Computers & Industrial Engineering 107 (2017) 105–108

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

On the dominance of permutation schedules for some ordered and proportionate flow shop problems

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

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

1. Introduction

It is generally agreed that Johnson (1954) pioneered research in shop scheduling with his seminal paper on the static, deterministic flow shop (see Panwalkar and Koulamas (2015) for a review of Johnson's paper). The flow shop defined by Johnson is now called the *classical* or the *pure* flow shop; see the explanation of pure flow shop in Emmons and Vairaktarakis (p. 10, 2012), or in Baker and Trietsch (Fig. 10.2, 2013). In the classical flow shop, there is a set of n jobs $j = 1, \dots, n$ (set N), all of them available at time zero; each job must be processed non-preemptively and sequentially on m machines $M_1, \dots, M_m$ (set M), with known *positive processing times* $p_{ij} > 0$, $i = 1, \dots, m$, $j = 1, \dots, n$. If jobs are processed in the same order on all machines, we have a *permutation schedule*. Permutation schedules are dominant if there is no non-permutation schedule with a better value of the objective function compared to the best permutation schedule.

Two important findings of Johnson (1954), while analyzing the minimum makespan flow shop problem, were stated as follows *"the first two machines have the same orders and the last two machines have the same orders"*. These findings have been generalized by Conway et al. (Theorems 5.1 and 5.2, 1967) and in the following two theorems (wording modified slightly) by Emmons and Vairaktarakis (p. 11, 2012).

Theorem 1. *For $Fm//any$, there exists an optimal permutation schedule on the first two machines.*

Theorem 2. *For $Fm//C_{\max}$, there exists an optimal permutation schedule on the last two machines.*

The term Fm above denotes a flow shop with m machines, "*any*" denotes any function of job completion times, i.e. all regular measures, and $C_{\max}$ denotes the minimum makespan.

Johnson (1954) also presented a problem instance for $F4//C_{\max}$ with an optimum non-permutation schedule. Conway, Maxwell, and Miller (1967, p. 82) presented a problem instance for $F3//\sum C$ with an optimum non-permutation schedule.

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

http://dx.doi.org/10.1016/j.cie.2017.03.013

The purpose of this paper is to present new flow shop models (involving ordered and proportionate flow shop) with dominant permutation schedules and examples of non-permutation optimal schedules for some other models. We will first present a summary of past work.

An *ordered flow shop* (see Smith, Panwalkar, & Dudek, 1975) assumes the following relationships among job processing times.

$p_{tj} > p_{tk}$ for $j, k \in N$ and for some $t \in M$ implies that $p_{ij} \geqslant p_{ik}, i = 1, \dots, m$;
$p_{ir} > p_{tr}$ for some $r \in N$ and for $i, t \in M$ implies that $p_{ij} \geqslant p_{tj}$, $j = 1, \dots, n$.

In an ordered problem, jobs can be numbered in the ascending order of processing times making $1, 2, \dots, n$ the shortest processing time (SPT) sequence. Also, machines can be ranked in the order of processing times. If $p_{ij} \geqslant p_{tj}$ we will simply indicate this by $M_i \geqslant M_t$ and the machine with the highest processing times will be the *maximal machine*. The ordered flow shop problems will be identified in this paper by adding the term "***ord***" to the second field of the problem definition.

A special case of the ordered flow shop is the *proportionate flow shop with unequal machine speeds* in which $p_{ij} = \frac{p_j}{s_i}$ where $s_i > 0$ is the speed of machine M_i $(i = 1, \dots, m)$. Note that $s_i \leqslant s_t$ corresponds with $M_i \geqslant M_t$. This shop is a generalization of the *proportionate flow shop with equal machine speeds* in which $p_{ij} = p_j$, that is, each job has the same processing time on all machines. A brief review of research involving ordered and proportionate problems can be found in Panwalkar, Smith, and Koulamas (2013).

Johnson (1959) showed that while a non-permutation schedule can be optimal for the $F2/l_j/C_{\max}$ problem (l_j representing an arbitrary time lag between the two operations of a job), permutation schedules are dominant under certain specified values l_j. Chin and Tsai (1981) showed (in Theorem 3) that all permutation schedules are optimal for the $Fm/p_{ij} = p_j/C_{\max}$ problem. They also showed that permutation schedules are dominant for a restricted class of $Fm/p_{ij} = \frac{p_j}{s_i}/C_{\max}$ problems with $m \geqslant 4$ and either $s_1 \neq s_2 = \dots = s_m$ or $s_1 = \dots = s_{m-1} \neq s_m$. Shakhlevich, Hoogeveen, and Pinedo (1998) showed that permutation schedules are dominant (with SPT schedule optimal, see Pinedo, 2016 p. 168) for the $Fm/p_{ij} = p_j/\sum C_j$ problem.

Choi, Yoon, and Chung (2007) proved the dominance of permutation schedules for the $Fm/p_{ij} = \frac{p_j}{s_i}/C_{\max}$ problem when $m \geqslant 4$ and $s_k < s_1 = \cdots = s_{k-1} = s_{k+1} = \cdots = s_m$ for some machine M_k, $1 \leqslant k \leqslant m$. These results indicate that permutation schedules are dominant when all machines have the same speed except one. Choi et al. also utilized a problem instance with the middle machine having three times the speed of the first and the last machine to show the optimality on a non-permutation schedule for the $F4/p_{ij} = \frac{p_j}{s_i}/C_{\max}$ problem.

It should be noted many other papers, not directly related to the present work but dealing with non-permutation schedules can be divided into two categories. (1) Flow shops with zero processing times or missing operations (see for example, Potts, Shmoys, & Williamson 1991). (2) Development of heuristics to generate non-permutation schedules; see for example, Liao and Huang (2010), Mehravaran and Logendran (2013), and Benavides and Ritt (2016).

