Information Processing Letters 111 (2011) 208–212

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Total completion time minimization in a 2-stage differentiation flowshop with fixed sequences per job type

Bertrand M.T. Lin *, F.J. Hwang

Institute of Information Management, Department of Information and Finance Management, National Chiao Tung University, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Received 27 January 2010
Received in revised form 22 September 2010
Accepted 25 November 2010
Available online 30 November 2010
Communicated by Wen-Lian Hsu

Keywords:
Design of algorithms
Scheduling
Differentiation flowshop
Fixed sequences
Dynamic programming

ABSTRACT

This paper addresses the total completion time minimization in a two-stage differentiation flowshop where the sequences of jobs per type are predetermined. The two-stage differentiation flowshop consists of a stage-1 common machine and m stage-2 parallel dedicated machines. The goal is to determine an optimal interleaved processing sequence of all jobs at the first stage. We propose an $O(m^2 \prod_{k=1}^{m} n_k^{m+1})$ dynamic programming algorithm, where n_k is the number of type-k jobs. The running time is polynomial when m is constant.

1. Introduction

In most scheduling problems, schedules can be easily derived if job/operation sequences on the machines are known. It is however non-trivial for some problems. In this paper, we consider a two-stage differentiation flowshop scheduling problem to minimize the total completion time, subject to the condition that job sequences per type are known a priori. The two-stage differentiation flowshop consists of a stage-1 common machine M_0 and m stage-2 parallel dedicated machines, $M_1, \dots, M_m$. Jobs are categorized into m types. All jobs are required to be processed on machine M_0 first, and then jobs of type l for $1 \leqslant l \leqslant m$ proceed to dedicated machine M_l for their second-stage process. Since the processing sequence of each job type is given, the goal is to find an interleaving processing sequence of all jobs on machine M_0 so as to minimize the sum of job completion times at stage 2 of all jobs. The problem under study is denoted by $F(1,m)|fixed\_seq|\sum C_j$, where $F(1,m)$ stands for a two-stage differentiation flow shop with m parallel dedicated machines at stage 2, *fixed_seq* for fixed sequences of jobs per type, and $\sum C_j$ for the total completion time minimization criterion. To solve $F(1,m)|fixed\_seq|\sum C_j$ optimally, we present a dynamic programming algorithm with a running time that is polynomial when the number of dedicated machines m is constant.

2. Literature review

Herrmann and Lee [4] first studied the $F(1,2)$ model (two job types) and showed the strong NP-hardness of three objectives, namely the makespan, the number of tardy jobs and the total completion time. An interesting problem arising from the machine configuration is to determine an optimal interleaving sequence on the stage-1 machine from fixed sequences for the two types of jobs. This interleaving problem of makespan minimization was reduced to the problem of minimizing the maximum lateness, which can be solved by Jackson's earliest due date (EDD) first rule [6] in $O(n \log n)$ time. Kyparisis and Koulamas [7] and Mosheiov and Yovel [11] proposed polynomial-time algorithms for the $F(1,m)$ problem of makespan minimization subject to the block assump-

* Corresponding author. Tel.: +886 3 5131472; fax: +886 3 5723792.
E-mail addresses: bmtlin@mail.nctu.edu.tw (B.M.T. Lin), fjhwang.iim95g@nctu.edu.tw (F.J. Hwang).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2010.11.021

tion that jobs of the same type must be processed adjacently on the stage-1 machine. With the block assumption, Mosheiov and Sarig [10] investigated the $F(1,m)$ model to minimize the weighted number of tardy jobs with a common due date and proposed a pseudo-polynomial dynamic programming algorithm to establish the ordinary NP-hardness. Cheng and Kovalyov [1] considered the $F(1,2)$ model incorporating batching decisions on the common machine, where setup times occur whenever the machine switches processing from a job of one type to a job of the other type. A polynomial-time dynamic programming algorithm for makespan minimization was presented. Cheng et al. [2] addressed a non-classical objective of minimizing the weighted sum of stage-2 machine completion times. They proved the strong NP-hardness and designed an $O(n^3)$ polynomial-time algorithm for the special case with given sequences of both types of jobs.

