Multicriteria Scheduling Problems to Minimize Total Tardiness Subject to Maximum Earliness or Tardiness

H. F Abdullah.
Department of Mathematics, Ibn Al-Haitham College of Education, University of Baghdad

Abstract

Scheduling problems have been treated as single criterion problems until recently. Many of these problems are computationally hard to solve three as single criterion problems. However, there is a need to consider multiple criteria in a real life scheduling problem in general.

In this paper, we study the problem of scheduling jobs on a single machine to minimize total tardiness subject to maximum earliness or tardiness for each job. And we give algorithm (ETST) to solve the first problem (p_1) and algorithm (TEST) to solve the second problem (p_2) to find an efficient solution.

1 Introduction

Since the beginning, most of the work in scheduling problems has concentrated on a single criterion. Hence numerous optimal and approximation algorithms have been developed for single-criterion problems [1].

However, scheduling problems often involve more than one aspect and therefore require multiple criteria analysis. Despite their importance, little attention has been given to multiple criteria scheduling problems. This is due to the extreme complexity of these combinatorial optimization problems. Obviously, the situation becomes more complicated when more criteria are involved, unless the criteria are not in conflict with each other; roughly speaking, two criteria are not in conflict if a solution that performs well on one criterion is likely to perform well on the other criterion [2].

The simplest multiobjective problems focus only on two criteria. In this paper, we let Lex(A,B) denotes a typical hierarchical problem where A and B are two performance measures. The notation Lex(A,B) will be used to mean that we want to find a schedule that minimizes criterion B subject to the constraint that criterion A is optimal. These problems are also called secondary criteria problems where the secondary criterion B refers to the less important criterion.

Throughout this paper, we use the three field notation scheme α / β / γ introduced by Graham et.al., [3] to denote the scheduling problem under consideration.

Some of the performance measures often used in scheduling are, sum of completion times ($\sum c_i$)total earliness ($\sum E_i$) total tardiness ($\sum T_i$) maximum lateness, $L_{max} = Max\{L_i\}$, $L_i = c_i - d_i$, maximum earliness, $E_{max} = Max\ \{E_i\}$, $E_i = Max\{d_i - c_i, 0\}$, and maximum tardiness, $T_{max} = Max\ \{T_i\}$, $T_i = Max\{ c_i - d_i, 0\}$.

Let $f = \sum c_i$ is the primary criterion and let $g \in \{\sum E_i, \sum T_i, L_{max}, E_{max}\}$ the secondary criterion, then the problem 1//Lex(f,g) can be solved in polynomial time, where Lex means Lexicographical (hierarchical) optimization. A detailed complexity analysis of hierarchical problems can be found in Lee and Vairaktarkis, [4].

Note that the hierarchical scheduling problem 1//Lex(f,g) is a special case of the simultaneous minimization 1//F(f,g) problem, where F is an increasing composite function of the two criteria and hence if 1//Lex(f,g) is NP-hard then 1//F(f,g) is also NP-hard.

Vanwassenhove and Gelder, [5] develop on algorithm to generate all the efficient solutions for $1//F(\sum c_i, L_{max})$ problem in polynomial time.

Hoogeveen, [6] shows that all efficient solutions for two different cost functions, f_{max}, g_{max} can be generated in polynomial time, where $f_{max} = Max\{f_i(c_i)$ and f_i is any non-decreasing function of $c_i\}$. Note that T_{max} is a special case of f_{max}. For detailed surveys on multicriteria scheduling, the reader can refer to Gupta and Kyparisis, [7], Fryet al., [8], Nagar et al., [9] and Hoogeveen, [10].

In the literature, there are three approaches that are applicable to scheduling problems, [11].

C1. Minimizing a weighted sum of the subcriteria and convert it to a single criterion problem.

C2. Regard some subcriteria as constraints which must be satisfied and optimize others.

C3. Generate all efficient (non-dominated) schedules then allow the decision maker to make explicit trade-off between these schedules.

In this paper, we will study some problems which belong to class C2.

