

Approximation algorithm for the parallel-machine scheduling problem with multiple servers

Guo-Sheng Liu

Guangdong University of Technology

Ting-Ting Zheng

zhengtingfy@126.com

Guangzhou City University of Technology

Fang-Qing Liu

Guangdong University of Technology

Research Article

Keywords: Parallel machine scheduling, Multiple servers, SWPT algorithm, Worst-case analysis, mixed integer programming

Posted Date: October 19th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11193291/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Approximation algorithm for the parallel-machine scheduling problem with multiple servers

Guo-Sheng Liu¹, Ting-Ting Zheng^{2*}, Fang-Qing Liu¹

¹School of Management, Guangdong University of Technology, 161
Yinglong Road, Guangzhou, 510520, Guangdong, China.

^{2*}School of Management, Guangzhou City University of Technology, 1
Xuefu Road, Guangzhou, 510800, Guangdong, China.

*Corresponding author(s). E-mail(s): zhengtingfy@126.com;
Contributing authors: gliu@gdut.edu.cn; peerfog@gdut.edu.cn;

Abstract

We study the identical parallel-machine scheduling problem with multiple servers. Each job has to be set up by a server before being processed on one of the machines. Preemption is not permitted, and the objective is to minimize the total weighted completion time. A time-indexed mixed-integer programming (MIP) formulation is presented to obtain exact optimal solutions for small- and medium-sized instances. As the MIP becomes intractable for large-scale instances, a shortest weighted processing time (SWPT) algorithm is proposed. We establish that SWPT yields a worst-case approximation bound of $3 - \frac{k+1}{m}$ for the general case and $3 - \frac{k}{m} - \frac{1}{k}$ for the special case with equal job processing times. Furthermore, the tightness of the second bound is verified for two-machine and one-server configurations by constructing an instance. Numerical experiments use MIP-derived optimal solutions as benchmarks and demonstrate that SWPT achieves an average optimality gap below 5% across test instances. In addition, we analyze the computational behavior of the time-indexed formulation with varying numbers of jobs, servers, and machines.

Keywords: Parallel machine scheduling, Multiple servers, SWPT algorithm, Worst-case analysis, mixed integer programming

1 Introduction

In this paper, we consider the problem of scheduling n jobs on m identical parallel machines with k identical servers, where $1 \leq k \leq m$. The setup and processing operations of each job must be performed non-preemptively on the same machine without interruption. Each job has to be loaded or set up onto a machine by a server before being processed. During the setup time, both the server and the machine are busy and unavailable for other jobs. Once a server has completed a job's setup on a machine, the job can be processed immediately, and the server becomes free to set up another job. Meanwhile, the server is free to perform the setup for another job. It is assumed that the travel times of servers between machines are zero. In industrial and business practices, the servers act like loading robots, humans or specific tools that assist the execution of multiple machines or assembly lines. With widespread applications of industrial robots, cloud services and centralized AI computing, these multi-server parallel-machine scheduling problems have become increasingly crucial.

Using the three-field notation proposed by Graham et al. [1], our problem can be denoted as $P_m S_k | s_j, p_j | \sum w_j C_j$. Hall and Potts [2] proved the NP-completeness of the scheduling problem $P_2 S_1 | s_j = 1, p_j | \sum w_j C_j$. Brucker [3] further complemented the complexity results, and verified the variant $P_2 S_1 | p_j = p | \sum C_j$ with constant processing and arbitrary set-up times is NP-hard. Such theoretical results imply that polynomial time optimal algorithms are unlikely to exist for our target problem. Therefore, approximation algorithms serve as a practical and promising way to tackle such scheduling problems. Existing approximation studies on multi-server parallel-machine scheduling mainly focus on two classical objectives. For the total completion time objective $\sum C_j$, Ou et al. [4] proved that the classical SPT rule yields a 2-approximation ratio for the problem $P_m S_k | p_j, u_j = 1 | \sum C_j$ and proposed improved SPT-based heuristics. For the makespan objective C_{max} , Werner et al. [5] considered the problem $P_m S_k | s_j, p_j | C_{max}$ and showed that arbitrary list scheduling (LS) algorithm and longest processing time (LPT) algorithm have worst-case ratios of $3 - \frac{k+1}{m}$ and $3 - \frac{k}{m} - \frac{1}{k}$, respectively. Building upon their work, this paper extends these performance results to the more general objective of minimizing the total weighted completion time $\sum w_j C_j$ and propose a approximation algorithm.

Approximation algorithms for server-constrained parallel-machine scheduling have attracted substantial research attention. For problems with equal processing times of jobs, Hall and Potts [2] pointed out that when $s_j \geq p_j = 1$, the single-server parallel machine scheduling problem degenerates into classic single-machine scheduling and can be solved in polynomial time. Furthermore, Zhang et al. [6] proved that the SPT algorithm achieves a $(2 - \frac{1}{m})$ -approximation for $P_m S_1 | p_j = p | C_{max}$ and obtains a $(1 + \frac{\sqrt{m-1}}{\sqrt{m} + \sqrt{m-1}})$ -approximation for problem $P_m S_1 | p_j = p | \sum C_j$. In contrast, for problems with distinct processing times, substantial results have been derived for minimizing the total weighted completion time $\sum w_j C_j$. For example, back in 2001, Wang and Cheng [7] provided the first $(5 - \frac{1}{m})$ -approximation algorithm. More than a decade later, Hasani et al. [8] proposed an SWPT algorithm that improves the performance bound to $3 - \frac{1}{m}$, where jobs are sequenced in non-decreasing order of $\frac{1}{w_j}(s_j + \frac{p_j}{m})$. Recently, Liu et al. [9] further showed that the SWPT yields a tighter ratio of $3 - \frac{2}{m}$. In this paper, we extend the SWPT algorithm to the multi-server setting and derive more general approximation ratios.