As aforementioned, problem $F(1,2)|\mathit{fixed\_seq}|C_{\max}$ can be solved in $O(n\log n)$ [4]. The solution approach developed by Hermann and Lee [4] actually can be further extended for the general $F(1,m)|\mathit{fixed\_seq}|C_{\max}$. Nevertheless, the objective of total completion time was not previously addressed. Due to the strong NP-hardness of the classical $F2||\sum C_j$ (equivalent to $F(1,1)||\sum C_j$), the $F(1,m)||\sum C_j$ problem is also intractable. We are interested in problem $F(1,m)|\mathit{fixed\_seq}|\sum C_j$, where job sequences on all dedicated machines are given beforehand. Our preliminary study suggests that even $F(1,2)|\mathit{fixed\_seq}|\sum C_j$ cannot be solved using the approach developed in [4] for $F(1,2)|\mathit{fixed\_seq}|C_{\max}$. In this study, we investigate the general m-machine setting, $F(1,m)|\mathit{fixed\_seq}|\sum C_j$, and propose a dynamic programming algorithm.

The assumption of fixed job sequences is justified from several aspects. Shafransky and Strusevich [13] considered the machine setting where a predetermined job sequence is retained on a specific machine in the manufacturing process owing to technological or managerial decisions. Another justification for the assumption of fixed job sequences is due to the Fist-Come-First-Served (FCFS) principle, which is regarded fair by customers [5]. Cheng et al. [3] and Lin et al. [9] solved some restricted cases, in which fixed job sequences are implied to derive lower bounds. Other studies on scheduling problems with the fixed-sequence assumption include [8] and [12].

3. Dynamic program for $F(1,m)|\mathit{fixed\_seq}|\sum C_j$

Denote $\mathcal{J}_l = \{J_{l,1},\ldots,J_{l,n_l}\}$ the set of type-l jobs, $1 \leqslant l \leqslant m$. Job $J_{l,j}$ requires a processing time $p_{l,j}$ and $q_{l,j}$ on machine M_0 and M_l, respectively. The processing sequence of jobs per type is already predetermined. Assume without loss of generality that the fixed sequence of type-l jobs is $(J_{l,1}, J_{l,2},\ldots,J_{l,n_l})$.

Let us first consider a special case where each type contains exactly one job, i.e. $n_l = 1$ for all types l. In this case, we denote p_l and q_l the processing times of the type-l job. Consider a particular sequence $\sigma = (\sigma_1,\sigma_2,\ldots,\sigma_m)$. The completion time of the j-th job is given by $\sum_{i=1}^{j} p_{\sigma_i} + q_{\sigma_j}$. The total completion time is thus given by

$$\sum_{j=1}^{m}\left(\sum_{i=1}^{j} p_{\sigma_i} + q_{\sigma_j}\right) = \sum_{j=1}^{m}\sum_{i=1}^{j} p_{\sigma_i} + \sum_{j=1}^{m} q_{\sigma_j}.$$

The second term is fixed once the instance is given. Therefore, the problem is equivalent to minimizing $\sum_{j=1}^{m}\sum_{i=1}^{j} p_{\sigma_i}$, which can be solved in $O(m\log m)$ time by the shortest processing time (SPT) first rule using p_l.

