HAL
open science

FPTASes for Scheduling a Distributed Flow-shop of Maximizing Total Early Work (awarded as the best paper)

Alexandre Dolgui, Mikhail Kovalyov, Bertrand M.T. Lin

To cite this version:

Alexandre Dolgui, Mikhail Kovalyov, Bertrand M.T. Lin. FPTASes for Scheduling a Distributed Flow-shop of Maximizing Total Early Work (awarded as the best paper). CMCT2022: 39th Workshop on Combinatorial Mathematics and Computation Theory, Taiwan, Jun 2022, Taipei, Taiwan. hal-03982287

HAL Id: hal-03982287

https://hal.science/hal-03982287v1

Submitted on 8 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FPTASes for scheduling a distributed flow-shop of maximizing total early work

Alexandre Dolgui[1], Mikhail Y. Kovalyov[2], Bertrand M.T. Lin[3]
[1]IMT Atlantique, LS2N-CNRS, France
[2]United Institute of Informatics Problems, National Academy of Sciences of Belarus, Belarus
[3]Institute of Information Management, National Yang Ming Chiao Tung University, Taiwan
alexandre.dolgui@imt-atlantique.fr; kovalyov_my@newman.bas-net.by; bmtlin@nycu.edu.tw

Abstract

The problem of maximizing total early work in a two-machine flow-shop is known to be NP-hard, for which an $O(n^2d^2)$ dynamic program and an $O(\frac{n^4}{\varepsilon^2})$ FPTAS exist in the literature. In this paper, we present new optimality properties, which pave the base for an $O(nd^2)$ time dynamic programming algorithm and an $O\left(\frac{n^3}{\varepsilon^2}\right)$ time FPTAS. We further generalize the problem to a distributed flow shop consisting of m parallel two-machine flow-shops. We develop an $O(mn(2d)^{3m})$ time dynamic programming algorithm and an $O\left(mn\left(\frac{4n}{\varepsilon}\right)^{3m}\right)$ time FPTAS.

1 Introduction

The most general problem studied in this paper can be formulated as follows. There are n jobs to be scheduled for processing in m parallel identical two-machine flow-shops. The jobs have a common due date d. Each flow-shop i consists of two machines $(1, i)$ and $(2, i)$. Any job j can be completely processed by any flow-shop. In flow-shop i, job j must be processed first on the machine $(1, i)$ during p_{1j} time units and then on the machine $(2, i)$ during p_{2j} time units. The values d, p_{1j} and p_{2j}, $j = 1, \ldots, n$, are positive integer numbers. No machine can process more than one job at the same time, no job can be processed by two machines at the same time, and processing of any job on any machine cannot be preempted and resumed at a later time.

Given a job schedule, let S_{hij} and C_{hij} denote start and completion times, respectively, of job j on machine (h, i). We have $C_{hij} = S_{hij} + p_{hj}$. The completion time of job j is defined as $C_j = \max_{1\le i\le m}\{C_{2ij}\}$, and the schedule *makespan* is defined as $C_{\max} = \max_{1\le j\le n}\{C_j\}$. Note that C_{hij} and S_{hij} are defined only if job j is assigned to the i-th flow-shop. The early work of machine (h, i) on job j is defined as $X_{hij} = \min\{p_{hj}, \max\{0, d - S_{hij}\}\}$, and the total early work on all jobs is defined as $X = \sum_{h\in\{1,2\}}\sum_{i=1}^{m}\sum_{j=1}^{n} X_{hij}$. The objective is to maximize the total early work.

We denote this problem as $mF2|d_j = d|\max\{X\}$ and the problem with $m = 1$ as $F2|d_j = d|\max\{X\}$. If there is a single flow-shop, that is, if $m = 1$, then we simplify the three-index notation by skipping the middle index i in it. The optimal objective function value is denoted as X^*.

