Minimizing the Sum of Makespan and Maximum Tardiness on Single Machine with Release Dates

E. O. Oyetunji
Academic Quality Assurance Unit
University for Development Studies
Ghana

Abstract

This paper considers the bicriteria scheduling problem of simultaneously minimizing makespan (C_{max}) and maximum tardiness (T_{max}) on a single machine with release dates. The problem is, by nature, NP-Hard. Therefore, one approximation algorithm (labeled as CTA1) was proposed for solving this problem. The CTA1 algorithm was compared with the Branch and Bounds (BB) procedure specifically implemented for this problem. The two criteria (makespan and maximum tardiness) were aggregated together into a linear composite objective function (LCOF). The performances of CTA1 and BB methods were evaluated based on effectiveness and efficiency. The two solution methods were tested on a set of 900 randomly generated single machine scheduling problems (problem sizes ranges from 10 to 500). Due to prohibitive long time, the BB procedure could not be applied to problems involving more than 100 jobs. Experimental results show that the CTA1 algorithm performed competitively (with respect to effectiveness) compared with the BB procedure when the number of jobs is less than 60. When the number of jobs exceeds 50, the BB outperformed (with respect to effectiveness) the CTA1 algorithm. However, the CTA1 outperformed (with respect to efficiency) the BB procedure for all the problem sizes considered.

Keywords
Makespan, Maximum Tardiness, Composite Objective Function, Scheduling, Approximation Algorithm, Single machine, Release dates

1. Introduction
Scheduling is concerned with efficient allocation of scarce resources (machines) over time to tasks (jobs) with the aim of minimizing or maximizing a set of objectives (also called criteria or performance measures). Ever since the realization of the fact that the total cost of a schedule is indeed not a function of one objective (criterion) but rather a function of two or more objectives (French, 1982), the bicriteria and multicriteria scheduling problems has attracted the attention of many researchers (Stein and Wein, 1997; Aslam et al., 1999; Rasala et al., 1999; Ehrgott and Grandibleux, 2000; Hoogeveen, 2005; Mansouri et al., 2009; Oyetunji, 2011; Oyetunji and Oluleye, 2012). Also, in practice, decision makers (DM) usually have to consider multiple criteria before arriving at a decision (Ehrgott and Grandibleux, 2000). A single criterion or objective represents only a component of the total cost of a schedule (French, 1982). Thus, considering scheduling problems with more than one criterion is more relevant in the context of real life scheduling problems (Nagar et al., 1995). Due to problem complexities, this paper considers the bicriteria scheduling problem. Several combinations of criteria have been explored by researchers (Stein and Wein, 1997; Aslam et al., 1999; Rasala et al., 1999). Because of their importance (Allahverdi, 2004), the makespan and maximum tardiness criteria are selected for study in this paper. Makespan is a measure of system utilization (which takes care of the manufacturer's concern) while maximum tardiness is a measure of performance in meeting customer due dates (which handles the customer's concern). Simultaneous minimization of two scheduling criteria involves aggregating the two criteria into a single function (called linear composite objective function) which is then minimized (Hoogeveen, 2005; Oyetunji and Oluleye, 2008; Oyetunji and Oluleye, 2010a; Oyetunji and Oluleye, 2010b). Usually, in this approach, the two criteria carry equal weights (indicating that both criteria are equally important to the decision maker). In real-life situations, not all the jobs usually arrive at time zero, hence the

introduction of release dates constraint. Also, problem complexities compelled us to limit the study to single machine problem. Therefore, this paper focuses on the bicriteria scheduling problem of simultaneously minimization of makespan and maximum tardiness on a single machine with release dates. The literature on simultaneous minimization of makespan and maximum tardiness criteria on a single machine with release dates appears sparse. To the best of our knowledge only Daniels and Chambers (1990), Chakravarthy and Rajendran (1999) and Allahverdi (2004) have studied both criteria on m-machine flowshop but with the constraint that the release dates are all zeros. Since we are unaware of any existing approximation algorithms or constructive heuristics for this class of scheduling problem under the particular shop environments that are of concern to us, one approximation algorithm is proposed and compared with the Branch and Bounds (BB) procedure specifically implemented for the problem. The paper is organized as follows: Section 1 deals with general introduction while the description of the scheduling problem being studied is given in section 2. The proposed approximation algorithm and the BB procedure implemented for the problem are discussed in section 3. Section 4 describes the data analysis while results and discussions are presented in section 5. The paper is concluded in section 6.

