Journal of the Operations Research Society of China (2019) 7:485–500
https://doi.org/10.1007/s40305-019-00253-0

Simultaneous Approximation Ratios for Parallel Machine Scheduling Problems

Long Wan[1] **· Jin-Jiang Yuan**[2]

Received: 31 July 2018 / Revised: 8 January 2019 / Accepted: 22 May 2019 / Published online: 11 July 2019

Abstract
Motivated by the problem to approximate all feasible schedules by one schedule in a given scheduling environment, we introduce in this paper the concepts of strong simultaneous approximation ratio and weak simultaneous approximation ratio. Then we study the two variants under various scheduling environments, such as non-preemptive, preemptive or fractional scheduling on identical, related or unrelated machines.

Keywords Scheduling · Simultaneous approximation ratio · Global fairness

Mathematics Subject Classification 90B35 · 90C27

1 Introduction

In the scheduling research, people always hope to find a schedule that achieves the balance of the loads of the machines well. To the end, some objective functions, such as minimizing makespan and maximizing machine cover, are designed to find

---

This paper is dedicated to Professor Ding-Zhu Du in celebration of his 70th birthday.

The first author was supported by the National Natural Science Foundation of China (Nos. 11601198 and 71761015). The second author was supported by the National Natural Science Foundation of China (No. 11671368).

✉ Jin-Jiang Yuan
yuanjj@zzu.edu.cn

Long Wan
cocu3328@163.com

1 School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 310013, China

2 School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China

a reasonable schedule. Representative publications can be found in Csirik et al. [1], Deuermeyer et al. [2], Graham [3], Graham [4] and McNaughton [5], among many others. But these objectives do not reveal the global fairness for the loads of all machines. Motivated by the problem to approximate all feasible schedules by one schedule in a given scheduling environment and thus realizing the global fairness, we introduce two new parameters: strong simultaneous approximation ratio (SAR) and weak simultaneous approximation ratio (WAR) for scheduling problems.

Our research is also enlightened from the research on global approximation of vector sets. Related work can be found in Bhargava et al. [6], Goel et al. [7], Goel et al. [8], Kleinberg et al. [9] and Kumar and Kleinberg [10]. Kleinberg et al. [9] proposed the notion of the coordinate-wise approximation for the fair vectors of allocations. Based on this notion, Kumar and Kleinberg [10] introduced the definitions of the global approximation ratio and the global approximation ratio under prefix sums.

For a given instance $\mathcal{I}$ of a minimization problem, we use $V(\mathcal{I})$ to denote the set of vectors induced by all feasible solutions of $\mathcal{I}$. For a vector $X = (X_1, X_2, \cdots, X_m) \in V(\mathcal{I})$, we use $\overleftarrow{X}$ to denote the vector in which the coordinates (components) of X are sorted in non-increasing order, that is, $\overleftarrow{X} = (X'_1, X'_2, \cdots, X'_m)$ is a resorting of $(X_1, X_2, \cdots, X_m)$ so that $X'_1 \geqslant X'_2 \geqslant \cdots \geqslant X'_m$. For two vectors $X, Y \in V(\mathcal{I})$, we write $X \preceq_c Y$ if $X_i \leqslant Y_i$ for all i. The global approximation ratio of a vector $X \in V(\mathcal{I})$, denoted by $c(X)$, is defined to be the infimum of α such that $\overleftarrow{X} \preceq_c \alpha \overleftarrow{Y}$ for all $Y \in V(\mathcal{I})$. Then the best global approximation ratio of instance $\mathcal{I}$ is defined to be $c^*(\mathcal{I}) = \inf_{X \in V(\mathcal{I})} c(X)$. For a vector $X \in V(\mathcal{I})$, we use $\sigma(X)$ to denote the vector in which the ith coordinate is equal to the sum of the first i coordinates of X. We write $X \preceq_s Y$ if $\sigma(\overleftarrow{X}) \preceq_c \sigma(\overleftarrow{Y})$. The global approximation ratio under prefix sums of a vector $X \in V(\mathcal{I})$, denoted by $s(X)$, is defined to be the infimum of α such that $X \preceq_s \alpha Y$ for all $Y \in V(\mathcal{I})$. Then the best global approximation ratio under prefix sums of instance $\mathcal{I}$ is defined to be $s^*(\mathcal{I}) = \inf_{X \in V(\mathcal{I})} s(X)$.

In terms of scheduling, the above concepts about the global approximation of vector sets can be naturally formulated as the simultaneous approximation of scheduling problems. Let $\mathcal{I}$ be an instance of a scheduling problem $\mathcal{P}$ on m machines $M_1, M_2, \cdots, M_m$, and let $\mathcal{S}$ be the set of all feasible schedules of $\mathcal{I}$. For a feasible schedule $S \in \mathcal{S}$, the *load* L_i^S of machine M_i is defined to be the time by which the machine finishes all the processing of the jobs and the parts of the jobs assigned to it. $L(S) = (L_1^S, L_2^S, \cdots, L_m^S)$ is called the *load vector* of machines under S. Then $V(\mathcal{I})$ is defined to be the set of all load vectors of instance $\mathcal{I}$. We write $c(S) = c(L(S))$ and $s(S) = s(L(S))$ for each $S \in \mathcal{S}$. Then $c^*(\mathcal{I}) = \inf_{S \in \mathcal{S}} c(S)$ and $s^*(\mathcal{I}) = \inf_{S \in \mathcal{S}} s(S)$. The *strong simultaneous approximation ratio* of problem $\mathcal{P}$ is defined to be $\mathrm{SAR}(\mathcal{P}) = \sup_{\mathcal{I}} c^*(\mathcal{I})$, and the *weak simultaneous approximation ratio* of problem $\mathcal{P}$ is defined to be $\mathrm{WAR}(\mathcal{P}) = \sup_{\mathcal{I}} s^*(\mathcal{I})$.

A scheduling problem is usually characterized by the machine type and the job processing mode. In this paper, the machine types under consideration are identical machines, related machines and unrelated machines, and the job processing modes under consideration are non-preemptive, preemptive and fractional. Let $\mathcal{J} = \{J_1, J_2, \cdots, J_n\}$ and $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$ be the set of jobs and the set of machines, respectively. The processing time of job J_j on machine M_i is given by p_{ij}, which is assumed to be a nonnegative integer. If $p_{ij} = p_{kj}$ for $i \neq k$, the machine type is *identical machines*. In this case, p_j is used to denote the processing time of job J_j on all machines. If $p_{ij} = \frac{p_j}{s_i}$ for all i, the machine type is *related machines*. In this case, p_j is called the standard processing time of job J_j and $s_i > 0$ is called the processing speed of machine M_i. If there is no restriction for p_{ij}, the machine type is *unrelated machines*. If each job must be non-preemptively processed on some machine, the processing mode is *non-preemptive*. If each job can be processed preemptively and can be processed on at most one machine at any time, the processing mode is *preemptive*. If each job can be partitioned into different parts which can be processed on different machines concurrently, the processing mode is *fractional*. We assume that each machine can process at most one job at any time under any processing mode.

Since we cannot avoid the worst schedule in which all jobs are processed on a common machine, it can be easily verified that, under each processing mode, $\mathrm{SAR}(\mathcal{P}) = m$ for identical machines, $\mathrm{SAR}(\mathcal{P}) = (s_1 + s_2 + \cdots + s_m)/s_1$ for related machines with speeds $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_m$ and $\mathrm{SAR}(\mathcal{P}) = +\infty$ for unrelated machines.

