Minimizing Total Weighted Completion Time Approximately for the Parallel Machine Problem with a Single Server

Keramat Hasani
Islamic Azad University, Malayer Branch, Malayer, Iran

Svetlana A. Kravchenko
United Institute of Informatics Problems,
Surganova St. 6, 220012 Minsk, Belarus

Frank Werner *
Otto-von-Guericke-Universität, Fakultät für Mathematik,
PSF 4120, 39016 Magdeburg, Germany

August 4, 2013

Abstract

In this paper, we consider the scheduling problem of minimizing total weighted job completion time when a set of jobs has to be processed on a set of $m$ parallel identical machines with a common server. We propose an approximation algorithm with a worst-case ratio of $3 - \frac{1}{m}$. This result improves an existing $(5 - \frac{1}{m})$ - approximation algorithm given by Wang and Cheng (2001).

Keywords: Parallel machines, Single server, Approximation algorithm, Worst-case analysis

*Corresponding author. Tel.:+49 391 6712025; fax:+49 391 6711171.
E-mail address: frank.werner@ovgu.de (F. Werner).
1 Introduction

In this paper, we consider a parallel machine scheduling problem with a single server and the total weighted completion time objective function to be minimized, i.e., problem $P, S_1||\sum w_j C_j$ using the standard scheduling notation. This problem can be formulated as follows. A set of $n$ jobs $1, \ldots, n$ has to be processed by a set of $m$ parallel machines $M_1, \ldots, M_m$. Before processing, each job $j$ has to be loaded on the machine, on which this job is processed at once after loading. The loading procedure requires both the server and the machine for $s_j$ time units. The processing time $p_j$ and the weight $w_j$ for each job $j$ are known in advance. Each job can be processed by an arbitrary machine, and each machine and also the server can perform only one job at a time. We want to find a feasible schedule minimizing the function $\sum w_j C_j$. This problem $P, S_1||\sum w_j C_j$ was considered in [4], where a $(5 - \frac{1}{m})$ - approximation algorithm was given. In this paper, we propose a $(3 - \frac{1}{m})$ - approximation algorithm for the same problem. In fact, we use the same approach as in [4], i.e., we replace the original problem by a relaxed problem, which can be easily solved and thus, an optimal order of the completion times can be found. Then, in the original problem, we schedule the jobs in the order determined and estimate the performance bound.

The problem $P, S_1||\sum w_j C_j$ is unary NP-hard, since it is known that the problem $P_2, S_1 | s_j = s | \sum C_j$ is unary NP-hard [2]. Some special cases of this problem were considered so far. The problem $P_2, S_1 | p_j = p | \sum C_j$ is binary NP-hard [1]. There exists a polynomial algorithm for the problem $P_2, S_1 | s_j = 1 | \sum C_j$ [2]. For the problem $P_3, S_1 | s_j = 1 | \sum C_j$, a polynomial algorithm was developed in [1]. For the unary NP-hard problem $P, S_1 | s_j = 1 | \sum C_j$, an algorithm was proposed which creates a schedule $\tilde{s}$ with the following estimation:

$$\sum_{j=1}^{n} C_j(\tilde{s}) - \sum_{j=1}^{n} C_j(s^*) \leq n'(m - 2),$$

where $s^*$ denotes an optimal schedule and $n' = |\{j | p_j < m - 1\}|$, see [3]. It was shown that the SPT (shortest processing time) schedule is a $\frac{3}{2}$ - approximation for the problem $P, S_1 | s_j = s | \sum C_j$, see [4]. The interested reader is referred to [5] for additional information on server scheduling models.

In the next section, we present the main result, and finally we give some concluding remarks in the last section.

2 Main result

Recall that for the problem $P, S_1 || \sum w_j C_j$, we know the set of machines $M_1, \ldots, M_m$, the set of jobs $1, \ldots, n$, and for each job $j$, we know the values $p_j, s_j, w_j$.