2. The Problem

The description of the scheduling problem studied in this paper is given as follows: Given a set *J* of *n* jobs to be sequenced on a single-machine in order to simultaneously minimize the completion time of the last scheduled job (also known as the makespan) and maximum tardiness. It is assumed that only one job can be processed at a time and the arrival time of every job J_i at the machine is known and denoted by r_i (release date). The processing time of each job on the machine is known and denoted by $p_{i.}$ Also, the expected date of delivery of each job is known and denoted by d_i. The time the processing of job J_i starts on the machine (start time) is designated as s_i with the property:

$$s_i \geq r_i \tag{1}$$

while its completion time (C_i) is defined as:

$$C_i = s_i + p_i \, . \tag{2}$$

also the makespan (C_{max}) is defined as:

$$C_{max} = max\ (C_1, C_2, \ldots, C_n) \tag{3}$$

and the tardiness of each job is defined as

$$T_i = max\ \{(C_i - d_i), 0\} \tag{4}$$

The maximum tardiness is defined as

$$T_{max} = max\ (T_1, T_2, \ldots, T_n) \tag{5}$$

Therefore, using the notations of Graham et al. (1979), the problem being explored is represented as

$$1 | r_i | (C_{\max}, T_{\max})$$

The linear composite objective function (LCOF) is defined as the aggregation of the two criteria (objectives) into a single function using scalar function.

Thus, LCOF is given as

$$F(C_{\max}, T_{\max}) = \alpha * C_{\max} + \beta * T_{\max} \tag{6}$$

$\alpha + \beta = 1$ (7)

Where α = relative weight of the makespan criterion and β = relative weight of the maximum tardiness

It is assumed that pre-emption is not allowed and that the problem is static and deterministic i.e. number of jobs, their processing times, and ready times and due dates are all known and fixed. It is also assumed that the two criteria (C_{max} and T_{max}) caries equal weight (i.e. α = β =0.5). The assumptions reflect many real-life problems which are often being encountered.

3. Solution Methods

The bicriteria scheduling problem of simultaneously minimizing the makespan and maximum tardiness is NP-Hard (Nagar et al., 1995; Chen et al. 1995), hence approximation algorithms are desired to solve the problem. Therefore, one approximation algorithm (labeled CTA1) is proposed for this problem. The CTA1 algorithm is discussed in section 3.1. Since we are unaware of any studies where approximation algorithms have been proposed for this bicriteria scheduling problem, the proposed algorithm was compared with the Branch and Bounds (BB) procedure described in section 3.2.

3.1 The CTA1 Algorithm

A proposed algorithm (called CTA1) constructs four partial schedules using different parameters of the given problem as well as four partial linear composite objective functions. The partial schedule that yields the least (for a minimization problem) value of the partial linear composite objective function is selected as the current solution. This is carried out iteratively until all the jobs have been assigned when the current solution (schedule) is taken as the final solution to the problem. The CTA1 steps are described below.