2 Literature review

Scheduling a two-machine flow-shop has been a popular research topic since the seminal paper of Johnson [13] in 1954 until the present time, see, for example, the most recent publications of Gmys et al. [11], Levorato et al. [17], Gilenson and Shabtay [9] and Zinder et al. [29]. The main application area for these studies is the short-term planning of industrial and service operations in a sequential two-stage system. We consider the common due date case, which often happens if the operations are fulfilled on the products that are delivered together at a given time or that have to be ideally completed at a given time due to the technological or economical reasons.

The scheduling objective of the total early work maximization was introduced by Sterna and Czerniachowska [28] in 2017, see also a survey of Sterna [27]. This objective is closely related, and, in the sense of optimal solutions, equivalent to the late work minimization, see Chen et al. [6] for the relevant discussions. For the problem

$mF2|d_j = d|\max\{X\}$, the late work of machine (h,i) on job j is defined as $p_{hj} - X_{hij}$. The late work objective was introduced by Błazewicz [3] in 1984. The applicability of both objectives was justified by the need of minimizing the information loss in communication systems (Błazewicz [3]), timely implementation of crop harvesting works (Potts and Van Wassenhove [24]), manufacturing with assignable due dates (Qi et al. [25], Kovalev et al. [15]) and training of airline pilots (Qi et al. [25]). Lin et al. [19] proved that the problem $F2|d_j = d|\max\{X\}$ is NP-hard. Błazewicz et al. [4] developed an $O(n^2d^4)$ time dynamic programming algorithm. For the un-weighted case, Chen et al. [5] proposed an $O(n^2d^2)$ time algorithm and proposed an $O\left(\frac{n^4}{\varepsilon^2}\right)$ time FPTAS.

The machine environment in the problem $mF2|d_j = d|\max\{X\}$ is known as a distributed flow-shop. This processing system has recently become popular in scheduling research, see, for example, Sevastyanov et al. [26], Naderi and Ruiz [21, 22], Ruiz et al. [23], and Ali et al. [1]. Distributed (cloud) manufacturing is considered as an application area for these studies. A state-of-the-art of scheduling research in cloud manufacturing is provided by Liu et al. [18]. Fot the problem $mF2||C_{\max}$, which differs from $mF2|d_j = d|\max\{X\}$ in that the due date is not given and the makespan $C_{\max}$ has to be minimized instead of maximizing the total early work, are given by Kovalyov [16] (see also [7]), and independently by Dong et al. [8].

3 Definitions and problem properties

We start this section with several useful definitions.

Early and non-early jobs: For a given solution of the problem $F2|d_j = d|\max\{X\}$, job j is called early if $S_{2j} \le d-1$ and it is called non-early if $S_{2j} \ge d$.

Property 1 *Maximization of the total early work,* $\max\{X\}$*, is a regular maximization criterion of job completion times on the machines.*

Property 2 *There exists an optimal solution of the problem* $F2|d_j = d|\max\{X\}$*, which is a non-delay permutation schedule.*

Corollary 1 *There exists an optimal solution of the problem* $F2|d_j = d|\max\{X\}$*, in which machine 1 does not stand idle in the time interval* $[0, \sum_{j=1}^n p_{1j}]$.

Corollary 2 *There exists an optimal solution of the problem* $F2|d_j = d|\max\{X\}$*, for which contribution of machine 1 to the objective function is equal to* $X_1^* = \min\{\sum_{j=1}^n p_{1j}, d\}$.

Property 3 *There exists an optimal solution of problem* $F2|d_j = d|\max\{X\}$*, which is a non-delay permutation schedule (hence, satisfying Corollary 2), and it is a Johnson's permutation schedule for the set of early jobs.*

Consider a schedule for the problem $F2|d_j = d|\max\{X\}$ satisfying Property 3. Denote by E and NE the sets of early and non-early jobs, respectively.

Property 4 *Contribution of non-early jobs to the total early work on machine 2 is equal to zero, and their contribution to the total early work on machine 1 is equal to*

$$\max\left\{0, \min\left\{d - \sum_{j\in E} p_{1j}, \sum_{j\in NE} p_{1j}\right\}\right\}. \qquad (1)$$

Corollary 3 *There exists an optimal solution of the problem* $F2|d_j = d|\max\{X\}$*, in which, for each flow-shop i, early jobs are sequenced in Johnson's order and followed by non-early jobs in any order, for example, in Johnson's order.*

Denote by C^* the makespan value of a Johnson's permutation of all jobs for the problem $F2|d_j = d|\max\{X\}$.