We then concentrate our research on the weak simultaneous approximation ratio $\mathrm{WAR}(\mathcal{P})$ of the scheduling problems defined as above. For convenience, we use P, Q and R to represent identical machines, related machines and unrelated machines, respectively, and use NP, PP and FP to represent non-preemptive, preemptive and fractional processing, respectively. Then the notation $Pm(\mathrm{NP})$ represents the scheduling problem on m identical machines under non-preemptive processing mode. Other notations for scheduling problems can be similarly understood. The main results are given in Table 1, where ρ is the value of WAR.

This paper is organizes as follows. In Sect. 2, we study the weak simultaneous approximation ratio for scheduling on identical machines. In Sect. 3, we study the weak simultaneous approximation ratio for scheduling on related machines. In Sect. 4, we study the weak simultaneous approximation ratio for scheduling on unrelated machines.

2 Identical Machines

For problem $P2(\mathrm{NP})$, we have $s(S) = 1$ for every schedule S which minimizes the makespan. So $\mathrm{WAR}(P2(\mathrm{NP})) = 1$. For problem $Pm(\mathrm{NP})$ with $m \geqslant 3$, the following instance shows that $\mathrm{WAR}(Pm(\mathrm{NP})) > 1$.

Table 1 Weak simultaneous approximation ratio of various scheduling problems

Model	Pm		Qm	Rm
NP	$\rho = 1$	for $m = 2$	$\frac{\sqrt{m}+1}{2} \leqslant \rho \leqslant \sqrt{m}$	$\frac{\sqrt{m}+1}{2} \leqslant \rho \leqslant \sqrt{m}$
	$1 < \rho \leqslant \sqrt{5} - 1$	for $m = 3$		
	$1 < \rho \leqslant \frac{3}{2}$	for $m \geqslant 4$		
PP	$\rho = 1$		$\frac{\sqrt{m}+1}{2} \leqslant \rho \leqslant \sqrt{m}$	$\frac{\sqrt{m}+1}{2} \leqslant \rho \leqslant \sqrt{m}$
FP	$\rho = 1$		$\rho = \frac{\sqrt{m}+1}{2}$	$\frac{\sqrt{m}+1}{2} \leqslant \rho \leqslant \sqrt{m}$

In the instance, there are m jobs with processing time $m-1$, $(m-1)(m-2)$ jobs with processing time m and a big job with processing time $(m-1)^2+r_m$, where

$$r_m = \frac{\sqrt{(m^3-m^2-m-2)^2+4m(m-1)(m-2)}-(m^3-m^2-m-2)}{2}.$$

It can be verified that $0 < r_m < m-2$. Let S be the schedule in which the m jobs with processing time $m-1$ are scheduled on one machine, the big job with processing time $(m-1)^2+r_m$ is scheduled on one machine, and the remaining $(m-1)(m-2)$ jobs with processing time m are scheduled on the remaining $m-2$ machines averagely. Let T be the schedule in which the big job is scheduled on one machine together with a job of processing time $m-1$, and each of the remaining machines has a job of processing time $m-1$ and $m-2$ jobs of processing time m. Then the makespan of schedule S is $m(m-1)$, and the $(m-1)$th prefix sum of $\overleftarrow{L(T)}$ is $m(m-1)^2-(m-2-r_m)$. Now we consider an arbitrary schedule R. If the big job is scheduled on one machine solely in R, then the $(m-1)$th prefix sum of $\overleftarrow{L(R)}$ is at least $m(m-1)^2$. Thus, by considering the $(m-1)$th prefix sums of $\overleftarrow{L(T)}$ and $\overleftarrow{L(R)}$, we have

$$s(R) \geqslant \frac{m(m-1)^2}{m(m-1)^2-(m-2-r_m)} = 1+\frac{r_m}{m(m-1)}.$$

If the big job is scheduled on one machine together with at least one other job, then the makespan of schedule R is at least $(m-1)+(m-1)^2+r_m$. Thus, by considering the makespans of S and R, we have $s(R) \geqslant 1+\frac{r_m}{m(m-1)}$. It follows that

$$\mathrm{WAR}(Pm(\mathrm{NP})) \geqslant 1+\frac{r_m}{m(m-1)} > 1 form \geqslant 3.$$

To establish the upper of WAR(Pm(NP)), we first present a simple but useful lemma.

Lemma 2.1 *Let X and Y be two vectors of n dimensions, and let X' and Y' be two vectors of two dimensions. If $X \preceq_s Y$ and $X' \preceq_s Y'$, then $(X, X') \preceq_s (Y, Y')$.*

Proof Suppose that $X' = (x_1, x_2)$ and $Y' = (y_1, y_2)$. Without loss of generality, we may further assume that $x_1 \geqslant x_2$ and $y_1 \geqslant y_2$. Then $x_1 \leqslant y_1$ and $x_1 + x_2 \leqslant y_1 + y_2$. Let $Z_x = (X, X')$ and $Z_y = (Y, Y')$. For $Z \in \{Z_x, Z_y\}$, we use $(\overleftarrow{Z})_k$ to denote the kth coordinate of $\overleftarrow{Z}$ and use $|\overleftarrow{Z}|_k$ to denote the sum of the first k coordinates of $\overleftarrow{Z}$ for $1 \leqslant k \leqslant n+2$. Similar notations are also used for X and Y. Given an index k with $1 \leqslant k \leqslant n+2$, we use $\delta(k, X')$ to denote the number of elements in $\{x_1, x_2\}$ included in the first k coordinates of $\overleftarrow{Z_x}$ and $\delta(k, Y')$ the number of elements in $\{y_1, y_2\}$ included in the first k coordinates of $\overleftarrow{Z_y}$. Then $0 \leqslant \delta(k, X'), \delta(k, Y') \leqslant 2$.

If $\delta(k, X') = \delta(k, Y')$, then we clearly have $|\overleftarrow{Z_x}|_k \leqslant |\overleftarrow{Z_y}|_k$.

If $\delta(k, X') = 0$, then $|\overleftarrow{Z_x}|_k = |\overleftarrow{X}|_k \leqslant |\overleftarrow{Y}|_k \leqslant |\overleftarrow{Z_y}|_k$.

If $\delta(k, Y') = 0$ and $\delta(k, X') \geqslant 1$, we suppose that x_1 is the ith coordinate of $\overleftarrow{Z_x}$. Then, for each j with $i \leqslant j \leqslant k$, $(\overleftarrow{Z_x})_j \leqslant x_1 \leqslant y_1 \leqslant (\overleftarrow{Z_y})_j$. Consequently, $|\overleftarrow{Z_x}|_k = |\overleftarrow{X}|_{i-1} + \sum_{i \leqslant j \leqslant k} (\overleftarrow{Z_x})_j \leqslant |\overleftarrow{Y}|_{i-1} + \sum_{i \leqslant j \leqslant k} (\overleftarrow{Z_y})_j = |\overleftarrow{Z_y}|_k$.

If $\delta(k, X') = 2$ and $\delta(k, Y') = 1$, then $(\overleftarrow{Y})_{k-1} \geqslant y_2$. Thus, $|\overleftarrow{Z_x}|_k = |\overleftarrow{X}|_{k-2} + x_1 + x_2 \leqslant |\overleftarrow{Y}|_{k-2} + y_1 + y_2 \leqslant |\overleftarrow{Y}|_{k-1} + y_1 = |\overleftarrow{Z_y}|_k$.

If $\delta(k, X') = 1$ and $\delta(k, Y') = 2$, then $(\overleftarrow{Y})_{k-1} \leqslant y_2$. Thus, $|\overleftarrow{Z_x}|_k = |\overleftarrow{X}|_{k-1} + x_1 \leqslant |\overleftarrow{Y}|_{k-1} + y_1 \leqslant |\overleftarrow{Y}|_{k-2} + y_1 + y_2 = |\overleftarrow{Z_y}|_k$.