CTA1 Algorithm Steps

Step 0: Initialization, k=0

Job_SetA=[J1, J2,…,Jn]; This is the set of given jobs
Job_SetB=[J1, J2,…,Jn] This is the set of unscheduled jobs
Job_SetC={ } This is the set of available jobs at time t
Current_Solution = { }
Current_Function_Value = 0
Partial_Schedule_1 = { }
Partial_Function_1_Value = 0
Partial_Schedule_2 = { }
Partial_Function_2_Value = 0
Partial_Schedule_3 = { }
Partial_Function_3_Value = 0
Partial_Schedule_4 = { }
Partial_Function_4_Value = 0
$t = \min_{J_i \in Job\_SetA} \quad (r_i)$ (i.e. the minimum ready time of all the given jobs)

Step 1: At time t

Update Job_SetC with jobs for which $r_i \leq t$
If $r_i \leq t$ then Job_SetC = [Ji]

Step 2: $k = k+1$. From among the jobs that are available at time t, construct four partial schedules and compute their corresponding partial function values as follows:

Partial_Schedule_1= Select the job with the least job allowance (J_A). J_A = $d_i - p_i - t$ (where di = due date, pi = processing time) from Job_SetC and schedule it to k^{th} position in Partial_Schedule_1.

Partial_Function_1_Value = $\alpha(C_{\max}) + \beta(T_{\max})$ where α=relative weight of C_{max} , β= relative weight of T_{max}.

Partial_Schedule_2= Select the job with the earliest due date (d_i) from Job_SetC and and schedule it to k^{th} position in Partial_Schedule_2.
Partial_Function_2_Value = $\alpha(C_{\max}) + \beta(T_{\max})$.

Partial_Schedule_3= Select the job with the earliest release date (r_i) from Job_SetC and and schedule it to k^{th} position in Partial_Schedule_3.
Partial_Function_3_Value = $\alpha(C_{\max}) + \beta(T_{\max})$.

Partial_Schedule_4= Select the job with the least processing time (p_i) from Job_SetC and and schedule it to k^{th} position in Partial_Schedule_4.
Partial_Function_4_Value = $\alpha(C_{\max}) + \beta(T_{\max})$.

Step 3: Accept the Partial schedule that yields the minimum Partial Function Value (for a minimization problem) as the current solution or maximum Partial Function Value (for a maximization problem) and remove from both Job_SetB and Job_SetC the job that was schedule to the k^{th} position of the Partial Schedule that yielded the minimum Partial Function Value in Step 2. Based on the Current Solution, compute Current_Function_Value= $\alpha(C_{\max}) + \beta(T_{\max})$, start time $Si = t$ and completion time $Ci = Si + pi$

Step 4: Compute new time as: $t = max\ (\ Ci,\ \min_{J_i \in Job\_SetB} r_i\ )$; max of the completion time or the minimum ready time of the remaining unscheduled jobs.

Step 5: If k is less than the number of jobs (n) go back to Step 1; otherwise go to Step 6.

Step 6: Accept the Current_Solution as the final solution
i.e. Final_Solution (CTA1) = Current_Solution.
Final_Objective_Function_Value = Current_Function_Value.

Step 7: Stop.

3.2 Branch and Bounds (BB) Procedure

The literature on the metrics (criteria) and the shop environments explored in this paper appears sparse. We are unaware of any existing polynomial time algorithm (s) with which to compare our proposed algorithm. Therefore, the proposed algorithm was compared with the branch and bounds (BB) method. The branch and bounds solution method for the problem was implemented following the procedure of Sayin and Karabati (1999). Two branch and bound methods (one for makespan called BB_{MS} the other for maximum tardiness called BB_{MT}) were implemented in order to obtain the values of makespan and maximum tardiness at each node. The two criteria (makespan and maximum tardiness) were aggregated into a single function called linear composite objective function. The value of the linear composite objective function was computed to obtain the lower bound at each node. The node that gave the best solution (that is the smallest value of the linear composite objective function) was used to determine the tree to branch to. At the terminal node, when all the jobs have been fully assigned, the node was noted and became the solution to the considered problem.

4 Data Analysis

The proposed algorithm (CTA1) and the BB procedure were tested on a set of 900 (18 problem sizes ranging from 10 to 500 jobs by 50 problem instances) randomly generated problems. The processing times of the jobs were randomly generated (using random number generator in Microsoft visual basic 6.0) with values ranging between 1 and 100 inclusive. The ready times were also randomly generated with values ranging between 0 and 49 inclusive while the due dates were also randomly generated with values ranging from r_i+p_i to r_i+2*p_i inclusive.