The work of Smith [12], on minimizing total completion time subject to no tardy jobs is the earliest work in this area. Recently [13] work on Lex $(C_{max}, \sum c_i)$ for the two machine flow shop problem.

In this paper, we address the following single machine multicriteria scheduling problem. A set of n independent jobs have to be scheduled on a single machine, which can handle only one job at a time. The machine is assumed to be continuously available from time 0 on words. Job $J_i(1,\ldots,n)$ requires a given positive processing time p_i and should be completed at a given due data d_i. A schedule defines for each job J_i its completion time C_i such that the jobs do not overlap in their execution. The cost of completing J_i at time C_i (i = 1, …,n) is measured by k (k = 3) functions f_i^k (k = 1, …,K); two of these functions are assumed to be non-decreasing in the job completion time; that is the value of $f_i^k(C_i)$ (i = 1, …,n; k = 1, …,K) does not decrease if we increase C_i and one of them (E_{max}) is not regular in our study. Hence for the hierarchical minimization problem, the performance criteria $f^1,\ldots,f^k$ are indexed in order of decreasing importance. In this paper, first f^1 is minimized. Next, f^2 is minimized subject to the constraint that the schedule has minimual f^1 value. If necessary, f^3 is minimized subject to the constraint that values for f^1 and f^2 are equal to the values determined in the previous step. If we use the three field notation, this problem is denoted by $1//Lex(f^1, f^2, f^3)$, where f^1, f^2 and $f^3 \in \{E_{max}, T_{max}, \sum T_i\}$.

The organization of this paper is as follows. In section 2, we provide the notation and basic concepts of the problems. In section 3, the proposed mathematical formulations for the problems is given. Also the proposed algorithms and the computational experience are given. Finally, in section 4 some of the conclusions that can be drawn from this research are outlined.

2 Notation and basic concepts

The following notation will be used:

n = number of jobs.

p_i = processing time of job i.

d_i = due data of job i.

c_i = completion time of job i.

$E_i = Max\{d_i - c_i, 0\}$; the earliness of job i.

$E_{max} = Max\{E_i\}$; the maximum earliness.

$T_i = Max\{c_i - d_i, 0\}$; the tardiness of job i.

$T_{max} = Max\{T_i\}$; the maximum tardiness.

$\sum T_i$ = the total tardiness.

We will use the following scheduling rules in this paper
EDD: jobs are sequenced in non-decreasing order of due dates, (this rule is known to minimize L_{max} and T_{max}) [14].
MST: jobs are sequenced according to non-decreasing order of minimum slack times, i.e. non-decreasing order of $s_i = d_i - p_i$, (this rule is known to minimize E_{max} subject to no machine idle time) [14].

Property 1, (4):
If for some criterion f the unconstrained problem 1//f is NP-complete then the hierarchical problem 1//Lex(f,g) is NP-complete for any criterion g.

A feasible schedule θ is pareto optimal, or non-dominated (efficient), with respect to the performance criteria f and g if there is no feasible schedule π such that both $f(\pi) \le f(\theta)$ and $g(\pi) \le g(\theta)$, where at least one of the inequalities is strict [17].

Suppose that we have selected the two performance criteria, say f and g, that we want to take into account [15]. If one performance criterion, say f, is for more important than the other one, then an obvious approach is to find the optimum value with respect to criterion f, which we denote by f*, and choose from among the set of optimum schedules for f the one that performs best on g, such an approach is called hierarchical optimization or Lexicographical optimization: in this type, we have to minimized the value of the more important criterion f, where in the second stage, the second criterion g is a minimized subject to the additional constraint that f = f*, where the criterion mentioned first in the argument of Lex is the more important one [16].

3 Three-criteria hierarchical problems and algorithms

In this section, we present the mathematical forms and the algorithms for generating solutions when one of the three criteria E_{max}, T_{max}, $\sum T_i$ is more important than the others. These hierarchical problems are also called secondary criteria problems where the secondary criterion refers to the less important criterion.