Exact optimization models for parallel-machine scheduling with server constraints have been extensively developed, mainly to generate optimal benchmark solutions for assessing approximation algorithms. Vilem Heinz [10] proposed a constraint programming (CP) model to obtain optimal solutions and designed constructive heuristics for large-scale instances. Li, Yantong [11] developed three novel mixed-integer linear programs for integrated discrete parallel-machine scheduling and location problems. For the single-server parallel-machine scheduling problem with the makespan objective, Elidrisi [12] and Silva [13] both proposed mixed-integer linear programming (MIP) formulations, whose time-indexed formulations motivated our modeling approach. Nevertheless, time-indexed formulations have not yet been developed for the multi-server parallel-machine scheduling problem with the total weighted completion time objective, which this work addresses by developing such a model.

In this paper, we consider the multi-server parallel machine scheduling problem with the objective of minimizing total weighted completion time, which has rarely been studied in existing literature. We develop a time-indexed MIP formulation to provide reliable optimal benchmarks for algorithm evaluation. Furthermore, we present a generalized SWPT algorithm, in which the job sequencing priority is determined by the weighted processing time $\frac{1}{w_j}(\frac{s_j}{k} + \frac{p_j}{m})$. We prove that the SWPT algorithm achieves a worst-case ratio of $3 - \frac{k+1}{m}$ for general cases and $3 - \frac{k}{m} - \frac{1}{k}$ when all jobs have equal processing times. In addition, we show that the bound of $3 - \frac{k}{m} - \frac{1}{k}$ is tight. The rest of the paper is organized as follows. Section 2 provides the problem description and some required notations. The time-indexed MIP formulation is developed in Section 3. In Section 4, we analyze the worst-case performance of the SWPT algorithm. Finally, the conclusions are provided in Section 5.

2 Problem Description

Let $\mathbb{N} = 1, 2, \dots, n$ denote the set of jobs which has to be processed on m identical parallel machines with k servers. Each job has a setup time s_j , a processing time p_j , and a weight w_j , and we assume that all setup times s_j and processing times p_j are positive integers. Furthermore, each job is restricted to commence its setup operation only at integer time instants. Prior to job processing on a machine, a server must first perform its setup, during which both the assigned server and machine are occupied for s_j time units. Once the setup is finished, the server is immediately released, and the machine continues to process the job for p_j time units. Each machine and each server can handle at most one job at a time, and preemption of jobs is not allowed. The objective is to minimize the total weighted completion time $\sum w_j C_j$.

In this paper, we adopt the notation that j , $\langle j \rangle$ and $[j]$ denote the job index, the j -th started job, and the j -th completed job under a given schedule, respectively. Note that $\langle 1 \rangle \rightarrow \langle 2 \rangle \dots \rightarrow \langle n \rangle$ and $[1] \rightarrow [2] \rightarrow \dots \rightarrow [n]$ can represent the same schedule. It is well known that no optimal schedule allows meaningless idle time for any machine or server. Consequently, each eligible job is assigned to the machine and server with the earliest available time, which reduces the original problem to job sequencing.

To illustrate the above problem and notation, we present a representative instance with 3 machines, 2 servers, and 6 jobs. The index, setup time, processing time, and weight of each job are listed in Columns 1-4 of Table 1. In Figure 1, W_{ij} denotes the waiting time of server i after completing job j , and I_j denotes the idel time of the machine after finishing job j . Under the SWPT algorithm, the job processing order will be $5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6$.

Table 1 A typical instance and its schedule $\tilde{\varphi}$

j	s_j	p_j	w_j	$\langle j \rangle$	S_j	C_j	$[j]$
1	1	6	12	5	0	7	2
2	2	2	6	1	1	5	5
3	4	2	6	2	5	11	1
4	3	3	5	3	6	12	3
5	2	4	12	4	0	6	4
6	1	2	2	6	9	12	6

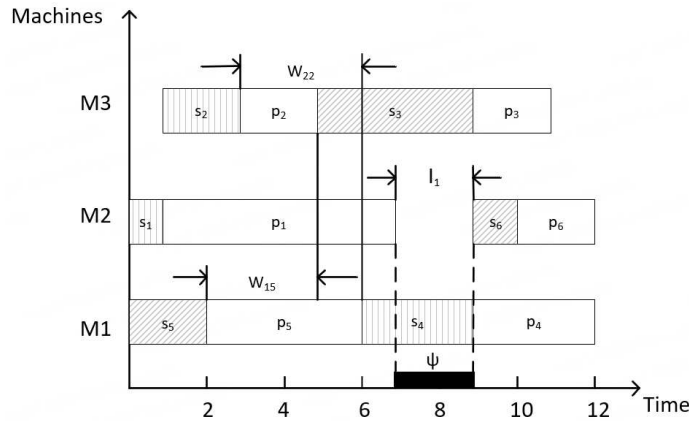


Fig. 1 Gantt Chart of the Instance Under Scheduling $\tilde{\varphi}$

3 Mathematical Formulation

In this section, we derive a MIP formulation based on completion-time variables and time-indexed variables for the problem $P_m S_k | s_j, p_j | \sum w_j C_j$. The time horizon is divided into T unit-length periods, where period t denotes the interval $[t, t + 1]$ for $t = 0, 1, 2, \dots, T$. And the value of T significantly affects the size of the model. Let

$\tau_j = s_j + p_j$ denote the total duration of job j , where s_j and p_j are its setup and processing times, respectively. The problem is to find a schedule that minimizes total weighted completion time.