The above discussion covers all possibilities. Then the lemma follows.

Theorem 2.1 WAR$(Pm(\text{NP})) \leqslant \frac{3}{2}$ *for* $m \geqslant 4$ *and* WAR$(P3(\text{NP})) \leqslant \sqrt{5} - 1 \approx 1.236$.

Proof Consider an instance of n jobs on $m \geqslant 4$ identical machines with $\mathcal{J} = \{J_1, J_2, \cdots, J_n\}$ and $\mathcal{M} = \{M_1, M_2, \cdots, M_m\}$. We assume that $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n$. Let S be a schedule produced by LPT algorithm (which is the LS algorithm with the jobs being given in the LPT order) such that $L_1^S \geqslant L_2^S \geqslant \cdots \geqslant L_m^S$. Then $L(S) = \overleftarrow{L(S)} = (L_1^S, L_2^S, \cdots, L_m^S)$. If $n \leqslant m$, it is easy to verify that $s(S) = 1$. Hence, we assume in the following that $n \geqslant m + 1$. Then some machine has at least two jobs in S.

Let i_0 be the smallest index such that either M_{i_0+1} has at least three jobs in S, or M_{i_0+1} has exactly two jobs in S and the size of the shorter job on M_{i_0+1} is at most half of the size of the longer job on M_{i_0+1}. If there is no such index, we set $i_0 = m$. Then $i_0 \geqslant 0$, and in the case $i_0 \geqslant 1$, each of $M_1, M_2, \cdots, M_{i_0}$ has at most two jobs in S. Let J_k be the shortest job scheduled on $M_1, M_2, \cdots, M_{i_0}$ and set $\mathcal{J}_k = \{J_1, J_2, \cdots, J_k\}$. Then $\mathcal{J}_k$ contains the jobs scheduled on $M_1, M_2, \cdots, M_{i_0}$. We use $M_{k'}$ to denote the machine occupied by J_k in S. Let T be the schedule derived from S by deleting $J_{k+1}, J_{k+2}, \cdots, J_n$. Then T is an LPT schedule for $\mathcal{J}_k$ with $L_i^T = L_i^S$, $i = 1, 2, \cdots, i_0$. We claim that $s(T) = 1$.

In the case $i_0 = 0$, the claim holds trivially. Hence, we assume in the following that $i_0 \geqslant 1$.

If each of $M_1, M_2, \cdots, M_{i_0}$ has only one job in S, then $i_0 = k \leqslant m$ and it is easy to see that $s(T) = 1$.

Suppose in the following that at least one of $M_1, M_2, \cdots, M_{i_0}$ has exactly two jobs in S. Then $m + 1 \leqslant k \leqslant 2m$ and the machine $M_{k'}$ has exactly two jobs, say J_t and J_k, in S. Note that there are at most two jobs on each machine in T. (Otherwise, some machine M_i with $i \geqslant i_0 + 1$ has $r \geqslant 3$ jobs, say $J_{h_1}, J_{h_2}, \cdots, J_{h_r}$, in T. By LPT algorithm, $p_t \geqslant \sum_{j=1}^{r-1} p_{h_j} \geqslant 2p_k$, contradicting the choice of i_0.) From the LPT algorithm, we have $t = 2m + 1 - k$. By the choice of i_0, we have $p_k > \frac{1}{2} p_{2m+1-k}$.

Let R be an arbitrary schedule for $\mathcal{J}_k$. If each machine has at most two jobs in R, we set $R_1 = R$. If some machine M_x has at least three jobs in R, by the pigeonhole principle, a certain machine M_y has either no job or exactly one job in $\{J_{2m+1-k}, J_{2m+2-k}, \cdots, J_k\}$. Let R' be the schedule obtained from R by moving the shortest job, say $J_{x'}$, on M_x to M_y. Then $L_x^{R'} \geqslant 2p_k > p_{2m+1-k} \geqslant L_y^R$ and $L_y^{R'} = L_y^R + p_{x'} \geqslant L_y^R$. Note that $L_x^R \geqslant L_x^{R'}$, $L_y^{R'} \geqslant L_y^R$ and $L_x^R + L_y^R = L_x^{R'} + L_y^{R'}$. Then we have $L(R') \preceq_s L(R)$ by Lemma 2.1. This procedure is repeated until we obtain a schedule R_1 so that each machine has at most two jobs in R_1. Then we have $L(R_1) \preceq_s L(R)$.

If $J_1, J_2, \cdots, J_m$ are processed on distinct machines, respectively, in R_1, we set $R_2 = R_1$. If some machine M_x has two jobs $J_{x'}, J_{x''} \in \{J_1, J_2, \cdots, J_m\}$ in R_1, by the pigeonhole principle, a certain machine M_y is occupied by at most two jobs in $\{J_m, J_{m+1}, \cdots, J_k\}$. Suppose that $p_{x'} \geqslant p_{x''}$ and $J_{y'}$ is the shorter job on M_y. Let R_1' be the schedule obtained from R_1 by shifting $J_{x''}$ to M_y and shifting $J_{y'}$ to M_x. Then $L_x^{R_1} \geqslant L_x^{R_1'}$, $L_y^{R_1'} \geqslant L_y^{R_1}$ and $L_x^{R_1} + L_y^{R_1} = L_x^{R_1'} + L_y^{R_1'}$. Consequently, by Lemma 2.1, $L(R_1') \preceq_s L(R_1)$. This procedure is repeated until we obtain a schedule R_2 so that $J_1, J_2, \cdots, J_m$ are processed on distinct machines, respectively, in R_2. Then we have $L(R_2) \preceq_s L(R_1)$.

Without loss of generality, we assume that J_j is processed on M_j in R_2, $1 \leqslant j \leqslant m$. Let $t = k - m$. Then the t jobs $J_{m+1}, J_{m+2}, \cdots, J_k$ are processed on t distinct machines in R_2. For convenience, we add another $m - t$ dummy jobs with sizes 0 in R_2 so that each machine has exactly two jobs. We define a sequence of t schedules $R_2^{(1)}, R_2^{(2)}, \cdots, R_2^{(t)}$ for $\mathcal{J}_k$ by the following way.

Initially we set $R_2^{(0)} = R_2$. For each i from 1 to t, the schedule $R_2^{(i)}$ is obtained from $R_2^{(i-1)}$ by exchanging the shorter job on M_{m-i+1} with job J_{m+i}.

We only need to show that $L(R_2^{(i)}) \preceq_s L(R_2^{(i-1)})$ for each i with $1 \leqslant i \leqslant t$. Note that the jobs $J_{m+1}, J_{m+2}, \cdots, J_{m+i-1}$ are processed on machines $M_m, M_{m-1}, \cdots, M_{m-i+2}$, respectively, in $R_2^{(i-1)}$. If J_{m+i} is processed on M_{m-i+1} in $R_2^{(i-1)}$, we have $R_2^{(i)} = R_2^{(i-1)}$, and so, $L(R_2^{(i)}) \preceq_s L(R_2^{(i-1)})$. Thus, we may assume that J_{m+i} is processed on a machine M_x with $x \leqslant m - i$ in $R_2^{(i-1)}$. Let J_j be the shorter job on M_{m-i+1} in $R_2^{(i-1)}$. Then $p_j \leqslant p_{m+i}$ and $p_x \geqslant p_{m-i+1}$. It is easy to see that $(L_x^{R_2^{(i)}}, L_{m-i+1}^{R_2^{(i)}}) = (p_x + p_j, p_{m-i+1} + p_{m+i}) \preceq_s (p_x + p_{m+i}, p_{m-i+1} + p_j) = (L_x^{R_2^{(i-1)}}, L_{m-i+1}^{R_2^{(i-1)}})$. Consequently, by Lemma 2.1, $L(R_2^{(i)}) \preceq_s L(R_2^{(i-1)})$.

