Computers & Industrial Engineering 115 (2018) 647–652

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

New index priority rules for no-wait flow shops

Christos Koulamas[a,*], S.S. Panwalkar[b]

[a] Department of Information Systems and Business Analytics, Florida International University, Miami, FL 33199, USA
[b] Carey Business School, Johns Hopkins University, 100 International Drive, Baltimore, MD 21202, USA

ARTICLE INFO

Keywords:
Scheduling
Flow shop
No-wait

ABSTRACT

We derive an index priority rule for an m-machine no-wait flow shop with the minimum makespan objective and specially-structured job processing times. Our rule generalizes the previously known rules for two special cases of the problem. We also derive an index priority rule for a two-machine no-wait flow shop with the minimum weighted total job completion time objective when all jobs have the same processing time on the first machine. We then show that additional index priority rules can be derived for the latter problem with other scheduling objectives. Finally, we discuss extensions to flow shops with blocking and no-wait job shops and open shops.

1. Introduction

It is well known that for certain scheduling objectives, a priority index can be computed for each job using only the parameters of that job. If an optimal sequence can be determined in $O(n\log n)$ time by sorting the jobs according to their indices, then, the addition or deletion of a job does not change the relative positions of the remaining jobs. These desirable properties have led to a theoretical analysis of index priority rules for single-machine scheduling problems. One of the earliest studies was by Rothkopf and Smith (1984); some of their results were extended by Kyparisis and Koulamas (2011), Kyparisis and Koulamas (2013) to single-machine scheduling problems with deteriorating jobs. In contrast, there is limited literature on the solvability of flow shop scheduling problems by index priority rules. The majority of index priority rules for flow shops are derived under the assumption of unlimited waiting (buffer) space between any two successive machines and are summarized by Achugbue and Chin (1982) and Monma and Rinnooy Kan (1983).

The objective of this paper is to investigate the solvability of no-wait flow shop scheduling problems using index priority rules. The motivation for our research stems from the observation that the early survey papers by Hall and Sriskandarajah (1996) and Bagchi, Gupta, and Sriskandarajah (2006) and the more recent survey paper by Allahverdi (2016) indicate that there is not a single paper focused on investigating the solvability of no-wait flow shops using index priority rules and there are very few papers (surveyed in subsequent sections) proposing an index priority rule for a no-wait flow shop. This is quite surprising given that Allahverdi (2016) surveys more than 400 papers on no-wait shops.

In the present paper, we propose solutions for two no-wait flow shop models using index priority rules. The first model with m machines and the minimum makespan objective generalizes two known special cases of the problem. Our second model has two machines and minimizes the weighted total job completion time ($\sum w_jC_j$) with arbitrary job weights and a common job processing time on the first machine; to the best of our knowledge, no index priority rule has been proposed for a flow shop with the $\sum w_jC_j$ objective and arbitrary job weights.

The paper is organized as follows: In Section 2, we present an index priority rule for a no-wait flow shop model with m machines and the minimum makespan objective. In Section 3, we present an index priority rule for a no-wait flow shop model with two machines, a common job processing time on the first machine and the $\sum w_jC_j$ objective. In Section 4, we discuss extensions to flow shops with blocking. In Section 5, we discuss extensions to no-wait job shops and open shops. The conclusions of this research along with some discussion are presented in Section 6. Numerical examples supporting our discussion are listed in the Appendix A.

2. An index priority rule for a no-wait flow shop with the minimum makespan objective

The m-machine no-wait flow shop $Fm|nwt|C_{\max}$ problem (with arbitrary job processing times and the minimum makespan objective) is strongly NP-hard when $m \geqslant 3$ (Rock, 1984). The strong NP-hardness of the $Fm|nwt|C_{\max}$ problem when $m \geqslant 3$ motivated the consideration of flow shops with specially-structured job processing times that can be solved in polynomial time.

On the other hand, the two-machine $F2|nwt|C_{\max}$ problem is solvable in $O(n\log n)$ time by the algorithm of Gilmore and Gomory (1964)

* Corresponding author.
E-mail addresses: koulamas@fiu.edu (C. Koulamas), Panwalkar@jhu.edu (S.S. Panwalkar).

https://doi.org/10.1016/j.cie.2017.12.015
Received 26 August 2017; Received in revised form 6 November 2017; Accepted 11 December 2017
Available online 13 December 2017
0360-8352/

when each job must visit both machines. The GG algorithm does not supply an index priority rule despite its $O(n\log n)$ running time. We demonstrate this (Example 1 in the Appendix A) by showing that the relative job ordering in an optimal sequence changes when a new job is added to the problem.