For the general case, we propose a dynamic programming algorithm in which two matrices, A and B are designed. Define matrix $\mathsf{A}_{1\times m} = [a_1, a_2, \ldots, a_m]$ with $0 \leqslant a_l \leqslant n_l$ for $1 \leqslant l \leqslant m$, where the element a_l is the number of the type-l job(s) considered. Given A, the objective is to find the optimal interleaving sequence of subsequences $(J_{1,1},\ldots,J_{1,a_1}),\ldots,(J_{m,1},\ldots,J_{m,a_m})$. In a given schedule of this problem, the last job having an idle time on its machine at stage 2 inserted in prior to its dedicated operation is called a *critical job* of its type. Matrix $\mathsf{B}_{m\times m}$ is defined with $0 \leqslant b_{l,r} \leqslant a_r$ for $1 \leqslant l, r \leqslant m$, where the element $b_{l,r}$ is defined as the number of the type-r job(s) arranged on machine M_0 before the stage-1 completion time of the critical job of type l. A schedule is associated with a state $(k,\mathsf{A},\mathsf{B})$ subject to the following conditions: (a) Subsequences $(J_{1,1},\ldots,J_{1,a_1}),\ldots,(J_{m,1},\ldots,J_{m,a_m})$ are considered; (b) Job J_{k,a_k} is the last job on machine M_0 and $a_k \neq 0$; (c) Job $J_{l,b_{l,l}}$ is the critical job of type l, $1 \leqslant l \leqslant m$, and its completion on machine M_0 is preceded by jobs of $\bigcup_{r=1}^{m}\{J_{r,1},\ldots,J_{r,b_{l,r}}\}$; (d) If $a_l \neq 0$, $1 \leqslant b_{l,l} \leqslant a_l$; otherwise $b_{l,r} = 0$. As depicted in Fig. 1, the configuration is aimed at determining the stage-2 completion time of the job scheduled last on M_0. With the parameter specifications, the completion time of job J_{k,a_k} is calculated as $C_{k,a_k} = \sum_{r=1}^{m}\sum_{j=1;b_{k,r}\neq 0}^{b_{k,r}} p_{r,j} + \sum_{j=b_{k,k}}^{a_k} q_{k,j}$. Consider an instance with $m = 3$: $(p_{1,1}, p_{1,2}, p_{1,3}) = (2,5,4)$, $(q_{1,1}, q_{1,2}, q_{1,3}) = (8,4,7)$, $(p_{2,1}, p_{2,2}, p_{2,3}, p_{2,4}) = (4,5,5,3)$, $(q_{2,1}, q_{2,2}, q_{2,3}, q_{2,4}) = (8,4,17,3)$, $(p_{3,1}, p_{3,2}, p_{3,3}) = (3,4,6)$, $(q_{3,1}, q_{3,2}, q_{3,3}) = (8,7,4)$. A schedule of the state with $k = 2$, $\mathsf{A} = [3,4,3]$, and $\mathsf{B} = \begin{bmatrix} 3 & 2 & 0 \\ 3 & 3 & 0 \\ 3 & 3 & 1 \end{bmatrix}$ is shown in Fig. 2, and we have $C_{2,4} = \sum_{r=1}^{3}\sum_{j=1;b_{2,r}\neq 0}^{b_{2,r}} p_{r,j} + \sum_{j=b_{2,2}}^{a_2} q_{2,j} = 25 + 20 = 45$.

The corresponding recursive function $f_k(\mathsf{A},\mathsf{B})$, $1 \leqslant k \leqslant m$ is defined as the minimum total completion time among all schedules associated with the same state $(k,\mathsf{A},\mathsf{B})$. From the above definition, we give the recursive formulation of our dynamic program as follows. Note that given a matrix A, the range of possible vales of $b_{l,r}$ for $1 \leqslant l, r \leqslant m$ can be obtained.

Algorithm DP. Initialization: For all $k \in \{1,\ldots,m\}$, all matrices A, and all matrices B,

$$f_k(\mathsf{A},\mathsf{B}) = \begin{cases} 0, & \text{if } \mathsf{A} \text{ and } \mathsf{B} \text{ both are zero matrices;} \\ \infty, & \text{otherwise.} \end{cases}$$

Recursion: For $1 \leqslant k \leqslant m$, each matrix A satisfying $1 \leqslant a_k \leqslant n_k$, and $0 \leqslant a_l \leqslant n_l$ for $1 \leqslant l \neq k \leqslant m$, and each possible matrix B corresponding to A, perform the recursion by removing the last job J_{k,a_k}.

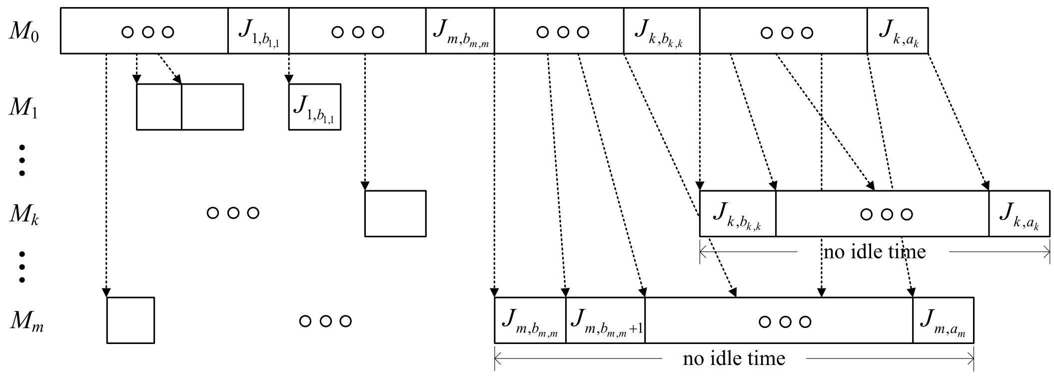

Fig. 1. Configuration of state $(k, \mathsf{A}, \mathsf{B})$.