The above discussion means that $L(R_2^{(t)}) \preceq_s L(R_2) \preceq_s L(R_1) \preceq_s L(R)$. Since $R_2^{(t)}$ is essentially an LPT schedule, we have $\overleftarrow{L(T)} = \overleftarrow{L(R_2^{(t)})}$, and so, $L(T) \preceq_s L(R_2^{(t)})$. It follows that $L(T) \preceq_s L(R)$. The claim follows.

Now let $\bar{S}$ be an arbitrary schedule for $\mathcal{J}$, and let $\bar{T}$ be the schedule for $\mathcal{J}_k$ derived from $\bar{S}$ by deleting jobs $J_{k+1}, J_{k+2}, \cdots, J_n$. Then $L(\bar{T}) \preceq_s L(\bar{S})$. Assume without loss of generality that $L_1^{\bar{S}} \geqslant L_2^{\bar{S}} \geqslant \cdots \geqslant L_m^{\bar{S}}$ and $L_{\pi(1)}^{\bar{T}} \geqslant L_{\pi(2)}^{\bar{T}} \geqslant \cdots \geqslant L_{\pi(m)}^{\bar{T}}$, where π is a permutation of $\{1, 2, \cdots, m\}$. For each i with $1 \leqslant i \leqslant i_0$, the above claim implies that $\sum_{j=1}^{i} L_j^S = \sum_{j=1}^{i} L_j^T \leqslant \sum_{j=1}^{i} L_{\pi(j)}^{\bar{T}} \leqslant \sum_{j=1}^{i} L_j^{\bar{S}}$.

Write $P = \sum_{j=1}^{n} p_j$, $Q = \sum_{i=1}^{i_0} L_i^S$ and $\bar{Q} = \sum_{i=1}^{i_0} L_i^{\bar{S}}$. Then $Q \leqslant \bar{Q}$. Note that, in the case $i_0 = 0$, we have $Q = \bar{Q} = 0$. Let J_d be the last job scheduled on machine M_{i_0+1} in S. By the choice of i_0, $p_d \leqslant \frac{1}{2}(L_{i_0+1}^S - p_d)$. From the LPT algorithm, we have $L_{i_0+1}^S - p_d \leqslant L_j^S$, $j = i_0 + 1, i_0 + 2, \cdots, m$. Hence,

$$L_{i_0+1}^S \leqslant \frac{3}{2}(L_{i_0+1}^S - p_d) \leqslant \frac{3}{2} \cdot \frac{\sum_{j=i_0+1}^{m} L_j^S}{m - i_0} = \frac{3}{2} \cdot \frac{1}{m - i_0}(P - Q).$$

Thus, for each i with $i_0 + 1 \leqslant i \leqslant m$, we have

$$\sum_{j=1}^{i} L_j^S \leqslant Q + (i - i_0) L_{i_0+1}^S \leqslant Q + \frac{3}{2} \cdot \frac{i - i_0}{m - i_0}(P - Q), \tag{2.1}$$

and

$$\sum_{j=1}^{i} L_j^{\bar{S}} \geqslant \bar{Q} + (i - i_0) \frac{\sum_{j=m}^{i_0+1} L_j^{\bar{S}}}{m - i_0} = \bar{Q} + \frac{i - i_0}{m - i_0}(P - \bar{Q}) \geqslant Q + \frac{i - i_0}{m - i_0}(P - Q). \tag{2.2}$$

From (2.1) and (2.2), we conclude that $\sum_{j=1}^{i} L_j^S \leqslant \frac{3}{2} \sum_{j=1}^{i} L_j^{\bar{S}}$. Consequently, $s(S) \leqslant \frac{3}{2}$. It follows that $\mathrm{WAR}(Pm(\mathrm{NP})) \leqslant \frac{3}{2}$ for $m \geqslant 4$.

Now let us consider problem $P3(\mathrm{NP})$. Let $\mathcal{I}$ be an instance. Denote by S the schedule which minimizes the makespan and by T the schedule which maximizes the machine cover. Without loss of generality, we may assume that

$$L_1^S \geqslant L_2^S \geqslant L_3^S, L_1^T \geqslant L_2^T \geqslant L_3^T \text{ and } L_1^S + L_2^S + L_3^S = L_1^T + L_2^T + L_3^T = 1.$$

Then

$$s(S) = \frac{L_1^S + L_2^S}{L_1^T + L_2^T} \text{ and } s(T) = \frac{L_1^T}{L_1^S}.$$

Consequently,

$$s^*(\mathcal{I}) \leqslant \min\{\frac{L_1^S + L_2^S}{L_1^T + L_2^T}, \frac{L_1^T}{L_1^S}\}.$$

Note that

$$L_1^T = 1 - L_2^T - L_3^T \leqslant 1 - 2L_3^T \text{ and } L_1^S \geqslant \frac{L_1^S + L_2^S}{2} = \frac{1 - L_3^S}{2}.$$

Then

$$s^*(\mathcal{I}) \leqslant \min\{\frac{1 - L_3^S}{1 - L_3^T}, \frac{1 - 2L_3^T}{\frac{1 - L_3^S}{2}}\}.$$

Set

$$x = 1 - 2L_3^T \text{ and } t = 1 - L_3^S.$$

Then $\frac{2}{3} \leqslant t \leqslant 1$ and

$$s^*(\mathcal{I}) \leqslant \min\{\frac{2t}{1+x}, \frac{2x}{t}\}.$$

If $x \geqslant \frac{\sqrt{1+4t^2}-1}{2}$, then

$$s^*(\mathcal{I}) \leqslant \frac{2t}{1+x} \leqslant \frac{2t}{1+\frac{\sqrt{1+4t^2}-1}{2}} = \frac{\sqrt{1+4t^2}-1}{t}.$$

If $x \leqslant \frac{\sqrt{1+4t^2}-1}{2}$, then

$$s^*(\mathcal{I}) \leqslant \frac{2x}{t} \leqslant \frac{\sqrt{1+4t^2}-1}{t}.$$

Note that $\frac{\sqrt{1+4t^2}-1}{t} \leqslant \sqrt{5}-1$ for all t with $\frac{2}{3} \leqslant t \leqslant 1$. It follows that $s^*(\mathcal{I}) \leqslant \sqrt{5}-1$. The result follows.

For problem Pm(PP), [5] presented an optimal algorithm to generate a schedule which minimizes the makespan. A slight modification of the algorithm can generate a schedule S with $s(S) = 1$.

Algorithm 1 (with input $\mathcal{M}$ and $\mathcal{J}$)

Step 1 Find the longest job J_h in $\mathcal{J}$. If $p_h \leqslant \frac{\sum_{J_j \in \mathcal{J}} p_j}{|\mathcal{M}|}$, then apply McNaughton's algorithm to assign all jobs in $\mathcal{J}$ to the machines in $\mathcal{M}$ evenly, and stop. Otherwise, assign J_h to an arbitrary machine $M_i \in \mathcal{M}$.

Step 2 Reset $\mathcal{M} = \mathcal{M} \backslash \{M_i\}$ and $\mathcal{J} = \mathcal{J} \backslash \{J_h\}$. If $|\mathcal{J}| \neq 0$, then go back to 1. Otherwise, stop.