Sets

$N = \{1, 2, \dots, n\}$: set of jobs
 $M = \{1, 2, \dots, m\}$: set of machines
 $K = \{1, 2, \dots, k\}$: set of servers

Parameters

w_j : weight of job j
 s_j : setup time of job j
 p_j : processing time of job j
 τ_j : the duration of job j , $\tau_j = s_j + p_j$

Decision variables

$x_j^t = \begin{cases} 1 & \text{if job } j \text{ starts processing at time } t \\ 0 & \text{otherwise} \end{cases}$

Auxiliary variables

C_j : completion time of job j

Model

$$\min \sum_{j \in N} w_j C_j \quad (1)$$

$$\sum_{t=0}^{T-\tau_j} x_j^t = 1, \forall j \in N \quad (2)$$

$$\sum_{j \in N} \sum_{s=\max(0, t-\tau_j+1)}^{\min(t, T-\tau_j)} x_j^s \leq m, \forall t \in [0, T] \quad (3)$$

$$\sum_{j \in N} \sum_{s=\max(0, t-s_j+1)}^{\min(t, T-\tau_j)} x_j^s \leq k, \forall t \in [0, T] \quad (4)$$

$$C_j = \sum_{t=0}^{T-\tau_j} (t + \tau_j) x_j^t, \forall j \in N \quad (5)$$

$$x_j^t \in \{0, 1\}, \forall j \in N, \forall t \in [0, T] \quad (6)$$

The objective function 1 minimizes the total weighted completion time. Constraints 2 ensures that each job is processed exactly once. Constraints (3) and (4) are the capacity constraints, which jointly limit the total workload assignable to the available resources at any time. Constraints 3 enforces that no more than m jobs are processed on all machines simultaneously, while Constraints 4 ensures that at most k jobs are served at any time. Equation 5 defines the completion time C_j of each job j , supporting the construction of scheduling objective functions. Finally, constraints 6 define the feasibility domain of the decision variables.

The proposed time-indexed MIP formulation yields optimal solutions, but its discrete-time structure causes the number of variables to grow rapidly with the problem size. Therefore, it is only applicable to small- and medium-sized instances and serves as a benchmark for the approximate algorithm developed in the following sections to handle large-scale cases.

4 The performance of the SWPT algorithm

The SWPT algorithm sequences jobs in non-decreasing order of $\frac{1}{w_j}(\frac{s_j}{k} + \frac{p_j}{m})$. We then schedule these jobs sequentially at the earliest feasible time slot where both a server and a machine are available. In addition, we derive a lower bound and an upper bound for the problem.

Lemma 1. Let φ be an arbitrary schedule for the problem $P_m S_k | s_j, p_j | \sum w_j C_j$. Then, the following inequality holds for all $j \in \{1, 2, \dots, n\}$:

$$C_{[j]}(\varphi) \geq \frac{1}{k} \sum_{q=1}^j s_{[q]}. \quad (7)$$

Proof By definition, the jobs $[1], [2], \dots, [j]$ are completed by time $C_{[j]}$. By that time, the total setup time of the first j completed jobs is no less than $\sum_{q=1}^j s_{[q]}$. These setup operations are handled by at most k servers. Hence, we have $C_{[j]}(\varphi) \geq \frac{1}{k} \sum_{q=1}^j s_{[q]}$. $\square$

Lemma 2. For the single-machine scheduling problem $1 | p'_j = \frac{s_j}{k} + \frac{p_j}{m} | \sum w_j C'_j$, let φ be an arbitrary schedule and $\tilde{\varphi}$ denote the schedule in which jobs are sequenced in non-decreasing order of $\frac{1}{w_j}(\frac{s_j}{k} + \frac{p_j}{m})$. Then, the following equality holds:

$$\begin{aligned} \min_{\varphi} \sum w_j C'_j(\varphi) &= \sum w_j C'_j(\tilde{\varphi}) = \\ &= \sum_{j=1}^n w_{\langle j \rangle} \left(\frac{1}{k} \sum_{q=1}^j s_{\langle q \rangle} + \frac{1}{m} \sum_{q=1}^j p_{\langle q \rangle} \right). \end{aligned} \quad (8)$$

Proof The conclusion of Lemma 2 is directly derived from the optimality of the SWPT rule for the single-machine scheduling problem to minimize total weighted completion time, as established by Smith [14]. It implies that the schedule in the non-decreasing order of $\frac{p'_j}{w_j}$ yields the minimum total weighted completion time. $\square$

Theorem 1. For the problem $P_m S_k | s_j, p_j | \sum w_j C_j$, let φ be any feasible schedule. The following inequality holds for all $j \in \{1, 2, \dots, n\}$:

$$\begin{aligned} C_{\langle j \rangle}(\varphi) &\leq \frac{1}{k} \sum_{q=1}^j s_{\langle q \rangle} + \frac{1}{m} \sum_{q=1}^j p_{\langle q \rangle} + \\ &\quad \left(1 - \frac{1}{k}\right) s_{\langle j \rangle} + \left(1 - \frac{1}{m}\right) p_{\langle j \rangle}. \end{aligned} \quad (9)$$

