Approximate Solution for Two Machine Flow Shop Scheduling Problem to Minimize the Total Earliness

H. F. Abdullah
Department of Mathematics-Ibn-Al-Haitham College of Education - University of Baghdad
Received in:4 May 2011 , Accepted in: 26 February 2012

Abstract

This paper proposes a new algorithm (F2SE) and algorithm (Alg(n – 1)) for solving the two-machine flow shop problem with the objective of minimizing total earliness. This complexity result leads us to use an enumeration solution approach for the algorithm (F2SE) and (DM) is more effective than algorithm Alg(n – 1) to obtain approximate solution.

Key Words: Flow-shops scheduling, approximate solution, two machine flow-shop, minize the total earliness, scheduling problem.

Introduction

For many problems in production scheduling, as in other areas of combinatorial optimization, it is unrealistic to attempt to find an optimal solution. In the problem under consideration, the objective is to find efficient solution to minimize the total earliness.

In the flow shop scheduling problem, indicated byFm//Cmax, or by F//Cmax for flow shop problem (Lawler etal. 1993), can be stated as follows. There are n jobs numbered 1, 2, …, n, each of which is to be processed on machines 1,2,…,m in that order. Each job j (j = 1,2,…,n) has a processing time P_{jk} on machine k (k = 1,2,…,m). Each machine can process not more than one job at a time and each job can be processed by not more than one machine a time. The order in which jobs are processed need not be the same on all machines. The objective is to find a processing order on each machine which minimizes Cmax, the maximum completion time of all the jobs.

Finally, it is well known that for m = 2, the resulting flow shop problem i.e., F2//Cmax, can be solved using Johnson's algorithm [1]. The objective function minimiz the $\sum E_j$, i.e. the sum of earliness of all jobs on all machines. For a schedules S, the value of the $\sum E_j$ is denoted by $\sum E_{j(s)}$. A schedule that minimize the $\sum E_j$ is called near optimal and is denoted by S*.

The problem denoted by F2//$\sum E_j$ problem. Let S be an arbitrary permutation of n jobs. For simplicity assume S = 1, 2, …, n. It is well known that the completion time C_j^k of job j (of sequence S) on machine M_k is given by [2]

$$C_j^k = \max_j\{C_j^{k-1}, C_{j-1}^k\} + P_{jk}$$

$$C_j^0 = C_0^k = 0,\ \forall 1 \le j \le n, 1 \le k \le m$$

where P_{jk} is the processing of job i on M_k, k = 1, …, m.

It is well known that many papers given by (Conway etal. 1967) [3], Rinnooy kan 1976, [4], Lenstra (1977) [5], Simulated annealing algorithms are proposed by O'sman and Potts [6] while the F2//Cmax problem is well known and ploynomially solvable by Johnson's algorithm [1], the F2//Tmax problem is strongly NP-hard, also for the Fm//Cmax problem, we need to consider only schedules with the same processing order on the first two machines and the same processing order on the last two machines.

Therefore for both problem F2//Cmax and F3//Cmax, there exists an optimal solution that is a permutation schedule for which all machines process the jobs according to the same job sequence (Conway etal. 1967) [3]. However, for Fm//Cmax, when $m \geq 4$, it can be the case that no optimal solution is permutation schedule (Conway etal.) [3] shows that this result can not be extended any further, for present polynomial time algorithms for solving a SLK due date assignment and the flow shop scheduling problems with objective to minimize the total earliness [7].

The organization of this paper is as follows. In section two, we provide the notation and basic concepts of the problems. In section three, the proposed mathematical formulations for the problem is given. Also the proposed algorithms and the computational experience are given, while section four contains some concluding remarks.

Notation and Basic Concepts

The following notation will be used:

n = number of jobs
P_j = processing time of job j
d_j = due date of job j
c_j = completion time of job j
E_j = Max $\{d_j - c_j, 0\}$; the earliness of job j
$\sum E_j$ = the total earliness
S_j = the slack time ($S_j = d_j - P_j$)
Fm = flow shop with m machine
m = the number of machines is equal to m (m is positive integer).