Property 5 *If* $\sum_{j=1}^n p_{1j} \le d$*, then Johnson's permutation of all jobs is optimal for the problem* $F2|d_j = d|\max\{X\}$*, and the maximum total early work is equal to* $\sum_{j=1}^n (p_{1j}+p_{2j}) - \max\{0, C^* - d\}$.

Consider an arbitrary instance I of the problem $F2|d_j = d|\max\{X\}$ and consider an instance of this problem obtained from I by re-setting $p_{1j} := \min\{d-1, p_{1j}\}$ and $p_{2j} := \min\{d-1, p_{2j}\}$, $j = 1, \ldots, n$, which we denote as $I_{p\le d-1}$. Denote optimal objective function values of these instances as $X^*(I)$ and $X^*(I_{p\le d-1})$, respectively.

Property 6 *Instances* I *and* $I_{p\le d-1}$ *are equivalent in the sense that an optimal solution of one instance is an optimal solution of the other instance and* $X^*(I) = X^*(I_{p\le d-1})$.

Based on Property 6, assume without loss of generality that $p_{1j} \le d-1$ and $p_{2j} \le d-1$,

$j = 1, \ldots, n$, in the problem $F2|d_j = d|\max\{X\}$. We continue with properties of the general problem $mF2|d_j = d|\max\{X\}$. We call a job as a contributing one if it starts by $d-1$ on either machine. Otherwise, a job is called as a non-contributing one.

It follows from the above definition that all early jobs are contributing ones. A non-early job can be either contributing or non-contributing.

Property 7 *There exists an optimal solution of the problem $mF2|d_j = d|\max\{X\}$, in which, for each flow-shop i, early jobs are sequenced in the Johnson's order, and they are followed by non-early jobs in any order, for example, in the Johnson's order.*

Consider an arbitrary instance $I^{(m)}$ of the problem $mF2|d_j = d|\max\{X\}$ and consider instance of this problem obtained from $I^{(m)}$ by re-setting $p_{1j} := \min\{d-1, p_{1j}\}$ and $p_{2j} := \min\{d-1, p_{2j}\}$, $j = 1, \ldots, n$, which we denote as $I^{(m)}_{p \le d-1}$. Denote optimal objective function values of the two instances as $X^*(I^{(m)})$ and $X^*(I^{(m)}_{p \le d-1})$, respectively.

Property 8 *Instances $I^{(m)}$ and $I^{(m)}_{p \le d-1}$ are equivalent in the sense that an optimal solution of one of these instances is an optimal solution of the other instance, and $X^*(I^{(m)}) = X^*(I^{(m)}_{p \le d-1})$.*

Assume that $n \ge m+1$ because otherwise assigning a single job to each flow-shop constitutes an optimal schedule. Besides, based on Property 8, assume without loss of generality that $p_{1j} \le d-1$ and $p_{2j} \le d-1$, $j = 1, \ldots, n$, in both problems $F2|d_j = d|\max\{X\}$ and $mF2|d_j = d|\max\{X\}$. These assumptions imply that there exists at least one early job in each flow-shop in any non-delay schedule for any of these problems.

4 Problem $F2|d_j = d|\max\{X\}$

In this section, we present a dynamic programming algorithm, denoted as DP($m = 1$), for the non-trivial case of the problem $F2|d_j = d|\max\{X\}$, in which $\sum_{j=1}^{n} p_{1j} \ge d+1$, $p_{1j} \le d-1$ and $p_{2j} \le d-1$, $j = 1, \ldots, n$.

Re-number jobs $1, \ldots, n$ according to a Johnson's permutation. In our dynamic programming algorithm DP($m = 1$) for the problem $F2|d_j = d|\max\{X\}$, partial non-delay permutation schedules of early jobs are constructed by considering jobs in the order $1, \ldots, n$, defining a current job as early or non-early, and adding an early job to the end of the sequence of early jobs. Non-early jobs follow early jobs in any order.