Next, we consider a relaxation of the given problem. In this relaxation, we add one additional machine $MS$ to the set of machines $M_1, \ldots, M_m$. The set of jobs $1, \ldots, n$ is
the same. However, each job \( j \) consists of two operations as follows. One operation, say \( j_1 \), has to be processed on (one or several of) the machines \( M_1, \ldots, M_m \) for \( p_j \) time units. It can be processed with preemptions and moreover, this operation can be performed at the same time on several machines. Another operation, say \( j_2 \), has to be processed on machine \( MS \) for \( s_j \) time units without preemptions. Both operations are independent, i.e., both operations can be processed at the same time.

**Example.** In Figure 1, there are given two schedules. The first one uses two machines, and it is feasible for the problem \( P, S_1 || \sum w_j C_j \). The second one uses three machines, and it is feasible for the relaxed model described above.

![Figure 1](image)

Figure 1: The left schedule is feasible for the model \( P, S_1 || \sum w_j C_j \), and the right schedule is feasible for the relaxed model \( RA || \sum w_j CA_j \).

We denote by \( sa \) a feasible schedule for the relaxed model and by \( CA_j \) the completion time of job \( j \), i.e., \( CA_{j1}(sa) \) denotes the completion time of the operation \( j_1 \) in the schedule \( sa \). We set

\[
CA_j(sa) = \max \{CA_{j1}(sa), CA_{j2}(sa)\}.
\]

We want to find a schedule which minimizes the weighted sum of completion times for all jobs. Denote the described relaxed model by \( RA || \sum w_j CA_j \).

**Lemma 1** For the model \( P, S_1 || \sum w_j C_j \) and for the corresponding relaxed model \( RA || \sum w_j CA_j \), the inequality

\[
\sum w_j CA_j^* \leq \sum w_j C_j^*
\]

holds, where \( \sum w_j C_j^* \) denotes the optimal value of the weighted sum of completion times for the problem \( P, S_1 || \sum w_j C_j \), and \( \sum w_j CA_j^* \) denotes the optimal value of the weighted sum of completion times for the corresponding relaxed model \( RA || \sum w_j CA_j \).

**Proof:** Take any feasible schedule \( s \) for the problem \( P, S_1 || \sum w_j C_j \). We construct a schedule \( sa \) in the following way. We add one additional machine \( MS \) and schedule all the jobs \( 1, \ldots, n \) for \( s_1, \ldots, s_n \) time units on machine \( MS \) in the same order as in the schedule \( s \). For the set of machines \( M_1, \ldots, M_m \), we schedule all the jobs \( 1, \ldots, n \) for \( p_1, \ldots, p_n \) time units at the same time intervals and on the same machines as in the schedule \( s \). Finally, we obtain the schedule \( sa \) which is feasible for the model \( RA || \sum w_j CA_j \), and the inequality

\[
\sum w_j CA_j(sa) \leq \sum w_j C_j(s)
\]

holds.
Thus, for the model $RA \| \sum w_j C_{A_j}$, we know the set of machines $M_1, \ldots, M_m, MS$, the set of jobs $1, \ldots, n$, each of the jobs $j$ has the weight $w_j$ and consists of two operations: operation $j1$ has to be processed on the machines $M_1, \ldots, M_m$ for $p_j$ time units, and operation $j2$ has to be processed on machine $MS$ for $s_j$ time units.

Next, we consider an alternative relaxation of the given problem. In this relaxation, we have only two machines $MP$ and $MS$. Each job $j$ consists of two operations $j1$ and $j2$. Operation $j1$ has to be processed on machine $MP$ for $p_{j1}$ time units with preemptions, and operation $j2$ has to be processed on machine $MS$ for $s_j$ time units without preemptions. Both operations are independent, i.e., both operations can be processed at any time. We denote by $sb$ a feasible schedule for the relaxed model and by $CB_j$ the completion time of the job $j$, i.e., $CB_{j1}(sb)$ denotes the completion time of the operation $j1$ in the schedule $sb$. We set

$$CB_j(s) = \max\{CB_{j1}(sb), CB_{j2}(sb)\}.$$

We want to find a schedule which minimizes the weighted sum of completion times for all jobs. Denote the described relaxed model by $RB \| \sum w_j CB_j$.