Lemma 2.2 *Assume $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n$ and let S be a preemptive schedule with $L_1^S \geqslant L_2^S \geqslant \cdots \geqslant L_m^S$. Then $\sum_{i=1}^k p_i \leqslant \sum_{i=1}^k L_i^S$, $k = 1, 2, \cdots, m$.*

Proof Let $\mathcal{J}_k = \{J_1, J_2, \cdots, J_k\}$. Then at most k jobs in $\mathcal{J}_k$ can be processed simultaneously in the time interval $[0, L_k^S]$ and at most $k - i$ jobs of $\mathcal{J}_k$ can be processed simultaneously in the time interval $[L_{k+1-i}^S, L_{k-i}^S]$, $i = 1, 2, \cdots, k-1$. Therefore,

$$\sum_{i=1}^{k} p_i \leqslant kL_k^S + \sum_{i=1}^{k-1}(k-i)(L_{k-i}^S - L_{k+1-i}^S) = \sum_{i=1}^{k} L_i^S.$$

The lemma follows.

Theorem 2.2 WAR$(Pm$(PP)$) = 1$.

Proof Assume that $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_n$. Let i_0 be the largest job index such that $p_{i_0} > \frac{\sum_{j=i_0}^{n} p_j}{m-i_0+1}$. If there is no such index, we set $i_0 = 0$. Let S be the preemptive schedule generated by Algorithm 1 with $L_1^S \geqslant L_2^S \geqslant \cdots \geqslant L_m^S$. Then we have

$$L_i^S = p_i, \quad i = 1, 2, \cdots, i_0, \tag{2.3}$$

and

$$L_i^S = \frac{\sum_{j=i_0+1}^{n} p_j}{m - i_0}, \quad i = i_0 + 1, i_0 + 2, \cdots, m. \tag{2.4}$$

Let T be a preemptive schedule with $L_1^T \geqslant L_2^T \geqslant \cdots \geqslant L_m^T$. If $1 \leqslant k \leqslant i_0$, by Lemma 2.2 and (2.3), we have $\sum_{i=1}^{k} L_i^S = \sum_{i=1}^{k} p_i \leqslant \sum_{i=1}^{k} L_i^T$. If $i_0+1 \leqslant k \leqslant m$, by noting that $\sum_{i=1}^{i_0} L_i^S \leqslant \sum_{i=1}^{i_0} L_i^T$, we have

$$\begin{aligned}
\sum\nolimits_{i=1}^{k} L_i^S &= \sum\nolimits_{i=1}^{i_0} L_i^S + \frac{k-i_0}{m-i_0}\left(\sum\nolimits_{i=1}^{n} p_i - \sum\nolimits_{i=1}^{i_0} L_i^S\right) \\
&= \left(1 - \frac{k-i_0}{m-i_0}\right)\sum\nolimits_{i=1}^{i_0} L_i^S + \frac{k-i_0}{m-i_0}\sum\nolimits_{i=1}^{n} p_i \\
&\leqslant \left(1 - \frac{k-i_0}{m-i_0}\right)\sum\nolimits_{i=1}^{i_0} L_i^T + \frac{k-i_0}{m-i_0}\sum\nolimits_{i=1}^{n} p_i \\
&= \sum\nolimits_{i=1}^{i_0} L_i^T + \frac{k-i_0}{m-i_0}\left(\sum\nolimits_{i=1}^{n} p_i - \sum\nolimits_{i=1}^{i_0} L_i^T\right) \\
&\leqslant \sum\nolimits_{i=1}^{k} L_i^T.
\end{aligned}$$

Hence, $\mathrm{WAR}(Pm(\mathrm{PP})) = 1$. The result follows.

For problem $Pm(\mathrm{FP})$, the schedule S averagely processing each job on all machines clearly has $s(S) = 1$. Then we have

Theorem 2.3 $\mathrm{WAR}(Pm(\mathrm{FP})) = 1$.

3 Related Machines

Assume that $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_m$. We first present the exact expression of $\mathrm{WAR}(Qm(\mathrm{FP}))$ on the machine speeds $s_1, s_2, \cdots, s_m$. Then we show that it is a lower bound for $\mathrm{WAR}(Qm(\mathrm{PP}))$ and $\mathrm{WAR}(Qm(\mathrm{NP}))$.

The fractional processing mode means that all jobs can be merged into a single job with processing time equal to the sum of processing times of all jobs. Thus, we may assume that $\mathcal{I}$ is an instance of $Qm(\mathrm{FP})$ with just one job $J_{\mathcal{I}}$. Suppose without loss of generality that $p_{\mathcal{I}} = 1$. A schedule S of $\mathcal{I}$ is called *regular* if $L_1^S \geqslant L_2^S \geqslant \cdots \geqslant L_m^S$. Then $\overleftarrow{L}(S) = L(S)$ if S is regular. The following lemma can be observed from the basic mathematical knowledge.

Lemma 3.1 *Suppose that* $x_1 \geqslant x_2 \geqslant \cdots \geqslant x_n \geqslant 0$ *and* $y_1 \geqslant y_2 \geqslant \cdots \geqslant y_n \geqslant 0$. *Then* $\sum_{i=1}^{n} x_i y_{\pi(i)} \leqslant \sum_{i=1}^{n} x_i y_i$ *for any permutation* π *of* $\{1, 2, \cdots, n\}$.

Lemma 3.2 *For any schedule* T *of* $\mathcal{I}$, *there exists a regular schedule* S *such that* $L(S) \preceq_c \overleftarrow{L(T)}$.

Proof Let T be a schedule of $\mathcal{I}$ and π a permutation of $\{1, 2, \cdots, m\}$ such that $L^T_{\pi(1)} \geqslant L^T_{\pi(2)} \geqslant \cdots \geqslant L^T_{\pi(m)}$. By Lemma 3.1, $\sum_{i=1}^{m} s_i L^T_{\pi(i)} \geqslant \sum_{i=1}^{m} s_{\pi(i)} L^T_{\pi(i)} \geqslant 1$. Let i_0 be the smallest machine index such that $\sum_{i=1}^{i_0} s_i L^T_{\pi(i)} \geqslant 1$. Let S be the schedule in which a part of processing time $s_i L^T_{\pi(i)}$ is assigned to M_i, $i = 1, 2, \cdots, i_0 - 1$, and the rest part of processing time $1 - \sum_{i=1}^{i_0-1} s_i L^T_{\pi(i)}$ is assigned to M_{i_0}. Then we have $L^S_i = L^T_{\pi(i)}$, for $i = 1, 2, \cdots, i_0 - 1$,

$$L^S_{i_0} = \frac{1 - \sum_{i=1}^{i_0-1} s_i L^T_{\pi(i)}}{s_{i_0}} \leqslant \frac{\sum_{i=1}^{i_0} s_i L^T_{\pi(i)} - \sum_{i=1}^{i_0-1} s_i L^T_{\pi(i)}}{s_{i_0}} = L^T_{\pi(i_0)},$$

and $L^S_i = 0 \leqslant L^T_{\pi(i)}$ for $i = i_0 + 1, i_0 + 2, \cdots, m$. It can be observed that S is regular and $L(S) \preceq_c \overleftarrow{L(T)}$. The lemma follows.

Let $f(i)$ be the infimum of the sum of the first i coordinates of $\overleftarrow{L(T)}$ in all feasible schedule T of $\mathcal{I}$, $i = 1, 2, \cdots, m$. By Lemma 3.2, we have $f(i) = \inf\{\sum_{k=1}^{i} L^S_k : S \text{ is regular}\}$, $i = 1, 2, \cdots, m$. Then, for each schedule T of $\mathcal{I}$ with $L^T_{\pi(1)} \geqslant L^T_{\pi(2)} \geqslant \cdots \geqslant L^T_{\pi(m)}$ for some permutation π of $\{1, 2, \cdots, m\}$, we have

$$s(T) = \max_{1 \leqslant i \leqslant m} \left\{ \frac{\sum_{k=1}^{i} L^{\tau}_{\pi(k)}}{f(i)} \right\}. \tag{3.1}$$

The following lemma gives the exact expression for each $f(i)$.