Each partial schedule is associated with a *state* (j, A, B), where j is the last considered job, and A and B are the completion times of the last early job on machines 1 and 2, respectively. Function $X_2(j, A, B)$ is recursively calculated for each state (j, A, B), which is the maximum total early work on early jobs on machine 2 in any partial schedule in this state, which satisfies Property 3. A partial schedule in the state (j, A, B), for which the value of $X_2(j, A, B)$ is attained, *dominates* all other partial schedules in this state.

Denote $p_{h,\min} = \min_{1 \le j \le n}\{p_{hj}\}$ and $p_{h,\max} = \max_{1 \le j \le n}\{p_{hj}\}$, $h = 1, 2$. Re-call that $p_{2,\max} \le d-1$ by Property 6. The initialization is $X_2(0,0,0) = 0$ and $X_2(0, A, B) = -\infty$ for $(A, B) \ne (0,0)$. We assume that an artificial schedule is associated with the state $(0,0,0)$, which consists of a single artificial early job 0 with processing times $p_{10} = p_{20} = 0$. The recursion for $j = 1, \ldots, n$, $A = 0, 1, \ldots, d-1$, and $B = A + p_{1,\min}, A + p_{1,\min} + 1, \ldots, d - 1 + p_{2,\max}$ is $X_2(j, A, B) = \max\{z_1, z_2, z_3\}$, where
Case 1, $A \le B \le d-1$: $z_1 = X_2(j-1, A, B)$;
Case 2, $A = B - p_{2j}$, $p_{1j} \le A \le d-1$: $z_2 = \max_{A - p_{1j} + p_{2,\min} \le B' \le A}\{X_2(j-1, A - p_{1j}, B') + p_{2j} - \max\{0, B-d\}\}$;
Case 3, $p_{1j} \le A < B - p_{2j} \le d-1$: $z_3 = X_2(j-1, A - p_{1j}, B - p_{2j}) + p_{2j} - \max\{0, B-d\}$.

The maximum total early work on machine 2 is equal to $X_2^* = \max\Big\{X_2(n, A, B) \mid A = p_{1,\min}, p_{1,\min}+1, \ldots, d-1,\ B = A + p_{2,\min}, A + p_{2,\min} + 1, \ldots, d - 1 + p_{2,\max}\Big\}$, and the maximum total early work (on both machines) is equal to $X^* = X_1^* + X_2^* = d + X_2^*$, because $X_1^* = \min\{\sum_{j=1}^{n} p_{1j}, d\}$ and $\sum_{j=1}^{n} p_{1j} \ge d+1$. The corresponding optimal schedule can be determined by tracing back the solutions of the recursive equation. The running time of the algorithm DP($m = 1$) can be evaluated as $O(nd(d + p_{2,\max} - p_{1,\min})) = O(nd^2)$.

5 Problem $mF2|d_j = d|\max\{X\}$

In this section, we present a dynamic programming algorithm, denoted as DP(m), for the problem $mF2|d_j = d|\max\{X\}$. Assume that the jobs are re-numbered $1, \ldots, n$ according to a Johnson's permutation, $p_{1j} \le d-1$, $p_{2j} \le d-1$, $j = 1, \ldots, n$,

and $n \geq m+1$. It follows from definitions and properties in Section 3 that there exists an optimal solution, in which the jobs in each flow-shop are sequenced in the following order: early jobs appear first in Johnson's order, they are followed by non-early contributing jobs in any order, and non-contributing jobs appear last in any order.