**Lemma 2** For the model $RA \| \sum w_j C_{A_j}$ and for the corresponding relaxed model $RB \| \sum w_j CB_j$, the inequality

$$\sum w_j C_{B_j}^* \leq \sum w_j C_{A_j}^*$$

holds, where $\sum w_j C_{A_j}^*$ denotes the optimal value of the weighted sum of completion times for the model $RA \| \sum w_j C_{A_j}$, and $\sum w_j C_{B_j}^*$ denotes the optimal value of the weighted sum of completion times for the corresponding relaxed model $RB \| \sum w_j CB_j$.

**Proof:** Take any feasible schedule $sa$ for the problem $RA \| \sum w_j C_{A_j}$. Using the schedule $sa$, we construct the following schedule $sb$ for the corresponding relaxed model $RB \| \sum w_j CB_j$. On machine $MS$, both schedules $sa$ and $sb$ coincide. Consider the machines $M_1, \ldots, M_m$. Without loss of generality let $C_{A11}(sa) \leq \ldots \leq C_{An1}(sa)$. Then for the schedule $sb$, we schedule the operations $11, \ldots, n1$ in the following way:

- operation $11$ is processed for $\frac{p_1}{m}$ time units on machine $MP$ in the interval $[0, CA_1]$;
- operation $21$ is processed for $\frac{p_2}{m}$ time units on machine $MP$ in the interval $[0, CA_2]$;
- \ldots
- operation $n1$ is processed for $\frac{p_n}{m}$ time units on machine $MP$ in the interval $[0, CA_n]$.

To show that the constructed schedule $sb$ is a feasible one, suppose that in some interval $[0, CA_k]$ it is impossible to process the operation $k1$ for $\frac{p_k}{m}$ time units. However, in this case, it is easy to see that the original schedule $sa$ cannot be feasible, since for the schedule $sa$ the following inequalities hold:

- $CA_1 \geq \frac{p_1}{m}$ since in the schedule $sa$, the operation $11$ is processed in the interval $[0, CA_1]$ on $m$ machines;
- $CA_2 \geq \frac{p_1 + p_2}{m}$ since in the schedule $sa$, the operations $11$ and $21$ are processed in the
interval \([0, CA_2]\) on \(m\) machines;

\[-\ldots\]

- \(CA_n \geq \frac{p_1 + \ldots + p_n}{m}\) since in the schedule \(sa\), the operations \(11, \ldots, n1\) are processed in the interval \([0, CA_n]\) on \(m\) machines.

Thus, for the model \(RB || \sum w_jCB_j\), we know the set of machines \(MP, MS\), the set of jobs \(1, \ldots, n\), each of the jobs \(j\) has the weight \(w_j\) and consists of two operations: operation \(j1\) has to be processed on machine \(MP\) for \(\frac{p_j}{m}\) time units, and operation \(j2\) has to be processed on machine \(MS\) for \(s_j\) time units.

Now, consider another relaxation of the given problem. In this relaxation, we have two machines \(MP\) and \(MS\). Each job \(j\) consists of two operations \(j1\) and \(j2\). Operation \(j1\) has to be processed on the machines \(MP, MS\) for \(\frac{p_j}{m}\) time units with preemptions, and operation \(j2\) has to be processed on the machines \(MP, MS\) for \(s_j\) time units with preemptions. Both operations are independent, i.e., both operations can be processed at any time. We denote by \(sc\) a feasible schedule for the relaxed model and by \(CC_j\) the completion time of the job \(j\), i.e., \(CC_j(sc)\) denotes the completion time of the operation \(j1\) in some schedule \(sc\). We set

\[CC_j(sc) = \max\{CC_{j1}(sc), CC_{j2}(sc)\}.\]

We want to find a schedule which minimizes the weighted sum of completion times for all jobs. Denote the described relaxed model by \(RC || \sum w_jCC_j\).