A program was written in Microsoft visual basic 6.0 to apply both the CTA1 and BB methods to the problems generated. The case of simultaneous minimization of the makespan and maximum Tardiness is considered. Therefore, the program computes the value of the linear composite objective function ($K=0.5*Cmax + 0.5*Tmax$) obtained by each of the solution methods under each problem. The data was exported to Statistical Analysis System (SAS version 9.2) for detailed analysis. The hardware used for the experiment is a 1.87 GHz P6000 Intel CPU with 4 GB of main memory. The test of means was carried out using the GLM procedure in SAS so as to determine whether or not the differences observed in the mean value of the composite objective function obtained by the BB and CTA1 methods are statistically significant. Approximation ratio (CTA1/BB) with respect to effectiveness and efficiency was also computed. The results obtained are presented and discussed in section 5.

5. Results and Discussions

The value of the linear composite objective function (LCOF) obtained by any solution method is a measure of the effectiveness of that solution method while the efficiency of the solution method is measured by the execution time taken to obtain solution to an instance of a problem. For a minimization problem, the smaller the value of the linear composite objective function the better (more effective) the solution method. Also, the smaller the execution time taken the better (more efficient) the solution method. Therefore, the mean value of the linear composite objective function obtained using the BB and CTA1 methods over the 50 problem instances solved under 18 different problem sizes ranging from 10 to 500 jobs is shown in Table 1. The BB method could not be applied to problem instances that exceed 100 jobs due to prohibitive execution time (exponential in nature) required to solve an instance of problem (BB required more than one hour to solve an instance of problem). As expected, the BB method (being an implicit enumeration technique) gave the minimum mean value of the linear composite objective function for all the problem sizes considered (10 to 100 jobs) while the CTA1 closely trailed behind (Table 1). The BB method is known to yield optimal solution. When the results presented in Table 1 were subjected to statistical test, the results obtained are summarized in Tables 2 and 3. When the number of jobs ranges between 10 and 50 inclusive, the differences observed in the mean value of the linear composite objective function obtained by the BB and CTA1 methods are not statistically significant (indicating competitive performance) at 5% level (Table 2). However, when the number of jobs exceeds 50, the mean value of the linear composite objective function obtained by BB is significantly different (indicating better performance of BB over CTA1) from that of the CTA1 method at 5% level (Table 3). Therefore, a firm whose focus is mainly on effectiveness will find the CTA1 algorithm useful for only small size problems (problems not involving more than 50 jobs).

In order to measure the closeness of CTA1 performance to that of BB, the approximation ratio (CTA1/BB) was computed and plotted in Figure 1. The performance of CTA1 was closest to that of BB when the number of job is 70 (about 1.2% away from BB) while the widest gap between CTA1 and BB occurred when the number of jobs is 90 (CTA1 is about 3.5% away from BB). This is an indication of the CTA1 competitive performance compared with the BB procedure (Fig. 1).

Table 4 shows the execution time (secs) taken by BB and CTA1 methods to obtain solution to an instance of a problem. As expected the CTA1 algorithm (being an approximation algorithm) took less time (faster) compared with the BB method for all the problem sizes considered. For example, while it took the CTA1 method 0.0933 secs to solve an instance of problem involving 100 jobs the BB method took about 4425 secs to solve the same instance of the problem (Table 4). The time taken by CTA1 method is significantly different (indication faster method compared with BB) from that of the BB method at 5% level (Table 5). In order to observe the nature of the time complexity function of each of the BB and CTA1 methods, the execution time taken by both methods were plotted and is as shown in Figure 2. It is observed that the BB method begins to exhibit exponential time complexity function when the number of jobs exceeds 50 whereas the CTA1 method exhibited polynomial time complexity function throughout (Figure 2).Also, a firm who focus more on efficiency will find the CTA1 algorithm very useful as it can obtain solution within few seconds. In order to measure by what factor is the CTA1 algorithm faster than the BB method, approximation ratio (BB/CTA1) was computed and summarized in Table 6. The factor ranges from about 64 to about 47,428. Overall, the CTA1 method is faster than the BB method by a factor of about 17, 650 (Table 6).