The rest of the paper is organized as follows. In Section 2, we will discuss models involving the $C_{\max}$ criterion and in Section 3 we focus on the $\sum C_j$ criterion. The conclusions of this research are summarized in Section 4.

2. Permutation vs. non-permutation schedules: new results for the $C_{\max}$ problems

Our new results for the makespan objective can be summarized as follows.

Lemma 1. *The SPT schedule is optimal (hence dominant) for the* $Fm/ord/C_{\max}$ *problem if* $M_1 \leqslant M_2 \ldots \leqslant M_{m-1} \leqslant M_m$.

Lemma 2. *An SPT-LPT schedule is optimal (hence dominant) for the* $Fm/ord/C_{\max}$ *problem if* $M_1 \ldots \leqslant M_{k-1} \leqslant M_k \geqslant M_{k+1} \geqslant \cdots \geqslant M_m$.

Lemma 3. *A non-permutation schedule can be optimal for the* $F4/p_{ij} = \frac{p_j}{s_i}/C_{\max}$ *problem when the middle machine speeds differ from the speeds of the first and the last machines by an arbitrarily small amount.*

To prove the above lemmas we will use the following results. Smith et al. (1975) and Smith, Panwalkar, and Dudek (1976) assumed that only permutation schedules are allowed and showed that the SPT (LPT) sequence is the optimal for the $Fm/ord/C_{\max}$ problem when the last (first) machine is maximal. They also showed that when the maximal machine is not the first or the last machine, an SPT-LPT schedule is optimal.

In the $Fm/ord/C_{\max}$ problem there is at least one "critical path" that passes through p_{kn} (the processing time of the largest job n on the maximal machine M_k). This applies only to permutation schedules; in a non-permutation schedule, no critical path may pass through p_{kn}.

Proof of Lemma 1. Consider any pair of machines M_i and M_j with $1 \leqslant i \leqslant j \leqslant m$. Since $M_i \leqslant M_j$, the SPT schedule is optimal for the two-machine problem comprising machines M_i and M_j. Since this is true for all machine pairs, Lemma 1 follows. □

Proof of Lemma 2. We use schematics for a 6 × 6 problem to facilitate the presentation of the proof. Consider an arbitrary non-permutation schedule $\sigma = \{\sigma_1, \ldots, \sigma_i, \ldots, \sigma_m\}$ (depicted on Fig. 1) with makespan value $C^{\sigma}_{\max}$ where σ_i denotes the sequence on machine M_i. Observe that $\sigma_i = \alpha_i \cup \{n\} \cup \beta_i$ for all $i = 1, \ldots, m$ where one of the job subsets α_i, β_i may be empty. Since M_k is the maximal machine, it has the largest processing time for every job. Using the sequence σ_k on all machines, we build the permutation schedule π on all machines (depicted in Fig. 2) with makespan value $C^{\pi}_{\max}$. □

	3	4	6	5	4	5	1	6	2	3	2	6	1	Sequence
M1	x	x	x	x			x		x					3,4,6,5,1,2
M2	x	x	x	x			x		x					3,4,6,5,1,2
M3				x	x		x	x	x	x				5,4,1,6,2,3
M4					x	x			x	x		x	x	4,5,2,3,6,1
M5					x	x				x	x	x	x	4,5,3,2,6,1
M6					x	x				x	x	x	x	4,5,3,2,6,1

Fig. 1. A non-permutation schedule σ.

For the non-permutation schedule σ, let $R^{\sigma}_{k,n}$ be the length of the longest path passing through p_{kn} (depicted by dark cells in Fig. 1) with $R^{\sigma}_{k,n} \leqslant C^{\sigma}_{\max}$. It is known that a critical path passes through p_{kn} for the permutation schedule π (depicted by dark cells in Fig. 2). This facilitates the comparison of σ and π outlined next. Consider the two sub-problems P1, P2 in schedule π defined as follows.

- P1 is defined on machines $M_1, \ldots, M_k$ and contains all jobs $j \in \{\alpha_k, n\}$.
- P2 is defined on machines $M_k, \ldots, M_m$ and contains all jobs $j \in \{n, \beta_k\}$.

Now consider the two sub-problems P3, P4 in schedule σ defined as follows:

- P3 is defined on machines $M_1, \ldots, M_k$ and contains all jobs $j \in \{\alpha_1 \cup \cdots \cup \alpha_{k-1} \cup \alpha_k, n\}$.
- P4 is defined on machines $M_k, \ldots, M_m$ and contains all jobs $j \in \{n, \beta_k \cup \beta_{k+1} \cup \cdots \cup \beta_m\}$.

The sub-problems P1 and P2 are depicted in Fig. 2 with thick-bordered rectangles and the sub-problems P3 and P4 are depicted in Fig. 1 also with thick-bordered rectangles.

Since $P1 \subseteq P3$ and $P2 \subseteq P4$, $C_{\max}(P1) \leqslant C_{\max}(P3)$ and $C_{\max}(P2) \leqslant C_{\max}(P4)$.

By construction, p_{kn} is the first element of $C_{\max}(P2)$ and $C_{\max}(P4)$; it is also the last element of $C_{\max}(P1)$ and $C_{\max}(P3)$. Since

	5	4	1	6	2	3
M1	x	x	x	x	x	x
M2	x	x	x	x	x	x
M3	x	x	x	x	x	x
M4	x	x	x	x	x	x
M5	x	x	x	x	x	x
M6	x	x	x	x	x	x

Fig. 2. A permutation schedule π.