Formulation for multicriteria problems are similar to that for the single criterion problems with additional constraints requiring that the optimal value of the primary objective is not violated. Let us consider the formulations for bicriterion problems.
There are two parts of the formulations

primary objective function
subjected to:
  primary problem constraints
secondary objective function
subjected to:
  secondary problem constraints
  primary objective function value constraints
  primary problem constraints

Hence, the bicriterion problem is solved in two steps. First, we optimize the primary criterion followed by the optimization of the secondary criterion subject to the primary objective value. This formulation in this paper can be generalized to the multicriteria problems. Hence we present the mathematical forms of four multicritria problems.

Let the first problem (P_1) is denoted by $1//Lex(E_{max},T_{max},\sum T_i)$.

The multicriteria scheduling problem (P_1) is defined as:

$$\left.\begin{array}{l} \text{Min}\sum_{i=1}^{n} T_i \\ \text{s.t.} \\ \quad E_{max} = E^*_{max}(MST)(\text{is the optimal}) \\ \quad T_{max} \leq T, T \in \{T_{max}(EDD), T_{max}(MST)\} \end{array}\right\} (P_1)$$

For this problem (P_1), E_{max} is the most important objective function and should be optimal for any feasible schedule.

The following algorithm (ETST) gives the best possible solution for (P_1).

Algorithm (ETST)

Step (1): Solve the problem $1//E_{max}$ to find $E^*_{max}(MST)$, by using MST rule.

Step (2): Let N ={1,2,…,n} be the set of unscheduled, $\theta = \phi$ be the sequence for the scheduled jobs and set k = 1.

Step (3): For each job $j \in N$ calculate a start time r_j, $r_j = \text{Max}\{d_j - p_j - E^*_{max}(MST), 0\}$.

Step (4): Find a job $j^* \in N$ with minimum r_{j^*} such that $r_{j^*} \leq C_{k-1}$ and if there exists a tie choose the job j* with smalled due date d_{j^*} (where C_{k-1} is the completion time of a job in position k – 1 and $C_0 = 0$ where k = 1). Assigin job j* in position k of θ (i.e. $\theta = (\theta, \theta(k))$).

Step (5): Set k = k + 1 and N = N – {j*}, if $N = \phi$ go to step (6) otherwise go to step (4).

Step (6): For the schedule jobs of $\theta = (\theta(1), \ldots, \theta(n))$ calculate E_{max}, T_{max}, $\sum T_i$ and stop.

Let the second problem (P_2) denoted by $1//Lex(T_{max}, E_{max}, \sum T_i)$.
The multicriteria scheduling problem (P_2) is defined as:

$$\left.\begin{array}{l} \text{Min}\sum_{i=1}^{n} T_i \\ \text{s.t.} \\ \quad T_{max} = T^*_{max}(EDD)(\text{is the optimal}) \\ \quad E_{max} \leq E, E \in \{E_{max}(MST), E_{max}(EDD)\} \end{array}\right\} (P_2)$$

For this problem (P_2) T_{max} is the most important objective function and should be optimal for any feasible schedule.

The following **algorithm (TEST)** gives the best possible solution.

Step (1): Solve the problem $1//T_{max}$ to find $T^*_{max}(EDD)$, by using EDD rule.

Step (2): Let N ={1,2,…,n} be the set of unscheduled jobs, $\theta = \phi$ be the sequence for the scheduled jobs and let k = n and $t = \sum_{j=1}^{n} P_j$.

Step (3): For each job $j \in N$ calculate a dead line $\overline{d_j}$, $\overline{d_j} = d_j + T^*_{max}(EDD)$ and $S_j = d_j - P_j$.

Step (4): Find a job $j^* \in N$ such that $d_{j^*} \geq t$, if there exists a tie choose the job with largest slack time s_{j^*}.

Step (5): Set $t = t - p_{j^*}$, $k = k - 1$, $N = N - \{j^*\}$ and assigin job j* in position k of θ (i.e. $\theta = (\theta(k),\theta))$, if $N = \phi$ go to step (6), else go to step (4).

Step (6): For the schedule jobs of $\theta = (\theta(1), \ldots,\theta(n))$ calculate T_{max}, E_{max}, $\sum T_i$.