Table 1. Mean value of linear composite objective function obtained using BB and CTA1 methods

Problem Size	Mean of linear composite objective function BB	CTA1
10x1	396.90	405.75
20x1	833.05	856.50
30x1	1379.72	1416.81
40x1	1819.04	1862.30
50x1	2365.30	2427.50
60x1	2759.48	2821.51
70x1	3390.36	3430.90
80x1	3822.90	3921.05
90x1	4223.77	4371.92
100x1	4820.87	4937.80
150x1	-	7406.90
200x1	-	9954.95
250x1	-	12556.47
300x1	-	15120.15
350x1	-	16152.79
400x1	-	15991.28
450x1	-	15536.30
500x1	-	15020.50

Sample size=50

Table 2. Test of means (probability values) with respect to effectiveness for $10 \leq n \leq 50$ problem sizes

	Solution Methods	
Solution Methods	**BB**	**CTA1**
BB	**-**	>0.2x
CTA1	>0.2x	-

Note x indicate non significant result at 5% level; Sample size = 50

- indicate not necessary

Table 3. Test of means (probability values) with respect to effectiveness for $60 \leq n \leq 100$ problem sizes

	Solution Methods	
Solution Methods	BB	CTA1
BB	-	<0.04*
CTA1	<0.04*	-

Note * indicate significant result at 5% level; Sample size = 50

- indicate not necessary

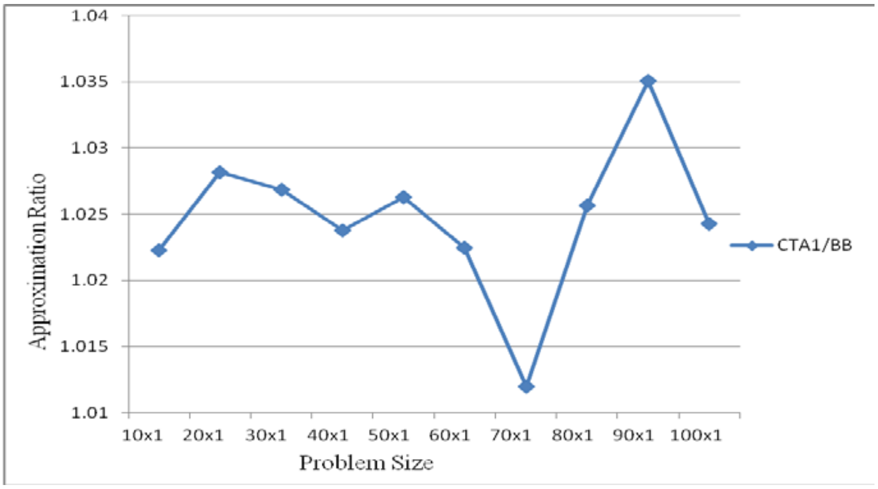

Figure 1: Approximation Ratio (CTA1/BB) with respect to effectiveness

Table 4. Average execution time (seconds) taken by BB and CTA1 methods

Problem	Mean of execution time (seconds)	
Size	BB	CTA1
10x1	0.4074	0.0063
20x1	6.8164	0.0118
30x1	99.3258	0.0190
40x1	123.1436	0.0260
50x1	276.4057	0.0357
60x1	604.7299	0.0451
70x1	1119.86	0.0571
80x1	2193.26	0.0667
90x1	3603.73	0.0804
100x1	4425.04	0.0933
150x1	-	0.1868
200x1	-	0.2801
250x1	-	0.4253
300x1	-	0.6698
350x1	-	0.9154
400x1	-	1.0995
450x1	-	1.3807
500x1	-	1.8999

