00083	0.00085	0.00196	0.00031	
9	80 ×1	0.00087	0.00089	0.00277	0.00035	
10	100 ×1	0.00092	0.00093	0.00362	3.7E-05	

5.5 The t-test

The t-tests was carried out to ascertain whether the differences observed between the execution times of the SPT and other heuristics were significant. The statistical t-test of paired two-samples for means was carried out using Spreadsheet 2013 data analysis. Table 9 shows the results of t-test for problem ranges; 5 ≤ n ≤ 100.

Table 9
t-test for mean execution time for 5 ≤n ≤ 100 problems

Solution methods	AA5	AA6	DMDD	SPT
AA5	---------	>0.05	<0.05*	>0.05
AA6	>0.05	-------	<0.05*	>0.05
DMDD	<0.05*	<0.05*	-------	<0.05*
SPT	>0.05	>0.05	<0.05*	---------

Note* indicates significant result; Sample size = 50 ; -----indicates not necessary

The results in Table 9 shows that the execution time of AA5, AA6 and the SPT are not significantly different ($p > 0.05$) except the DMDD heuristic. Other solution methods are significantly faster than the DMDD ($p < 0.05$).

5.6 The Approximation Ratio Test for Efficiency

The approximation ratio test was used to compare the efficiency of all the solution methods. In this case, the SPT algorithm was used as the standard solution method. Fig. 3 shows the plots of approximation ratio for all the solution methods.

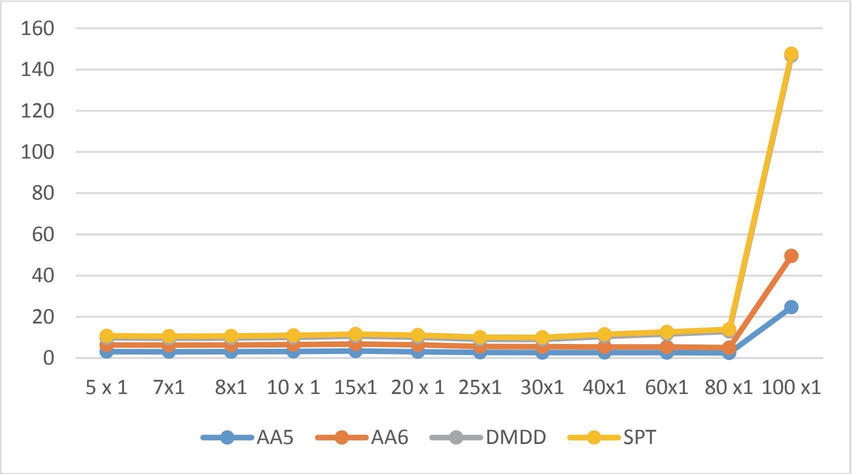

Fig. 3. Plots of approximation ratio of solution methods for 5 ≤n ≤ 100

The figure shows that at the initial stages (5 ≤ n ≤ 80), all the heuristics aligned. This implies that they show a very little differences in the execution time. However, as the problem sizes increase, the DMDD solution method diverges away. (See APPENDIX B, Table B.1 for the approximation ratio table for Fig. 3). Furthermore, the overall means of approximation ratios (of the execution time) for all the solution methods are shown in Table 10.

Table 10
The overall means of the approximation ratio

Solution methods	Overall means of approximation ratio
AA5	5.08
AA6	5.13
DMDD	42.4

The results in the table 10 show that the overall means of the AA5, AA6, and DMDD were 5.08, 5.13 and 42.4 respectively. This implies that that SPT solution method is 5.08, 5.13, and 42.4 times faster than the AA5, AA6 and the DMDD heuristics respectively on the average.

6. Conclusion

Based on the results obtained it can be concluded that the heuristic AA6 is recommended for solving SPTTP if the number of job does not exceed 30. This is because the method produced a result (in terms of effectiveness) that is not significantly different from the optimal for small problem sizes. The method also shows no significant difference in terms of execution time to the most efficient heuristic in the literature (SPT). However, if the number of accumulated jobs is greater than thirty (30), the heuristic AA5 is recommended.

References

Anderson, E. J., & Nyirenda, J. C. (1990). Two new rules to minimize tardiness in a job shop. *The International Journal of Production Research*, *28*(12), 2277-2292.

Baptiste, P., Carlier, J., & Jouglet, A. (2004). A branch-and-bound procedure to minimize total tardiness on one machine with arbitrary release dates. *European Journal of Operational Research*, *158*(3), 595-608.