We describe DP(m) as an algorithm to find a longest path in a special multi-layered directed acyclic graph. Denote this graph as $G(V,U)$, where V is the set of nodes and U is the set of arcs. A node in this graph is represented by a $(3m+1)$-tuple $(j, A_1, \ldots, A_m, B_1, \ldots, B_m, D_1, \ldots, D_m)$, which is associated with a partial non-delay permutation schedule of jobs $1, \ldots, j$, and plays the role of a state in the dynamic programming algorithm. Values A_i and B_i are completion times of the last early job on machines $(1,i)$ and $(2,i)$, respectively, and D_i is the total processing time of contributing non-early jobs on machine $(1,i)$, $i = 1, \ldots, m$. Based on definitions of the job types and properties in Section 3, we require $A_i \leq d-1$, $A_i \leq B_i$, $B_i \leq d-1+p_{2,\max}-p_{1,\min}$ and $D_i \leq d-1+p_{1,\max}-p_{1,\min}$, $i = 1, \ldots, m$.

Graph $G(V,U)$ is recursively constructed as follows. Layer 0 consists of a single $(3m+1)$-component node $E = (0, \ldots, 0)$, which represents an artificial empty schedule with $A_i = B_i = D_i = 0$, $i = 1, \ldots, m$. Any arc goes from a node of a layer $j-1$ to a node of the layer j, $j = 1, \ldots, n$. A node $P = (j-1, A_1, \ldots, A_m, B_1, \ldots, B_m, D_1, \ldots, D_m)$ of the layer $j-1$ is linked to the following nodes of layer j: (a) at most m nodes $Q^{(i)}_{early} = (j, A^{(i)}_1, \ldots, A^{(i)}_m, B^{(i)}_1, \ldots, B^{(i)}_m, D_1, \ldots, D_m)$, $i \in \{1, \ldots, m\}$, (2) at most m nodes $Q^{(i)}_{contr} = (j, A_1, \ldots, A_m, B_1, \ldots, B_m, D^{(i)}_1, \ldots, D^{(i)}_m)$, $i \in \{1, \ldots, m\}$, and (3) one node $P_{non} = (j-1, A_1, \ldots, A_{m-1}, B_1, \ldots, B_m, D_1, \ldots, D_m)$. A-components and B-components of the node $Q^{(i)}_{early}$, $1 \leq i \leq m$, are defined such that $A^{(i)}_h = A_h$, $B^{(i)}_h = B_h$ for $h \neq i$, and $A^{(i)}_i = A_i + p_{1j}$, $B^{(i)}_i = \max\{A^{(i)}_i + p_{1j}, B_i\} + p_{2j}$. An arc $(P, Q^{(i)}_{early})$ is introduced if and only if $\max\{A^{(i)}_i + p_{1j}, B_i\} \leq d-1$. This arc is associated with an assignment of job j to the flow-shop i as an early job, $i = 1, \ldots, m$. D-components of the node $Q^{(i)}_{contr}$, $1 \leq i \leq m$, are defined such that $D^{(i)}_h = D_h$ for $h \neq i$ and $D^{(i)}_i = D_i + p_{1j}$. An arc $(P, Q^{(i)}_{contr})$ is introduced if and only if $D^{(i)}_i + p_{1j} \leq d-1+p_{1,\max}-p_{1,\min}$. This arc is associated with an assignment of job j to the flow-shop i as a contributing non-early job, $i = 1, \ldots, m$. Arc (P, P_{non}) is always introduced. It is associated with the decision that job j is a non-contributing job. The length of an arc (P,Q) is denoted as $l(P,Q)$. If $Q = Q^{(i)}_{early}$, then it represents contribution of the early job j to the total early work and it is calculated as $l(P, Q^{(i)}_{early}) = p_{1j} + \min\{p_{2j}, \max\{0, d-(B^{(i)}_i - p_{2j})\}\}$. If $Q \in \{Q^{(i)}_{contr}, P_{non}\}$, then we set $l(P,Q) = 0$, which means that we postpone calculation of the contribution of the non-early job j to the total early work until all jobs are scheduled.

In the graph, a path in the graph $G(V,U)$ from the node E to a node $(j, A_1, \ldots, A_m, B_1, \ldots, B_m, D_1, \ldots, D_m)$ with the maximum length determines an optimal solution to a sub-problem of problem $mF2|d_j = d|\max\{X\}$, in which jobs $1, \ldots, j$ are scheduled, A_i and B_i are the completion times of the last early job on machines $(1,i)$ and $(2,i)$, respectively, and D_i is the total processing time of contributing non-early jobs on machine $(1,i)$, $i = 1, \ldots, m$. The total early work of this solution is equal to the length of this path plus $\sum_{i=1}^{m} \max\{0, \min\{d-A_i, D_i\}\}$.