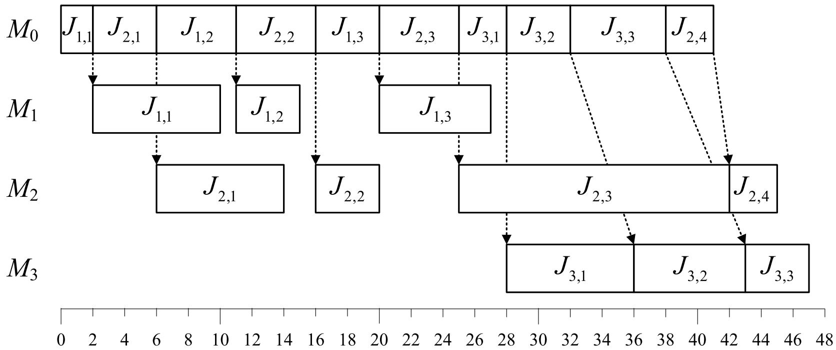

Fig. 2. A schedule of the state $(2, \mathsf{A}, \mathsf{B})$ in the instance with $m = 3$.

Define the updated matrix A' by letting $a'_k = a_k - 1$, and $a'_l = a_l$ for $1 \leqslant l \neq k \leqslant m$.

Case 1 ($b_{k,k} = a_k$):

In this case, J_{k,a_k} is the critical job of its type. After each recursive call, the critical job of type k needs to be updated. Construct the updated matrix B' by letting $b'_{l,r} = b_{l,r}$, $1 \leqslant l \neq k \leqslant m$ and $b'_{k,r}$ denote the number of the type-r job(s) arranged on machine M_0 before the stage-1 completion time of the updated critical job of type k for $1 \leqslant r \leqslant m$.

Subcase 1-1 ($b_{k,k} = a_k > 1$):

Denote the set $\mathcal{D} = \{[b'_{k,1}, b'_{k,2}, \ldots, b'_{k,m}]\colon\ 1 \leqslant b'_{k,k} < b_{k,k},\ \text{and}\ 0 \leqslant b'_{k,r} \leqslant b_{k,r}\ \text{for}\ 1 \leqslant r \neq k \leqslant m\}$. In this subcase, we have the k-th row of B', $[b'_{k,1}, b'_{k,2}, \ldots, b'_{k,m}] \in \mathcal{D}$. The configuration of the subcase is presented in Fig. 3.

$$f_k(\mathsf{A}, \mathsf{B}) = \min_{1 \leqslant l \leqslant m; \mathcal{D}} \left\{ f_l(\mathsf{A}', \mathsf{B}')\colon \sum_{r=1}^{m} \sum_{j=b'_{k,r}+1;\, b'_{k,r} \neq a'_r}^{a'_r} p_{r,j} > \sum_{j=b'_{k,k}}^{a'_k} q_{k,j} \right\} + C_{k,a_k}.$$

Subcase 1-2 ($b_{k,k} = a_k = 1$):

In this subcase, J_{k,a_k} is the unique job of its type as well. After removing J_{k,a_k} for recursion, there exists no critical job of type k, i.e. $a'_k = 0$. By virtue of the aforementioned condition (d), we have $b'_{k,r} = 0$ for $1 \leqslant r \leqslant m$.

$$f_k(\mathsf{A}, \mathsf{B}) = \min_{1 \leqslant l \leqslant m} \left\{ f_l(\mathsf{A}', \mathsf{B}') \right\} + C_{k,a_k}.$$

Case 2 ($b_{k,k} < a_k$):

For this case, J_{k,a_k} is not the critical job of its type. As illustrated in Fig. 4, we simply remove J_{k,a_k} in the recursion.

$$f_k(\mathsf{A}, \mathsf{B}) = \min_{1 \leqslant l \leqslant m} \left\{ f_l(\mathsf{A}', \mathsf{B})\colon \sum_{r=1}^{m} \sum_{j=b_{k,r}+1;\, b_{k,r} \neq a'_r}^{a'_r} p_{r,j} \leqslant \sum_{j=b_{k,k}}^{a'_k} q_{k,j} \right\} + C_{k,a_k}.$$

Goal: Let $a_l = n_l$ for $1 \leqslant l \leqslant m$. Find $\min_{1 \leqslant k \leqslant m}\{f_k(\mathsf{A}, \mathsf{B})\colon\ 0 \leqslant b_{l,r} \leqslant n_r$ and $1 \leqslant b_{l,l} \leqslant n_l$ for $1 \leqslant l, r \leqslant m, r \neq l\}$.