One shop with specially-structured processing times is the ordered flow shop introduced by Smith, Panwalkar, and Dudek (1975); in an ordered shop, the following conditions hold. Let p_{ij} denote the processing time of job $j = 1,\ldots,n$ on machine M_i, $i = 1,\ldots,m$; then,

$$\text{if } p_{ij} < p_{ik} \text{ for any two jobs } j,\ k \text{ on any machine } M_i, \text{ then } p_{ij} \leqslant p_{ik} \text{ on all machines } M_i, i = 1,\ldots,m \text{ (ordered jobs)} \tag{1}$$

$$\text{if } p_{ij} < p_{kj} \text{ for any job } j \text{ on any two machines } M_i,\ M_k, \text{ then } p_{ij} \leqslant p_{kj} \text{ for all jobs } j,\ j = 1,\ldots,n \text{ (ordered machines)} \tag{2}$$

We denote the fully-ordered problem with (1) and (2) in effect by inserting the term "*ord*" in the problem definition. If only (1) holds, we have a semi-ordered flow shop with ordered jobs (denoted as "*j-ord*") and the shortest processing (SPT) and longest processing time (LPT) sequences are well defined and identical on all machines. If only (2) holds, we have a semi-ordered flow shop with ordered machines (denoted as "*m-ord*") and the machines can be ranked from maximal to minimal.

Panwalkar and Woollam (1979) derived the first index priority rule for a special case of the no-wait flow shop by showing that the SPT sequence is optimal for the $Fm|nwt,ord|C_{\max}$ problem when the last machine is maximal and by reversibility that the LPT sequence is optimal for the $Fm|nwt,ord|C_{\max}$ problem when the first machine is maximal. Arora and Rana (1980) considered the more general $Fm|nwt,j{-}ord|C_{\max}$ problem and showed that the optimal sequence is a pyramidal (SPT-LPT) sequence with all jobs preceding the longest job sequenced in the SPT order and all jobs following the longest job sequenced in the LPT order. They proposed an $O(n^3)$ solution procedure for the problem and also showed that the LPT (SPT) sequence is optimal for the $Fm|nwt,j{-}ord|C_{\max}$ problem with a maximal first (last) machine.

Axsater (1982) utilized the findings of Arora and Rana (1980) to solve the $Fm|nwt,j{-}ord|C_{\max}$ problem in $O(n^2)$ time by dynamic programming (DP). In the DP algorithm, the jobs are indexed in the LPT order and the optimal partial sequence of jobs $\{1,\ldots,j-1\}$ is augmented to the optimal partial sequence of jobs $\{1,\ldots,j\}$ by appending the next job in the LPT order (job j) either to the right or to the left of the optimal partial sequence for jobs $\{1,\ldots,j-1\}$. The initial LPT indexing of the jobs ensures that all partial $\{1,\ldots,j\}$ sequences are "pyramidal" SPT-LPT sequences.

We now consider the $Fm|nwt,j{-}ord|C_{\max}$ model when each job has its maximal processing time either on the first or on the last machine. In this case, the jobs can be partitioned into two groups; for the jobs in group I, $p_{mj} = \max_{i=1,\ldots,m}\{p_{ij}\}$ and for the remaining jobs in group II, $p_{1j} = \max_{i=1,\ldots,m}\{p_{ij}\}$. The following proposition can be stated.

Proposition 1. The $Fm|nwt,j{-}ord|C_{\max}$ problem with each job j having its maximal processing time either on the first or on the last machine can be solved optimally by first assigning each job j the priority index

$$\operatorname{sgn}(p_{1j}-p_{mj})[M-\min(p_{1j},p_{mj})] \tag{3}$$

where $M > \min(p_{1j},p_{mj})$ for all $j = 1,\ldots,n$ and then sorting the jobs in the non-decreasing order of their indices given by (3).

Proof. The implementation of the index priority rule given by (3) yields an SPT – LPT sequence in which all group I jobs are sequenced first in the SPT order followed by all group II jobs sequenced in the LPT order. It suffices to show that this particular SPT-LPT sequence is the optimal sequence obtained by the DP algorithm of Axsater (1982).