Denote by L_n the set of nodes of layer n in the graph $G(V,U)$. Our dynamic programming algorithm DP(m) is an algorithm that finds a maximum length path from the start node E to each node $Q \in L_n$. Its running time can be evaluated as $O(|U|) = O(m|V|) = O(mnd^m(d+p_{2,\max}-p_{1,\min})^m(d-1+p_{1,\max}-p_{1,\min})^m) = O(mn(2d)^{3m})$. Taking into account (1), an optimal solution to the problem $mF2|d_j = d|\max\{X\}$ is determined by a path with the value $\max\Big\{l(E,Q) + \sum_{i=1}^{m} \max\{0, \min\{d-A_i, D_i\}\} \mid Q = (n, A_1, \ldots, A_m, B_1, \ldots, B_m, D_1, \ldots, D_m)\Big\}$.

6 FPTASes

Assume that a lower bound L is known. For a given relative error ε, $0 < \varepsilon < 1$, calculate scaling factor $\delta = \frac{\varepsilon L}{4n}$. For any instance I^* of $F2|d_j = d|\max\{X\}$ and $mF2|d_j = d|\max\{X\}$, consider a modified instance I^0, in which original job processing times p_{1j} and p_{2j}, $j = 1, \ldots, n$, and the due date d are replaced by the rounded job processing times $p^0_{1j} = \delta\left\lfloor \frac{p_{1j}}{\delta} \right\rfloor$, $p^0_{2j} = \delta\left\lfloor \frac{p_{2j}}{\delta} \right\rfloor$, $j = 1, \ldots, n$, and the due date $d^0 = \delta\left\lfloor \frac{d}{\delta} \right\rfloor$, respectively.

Lemma 1 *An optimal solution of the modified instance I^0 is a $(1-\varepsilon)$-approximation solution for the original instance I^*.*

Denote by $X(S,I)$ the total early work determined by a solution $S \in \{T,O\}$ for an instance $I \in \{I^*, I^0\}$. Statement of the lemma follows from the following chain of relations: $X(O,I^*) \geq X(O,I^0) - n\delta \geq X(T,I^0) - n\delta \geq$ $\sum_{i=1}^{m} \min\left\{\delta\left\lfloor\frac{d}{\delta}\right\rfloor, \delta \sum_{j\in T_A^{(i)}} \left\lfloor\frac{p_{1j}}{\delta}\right\rfloor\right\}+$ $\sum_{i=1}^{m} \min\left\{\delta\left\lfloor\frac{d}{\delta}\right\rfloor, \delta \sum_{j\in T_B^{(i)}} \left\lfloor\frac{p_{2j}}{\delta}\right\rfloor\right\} - n\delta \geq$ $\sum_{i=1}^{m} \min\left\{d-\delta|T_A^{(i)}|, \delta \sum_{j\in T_A^{(i)}} \left\lfloor\frac{p_{1j}}{\delta}\right\rfloor\right\}+$ $\sum_{i=1}^{m} \min\left\{d-\delta|T_B^{(i)}|, \delta \sum_{j\in T_B^{(i)}} \left\lfloor\frac{p_{2j}}{\delta}\right\rfloor\right\} - n\delta =$ $\sum_{i=1}^{m} \min\left\{d, \delta \sum_{j\in T_A^{(i)}} (\left\lfloor\frac{p_{1j}}{\delta}\right\rfloor + 1)\right\}+$ $\sum_{i=1}^{m} \min\left\{d, \delta \sum_{j\in T_B^{(i)}} (\left\lfloor\frac{p_{2j}}{\delta}\right\rfloor + 1)\right\} - \delta\sum_{i=1}^{m}(|T_A^{(i)}| +$ $|T_B^{(i)}|) - n\delta \geq$ $\sum_{i=1}^{m} \min\left\{d, \delta \sum_{j\in T_A^{(i)}} (\left\lfloor\frac{p_{1j}}{\delta}\right\rfloor + 1)\right\}+$ $\sum_{i=1}^{m} \min\left\{d, \delta \sum_{j\in T_B^{(i)}} (\left\lfloor\frac{p_{2j}}{\delta}\right\rfloor + 1)\right\} -$ $3n\delta \geq X(T,I^*) - 4n\delta = X(T,I^*) - \varepsilon L \geq$ $X(T,I^*)(1-\varepsilon)$.