Proof By definition, let $C_{\langle j \rangle}$ denote the completion time of the j -th started job, and t be its earliest available start time. Then, $C_{\langle j \rangle} = t + s_{\langle j \rangle} + p_{\langle j \rangle}$. As shown in Figure 1, Ψ denotes the total projection length of machine waiting times on the horizontal axis. During the machine idle period, at most k servers are in operation, which implies $k\Psi \leq \sum_{q=1}^{j-1} s_{\langle q \rangle}$. In this case, at most $(m-k)$ parallel machines are idle when all servers are busy, from which we derive $\frac{1}{m-k} \sum_{q=1}^{j-1} I_{\langle q \rangle} \leq \Psi \leq \frac{1}{k} \sum_{q=1}^{j-1} s_{\langle q \rangle}$. Equivalently, we have $\sum_{q=1}^{j-1} I_{\langle q \rangle} \leq \frac{(m-k)}{k} \sum_{q=1}^{j-1} s_{\langle q \rangle}$. Since job $\langle j \rangle$ is assigned to the machine with earliest available time, we can obtain $t \leq \frac{1}{m} (\sum_{q=1}^{j-1} (s_{\langle q \rangle} + p_{\langle q \rangle}) + \sum_{q=1}^{j-1} I_{\langle q \rangle})$, in which the first term on the right-hand side is the total setup and processing time of the first $j-1$ jobs, and the second term denotes their total idle time. And then we derive that $t \leq \frac{1}{m} ((\sum_{q=1}^{j-1} (s_{\langle q \rangle} + p_{\langle q \rangle}) + \frac{(m-k)}{k} \sum_{q=1}^{j-1} s_{\langle q \rangle})) \leq \frac{1}{m} \sum_{q=1}^{j-1} p_{\langle q \rangle} + \frac{1}{k} \sum_{q=1}^{j-1} s_{\langle q \rangle}$. Finally, the following inequality holds:

$$\begin{aligned} C_{\langle j \rangle} &= t + s_{\langle j \rangle} + p_{\langle j \rangle} \leq \frac{1}{m} \sum_{q=1}^{j-1} p_{\langle q \rangle} + \frac{1}{k} \sum_{q=1}^{j-1} s_{\langle q \rangle} + \\ &\quad s_{\langle j \rangle} + p_{\langle j \rangle} \leq \frac{1}{k} \sum_{q=1}^j s_{\langle q \rangle} + \frac{1}{m} \sum_{q=1}^j p_{\langle q \rangle} + \\ &\quad \left(1 - \frac{1}{k}\right) s_{\langle j \rangle} + \left(1 - \frac{1}{m}\right) p_{\langle j \rangle}. \end{aligned} \quad (10)$$

$\square$

Therefore, Theorem 1 is proved.

Theorem 2. For the problem $P_m S_k |s_j, p_j| \sum w_j C_j$, let φ^* be the optimal schedule and $\tilde{\varphi}$ denote the schedule obtained by sequencing jobs in the non-decreasing order of $\frac{1}{w_j}(\frac{s_j}{k} + \frac{p_j}{m})$. Then, the following inequality holds:

$$\sum w_j C_j(\tilde{\varphi}) \leq (3 - \frac{k+1}{m}) \sum w_j C_j(\varphi^*). \quad (11)$$

Proof Based on [9], we obtain the following result:

$$C_{[j]}(\varphi) \geq s_{[j]} + p_{[j]}, \quad (12)$$

$$C_{[j]}(\varphi) \geq \frac{1}{m} \left(\sum_{q=1}^j s_{[q]} + \sum_{q=1}^j p_{[q]} \right). \quad (13)$$

Combining Lemma 1 with inequalities (12) and (13) for each $j \in \{1, 2, \dots, n\}$, we derive the following inequality:

$$\begin{aligned} (x + y + z)C_{[j]}(\varphi^*) &\geq \frac{x}{k} \sum_{q=1}^j s_{[q]} + y(s_{[j]} + p_{[j]}) + \\ &\frac{z}{m} \left(\sum_{q=1}^j s_{[q]} + \sum_{q=1}^j p_{[q]} \right) = \left(\frac{x}{k} + \frac{z}{m} \right) \sum_{q=1}^j s_{[q]} + \\ &\frac{z}{m} \sum_{q=1}^j p_{[q]} + y(s_{[j]} + p_{[j]}), \end{aligned} \quad (14)$$

where $x, y, z \geq 0$. Setting $x = 1 - \frac{k}{m}, y = 1 - \frac{1}{m}, z = 1$, we obtain following inequality can be obtained for all $j \in \{1, 2, \dots, n\}$:

$$\begin{aligned} \left(3 - \frac{k+1}{m}\right) C_{[j]}(\varphi^*) &\geq \frac{1}{k} \sum_{q=1}^j s_{[q]} + \\ &\frac{1}{m} \sum_{q=1}^j p_{[q]} + \left(1 - \frac{1}{m}\right) (s_{[j]} + p_{[j]}). \end{aligned} \quad (15)$$

Multiply both sides of (15) by the weight $w_{[j]}$, and sum the resulting inequalities over all $j \in \{1, 2, \dots, n\}$. This yields

$$\begin{aligned} \left(3 - \frac{k+1}{m}\right) \sum_{j=1}^n w_{[j]} C_{[j]}(\varphi^*) &\geq \sum_{j=1}^n w_{[j]} \left(\frac{1}{k} \sum_{q=1}^j s_{[q]} + \right. \\ &\left. \frac{1}{m} \sum_{q=1}^j p_{[q]} + \left(1 - \frac{1}{m}\right) (s_{[j]} + p_{[j]}) \right). \end{aligned} \quad (16)$$