Now consider the following (P_3) and (P_4) problems $1//Lex(\sum_{j=1}^{n} T_j, E_{max}, T_{max})$ and $1//Lex(\sum_{j=1}^{n} T_j, T_{max}, E_{max})$ respectively.

$$\left.\begin{array}{l} \text{Min } T_{max} \\ \text{s.t.} \\ \quad \sum_{i=1}^{n} T_i = \sum_{j=1}^{n} T_j^* \text{(is the optimal)} \\ \quad E_{max} \le E, E \in \{E_{max}(MST), E_{max}(EDD)\} \end{array}\right\} (P_3)$$

$$\left.\begin{array}{l} \text{Min } E_{max} \\ \text{s.t.} \\ \quad \sum_{i=1}^{n} T_i = \sum_{j=1}^{n} T_j^* \text{(is the optimal)} \\ \quad T_{max} \le T, T \in \{T_{max}(EDD), T_{max}(MST)\} \end{array}\right\} (P_4)$$

Since the unconstrained total tardiness problem ($1//\sum T_i$) is NP-complete in ordinary sence (Due and lenug) [17].

Consequently by property 1, the corresponding hierarchical optimization problems (P_3) and (P_4) are NP-complete.

This means that all the hierarchical problems with primary criterion $//\sum T_i$ (total tardiness) are strongly NP-complete because $1//\sum T_i$ problem is NP-complete.

3.1 Computational results

We first present how tests problem can be randomly generated. The processing time P_i is uniformly distributed in the interval [1,10]. The due date d_i are uniformly distributed in the interval $[p(1 - TF - \frac{RDD}{2}), P(1 - TF + \frac{RDD}{2})$; where $P = \sum_{i=1}^{n} P_i$, depending on the relative range of due date (RDD) and on the average tardiness factor (TF). For both parameters, the values 0.2, 0.4, 0.6, 0.8 and 1.0, are considered. For each selected value of n, one problem was generated for each of five values of parameters producing five problems for each value of n.

The complete enumeration (CE), (ETST) and (TEST) algorithms were tested by coding them in matlab7 and running Pentium IV at 2800MHZ with Ram 512MB computer. It is well known that (CE) algorithm gives optimal solutions which are tested on problems with size (3,4,5,6,7,8) for problems (p_1) and (p_2) respectively. For problems (with $n > 8$) that are not solved optimality by (CE) algorithm because the execution time exceeds 30 minutes, the near optimal solution for these unsolved problems was found by our algorithms (ETST) and (TEST) respectively.

Tables (1) and (2) show the results for problems (p_1) and (p_2) obtained by (CE), (ETST) and (TEST) algorithms respectively.

References

1. Steure Ralph, E., (1986), "Multiple Criteria Optimization: Theory, Comutation and Application", Willey.
2. Hoogeveen, J. A., (1995),"Single-Machine Scheduling to Minimize a Function of Two or Three Maximum Cost Criteria", Department of Math. And Computing Science, Eindhoveen, the Netherlands.
3. Graham, R. L.; Lawler, E. L.; Lenstra, J. K. and Rinnooykan, A. H. G., (1979), Ann. Discrete Math. 5: 287-326.
4. Lee, C. Y. & Vairaktarakis, G. L., (1993), World scientific, 19: 269-298.
5. Van Wassenhove, L. N. & Gelders, F., (1980), European Journal of Operational Research, 4: 42-48.
6. Hoogeveen, J. A., (1992), "Single-Machine Bicriteria Scheduling", Ph.D. Thesis, Eindhoveen.
7. Gupta, S. K. and Hyparisis, J., (1987), Omega 15: 207-227.
8. Fry, T. D.; Armstrong, R. D. and Lewis, H., (1989), Research Omega 17: 54-607.
9. Nagar, A.; Haddock, J. and Heragu, S., (1995), European Journal of Operations Research, 81: 88-104.
10. Hoogeveen, J. A., (2005), Department of Computer Science, Europen Journal of Operational Research, 167.
11. Tsiushuang, C.; Xiangtong, Q. & Fengsheng, T., (1996), "Single Machine Scheduling to Minimize Weighted Earliness Subject to Maximum Tardiness", Department of Computers and System Sciences, Nankai University, Tianjin, 300071, P.R. China Qi.
12. Smith, W. E., (1956), "Various Optimizers for Single-Stage Production", Naval Res-Logist Quar.
13. Trao, H. M., (2006), "Optimal Solution for Two-Stages Flow Shop Scheduling Problem with Secondary Criterion", M.Sc. Thesis University of Al-Mustansiriyah, College of Science Dep. of Math.
14. Jakson, J. R., (1995), "Scheduling a Production Line to Minimize Maximum Tardiness", Res. Report 43, Management Science, Res.Project, University of California, Loss Angles, CA.
15. Hoogeveen, J. A., (1996), Math. Operat. Res. 21: 100-114.
16. Evans, G. W., Management science 30: 1268-1282.
17. Du., J and Lenug, J. Y. T., (1990), Math. operat. Res. 15: 483-495.