As for the complexity of Algorithm DP, the running times for Subcase 1-1, Subcase 1-2, Case 2, and the Goal phase are analyzed as follows. For Case 1, $b_{k,k} = a_k$ implies that $b_{k,r} = a_r$, $1 \leqslant r \leqslant m$. Hence, there are $O(m \prod_{k=1}^{m} n_k^m)$ states, each of which takes $O(m \prod_{k=1}^{m} n_k)$ time in Subcase 1-1. The running time is $O(m^2 \prod_{k=1}^{m} n_k^{m+1})$. In Subcase 1-2, $a_k = 1$ implies that $b_{l,k} = 0$, $1 \leqslant l \neq k \leqslant m$, and there are less than $O(m \prod_{k=1}^{m} n_k^m)$ states, each of which takes $O(m)$ time. The running time is thus $O(m^2 \prod_{k=1}^{m} n_k^m)$. In Case 2, the size of the state space is $O(m \prod_{k=1}^{m} n_k^{m+1})$ and the computation required for each state takes $O(m)$

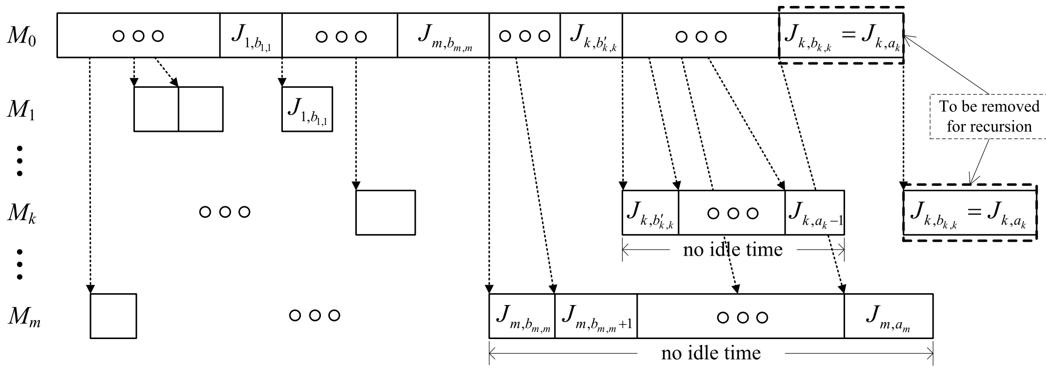

Fig. 3. Recursion for the case $b_{k,k} = a_k$.

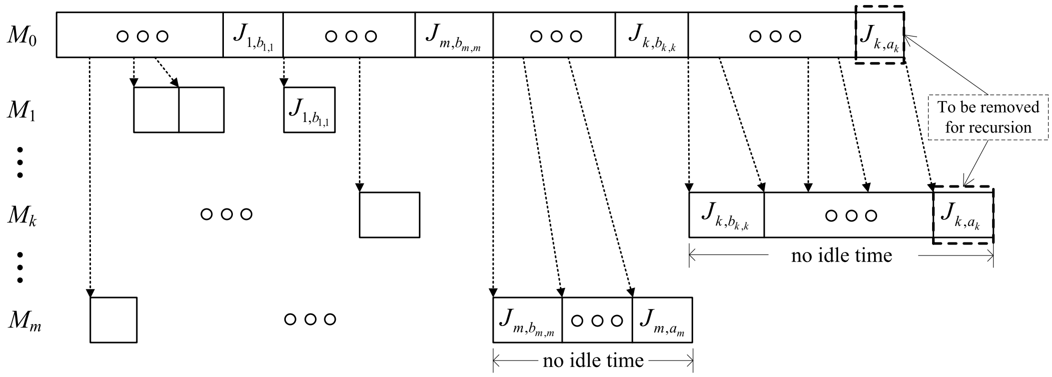

Fig. 4. Recursion for the case $b_{k,k} < a_k$.