Note that $\sum_{j=1}^n w_{[j]} C_{[j]}(\varphi^*) = \sum_{j=1}^n w_j C_j(\varphi^*)$ and $\sum_{j=1}^n w_{[j]} ((1 - \frac{1}{m})(s_{[j]} + p_{[j]})) = \sum_{j=1}^n w_j ((1 - \frac{1}{m})(s_j + p_j))$, because any schedule of the problem is a permutation of the job set N . Therefore, we have

$$\begin{aligned} \left(3 - \frac{k+1}{m}\right) \sum_{j=1}^n w_j C_j(\varphi^*) &\geq \sum_{j=1}^n w_{[j]} \left(\frac{1}{k} \sum_{q=1}^j s_{[q]} + \right. \\ &\left. \frac{1}{m} \sum_{q=1}^j p_{[q]} + \left(1 - \frac{1}{m}\right) (s_{[j]} + p_{[j]}) \right) = \\ &\sum_{j=1}^n w_j C'_j(\varphi') + \left(1 - \frac{1}{m}\right) \sum_{j=1}^n w_j (s_j + p_j), \end{aligned} \quad (17)$$

where φ' is the schedule obtained by sorting jobs in non-decreasing order of $C_j(\varphi^*)$, and $C'_j(\varphi')$ denotes the completion time of job j in the problem $1|p'_j = \frac{s_j}{k} + \frac{p_j}{m}| \sum w_j C'_j$ under schedule φ' .

Meanwhile, let $\tilde{\varphi}$ define the schedule constructed by sorting jobs in non-decreasing order of $\frac{1}{w_j}(\frac{s_j}{k} + \frac{p_j}{m})$. Substituting $\tilde{\varphi}$ into Theorem 1 yields the following inequality for all $j \in \{1, 2, \dots, n\}$:

$$\begin{aligned}
C_{\langle j \rangle}(\tilde{\varphi}) &\leq \frac{1}{k} \sum_{q=1}^j s_{\langle q \rangle} + \frac{1}{m} \sum_{q=1}^j p_{\langle q \rangle} + \\
&\quad (1 - \frac{1}{k})s_{\langle j \rangle} + (1 - \frac{1}{m})p_{\langle j \rangle}.
\end{aligned} \tag{18}$$

Similarly, we can derive

$$\begin{aligned}
\sum_{j=1}^n w_j C_j(\tilde{\varphi}) &\leq \sum_{j=1}^n w_j (\frac{1}{k} \sum_{q=1}^j s_q + \frac{1}{m} \sum_{q=1}^j p_q) + \\
&\quad (1 - \frac{1}{k}) \sum_{j=1}^n w_j s_j + (1 - \frac{1}{m}) \sum_{j=1}^n w_j p_j = \\
&\quad \sum_{j=1}^n w_j C'_j(\tilde{\varphi}) + (1 - \frac{1}{k}) \sum_{j=1}^n w_j s_j + (1 - \frac{1}{m}) \sum_{j=1}^n w_j p_j,
\end{aligned} \tag{19}$$

where $C'_j(\tilde{\varphi})$ represents the completion time of job j under SWPT schedule $\tilde{\varphi}$ for the problem $1|p'_j = \frac{s_j}{k} + \frac{p_j}{m}| \sum w_j C'_j$. By Lemma 2, $\tilde{\varphi}$ is an optimal schedule for problem $1|p'_j = \frac{s_j}{k} + \frac{p_j}{m}| \sum w_j C'_j$, so we have $\sum_{j=1}^n w_j C'_j(\tilde{\varphi}) \leq \sum_{j=1}^n w_j C'_j(\varphi^*)$. Meanwhile, for $m \geq k \geq 1$, the inequality $1 - \frac{1}{m} \geq 1 - \frac{1}{k}$ holds, which means the right-hand side of Equation (13) is no greater than that of Equation (11). Therefore, we derive $\sum w_j C_j(\tilde{\varphi}) \leq (3 - \frac{k+1}{m}) \sum w_j C_j(\varphi^*)$. $\square$

Theorem 3. For problem $P_m S_k | s_j, p_j = p | \sum w_j C_j$, let φ^* be the optimal schedule, and $\tilde{\varphi}$ the schedule obtained by sorting the jobs in the non-decreasing order of $\frac{1}{w_j}(\frac{s_j}{k} + \frac{p_j}{m})$. Then the following inequality holds:

$$\sum w_j C_j(\tilde{\varphi}) \leq (3 - \frac{k}{m} - \frac{1}{k}) \sum w_j C_j(\varphi^*). \tag{20}$$

Proof Since all jobs have equal processing times, the completion order of jobs is the same as the completion order of their setup times. Let $V_{[j]}$ denote the setup completion time of the j -th completed job. At most k servers can perform setup operations in parallel, so $V_{[j]} = \max(V_{[1]}, V_{[2]}, \dots, V_{[j]}) \geq \frac{1}{k} \sum_{q=1}^j V_{[q]} \geq \frac{1}{k} \sum_{q=1}^j s_{[q]}$. Therefore, we can have

$$C_{[j]}(\varphi) = V_{[j]} + p_{[j]} \geq \frac{1}{k} \sum_{q=1}^j s_{[q]} + p_{[j]} \tag{21}$$

Multiplying (21) by $(1 - \frac{k}{m})$ and (12) by $(1 - \frac{1}{k})$, and then adding them to (13), we obtain

$$\begin{aligned}
(3 - \frac{k}{m} - \frac{1}{k})C_{[j]}(\varphi) &\geq \frac{1}{k} \sum_{q=1}^j s_{[q]} + \frac{1}{m} \sum_{q=1}^j p_{[q]} + \\
&\quad (2 - \frac{1}{k} - \frac{k}{m})p_{[j]} + (1 - \frac{1}{k})s_{[j]}.
\end{aligned} \tag{22}$$