We will show that, while implementing the DP algorithm of Axsater (1982) at a state comprising jobs $\{1,\ldots,j-1\}$, the next job j in the LPT sequence is always appended to the right if it is a group II job and is always appended to the left if it is a group I job effectively yielding the SPT-LPT sequence according to the index priority rule (3).

Suppose that job j is a group II job with $p_{1j} = \max_{i=1,\ldots,m}\{p_{ij}\}$. If j is appended to the left of the optimal sequence for jobs $\{1,\ldots,j-1\}$, the makespan will increase by at least p_{1j}. If j is appended to the right of the optimal sequence for jobs $\{1,\ldots,j-1\}$, the makespan will increase by $\Delta = \max_{r=1,\ldots,m}\{p_{rj} + \sum_{i=r+1}^{m}(p_{ij}-p_{ik})\}$ where job k is the rightmost job in the optimal sequence for jobs $\{1,\ldots,j-1\}$. Since $p_{ij} \leqslant p_{ik}$ for all $i = 1,\ldots,m$ due to the initial LPT indexing of the jobs, $\Delta \leqslant \max_{r=1,\ldots,m}\{p_{rj}\} \leqslant p_{1j}$; therefore, job j is always appended to the right. The $p_{mj} = \max_{i=1,\ldots,m}\{p_{ij}\}$ case (where job j is a group I job) is symmetric and can be proved analogously. □

The case discussed in Proposition 1 is a generalization of two cases discussed in Arora and Rana (1980). If either group I or group II is empty, Proposition 1 corresponds to the findings of their Theorems 4 and 3 respectively.

The index (3) also solves a generalization of the $F3|ord|C_{\max}$ problem (without the no-wait restriction) in which $M_1 < M_2 < M_3$ for a group of jobs (group I) and $M_1 > M_2 > M_3$ for the remaining jobs comprising group II. This problem has neither ordered machines nor ordered jobs and has not been previously identified in the literature as a solvable case of the $F3||C_{\max}$ problem. The optimality of the sequence according to index (3) follows from the findings of Burns and Rooker (1976) for the $F3||C_{\max}$ problem stating that when a sequence is optimal for all three embedded two-machine flow shop problems ($M_1 \to M_2$, $M_2 \to M_3$ and $M_1 \to M_3$ respectively), it is also optimal for the three-machine problem.

If we consider the problem addressed in Proposition 1 for the two-machine case, each job will have its largest processing time either on the first or on the second machine and the following corollary can be stated.

Corollary 1. The $F2|nwt,j{-}ord|C_{\max}$ problem can be solved optimally by first assigning each job j the priority index

$$\operatorname{sgn}(p_{1j}-p_{2j})[M-\min(p_{1j},p_{2j})] \tag{4}$$

where $M > \min(p_{1j},p_{2j})$ for all $j = 1,\ldots,n$ and then sorting the jobs in the non-decreasing order of their indices given by (4).

Corollary 1 supplies an index priority rule for the $F2|nwt,j{-}ord|C_{\max}$ problem in $O(n\log n)$ time. Since the $F2|nwt,j{-}ord|C_{\max}$ problem does not have a maximal machine, it is solvable in $O(n^2)$ time by the DP algorithm of Axsater (1982). As a result, we propose a faster algorithm for the $F2|nwt,j{-}ord|C_{\max}$ problem.

The index given by (4) coincides with the index of Johnson's (1954) rule for the $F2||C_{\max}$ problem and is also stated in Potts and Strusevich (2009). An analogous observation cannot be made when the ordered jobs are replaced by ordered machines. We can show (Example 2 in the Appendix A) that the optimal sequence for the $F2|nwt,m{-}ord,M_2{-}\max|C_{\max}$ problem with a maximal second machine does not coincide with the SPT sequence on M_1 which is the optimal sequence for the corresponding $F2|m{-}ord,M_2{-}\max|C_{\max}$ problem.

We conclude this section by discussing a proportionate flow shop model in which each job has the same processing time on all machines. Pinedo (Theorem 6.2.4, 2008) states that any SPT-LPT sequence is optimal for the proportionate flow shop with blocking where there is no waiting (buffer) space between any two successive machines. It is shown in the proof of Theorem 6.2.4 that, in any SPT-LPT sequence, no machine is blocked which implies that any SPT-LPT sequence is also optimal for the no-wait proportionate flow shop problem.

3. An index priority rule for a no-wait flow shop with the minimum weighted total job completion time objective

The list of no-wait flow shops with the total job completion time