For $F2|d_j = d|\max\{X\}$, relation $d \leq X^*$ is satisfied, and we can set $L = d$. For $mF2|d_j = d|\max\{X\}$, consider a feasible solution in which all jobs are processed by a single flow-shop according to Johnson's order. If the minimum makespan, C^*, of this solution satisfies $C^* \leq d$, then this solution is optimal for problem $mF2|d_j = d|\max\{X\}$ with the total early work $X^* = \sum_{i=1}^{n}(p_{1j} + p_{2j})$. Consider the non-trivial case, in which $C^* \geq d+1$, implying that the total early work of the Johnson's solution, X^0, satisfies $X^0 \geq d$. In this case, since this solution is feasible for the general problem $mF2|d_j = d|\max\{X\}$, we have $d \leq X^0 \leq X^*$. Therefore, we can set $L = d$ for the problem $mF2|d_j = d|\max\{X\}$.

Our FPTASes $\{G_\varepsilon\}$ and $\{H_\varepsilon\}$ for the problems $F2|d_j = d|\max\{X\}$ and $mF2|d_j = d|\max\{X\}$, respectively, have running times $O\left(\frac{n^3}{\varepsilon^2}\right)$ and $O\left(mn\left(\frac{4n}{\varepsilon}\right)^{3m}\right)$, respectively.

7 Conclusions and suggestions for future research

In this paper, we investigated a flow-shop scheduling problem of maximizing total early work. Given a two-machine flow-shop, we presented an $O(nd^2)$ time dynamic programming algorithm and an $O\left(\frac{n^3}{\varepsilon^2}\right)$ time FPTAS. We then generalized the problem setting to m parallel two-machine flow-shops. We developed an $O(mn(2d)^{3m})$ time dynamic programming algorithm, and an $O\left(mn\left(\frac{4n}{\varepsilon}\right)^{3m}\right)$ time FPTAS.

References

[1] Ali, A., Gajpal, Y., Elmekkawy, T.Y. (2021). Distributed permutation flowshop scheduling problem with total completion time objective. *OPSEARCH*, 58, 425-447.

[2] Baker, K.R. (1974). *Introduction to Sequencing and Scheduling.* Wiley, New York.

[3] Błazewicz, J. (1984). Scheduling preemptible tasks on parallel processors with information loss. *Technique et Science Informatiques*, 3(6), 415-420.

[4] Błazewicz, J., Pesch, E., Sterna, M., Werner, F. (2005). The two-machine flow-shop problem with weighted late work criterion and common due date. *European Journal of Operational Research*, 165 (2), 408-415.

[5] Chen, X., Miao Q., Lin, B.M.T., Sterna, M., Błazewicz, J. (2022). Two-machine flow shop scheduling with a common due date to maximize total early work. *European Journal of Operational Research*, 300(2), 504–511.

[6] Chen, X., Kovalev, S., Sterna, M., Błazewicz, J. (2021). Mirror scheduling problems with early work and late work criteria. *Journal of Scheduling*, 24, 483-487.

[7] Dong, J., Hu, J., Kovalyov, M.Y., Lin, G., Luo, T., Tong, W., Wang, X., Xu, Y. (2017). Corrigendum to "An FPTAS for the parallel two-stage flowshop problem" [Theoret. Comput. Sci. 657 (2017) 64-72]. *Theoretical Computer Science*, 687, 93-94.