time. It results in a total running time of $O(m^2 \prod_{k=1}^{m} n_k^{m+1})$ for the Recursion phase. When the Recursion phase is done, the Goal phase requires $O(m \prod_{k=1}^{m} n_k^{m})$ comparisons, each of which takes constant time. Therefore, the overall running time of Algorithm DP is $O(m^2 \prod_{k=1}^{m} n_k^{m+1})$, which is polynomial when m is not part of the input. For the specific case of $m = 2$, the complexity is $O(n_1^3 n_2^3)$.

4. Concluding remarks

A two-stage differentiation flowshop scheduling problem with predetermined job sequences per type for the minimization of total completion time has been addressed in this study. For the minimization of the total completion time, we designed an $O(m^2 \prod_{k=1}^{m} n_k^{m+1})$-time dynamic programming algorithm, where n_k is the number of type-k jobs. The running time is polynomial when the number of dedicated machines m is constant.

Two directions are suggested for further extensions of our research. First, since the stage-1 machine is common for all product types, in the aspect of mass customization the processing is mostly carried out in batches. It would be interesting to consider different batching modes, including max-batch (parallel-batch) and sum-batch (sequential-batch) on the common machine. Second, we can consider the reverse model that has two dedicated machines in stage one and a common machine in stage two. The differentiation flowshop and its reverse model are equivalent for makespan minimization, but exhibit different characteristics for total completion time minimization.

Acknowledgements

The authors are grateful to the reviewers for their constructive comments that have improved the presentation of the paper and clarified some technical points. This research was partially supported by the National Science Council of Taiwan under grant NSC98-2410-H-009-011-MY2.

References

[1] T.C.E. Cheng, M.Y. Kovalyov, An exact algorithm for batching and scheduling two part types in a mixed shop: A technical note, International Journal of Production Economics 55 (10) (1998) 53–56.

[2] T.C.E. Cheng, B.M.T. Lin, Y. Tian, Scheduling of a two-stage differentiation flowshop to minimize weighted sum of machine completion times, Computers Operations Research 36 (11) (2009) 3031–3040.

[3] T.C.E. Cheng, B.M.T. Lin, A. Toker, Makespan minimization in the two-machine flowshop batch scheduling problem, Naval Research Logistics 47 (2) (2000) 128–144.

[4] J.W. Herrmann, C.Y. Lee, Three-machine look-ahead scheduling problems, Research Report No. 92–93, Department of Industrial Engineering, University of Florida, FL, 1992.

[5] F.J. Hwang, M.Y. Kovalyov, B.M.T. Lin, Minimization of total completion time in flowshop scheduling subject to fixed job sequences, in: Proceedings of the 12th International Conference on Project Management and Scheduling, Tours, France, 2010, pp. 249–252.

[6] J.R. Jackson, Scheduling a production line to minimize maximum lateness, Research Report 43, Management Science Research Report, University of California, LA, 1955.
[7] G.J. Kyparisis, C. Koulamas, Flow shop and open shop scheduling with a critical machine and two operations per job, European Journal of Operational Research 127 (1) (2000) 120–125.
[8] B.M.T. Lin, T.C.E. Cheng, Two-machine flowshop scheduling with conditional deteriorating second operations, International Transactions in Operational Research 13 (2) (2006) 91–98.
[9] B.M.T. Lin, T.C.E. Cheng, A.S.C. Chou, Scheduling in an assembly-type production chain with batch transfer, Omega 35 (2) (2007) 143–151.
[10] G. Mosheiov, A. Sarig, Minimum weighted number of tardy jobs on an m-machine flowshop with a critical machine, European Journal of Operational Research 201 (2) (2010) 404–408.
[11] G. Mosheiov, U. Yovel, Comments on “Flow shop and open shop scheduling with a critical machine and two operations per job”, European Journal of Operational Research 157 (1) (2004) 257–261.
[12] C.T. Ng, M.Y. Kovalyov, Batching and scheduling in a multi-machine flow shop, Journal of Scheduling 10 (6) (2007) 353–364.
[13] Y.M. Shafransky, V.A. Strusevich, The open shop scheduling problem with a given sequence of jobs on one machine, Naval Research Logistics 45 (7) (1998) 705–731.