Table (1) The Performance of (CE) and (ETST) algorithm for Problem (p_1)

n	no. of ex.	(CE) Alg. Opt.val.			(ETST) Alg.		
		E	T	ST	E	T	ST
3	1	0	11	14	0	13	22
	2	0	12	14	0	12	14
	3	0	8	9	0	8	9
	4	0	15	22	0	15	22
	5	0	2	3	0	2	3
4	1	5	4	4	5	4	4
	2	0	6	15	0	6	15
	3	0	12	20	0	12	20
	4	0	9	14	0	9	14
	5	0	11	13	0	18	31
5	1	0	9	27	0	9	27
	2	0	19	44	0	19	49
	3	0	14	34	0	14	39
	4	0	5	10	0	5	10
	5	0	11	20	0	13	28
6	1	0	9	24	0	11	44
	2	0	13	32	0	14	47
	3	0	8	27	0	9	29
	4	0	23	52	0	27	94
	5	0	9	36	0	9	40
7	1	0	30	104	0	30	127
	2	0	13	52	0	13	59
	3	0	16	53	0	16	70
	4	0	32	113	0	32	121
	5	0	18	60	0	18	60
8	1	0	56	204	0	56	204
	2	2	19	45	2	21	59
	3	0	21	93	0	21	107
	4	0	24	105	0	24	113
	5	11	0	0	11	0	0

where E = E_{max}, T = T_{max} and ST =

Table (2) The Performance of (CE) and (TEST) algorithm for Problem (p_2)

n	no. of ex.	(CE) Alg. Opt.val.			(TEST) Alg.		
		T	E	ST	T	E	ST
3	1	11	0	14	11	6	13
	2	12	0	14	12	0	14
	3	8	0	9	8	0	9
	4	15	0	22	15	0	23
	5	2	0	3	2	2	3
4	1	4	5	4	4	17	4
	2	6	0	15	6	0	15
	3	12	0	20	12	0	28
	4	9	0	14	9	0	16
	5	11	0	13	11	2	20
5	1	9	0	27	9	7	34
	2	19	0	44	19	2	45
	3	14	0	34	14	6	37
	4	5	0	10	5	0	12
	5	11	0	20	11	8	28
6	1	9	0	24	9	5	13
	2	13	0	32	13	12	34
	3	8	0	27	8	8	27
	4	23	0	52	23	3	52
	5	7	0	19	7	4	20
7	1	30	0	104	30	3	109
	2	13	0	52	13	5	51
	3	16	0	53	16	11	59
	4	32	0	113	32	3	124
	5	18	0	60	18	0	69
8	1	56	0	204	56	0	235
	2	19	2	45	19	23	74
	3	21	0	93	21	7	84
	4	24	0	105	24	4	117
	5	0	11	0	0	25	0

Where $E = E_{max}$, $T = T_{max}$ and $ST = \sum T_i$

Table (1) and table (2) show (12) problems, (ETST) algorithm give the optimal solution from (30) problems to (p_1). Also (TEST) algorithm gives optimal solution to (3) problems from (30) problems to (p_2).