Sample size=50

Table 5. Test of means (probability values) with respect to efficiency for $10 \leq n \leq 500$ problem sizes

	Solution Methods	
Solution Methods	BB	CTA1
BB	-	<0.0001*
CTA1	<0.0001*	-

Note * indicate significant result at 5% level; Sample size = 50

- indicate not necessary

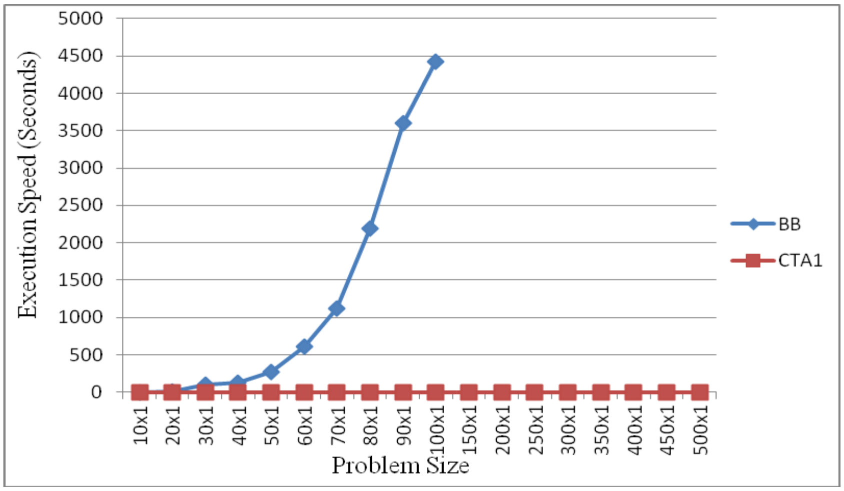

Figure 2: Execution time (seconds) taken by BB and CTA1 methods under the various problem sizes

Table 6. Approximation Ratio (BB/CTA1) with respect to efficiency

Problem Size	Approximation Ratio BB/CTA1
10x1	64.67
20x1	577.66
30x1	5,227.67
40x1	4,736.29
50x1	7,742.46
60x1	13,408.65
70x1	19,612.26
80x1	32,882.46
90x1	44,822.51
100x1	47,428.08
Average	**17,650.27**

Sample size=500

6. Conclusion

In this paper, the bicriteria scheduling problem of simultaneously minimizing the makespan (C_{max}) and maximum tardiness (T_{max}) on a single machine with release dates has been explored. The makespan and maximum tardiness criteria were aggregated into a single function called the linear composite objective function. In view of the NP-Hard nature of the problem, one approximation algorithm (labeled CTA1) was proposed for solving the problem. The CTA1 algorithm was compared with the Branch and Bounds (BB) procedure specifically implemented for the bicriteria scheduling problem. The performances of the CTA1 and BB methods were evaluated with respect to both effectiveness (closeness of the value of the linear composite objective function to the optimal) and efficiency (how fast solution can be obtained to an instance of problem i.e. a measure of execution speed). Experimental results (with respect to effectiveness) show that the CTA1 algorithm performed competitively (differences not significant) compared with the BB procedure when the number of jobs ranges from 10 to 50. The BB procedure outperformed (superior to) the CTA1 algorithm when the number of jobs exceeds 50. With respect to efficiency, the CTA1 algorithm performed exceptional better than (this is expected) the BB procedure for all the problem sizes considered. Therefore, based on both effectiveness and efficiency, the CTA1 algorithm is recommended for the bicriteria scheduling problem (involving at most 50 jobs) of minimizing the makespan and maximum tardiness on a single machine with release dates. Also, in view of the fact that the author is unaware of any studies where approximation algorithm has been proposed for the bicriteria scheduling problem explored in this paper and under the considered shop environments, the CTA1 algorithm thus become a benchmark/reference for future researchers. Future research efforts should be directed towards improving the effectiveness of CTA1 algorithm especially for large size problems (problems involving more than 50 jobs).