Complete Enumeration Method (CE)

Enumeration method generates schedules one by one to find optimal solutions, lists all possible schedules and then eliminates the non-optimal schedules form the list leaving those that are optimal. Clearly searching for an optimal schedules among all possible schedules using complete enumeration is not appropriate even for problem of small size, thus the complete enumeration method may be rejected immediately [8].

Flow Shop Problem

In each job exactly one operation for every machine, all jobs go through all the machines in the same order, [9].

Mathematical Formula

The scheduling problem (P) is defined as:

M is $\sum E_j$

s.t.

$$C_j^B = \max_j\{C_{j-1}^B, C_j^A\} + b_j \quad \ldots(1)$$

$$S_j^A = d_j - a_j \quad , \quad j = 1, \ldots, n \quad \ldots(2)$$

$$S_j^B = d_j - b_j \quad , \quad j = 1, \ldots, n \quad \ldots(3)$$

$E_j \geq d_j - C_j^B$, j = 1, …, n …(4)
$E_j \geq 0$

Constraint (1) specifies that the completion time of job j on machine B is, constraint (2) assure that the slack time of machine A is equal to the difference between due date and the processing time of job j on machine A, constraint (3) assures that the slack time of machine B is equal to the difference between due date and the processing time of job j on machine B, constraint grater than or equal to $d_j - C_j^B$.

Algorthim (F2SE)

Step (1): Find slack time for each job $j \in N$, for machine A and B ($S_j^A = d_j - a_j, S_j^B = d_j - b_j$).

Step (2): This, rule can be described as sequence the jobs with $S_j^A \geq S_j^B$ in the first, in non-increasing order of S_j.

Step (3): Followed by the jobs with $S_j^B \leq S_j^B$ (for the same machine B) in non-decreasing order of S_j^B.

Step (4): For the schedule job of θ = (θ(1), …,θ(n)) calculate $\sum E_j$ of machine B.

Algorithm (Alg(n – 1))

Scheduling the jobs of θ = (θ(1), …,θ(n)), obtained by algorithm (F2SE) and calculate $\sum E_j$ of machine B, changing jobs of the schedule θ to (n – 1) positions to produce θ*, calculate $\sum E_j$ of machine B and choose to the minimum value.

Descent Method (DM):

It is the simplest type of neighborhood search, which is sometimes known as iterative local improvement. In this method only moves that result in an improvement in the objective function value are accepted [10].

Under a first improve search, the first move that improves the objective function value is accepted. On the other hand, best improve selects a move that yields the best objective function value among all neighbors, when no further improvement can be achieved, a descent method terminates with a solution that is a local optimum. The local optimum is not necessarily the true global optimum. A widely used remedy for this drawback is to use multi-start descent method (F2DM) in which multiple runs of descent from different starting solution are performed, and the best overall solution is selected [10].

F2DM can be executed for our problem as follows:

Step (1): Initialization

In this step a feasible solution θ = (θ(1), …,θ(n)) obtained from EDD rule (heuristic method) is chosen to be the initial current solution for F2DM.

Step (2): Neighborhood Generation

We use variable neighborhood search which is a simple change of neighborhood within the search. In order to improve the sequence θ the traveling between different neighborhoods gives a new sequence θ*, that will be obtained with its objective function value S(θ*).

Step (3): Evaluation

(1) S(θ*) < S(θ) then θ* is accepted as the current solution and set θ = θ*.
Go to step (2).

(2) otherwise S(θ*) ≥ S(θ),θ is retained as the current solution and go to step (2).

Step (4): Termination

This algorithm is terminated after (100) iteration at near optimal solution.

Computational Results

In this section we first present how tests problem can be randomly generated. The processing time a_j and bj is uniformly distributed in the interval [1,10]. The due date d_j are uniformly distributed in the interval $[P(1 - TF - \frac{RDD}{2}), P(1 - TF + \frac{RDD}{2})]$, $T = \sum a_i + b_i$ 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), algorithm (F2SE) (DM) and algorithm (Alg(n – 1)) were tested by coding them in matlab 7 and running Pentium IV at 2800MHZ with Ram 1GB 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). 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 (F2SE) and algorithm (Alg(n – 1)) respectively.