Table (3) The Performance of (ETST) and (TEST) algorithm for Problems (p_1) and (p_2) respectively

n	no. of ex.	(CE) Alg. Opt.val.			time	(TEST) Alg.			time
		E	T	ST		T	E	ST	
100	1	179	0	0	1.12647036	0	752	0	0.74858
	2	2	363	16796	0.11953005	358	229	18668	0.20988
	3	0	503	25959	0.26942652	502	53	24211	0.27051
	4	143	11	39	0.13075979	11	608	392	0.10606
	5	0	338	17138	0.21426937	338	211	16958	0.38384
200	1	319	45	1103	0.62734318	45	1258	5263	0.2926
	2	17	360	27598	0.37524356	360	808	49141	0.50081
	3	1	533	53662	0.66433407	531	499	59127	0.69638
	4	339	0	0	0.49009237	0	1465	0	0.30636
	5	0	558	61168	0.66032482	552	531	60117	0.67944
300	1	20	208	32476	0.62758178	207	1287	49249	0.54612
	2	47	107	18968	0.57598413	166	1444	37619	0.59398
	3	0	1165	172992	1.19656747	1163	493	166785	2.6832
	4	0	1334	205262	1.23098989	1334	330	188096	2.76682
	5	98	17	56	0.37925246	14	1626	2183	0.55104
400	1	640	0	0	1.15046111	0	2806	0	1.459
	2	89	86	7540	0.91170198	86	1087	25689	0.67944
	3	0	2026	415782	6053897226	2025	223	383686	6.1565
	4	447	39	782	3.12389536	39	2556	9151	0.98865
	5	11	231	36156	0.86524203	230	2016	71836	1.03529
500	1	279	160	17036	2.15753627	160	2215	57682	1.42939
	2	20	568	106437	2.15590147	561	2098	206248	2.42633
	3	0	2520	632436	7.90571441	2520	278	610651	10.1674
	4	0	1677	432201	5.54392977	1675	1104	445034	6.5482
	5	0	2425	617604	8.8913358	2424	266	578491	9.92506
600	1	659	71	2065	2.26473278	71	3400	27707	2.03721
	2	7	323	96218	3.15671991	323	2854	158962	2.20158
	3	0	1332	423809	5.56702833	1331	1966	490221	5.63833
	4	0	2306	672134	9.97153693	2305	967	690366	13.2039
	5	0	1328	378006	4.46032238	1328	1959	490624	7.67578
700	1	0	1880	616513	9.87129298	1879	1869	718391	13.4187
	2	69	69	12836	1.15406665	69	3025	35145	1.18976
	3	1	1895	674347	8.92376825	1895	1844	735889	1.22257
	4	0	1896	663143	7.90197834	1896	1882	753681	14.2562
	5	780	7	11	2.28917996	7	4592	1668	5.30548
800	1	45	1349	497654	7.66431661	1349	3062	715853	11.0311
	2	0	2634	1032408	14.5439547	2632	1748	1087947	20.0141
	3	0	3062	1193374	17.8258838	3060	1308	1197792	23.4111
	4	449	53	1151	3.28143218	53	4731	27641	2.28639
	5	0	3813	1531458	2.61768531	3811	421	1473201	28.971

where $E = E_{max}$, $T = T_{max}$ and $ST = \sum T_i$

مسائل الجدولة للمعايير المتعددة لتصغير مجموع التأخير مشروطة الى أكبر تبكير أو تأخير

هند فالح عبدالله
قسم الرياضيات , كلية التربية ابن الهيثم ، جامعة بغداد

الخلاصة

عدت مسائل الجدولة مسائل معيار مفردة حتى الآن. العديد من هذه المسائل تعد صعبة الحل لثلاثة مسائل معيارية مفردة مع ذلك هنالك حاجة الى اعتبار المعايير المتعددة في مسائل الحياة الحقيقية على العموم.

في هذا البحث درست مسائل جدولة n من الاعمال على ماكنة واحدة لتصغير مجموع التأخير مشروطة الى اكبر تكبير أو تأخير لكل عمل، وقد اعطيت خوارزمية (ETST) لحل المسألة الاولى (p_1) ، وخوارزمية (TEST) لحل المسألة الثانية (p_2) لايجاد حل كفوء.