Bean, J.C., & Hall, D.H. (1985). Accuracy of the modified due date rule. *Technical Report 85-10, Department of Industrial and Production Engineering, Michigan University*. 1-7.

Baker, K. R., & Bertrand, J. W. M. (1982). A dynamic priority rule for scheduling against due-dates. *Journal of Operations Management*, *3*(1), 37-42.

Baker, K. R., & Trietsch, D. (2013). *Principles of sequencing and scheduling*. John Wiley & Sons.

Chu, C., & Portmann, M.C. (1992). Some new efficient methods to solve the 1/ri/ $\sum_{i=1}^{n} T_i$ (min) scheduling problem. *European Journal of Operational Research,* 58, 404–413.

Chu, C. (1992). A branch-and-bound algorithm to minimize total flow time with unequal release dates. *Naval Research Logistics (NRL)*, *39*(6), 859-875.

Du, J., & Leung, J. Y. T. (1990). Minimizing total tardiness on one machine is NP-hard. *Mathematics of operations research*, *15*(3), 483-495.

Erenay, F. S., Sabuncuoglu, I., Toptal, A., & Tiwari, M. K. (2010). New solution methods for single machine bicriteria scheduling problem: Minimization of average flowtime and number of tardy jobs. *European Journal of Operational Research*, *201*(1), 89-98.

French, S. (1982). *Sequencing and Scheduling*, 1st ed. Ellis USA: Horwood Limited.

Hahn, B., & Valentine, D. T. (2016). *Essential MATLAB for engineers and scientists*. Academic Press.

Kanet, J. J., & Hayya, J. C. (1982). Priority dispatching with operation due dates in a job shop. *Journal of operations Management*, *2*(3), 167-175.

Muhlemann, A.P., Lockett, A. G., and. Farn, C. I. (1982). Job shop scheduling heuristics and frequency of scheduling. *International Journal of Production Research, 20*(2), 227–241.

Oyawale, F. A. (2006). *Statistical methods: An introduction*.1st ed. Nigeria: International Publisher Ltd.

Oyetunji, E.O. (2009). Some common performance measures in scheduling problems. *Research Journal of Applied Science, Engineering and Technology, 1*(2), 6-9.

Pinedo, M.L., (2008). *Scheduling – Theory, Algorithms and Systems*.1st ed. New York : Springer.

Naidu, J. T. (2003). A note on a well-known dispatching rule to minimize total tardiness. *Omega*, *31*(2), 137-140.

Süer, G. A., Yang, X., Alhawari, O. I., Santos, J., & Vazquez, R. (2012). A genetic algorithm approach for minimizing total tardiness in single machine scheduling. *International Journal of Industrial Engineering and Management (IJIEM)*, *3*(3), 163-171.

Appendix A

Table A1
Approximation ratio (Effectiveness) for small-sized problems

S/No	Sizes	AA5	AA6	DMDD	SPT	BB
1	5 × 1	16.62	3.69	10.85	147.77	1
2	7×1	2.51	1.197	3.04	15.98	1
3	8×1	1.42	1.17	1.7	10.99	1
4	10 × 1	1.57	1.35	2.08	8.13	1
3	15×1	1.78	1.73	1.99	4.85	1
4	20 × 1	1.43	1.38	1.43	2.55	1
5	25×1	1.56	1.54	1.57	2.39	1

Table A2
Approximation ratio (Effectiveness) for large-sized problems

S/No	Sizes	AA5	AA6	DMDD	SPT
6	30×1	1	1.17	1.1	1.53
7	40×1	1	1.06	1.02	1.22
8	60×1	1	1.11	1.07	1.12
9	80 ×1	1	1.1	1.06	1.05
10	100 ×1	1	1.15	1.11	1.06

Appendix B

Table B.1
Approximation ratio for execution time for problem-sized 5 ≤n ≤ 100

S/No	Sizes	AA5	AA6	DMDD	SPT	BB
1	5 × 1	3.13	3.23	3.44	1	1476923
2	7×1	3.05	3.2	3.35	1	4810000
3	8×1	3.1	3.25	3.35	1	7800000
4	10 × 1	3.23	3.32	3.47	1	1.5E+07
5	15×1	3.38	3.43	3.82	1	2.2E+07
6	20 × 1	3.07	3.31	3.72	1	3.8E+07
7	25×1	2.76	2.83	3.52	1	
8	30×1	2.66	2.84	3.51	1	
9	40×1	2.71	2.71	5.09	1	
10	60×1	2.66	2.72	6.28	1	
11	80 ×1	2.48	2.54	7.89	1	
12	100 ×1	24.66	24.93	97.05	1	

© 2018 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).