[8] Dong, J., Tong, W., Luo, T., Wang, X., Hu, J., Xu, Y., Lin, G. (2017). An FPTAS for the parallel two-stage flowshop problem. *Theoretical Computer Science*, 657, 64-72.

[9] Gilenson, M., Shabtay, D. (2021). Multi-scenario scheduling to maximise the weighted number of just-in-time jobs. *Journal of the*

Operational Research Society, 72(8), 1762-1779.

[10] Garey, M., Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., San Francisco.

[11] Gmys, J., Mezmaz, M., Melab, N., Tuyttens, D. (2020). A computationally efficient Branch-and-Bound algorithm for the permutation flow-shop scheduling problem. *European Journal of Operational Research*, 284(3), 814-833.

[12] Hermelin, D., Shabtay, D., Talmon, N. (2019). On the parameterized tractability of the just-in-time flow-shop scheduling problem. *Journal of Scheduling*, 22, 663-676.

[13] Johnson, S. (1954). Optimal two and three stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1(1), 61-68.

[14] Kelley, J.E. (1961). Critical-path planning and scheduling: mathematical basis. *Operations Research*, 9(3), 296-320.

[15] Kovalev, S., Kovalyov, M.Y., Pesch, E. (2021). Single machine scheduling with assignable due dates to minimize maximum and total late work, in submission.

[16] Kovalyov, M.Y. (1985). Efficient epsilon-approximation algorithm for minimizing the makespan in a parallel two-stage system. *Vesti Academii navuk Belaruskai SSR, Ser. Phiz.-Mat. Navuk*, 3, 119 (in Russian).

[17] Levorato, M., Figueiredo, R., Frota, Y. (2022). Exact solutions for the two-machine robust flow shop with budgeted uncertainty. *European Journal of Operational Research*, in press.

[18] Liu, Y., Wang, L., Wang, X.V., Xu, X., Zhang, L. (2019). Scheduling in cloud manufacturing: state-of-the-art and research challenges. *International Journal of Production Research*, 57(15-16), 4854-4879.

[19] Lin, B.M.T., Lin, F.-C. , Lee, R.C.T. (2006). Two-machine flow-shop scheduling to minimize total late work. *Engineering Optimization*, 38 (04), 501-509.

[20] Mnich, M., van Bevern, R. (2018). Parameterized complexity of machine scheduling: 15 open problems. *Computers & Operations Research*, 100, 254-261.

[21] Naderi, B., Ruiz, R. (2010). The distributed permutation flowshop scheduling problem. *Computers and Operations Research*, 37(4), 754-768.

[22] Naderi, B., Ruiz, R. (2014). A scatter search algorithm for the distributed permutation flowshop scheduling problem. *European Journal of Operational Research*, 239, 323-334.

[23] Ruiz, R., Pan, Q.K., Naderi, B. (2019). Iterated Greedy methods for the distributed permutation flowshop scheduling problem. *Omega*, 83, 213-222.

[24] Potts, C. N. , Van Wassenhove, L. N. (1992). Single machine scheduling to minimize total late work. *Operations Research*, 40(3), 586-595.

[25] Qi, X., Yu, G., Bard, J.F. (2002). Single machine scheduling with assignable due dates. *Discrete Applied Mathematics*, 122, 211-233.

[26] Sevastyanov, S.V., Lin, B.M.T., Hwang, F.J. (2009) Makespan minimization in parallel flow shops. *Proceedings of The 9th Workshop on Models and Algorithms for Planning and Scheduling Problems* (MAPSP 2009), July 2009, Abbey Rolduc, The Netherlands.

[27] Sterna, M. (2021). Late and early work scheduling: A survey. *Omega*, 104, 102453.

[28] Sterna, M., Czerniachowska, K. (2017). Polynomial time approximation scheme for two parallel machines scheduling with a common due date to maximize early work. *Journal of Optimization Theory and Applications*, 174(3), 927-944.

[29] Zinder, Y., Kononov, A., Fung, J. (2021). A 5-parameter complexity classification of the two-stage flow shop scheduling problem with job dependent storage requirements. *Journal of Combinatorial Optimization*, 42(2), 276-309.