Table (1) shows the results of problem (P) obtained by algorithm (CE) comper to algorithm (F2SE) and (DM) algorithm (Alg(n – 1)) respectively.

Table (2) shows the results of problem (P) of comberison (F2SE) and (Alg (n – 1)).

Conclusion

In this paper, we have developed exact for n ≤ 8 and approximate solutions for two machine flow shop scheduling to minimize the total earliness.

This paper reports on the results of extensive computational test for the following developed algorithms (F2SE) and algorithm (Alg (n – 1)) comparing it with the (DM) and optimal solution (obtained by (CE) algorithm). The main conclusion to be drawn from our comparison of computational results is that F2SE and (DM) is more effective than algorithm (Alg (n – 1)) for the large problem instances.

Finally, the algorithm (F2SE) proposed here has been shown to perform well when tested against algorithm (Alg (n – 1)) to obtain approximate solution.

References

1. Breit, J. (2004), An Improved Approximation Algorithm for Two-Machine Flow Shop Scheduling with an availability constraint, Department of Information and Technology Management, Saarland University, Saarbraken, Germany.
2. Abdul-Razaq, T.S. and Al-Harby, M. H. (2006), Exact and Near Optimal Solution for Three Machines Flow Shop Scheduling, Mathematics Department, Journal of College of Education, No. 3:50-62
3. Conway, R.W.; Maxwell, W.L. and Miller, L.W. (1967), Theory of Scheduling Addision Wesley, Reading MA.
4. Lageweg, B. J.; Lawler, E. L.; Lenstra, J. K. and Rinnooy Kan, A. H. CT, (1981), Computer Aided Complexity Classification of deterministic Scheduling Problems, Mathematics Contrum, Amsterdam, The Netherlands, Report BW 138.

5. Shi-Ling and Cheng Xue-Guang, (2011), Complexity Reesults for Flow-Shop Scheduling Problems with Transportation Delays and a Single Robot, Journal of Applied Mathematics and Bioinformatics, No.1:135-142.
6. Ishibuchi, H.; Misski, S. and Tanaka, H., (1995), Modified Simulated Annealing Algorithms for Flow Shop Sequencing Problem, European Journal of Operation, 81:388-398.
7. Abdul-Razaq, T.S. (2005), Solvable Case of a Two Machine Flow Shop Scheduling Problem with No-Idle in Process, Mathematics Department, College of Science, Al-Mustansiriyah Journal Sci., 16, No.4:77-94.
8. Al-Shebani, H. M. T. (2006), Optimal Solution for Two-Stage Flow Shop Scheduling Problem with Secondary Criterion, M.Sc. Thesis, College of Science, Dep. Of Mathematics, University of Al-Mustansiriyah.
9. Bellman, R.; Esogbue, A.O. and Nabeshima, I., (1982), Mathematical Aspects of Scheduling and Applications, Perganon Press, Great Britain.
10. Al-Salihi, V.A.; Abdullah, H. F., and Abdul-Razaq, T.S., (2009), Exact and Local Algorithms for Single Machine Sequencing to Minimize the Multi Objective Functions of Total Tardiness, Maximum Earliness and Tardiness, Proceeding of 3rd Scientific Conference of the College of Science, University of Baghdad.564-575.

Table (1): The Performance of (CE) and (F2SE), Alg.(n – 1) algorithms for Problem (P

n	no. of ex.	(CE) Alg. Opt.Val.	(F2SE) Alg.	Alg(n – 1)	DM
3	1	143	146	148	147
	2	102	112	102	112
	3	109	118	120	118
	4	100	100	101	101
	5	116	119	118	116
4	1	0	0	1	0
	2	141	143	141	143
	3	84	87	88	88
	4	175	179	180	179
	5	141	151	151	151
5	1	33	33	35	35
	2	149	161	161	161
	3	189	206	205	205
	4	25	29	31	29
	5	89	92	94	94
6	1	169	188	190	169
	2	211	216	216	216
	3	138	146	147	147
	4	158	163	163	163
	5	352	352	352	352
7	1	129	149	149	149
	2	177	193	194	194
	3	141	156	160	156
	4	108	125	124	124
	5	344	358	358	358
8	1	61	72	73	73
	2	98	118	118	118
	3	82	106	106	107
	4	73	81	82	82
	5	121	121	121	121