Since $2 - \frac{1}{k} - \frac{k}{m} \geq 1 - \frac{1}{m}$, (22) can be rewritten as

$$\begin{aligned}
(3 - \frac{k}{m} - \frac{1}{k})C_{[j]}(\varphi) &\geq \frac{1}{k} \sum_{q=1}^j s_{[q]} + \frac{1}{m} \sum_{q=1}^j p_{[q]} + \\
&\quad (1 - \frac{1}{m})p_{[j]} + (1 - \frac{1}{k})s_{[j]}.
\end{aligned} \tag{23}$$

Substituting the optimal schedule φ^* into (23), we obtain

$$\begin{aligned}
(3 - \frac{k}{m} - \frac{1}{k}) \sum_{j=1}^n w_j C_j(\varphi^*) &= (3 - \frac{k}{m} - \frac{1}{k}) \sum_{j=1}^n w_{[j]} C_{[j]}(\varphi^*) \geq \\
&\sum_{j=1}^n w_{[j]} (\frac{1}{k} \sum_{q=1}^j s_{[q]} + \frac{1}{m} \sum_{q=1}^j p_{[q]}) + (1 - \frac{1}{m}) \sum_{j=1}^n w_{[j]} p_{[j]} + \\
&(1 - \frac{1}{k}) w_{[j]} s_{[j]} \geq \sum_{j=1}^n w_{[j]} (\frac{1}{k} \sum_{q=1}^j s_{[q]} + \frac{1}{m} \sum_{q=1}^j p_{[q]}) + \\
&(1 - \frac{1}{m}) \sum_{j=1}^n w_j p_j + (1 - \frac{1}{k}) w_j s_j = \sum_{j=1}^n w_j C'_j(\varphi') + \\
&(1 - \frac{1}{m}) \sum_{j=1}^n w_j p_j + (1 - \frac{1}{k}) \sum_{j=1}^n w_j s_j,
\end{aligned} \tag{24}$$

where φ' and $C'_j(\varphi')$ follow the same definition as in (17). By a similar reasoning as in the proof for Theorem 2, we find $\sum_{j=1}^n w_j C'_j(\tilde{\varphi}) \leq \sum_{j=1}^n w_j C'_j(\varphi')$ and then the right-hand side of (19) is not greater than the right-hand side of (24). Therefore, $\sum w_j C_j(\tilde{\varphi}) \leq (3 - \frac{k}{m} - \frac{1}{k}) \sum w_j C_j(\varphi^*)$ holds for problem $P_m S_k | s_j, p_j = p | \sum w_j C_j$. $\square$

We proceed to prove that the second bound is tight. By Theorem 3, for the case with $m = 2$ and $k = 1$, the problem achieves a tight bound of $\frac{3}{2}$, which is verified by the constructed instance. Hence, this bound cannot be further improved.

Consider two job groups, A and B, each containing $2n$ jobs. Each job in group A has a setup time of 1, a processing time of 0, and a weight of 2, while each job in group B has a setup time of 0, a processing time of 1, and a weight of 1. Since $\frac{1}{w_j}(\frac{s_j}{k} + \frac{p_j}{m})$ is equal to $\frac{1}{2}$ for all $j \in \{1, 2, \dots, n\}$, every schedule satisfies the non-decreasing order of $\frac{1}{w_j}(\frac{s_j}{k} + \frac{p_j}{m})$. An interleaved schedule φ^* of A and B jobs achieves a minimum total weighted completion time of $6n^2 + 3n$, whereas the schedule $\tilde{\varphi}$ that processes all B jobs first followed by all A jobs yields a maximum of $\frac{9n^2+3n}{6n^2+3n}$. The worst-case ratio $\frac{9n^2+3n}{6n^2+3n}$ tends to $\frac{3}{2}$ if $n \rightarrow \infty$. These two schedules are shown in Figures 3 and 2.