References

Aslam, J., Rasala, A., Stein, C. and Young, N., Improved Bicriteria Existence Theorems for Scheduling. In proceedings of the 10th annual ACM-SIAM Symposium on Discrete Algorithms, pp. 846-847, 1999.

Allahverdi, A., A new heuristic for m-machine &owshop scheduling problem with bicriteria of makespan and maximum tardiness, Computers & Operations Research, Vol. 31, pp. 157–180, 2004.

Chen, H., Chu, C. and Proth, J. M., A branch and bound approach for earliness-tardiness scheduling problems with different due dates, INRIA Report, 1995.

Chakravarthy, K, and Rajendran, C., A heuristic for scheduling in a flowshop with the bicriteria of makespan and maximum tardiness minimization, Production Planning and Control, Vol. 10, pp. 707–714, 1999.

Daniels, R.L, and Chambers, R.J., Multiobjective Flow-shop scheduling, Naval Research Logistics, Vol. 37, pp. 981–995, 1990.

Ehrgott, M. and Grandibleux, X., An Annotated Bibliography of Multiobjective Combinatorial Optimization, Report in Wirtschaftsmathematik, No. 62, Fachbereich Mathematik - Universitat Kaiserslautern, 2000.

French, S., Sequencing and Scheduling. Ellis Horwood Limited, 1982.

Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann. Discrete Math. Vol. 5, pp. 287-326, 1979.

Hoogeveen, J.A., Multicriteria Scheduling. European Journal of Operational research, Vol. 167, no. 3, pp. 592-623, 2005.

Mansouri, S.A., Hendizadeh, S.H. and Salmasi, N., Bicriteria scheduling of a two-machine flowshop with sequence-dependent setup times, The International Journal of Advanced Manufacturing Technology, Vol. 40, No. 11-12, pp. 1216-1226, 2009.

Nagar, A., Haddock J., Heragu,S., Multiple and bicriteria scheduling: A literature survey, European Journal of Operational Research, Vol. 81, no. 1, pp. 88-104, 1995.

Oyetunji, E.O.; Assessing solution methods to mixed multi-objectives scheduling problems, International Journal of Industrial and Systems Engineering Vol. 9, no. 2, pp. 213-226, 2011.

Oyetunji, E.O. and Oluleye, A.E., Heuristics for Minimizing Total Completion Time and Number of Tardy Jobs Simultaneously on Single Machine with Release Time, Research Journal of Applied Sciences, Vol. 3, no. 2, pp. 147-152, 2008.

Oyetunji, E.O. and Oluleye, A.E., Minimizing makespan and total completion time criteria on a single machine with release dates, Journal of Emerging Trends in Engineering and Applied Sciences, Vol. 1, no. 1, pp. 100-108, 2010a.

Oyetunji, E.O. and Oluleye, A.E., New Heuristics for Minimizing Total Completion Time and Number of

Tardy Jobs Criteria on Single Machine with Release Time, South African Journal of Industrial Engineering, Vol. 21, no. 2, pp. 101-113, 2010b.

Oyetunji, E.O. and Oluleye, A.E., A Generalized Algorithm for Solving Multicriteria Scheduling Problems, Advanced Materials Research, Trans Tech Publications Ltd., Switzerland Vol. 367 pp 653-666, 2012.

Rasala, A., Stein, C., Torng, E. and Uthaisombut, P., Existence Theorems, Lower Bounds and Algorithms for Scheduling to Meet Two Objectives, Technical Report, Dartmouth College, Computer Science Technical Report, PCS-TR99 . 347, 1999.

Sayin, S. and Karabati, S., A Bicriteria Approach to the 2-machine flowshop Scheduling problem, European Journal of Operational Research, Vol. 113, pp. 435–449, 1999.

Stein, C. and Wein, J., On the Existence of Schedules that are Near-Optimal for both Makespan and Total Weighted Completion Time. Operations Research Letters, Vol. 21, no. 3, pp. 115-122, 1997.