This table shows (5) problems, (F2SE) algorithm give the optimal solution from (30) problems to (P), (Alg (n – 1)) algorithm gives optimal solution to (3) problems from (30) problems to (P), also DM gives optimal solution to (5) problems from (30) problems to (P).

Table (2): The Performance of (F2SE) and Alg(n – 1) algorithms for Problem (P

n	no. of ex.	(F2SE) Alg.	Time	Alg(n – 1)	Time
100	1	0	0.0300	1	1.12164
	2	0	0.0213	0	0.2705
	3	0	0.0219	0	0.6963
	4	0	0.2194	0	0.5759
	5	0	0.263	1	0.9117
200	1	0	0.3005	1	0.8652
	2	0	0.0254	1	0.5461
	3	0	0.0635	1	0.1234
	4	5	0.0532	5	0.8913
	5	5	0.0233	6	1.1505
300	1	0	0.0272	1	0.5758
	2	0	0.275	0	0.5463
	3	7	0.290	8	0.7667
	4	0	0.0280	1	0.5511
	5	0	0.0346	0	0.1967
400	1	0	0.0939	1	0.1239
	2	0	0.0332	0	0.8653
	3	0	0.0319	1	0.1566
	4	1	0.0345	0	0.0353
	5	7	0.0330	7	0.6795
500	1	5	0.0664	6	0.1575
	2	0	0.0356	1	2.1528
	3	0	0.0369	0	0.4293
	4	0	0.0405	0	0.5482
	5	1	0.0342	1	0.0351
600	1	7	0.0645	7	0.2648
	2	1	0.0345	0	2.4603
	3	0	0.0375	1	1.4604
	4	0	0.0374	1	0.2038
	5	0	0.0390	0	0.0371
700	1	0	0.0407	0	1.1898
	2	0	0.0362	0	2.2891
	3	0	0.0365	1	2.9018
	4	6	0.0412	8	1.1541
	5	9	0.0365	9	2.6384
800	1	0	0.0378	0	3.2851
	2	0	0.0400	1	4.2864
	3	1	0.0402	0	2.6177
	4	0	0.0407	1	3.9012
	5	1	0.0400	0	1.8971

This table shows (27) problems, (F2SE) algorithm give the optimal solution from (40) problems to (P). Also (Alg (n – 1)) algorithm gives optimal solution to (16) problems from (40) problems to (P) (0 is optimal solution because $E_j \geq 0$).

الحل الكفوء لمسألة الجدولة الانسيابية ذات الماكنتين لتصغير مجموع التكبير

هند فالح عبدالله
قسم الرياضيات – كلية التربية (ابن الهيثم) – جامعة بغداد
استلم البحث في:22 كانون الثاني 2012 قبل البحث في: 21 ايار 2012

الخلاصة

في هذا البحث تطرقنا الى خوارزمية جديدة (F2SE) وخوارزمية ((Alg(n – 1 لحل مسالة الجدولة الانسيابية للنتاجات (jobs) على ماكنتين والهدف هو تصغير مجموع التبكير للنتاجات. وتكون المسألة من نوع NP–hard قادتنا الى استعمال خوارزمية العد التام لايجاد الحل الامثل الى (n < 8) واستعملنا الخوارزمية (F2SE) ،وخوارزمية Alg(n), ((1 – و (DM) الى (n > 8). و وجدنا ان (F2SE) اكثر كفاية من ((Alg(n – 1 لايجاد الحل الكفوء.

الكلمـات المفتاحيـة: جدول المشغل الانسيابي، الحل التقريبي، المشغل الانسيابي للماكنتين، تصغير مجموع التبكير، مسالة الجدولة.