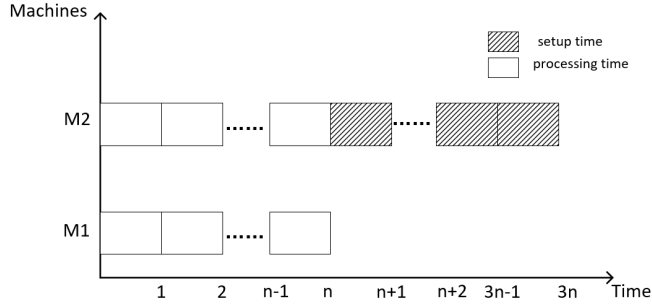


Fig. 2 Schedule $\tilde{\varphi}$: Group B Jobs Precede Group A Jobs

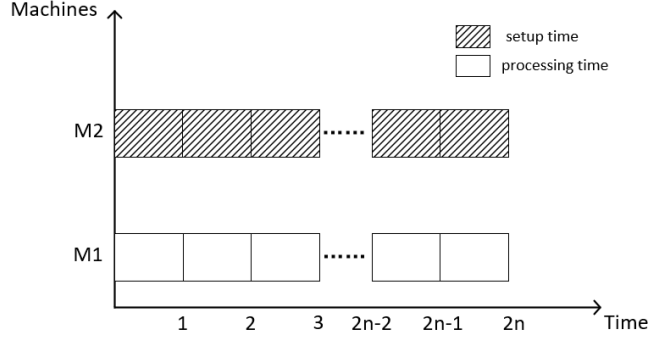


Fig. 3 Schedule φ^* : Interleaved A-B Group Processing

5 Experimental results

We evaluate the performance of the SWPT algorithm by benchmarking its approximate solutions against the exact optima obtained by solving the MIP model with Gurobi. The experimental design comprises 12 scenarios with varying J , M , and K , each containing 20 random instances. Table 2 lists the parameter settings. Each job is assigned a random weight from Uniform $[0.1, 1)$ to reflect differences in job priority. The setup times s_j and processing times p_j are drawn from Uniform $[1, 100]$. Furthermore, since T directly affects the size of the MIP model, we set its value to the minimum feasible level to maintain computational tractability. By Theorem 1, we adopt the upper bound $T \leq \frac{1}{k} \sum_{q=1}^n s_q + \frac{1}{m} \sum_{q=1}^n p_q + (1 - \frac{1}{m}) \max_{j \in N} \tau_j$, which guarantees that the start time of any job doesn't exceed this limit. The MIP model is solved using Gurobi 10.0.1. All experiments are performed on a workstation with an Intel(R) Core(TM) i5-12600KF, 4.90 GHz, 32.0 GB of RAM, 64-bit Windows 11. The maximum computational time allocated for each instance is set to 900 seconds.

Table 2 Instance factors and their corresponding values

Influencing factors	Ranges of values
Number of jobs	$J = 10, J = 20, J = 30$
Number of machines	$M = 3, M = 4, M = 5$
Number of servers	$K = 2, M = 3$
Range of job weights	$[0.1, 1)$
The range of setup times	$[1, 100]$
The range of processing times	$[1, 100]$

Table 3 shows the maximum, minimum, and average optimality ratios of the SWPT heuristic for each scenario, and the number of instances solved to optimality. The average optimality ratios range from 1.0108 to 1.0423, indicating that the SWPT solutions deviate from the Gurobi optimal solutions by no more than 4.23%. And the minimum ratio of 1.0000 indicates that SWPT can yield the optimal solution for certain small-scale instances. For example, SWPT attains the optimum in 3 out of 20 instances for the scenario with $J = 10, M = 3, K = 2$. For larger instances, although the SWPT does not achieve the optimal solution in some cases, the maximum optimality ratios remain below 1.08, which attests to the heuristic's robustness. Moreover, the average ratios tend to increase with the number of jobs, indicating a slight degradation in solution quality for larger instances. Nevertheless, the overall gaps remain within 5% on average and 8% at worst, confirming SWPT as a reliable approximation method for the considered scheduling problem.

Table 3 Solution quality comparison between the SWPT algorithm and the Gurobi solver

Scenario	Max Optimality Ratio	Min Optimality Ratio	No. of optimal solutions	Avg. Optimality Ratio
J10-M3-K2	1.0588	1.0000	3	1.0238
J20-M3-K2	1.0795	1.0063	0	1.0323
J30-M3-K2	1.0649	1.0177	0	1.0397
J10-M4-K3	1.0382	1.0000	4	1.0108
J20-M4-K3	1.0330	1.0007	0	1.0153
J30-M4-K3	1.0315	1.0036	0	1.0167
J10-M5-K2	1.0564	1.0000	2	1.0131
J20-M5-K2	1.0668	1.0055	0	1.0278
J30-M5-K2	1.0736	1.0147	0	1.0378
J10-M5-K3	1.0569	1.0000	1	1.0130
J20-M5-K3	1.0617	1.0101	0	1.0345
J30-M5-K3	1.0677	1.0202	0	1.0423

Note: J , M , and K denote the number of jobs, machines, and servers, respectively. A ratio closer to 1 indicates better solution quality. Each scenario contains 20 instances.

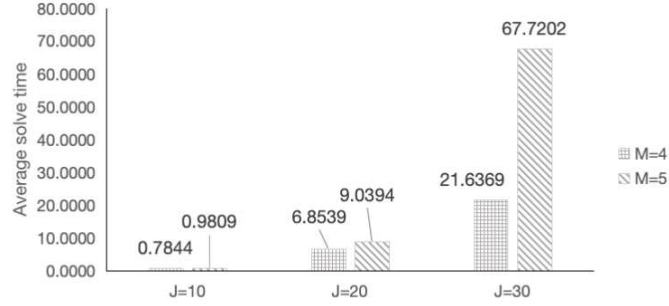


Fig. 4 Dependence of Gurobi Solve Time on Machine Count with Fixed Server Number ($K=3$)

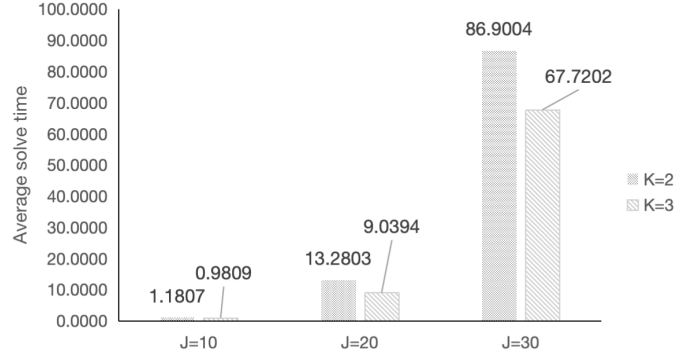


Fig. 5 Dependence of Gurobi Solve Time on Server Count with Fixed Machine Number ($M=5$)

To explore the computational limits of the exact solver, we analyze the impacts of problem parameters on Gurobi's runtime. As shown in Figure 4, with the numbers of servers and jobs fixed, the solution time increases with the number of machines. This growth can be attributed to the expansion of the solution space, as larger M results in more feasible assignments and consequently higher computational cost. In contrast, Figure 5 shows that, with the numbers of machines and jobs fixed, the solution time decreases as the number of servers increases, likely because additional servers facilitate parallel processing and reduce computational load. Moreover, both figures show that, for fixed M and K , the solution time rises markedly with the number of jobs, owing to the growing number of decision variables and constraints.