Lemma 3.3 $f(i) = \begin{cases} \frac{i}{\sum_{k=1}^{m} s_k}, & \text{if } i \leqslant \frac{\sum_{k=1}^{m} s_k}{s_1}, \\ \frac{1}{s_1}, & \text{if } i > \frac{\sum_{k=1}^{m} s_k}{s_1}. \end{cases}$

Proof Fix index i and let S be a regular schedule. Then we have

$$L^S_1 \geqslant L^S_2 \geqslant \cdots \geqslant L^S_m \tag{3.2}$$

and

$$\sum_{i=1}^{m} s_i L^S_i \geqslant 1. \tag{3.3}$$

So we only need to find a regular schedule S meeting (3.2) and (3.3) such that $\sum_{k=1}^{i} L^S_k$ reaches the minimum.

If $i \leqslant \frac{\sum_{k=1}^{m} s_k}{s_1}$, by (3.2) and (3.3), we have

$$\begin{aligned}
&\sum_{t=1}^{i}\left(\frac{\sum_{k=1}^{m} s_k}{i}\right) L_t^S \\
&= \sum_{t=1}^{i} s_t L_t^S + \sum_{t=1}^{i}\left(\frac{\sum_{k=1}^{m} s_k}{i} - s_t\right) L_t^S \\
&\geqslant \sum_{t=1}^{i} s_t L_t^S + \sum_{t=1}^{i}\left(\frac{\sum_{k=1}^{m} s_k}{i} - s_t\right) L_{i+1}^S \\
&= \sum_{t=1}^{i} s_t L_t^S + \left(\sum_{t=i+1}^{m} s_t\right) L_{i+1}^S \\
&\geqslant \sum_{t=1}^{i} s_t L_t^S + \sum_{t=i+1}^{m} s_t L_t^S \\
&= \sum_{t=1}^{m} s_t L_t^S \\
&\geqslant 1.
\end{aligned}$$

The equality holds if and only if $L_1^S = L_2^S = \cdots = L_m^S = \frac{1}{\sum_{k=1}^{m} s_k}$. Then the regular schedule S can be defined by the way that a part of processing time $\frac{s_k}{\sum_{i=1}^{m} s_i}$ is assigned to $M_k, k = 1, 2, \cdots, m$. Thus, $f(i) = \frac{i}{\sum_{k=1}^{m} s_k}$.

If $i > \frac{\sum_{k=1}^{m} s_k}{s_1}$, we can similarly deduce

$$\begin{aligned}
\sum_{k=1}^{i} s_1 L_k^S &= \sum_{k=1}^{i} s_k L_k^S + \sum_{k=1}^{i} (s_1 - s_k) L_k^S \\
&\geqslant \sum_{k=1}^{i} s_k L_k^S + \sum_{k=1}^{i} (s_1 - s_k) L_i^S \\
&= \sum_{k=1}^{i} s_k L_k^S + \left(i s_1 - \sum_{k=1}^{i} s_k\right) L_i^S \\
&\geqslant \sum_{k=1}^{i} s_k L_k^S + \left(\sum_{k=1}^{m} s_k - \sum_{k=1}^{i} s_k\right) L_i^S \\
&\geqslant \sum_{k=1}^{i} s_k L_k^S + \sum_{k=i+1}^{m} s_k L_k^S \\
&= \sum_{k=1}^{m} s_k L_k^S \\
&\geqslant 1.
\end{aligned}$$

The equality holds if and only if $L_1^S = \frac{1}{s_1}$, $L_2^S = \cdots = L_m^S = 0$. Then the regular schedule S can be defined by the way that $J_{\mathcal{I}}$ is scheduled totally on M_1 in S. Thus, $f(i) = \frac{1}{s_1}$. The lemma follows.

By Lemma 3.2, $s^*(\mathcal{I}) = \inf\{s(S) : S \text{ is regular}\}$. For each regular schedule S, by (3.1) and Lemma 3.3, we have $\sum_{k=1}^{i} L_k^S \leqslant s(L(S)) f(i)$ for $i = 1, 2, \cdots, m$.

Let $s_{m+1} = 0$ and $\frac{\sum_{i=1}^{m} s_i}{s_1} = t + \Delta$, where t with $1 \leqslant t \leqslant m$ is a positive integer and $0 \leqslant \Delta < 1$. By Lemma 3.3, we have

$$i \cdot \frac{s(L(S))}{\sum_{k=1}^{m} s_k} \geqslant \sum_{k=1}^{i} L_k^S, \quad i = 1, 2, \cdots, t \tag{3.4}$$

and

$$\frac{s(L(S))}{s_1} \geqslant \sum_{k=1}^{i} L_k^S, \quad i = t+1, t+2, \cdots, m. \tag{3.5}$$

From (3.4) and (3.5), we have

$$\sum_{i=1}^{t}(s_i - s_{i+1}) \cdot i \cdot \frac{s(L(S))}{\sum_{i=1}^{m} s_i} + \sum_{i=t+1}^{m}(s_i - s_{i+1})\frac{s(L(S))}{s_1} \geqslant \sum_{i=1}^{t}(s_i - s_{i+1})\sum_{t=1}^{i} L_t^S$$
$$+ \sum_{i=t+1}^{m}(s_i - s_{i+1})\sum_{t=1}^{i} L_t^S = \sum_{i=1}^{m} s_i L_i^S = 1.$$

Hence, $s(S) \geqslant \frac{\sum_{i=1}^m s_i}{\sum_{i=1}^t s_i + \left(\frac{\sum_{i=1}^m s_i}{s_1} - t\right)s_{t+1}} = \frac{\sum_{i=1}^m s_i}{\sum_{i=1}^t s_i + \Delta s_{t+1}}$. Note that the equality holds if and only if $L_1^S = L_2^S = \cdots = L_t^S = \frac{1}{\sum_{i=1}^t s_i + \Delta s_{t+1}}$, $L_{t+1}^S = \frac{\Delta}{\sum_{i=1}^t s_i + \Delta s_{t+1}}$ and $L_{t+2}^S = L_{t+3}^S = \cdots = L_m^S = 0$. Then the corresponding regular schedule S can be defined by the way that a part of processing time $\frac{s_i}{\sum_{k=1}^t s_k + \Delta s_{t+1}}$ is assigned to M_i, $i = 1, 2, \cdots, t$, and the rest part of processing time $\frac{\Delta s_{t+1}}{\sum_{i=1}^t s_i + \Delta s_{t+1}}$ is assigned to M_{t+1}. Hence, $s^*(\mathcal{I}) = \frac{\sum_{i=1}^m s_i}{\sum_{i=1}^t s_i + \Delta s_{t+1}}$. Consequently, $WAR(Qm(FP)) = \frac{\sum_{i=1}^m s_i}{\sum_{i=1}^t s_i + \Delta s_{t+1}}$ if the machine speeds are fixed.