In summary, the SWPT consistently delivers high-quality solutions across all tested scenarios, with average optimality gaps within 5% and maximum gaps below 8%.

Meanwhile, the runtime of the Gurobi solver increases rapidly with problem size, particularly with the numbers of jobs and machines, while additional servers reduce solution time through parallel processing. These results confirm SWPT as a reliable approximation method, especially for larger instances where exact solvers become computationally prohibitive.

6 Conclusion

In this paper, we studied a parallel-machine scheduling problem with multiple servers to minimize the total weighted completion time. Our time-indexed MIP formulation yielded optimal solutions for small- and medium-sized instances and served as a reliable benchmark for algorithm evaluation. However, exponential growth in decision variables limited its applicability to larger-sized cases and motivated the development of approximation algorithms.

Accordingly, We analyzed the performance of the SWPT algorithm and proved that it has worst-case ratio of $3 - \frac{k+1}{m}$ for the problem $P_m S_k | s_j, p_j | \sum w_j C_j$. When all jobs have equal processing time, the worst-case ratio improves to $3 - \frac{k}{m} - \frac{1}{k}$. Moreover, we provided an example to show that the bound $3 - \frac{k}{m} - \frac{1}{k}$ is tight.

Numerical experiments demonstrated that the theoretically derived upper bounds were relatively conservative. In practice, the SWPT algorithm achieved promising scheduling performance, with an average optimality gap no more than 5% and a maximum gap below 8% across various test instances. Further computational sensitivity tests revealed that larger job and machine counts increased the computational runtime of the MIP model, whereas an increase in the number of servers reduced runtime. These tests clearly exposed the inherent scalability bottleneck of the time-indexed formulation.

This study relied on assumptions of static job arrivals and homogeneous machines and servers, which may limit the generalization of the derived theoretical results to practical industrial settings. Furthermore, the tightness of the proven bound for general instances remains unproven. These aspects open promising avenues for further investigation.

Acknowledgments. The authors would like to thank the anonymous reviewers for their constructive comments and suggestions, which have greatly improved the quality of this paper.

Authors' contributions. **Guo-Sheng Liu:** Conceptualization, Methodology, Software, Supervision, Funding acquisition. **Ting-Ting Zheng:** Conceptualization, Writing - original draft, Writing- reviewing and editing, Data curation, Validation. **Fang-Qing Liu:** Conceptualization, Investigation, Supervision.

Conflict of interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R.: Optimization and approximation in deterministic sequencing and scheduling: a survey. In: Hammer, P.L., Johnson, E.L., Korte, B.H. (eds.) *Discrete Optimization II*. Annals of Discrete Mathematics, vol. 5, pp. 287–326. Elsevier, ??? (1979)
- [2] Hall, N.G., Potts, C.N., Sriskandarajah, C.: Parallel machine scheduling with a common server. *Discrete Applied Mathematics* **102**(3), 223–243 (2000)
- [3] Brucker, P., Dhaenens-Flipo, C., Knust, S., Kravchenko, S.A., Werner, F.: Complexity results for parallel machine problems with a single server. *Journal of Scheduling* **5**(6), (2002)
- [4] Ou, J., Qi, X., Lee, C.Y.: Parallel machine scheduling with multiple unloading servers. *Journal of Scheduling* **12**(1), 79–94 (2009)

- [5] Werner, F., Kravchenko, S.A.: Scheduling with multiple servers. *Automation and Remote Control* **71**(10), 2109–2121 (2010)
- [6] Zhang, A., Wang, H., Chen, Y., Chen, G.: Scheduling jobs with equal processing times and a single server on parallel identical machines. *Discrete Applied Mathematics* **213**, 196–206 (2016)
- [7] Wang, G., Cheng, T.E.: An approximation algorithm for parallel machine scheduling with a common server. *Journal of the Operational Research Society* **52**(2), 234–237 (2001)
- [8] Hasani, K., Kravchenko, S.A., Werner, F.: Minimizing total weighted completion time approximately for the parallel machine problem with a single server. *Information Processing Letters* **114**(9), 500–503 (2014)
- [9] Liu, G.-S., Li, J.-J., Yang, H.-D., Huang, G.Q.: Approximate and branch-and-bound algorithms for the parallel machine scheduling problem with a single server. *Journal of the Operational Research Society* **70**(9), 1554–1570 (2019)
- [10] Heinz, V., Novák, A., Vlk, M., Hanzálek, Z.: Constraint programming and constructive heuristics for parallel machine scheduling with sequence-dependent setups and common servers. *Computers & Industrial Engineering* **172**, 108586 (2022)
- [11] Li, Y., Côté, J.-F., Callegari-Coelho, L., Wu, P.: Novel formulations and logic-based benders decomposition for the integrated parallel machine scheduling and location problem. *INFORMS Journal on Computing* **34**(2), 1048–1069 (2022)
- [12] Elidrissi, A., Benmansour, R., Benbrahim, M., Duvivier, D.: Mathematical formulations for the parallel machine scheduling problem with a single server. *International Journal of Production Research* **59**(20), 6166–6184 (2021)
- [13] Silva, J.M.P., Subramanian, A., Uchoa, E.: On time-indexed formulations for the parallel machine scheduling problem with a common server. *Engineering Optimization* **56**(10), 1561–1578 (2024)
- [14] Smith, W.E., *et al.*: Various optimizers for single-stage production. *Naval Research Logistics Quarterly* **3**(1-2), 59–66 (1956)