If machine speeds are parts of the input, by the fact that $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_m$, we have

$$\frac{\sum_{i=2}^t s_i + \Delta s_{t+1}}{t - 1 + \Delta} \geqslant \frac{\sum_{i=2}^m s_i}{m-1}. \tag{3.6}$$

Let $\theta = \frac{\sum_{i=2}^m s_i}{m-1}$ and $\vartheta = \frac{s_1}{\theta} > 1$. Then

$$t + \Delta = \frac{\sum_{i=1}^m s_i}{s_1} = \frac{s_1 + (m-1)\theta}{s_1} = \frac{\vartheta + m - 1}{\vartheta}. \tag{3.7}$$

Obviously, $\frac{m}{\vartheta - 1} + (\vartheta - 1) \geqslant 2\sqrt{\frac{m}{\vartheta-1}(\vartheta - 1)} = 2\sqrt{m}$. By (3.6) and (3.7), we have

$$\frac{\sum_{i=1}^m s_i}{\sum_{i=1}^t s_i + \Delta s_{t+1}}$$
$$= \frac{s_1 + (m-1)\frac{\sum_{i=2}^m s_i}{m-1}}{s_1 + (t-1+\Delta)\frac{\sum_{i=2}^t s_i + \Delta s_{t+1}}{t-1+\Delta}}$$

$$
\begin{aligned}
&\leqslant \frac{s_1 + (m-1)\frac{\sum_{i=2}^{m} s_i}{m-1}}{s_1 + (t-1+\Delta)\frac{\sum_{i=2}^{m} s_i}{m-1}} \\
&= 1 + \frac{m-1}{(\frac{m}{\vartheta-1} + (\vartheta-1)) + 2} \\
&\leqslant 1 + \frac{m-1}{2\sqrt{m}+2} = \frac{\sqrt{m}+1}{2}.
\end{aligned}
$$

So we have $s^*(\mathcal{I}) \leqslant \frac{\sqrt{m}+1}{2}$, and therefore, $\mathrm{WAR}(Qm(\mathrm{FP})) \leqslant \frac{\sqrt{m}+1}{2}$.

To show that $WAR(Qm(\mathrm{FP})) = \frac{\sqrt{m}+1}{2}$, we consider the following instance $\mathcal{I}$ with $p_{\mathcal{I}} = 1$, $s_1 = s = \sqrt{m} + 1 > 1$ and $s_2 = s_3 = \cdots = s_m = 1$. Let S be a regular schedule and write $x = sL_1^S$. Then $\sum_{t=2}^{m} L_t^S = 1 - x$. By Lemma 3.3 and (3.1), we have

$$
\begin{aligned}
&s(S) \\
&\geqslant \max\left\{\frac{L_1^S}{f(1)}, \frac{\sum_{i=1}^{m} L_i^S}{f(m)}\right\} \\
&= \max\left\{\frac{x(s+m-1)}{s}, x + s(1-x)\right\} \\
&\geqslant \frac{s^2+sm-s}{s^2+m-1} \\
&= \frac{\sqrt{m}+1}{2},
\end{aligned}
$$

where the inequality follows from the fact that $\frac{x(s+m-1)}{s}$ is an increasing function in x, while $x + s(1-x)$ is a decreasing function in x and they meet with $\frac{s^2+sm-s}{s^2+m-1}$ when $x = \frac{s^2}{s^2+m-1}$. Then $s^*(\mathcal{I}) \geqslant \frac{\sqrt{m}+1}{2}$. Consequently, $\mathrm{WAR}(Qm(\mathrm{FP})) = \frac{\sqrt{m}+1}{2}$.

The above discussion leads to the following conclusion.

Theorem 3.1 *If the machine speeds $s_1, s_2, \cdots, s_m$ are fixed, then* $\mathrm{WAR}(Qm(\mathrm{FP})) = \frac{\sum_{i=1}^{m} s_i}{\sum_{i=1}^{t} s_i + \Delta s_{t+1}}$, *where* $\frac{\sum_{i=1}^{m} s_i}{s_1} = t + \Delta$, *$t$ with $1 \leqslant t \leqslant m$ is a positive integer and* $0 \leqslant \Delta < 1$. *Alternatively, if the machine speeds $s_1, s_2, \cdots, s_m$ are parts of the input, then $WAR(Qm(\mathrm{FP})) = \frac{\sqrt{m}+1}{2}$.*

Lemma 3.4 *If the machine speeds $s_1, s_2, \cdots, s_m$ are fixed, then* $\mathrm{WAR}(Qm(\mathrm{NP})) \geqslant \mathrm{WAR}(Qm(\mathrm{FP}))$ *and* $\mathrm{WAR}(Qm(\mathrm{PP})) \geqslant \mathrm{WAR}(Qm(\mathrm{FP}))$.

Proof We only consider the non-preemptive processing mode. For the preemptive processing mode, the result can be similarly proved. Given a schedule S, we denote by π^S the permutation of $\{1, 2, \cdots, m\}$ such that $L_{\pi^S(1)}^S \geqslant L_{\pi^S(2)}^S \geqslant \cdots \geqslant L_{\pi^S(m)}^S$.

Suppose without loss of generality that $s_m = 1$. Write $\eta = \mathrm{WAR}(Qm(\mathrm{NP}))$. Let $\mathcal{I}$ be an instance of $Q_m(\mathrm{FP})$ with only one job $J_{\mathcal{I}}$ of processing time 1. For each i, set

$f(i)$ to be the infimum of $\sum_{k=1}^{i} L^S_{\pi^S(k)}$ of schedule S over all fractional schedules of $\mathcal{I}$. We only need to show that $s^*(\mathcal{I}) \leqslant \eta$.

Assume to the contrary that $s^*(\mathcal{I}) > \eta$. Let $\varepsilon > 0$ be a sufficiently small number such that $\eta(f(i)+i\varepsilon) < s^*(\mathcal{I})f(i), i = 1, 2, \cdots, m$. Let $\mathcal{H}$ be an instance of Q_m(NP) such that the total processing time of jobs is equal to 1 and the processing time of each job is at most ε. For each i, let $g(i)$ be the infimum of $\sum_{k=1}^{i} L^S_{\pi^S(k)}$ of schedule S over all feasible schedules of $\mathcal{H}$. We assert that

$$g(i) \leqslant f(i) + i\varepsilon, \quad i = 1, 2, \cdots, m. \tag{3.8}$$

To the end, let S_i be the regular schedule of $\mathcal{I}$ such that $\sum_{k=1}^{i} L_k^{S_i} = f(i)$, $i = 1, 2, \cdots, m$. Fix index i, we construct a non-preemptive schedule S of $\mathcal{H}$ such that $\sum_{k=1}^{i} L^S_{\pi^S(k)} \leqslant f(i)+i\varepsilon$. This leads to $g(i) \leqslant \sum_{k=1}^{i} L^S_{\pi^S(k)} \leqslant f(i)+i\varepsilon$ and therefore proves the assertion. The construction of S is stated as follows. First, we assign jobs to M_i one by one until $L_1^S \geqslant L_1^{S_i}$. Then we assign the rest jobs to M_2 one by one until $L_2^S \geqslant L_2^{S_i}$. This procedure is repeated until all jobs are assigned. According to the construction of S, we have $L_k^S \leqslant L_k^{S_i} + \frac{\varepsilon}{s_k} \leqslant L_k^{S_i} + \varepsilon$, $k = 1, 2, \cdots, m$. Note that $L_1^{S_i} \geqslant L_2^{S_i} \geqslant \cdots \geqslant L_m^{S_i}$. Then $\sum_{k=1}^{i} L^S_{\pi^S(k)} \leqslant \sum_{k=1}^{i}(L^{S_i}_{\pi^S(k)}+\varepsilon) \leqslant \sum_{k=1}^{i} L_k^{S_i}+i\varepsilon = f(i) + i\varepsilon$.

Let R be the schedule of $\mathcal{H}$ such that $s(R) = s^*(\mathcal{H})$. It can be observed that there exists a schedule T of $\mathcal{I}$ such that $L(T) \preceq_c L(R)$. Hence, for each i with $1 \leqslant i \leqslant m$, we have

$$\begin{aligned}
&\sum_{k=1}^{i} L^T_{\pi^T(k)} \leqslant \sum_{k=1}^{i} L^R_{\pi^T(k)} \leqslant \sum_{k=1}^{i} L^R_{\pi^R(k)} \leqslant s(R)g(i) \\
&\leqslant s^*(\mathcal{H})(f(i) + i\varepsilon) \leqslant \eta(f(i) + i\varepsilon) < s^*(\mathcal{I})f(i).
\end{aligned}$$

This contradicts the definition of $s^*(\mathcal{I})$. So $s^*(\mathcal{I}) \leqslant \eta$. The result follows.

By Theorem 3.1 and Lemma 3.4, the following theorem holds.

Theorem 3.2 *If the machine speeds $s_1, s_2, \cdots, s_m$ are fixed, then*

$$\mathrm{WAR}(\mathcal{P}) \geqslant \frac{\sum_{i=1}^{m} s_i}{\sum_{i=1}^{t} s_i + \Delta s_{t+1}} \text{ for } \mathcal{P} \in \{Qm(\mathrm{NP}), Qm(\mathrm{PP})\},$$

where $\frac{\sum_{i=1}^{m} s_i}{s_1} = t + \Delta$, t is a positive integer with $1 \leqslant t \leqslant m$ and $0 \leqslant \Delta < 1$. If the machine speeds $s_1, s_2, \cdots, s_m$ are parts of the input, then $\mathrm{WAR}(\mathcal{P}) \geqslant \frac{\sqrt{m}+1}{2}$ for $\mathcal{P} \in \{Qm(\mathrm{NP}), Qm(\mathrm{PP})\}$.

4 Unrelated Machines

Since Qm is a special version of Rm, from the results in the previous section, the weak simultaneous approximation ratio is at least $\frac{\sqrt{m}+1}{2}$ for each of $Rm(\text{NP})$, $Rm(\text{PP})$ and $Rm(\text{FP})$. The following lemma establishes an upper bound of the weak simultaneous approximation ratio for the three problems.

Lemma 4.1 $\text{WAR}(\mathcal{P}) \leqslant \sqrt{m}$ *for* $\mathcal{P} \in \{Rm(\text{NP}), Rm(\text{PP}), Rm(\text{FP})\}$.

Proof Let $\mathcal{I}$ be an instance of $R_m(\text{NP})$, $R_m(\text{PP})$ or $R_m(\text{FP})$. Let S be a schedule which minimizes the makespan with $L_1^S \geqslant L_2^S \geqslant \cdots \geqslant L_m^S$. Let $p_{[j]} = \min_{1\leqslant i\leqslant m}\{p_{ij}\}$.

If $L_1^S \leqslant \frac{\sum_{j=1}^n p_{[j]}}{\sqrt{m}}$, let T be a feasible schedule with $L_{\pi(1)}^T \geqslant L_{\pi(2)}^T \geqslant \cdots \geqslant L_{\pi(m)}^T$ for some permutation π of $\{1, 2, \cdots, m\}$. For each i, we have

$$\sum_{k=1}^{i} L_k^S \leqslant iL_1^S \leqslant \sqrt{m}\cdot\frac{i}{m}\sum_{j=1}^{n} p_{[j]} \leqslant \sqrt{m}\sum_{k=1}^{i} L_{\pi(k)}^T.$$

This means that $s^*(\mathcal{I}) \leqslant \sqrt{m}$.

If $L_1^S > \frac{\sum_{j=1}^n p_{[j]}}{\sqrt{m}}$, let R be the schedule, in which each job J_j is assigned to the machine M_i with $p_{ij} = p_{[j]}$. Let O be an arbitrarily feasible schedule, and let π_1 and π_2 be two permutations of $\{1, 2, \cdots, m\}$ such that $L_{\pi_1(1)}^R \geqslant L_{\pi_1(2)}^R \geqslant \cdots \geqslant L_{\pi_1(m)}^R$ and $L_{\pi_2(1)}^O \geqslant L_{\pi_2(2)}^O \geqslant \cdots \geqslant L_{\pi_2(m)}^O$. For each i, we have

$$\sum_{k=1}^{i} L_{\pi_1(k)}^R \leqslant \sum_{k=1}^{m} L_{\pi_1(k)}^R = \sum_{j=1}^{n} p_{[j]} < \sqrt{m}L_1^S \leqslant \sqrt{m}L_{\pi_2(1)}^O \leqslant \sqrt{m}\sum_{k=1}^{i} L_{\pi_2(k)}^O.$$

This also means that $s^*(\mathcal{I}) \leqslant \sqrt{m}$. The lemma follows.

Combining with the results of the previous section, we have the following theorem.

Theorem 4.1 *For each problem* $\mathcal{P} \in \{Qm(\text{NP}), Qm(\text{PP}), Qm(\text{FP}), Rm(\text{NP}), Rm(\text{PP}), Rm(\text{FP})\}$, *we have* $\frac{\sqrt{m}+1}{2} \leqslant \text{WAR}(\mathcal{P}) \leqslant \sqrt{m}$.

5 Conclusion

We introduced and studied the strong and weak simultaneous approximation ratios, denoted by $\text{SAR}(\mathcal{P})$ and $\text{WAR}(\mathcal{P})$, of various parallel machine scheduling problems $\mathcal{P}$. Since determining $\text{SAR}(\mathcal{P})$ is trivial for most standard problems, we mainly presented research on the values $\text{WAR}(\mathcal{P})$. Our contributions are summarized in Table 1.

For further research, it is worth studying to determine the exact value of $\text{WAR}(\mathcal{P})$ or improve the bounds of $\text{WAR}(\mathcal{P})$ for

$$\mathcal{P} \in \{Pm(\text{NP}), Qm(\text{NP}), Rm(\text{NP}), Qm(\text{PP}), Rm(\text{PP}), Rm(\text{FP})\}.$$

Acknowledgements The authors would like to thank the associate editor and two anonymous referees for their constructive comments and helpful suggestions.

References

[1] Csirik, J., Kellerer, H., Woeginger, G.: The exact LPT-bound of maximizing the minimum completion time. Oper. Res. Lett. **11**, 281–287 (1992)
[2] Deuermeyer, B.L., Friesen, D.K., Langston, A.M.: Scheduling to maximize the minimum processor finish time in a multiprocessor system. SIAM J. Discrete Math. **3**, 190–196 (1982)
[3] Graham, R.L.: Bounds for certain multiprocessing anomalies. Bell Syst. Tech. J. **45**, 1563–1581 (1966)
[4] Graham, R.L.: Bounds for multiprocessing timing anomalies. SIAM J. Appl. Math. **17**, 416–429 (1969)
[5] McNaughton, R.: Scheduling with deadlines and loss functions. Manag. Sci. **6**, 1–12 (1959)
[6] Bhargava, R., Goel, A., Meyerson, A.: Using approximate majorization to characterize protocol fairness. In: Proceedings of the 2001 ACM Special Interest Group for the computer systems performance evaluation community. International Conference on Measurement and Modeling of Computer Systems, pp. 330–331. ACM, New York (2001)
[7] Goel, A., Meyerson, A., Plotkin, S.: Combining fairness with throughput: online routing with multiple objectives. J. Comput. Syst. Sci. **63**, 62–79 (2001)
[8] Goel, A., Meyerson, A., Plotkin, S.: Approximate majorization and fair online load balancing. ACM Trans. Algorithms **1**, 338–349 (2005)
[9] Kleinberg, J., Rabani, Y., Tardos, É.: Fairness in routing and load balancing. J. Comput. Syst. Sci. **63**, 2–20 (2001)
[10] Kumar, A., Kleinberg, J.: Fairness measures for resource allocation. SIAM J. Comput. **36**, 657–680 (2006)