**Lemma 3** For the model \(RB || \sum w_jCB_j\) and for the corresponding relaxed model \(RC || \sum w_jCC_j\), the inequality

\[\sum w_jCC_j^* \leq \sum w_jCB_j^*\]

holds, where \(\sum w_jCB_j^*\) denotes the optimal value of the weighted sum of completion times for the model \(RB || \sum w_jCB_j\), and \(\sum w_jCC_j^*\) denotes the optimal value of the weighted sum of completion times for the corresponding relaxed model \(RC || \sum w_jCC_j\).

**Proof:** The above inequality is evident, since any schedule feasible for the model \(RB || \sum w_jCB_j\) is also feasible for the corresponding model \(RC || \sum w_jCC_j\). 

Further we suppose that all the jobs are enumerated in such a way that

\[\frac{1}{2w_1} \left( s_1 + \frac{p_1}{m} \right) \leq \ldots \leq \frac{1}{2w_n} \left( s_n + \frac{p_n}{m} \right).\]

**Lemma 4** If the inequalities

\[\frac{1}{2w_1} \left( s_1 + \frac{p_1}{m} \right) \leq \ldots \leq \frac{1}{2w_n} \left( s_n + \frac{p_n}{m} \right)\]
hold, then an optimal schedule for the model RC \( \mid \mid \sum w_j CC_j \) can be constructed by processing each job \( j \) in the interval

\[
\left[ CC_{j-1}, CC_{j-1} + \frac{1}{2} \left( s_j + \frac{p_j}{m} \right) \right],
\]

where \( CC_0 = 0 \).

**Proof:** By induction one can prove that in an optimal schedule each job \( j \) has to be processed in the interval

\[
\left[ t, t + \frac{1}{2} \left( s_j + \frac{p_j}{m} \right) \right],
\]

where \( t \) is some time point. Using simple interchanging arguments, one can prove that in an optimal schedule the first finishing job, say \( a \), has to be processed in the interval

\[
\left[ 0, \frac{1}{2} \left( s_a + \frac{p_a}{m} \right) \right].
\]

Suppose that for each of the first \( k \) finishing jobs this statement holds. Denote the interval occupied by the first \( k \) finishing jobs by \([0, T]\). Consider the \((k+1)\)th finishing job, say \( b \). Using interchanging arguments, it is easy to prove that job \( b \) has to be processed in the interval

\[
\left[ T, T + \frac{1}{2} \left( s_b + \frac{p_b}{m} \right) \right].
\]

Now, one can easily see that an optimal schedule for the model RC \( \mid \mid \sum w_j CC_j \) can be constructed by scheduling all jobs in non-decreasing order of the values \( \frac{1}{2w_j}(s_j + \frac{p_j}{m}) \) in such a way that each job \( j \) is processed in the interval

\[
\left[ CC_{j-1}, CC_{j-1} + \frac{1}{2} \left( s_j + \frac{p_j}{m} \right) \right],
\]

where \( CC_0 = 0 \). \( \square \)

Consider now the original problem \( P, S_1 \mid \mid \sum w_j C_j \). Let the jobs be indexed according to non-decreasing ratios of the values \( \frac{1}{2w_j}(s_j + \frac{p_j}{m}) \), that is, we have

\[
\frac{1}{2w_1} \left( s_1 + \frac{p_1}{m} \right) \leq \ldots \leq \frac{1}{2w_n} \left( s_n + \frac{p_n}{m} \right).
\]

Denote by \( \tilde{s} \) the schedule obtained by scheduling the jobs according to non-decreasing values \( \frac{1}{2w_j}(s_j + \frac{p_j}{m}) \), i.e., in each step \( k \), \( k = 1, \ldots, n \), the \( k \)-th job in the list \( 1, \ldots, n \) is assigned to the machine available at the earliest possible time.

**Theorem 1** Let \( s^* \) denote an optimal schedule for the problem \( P, S_1 \mid \mid \sum w_j C_j \). Then the inequality

\[
\sum w_j C_j(\tilde{s}) \leq \left( 3 - \frac{1}{m} \right) \sum w_j C_j(s^*)
\]

holds.
Proof: For each job \( j \), we have

\[
C_j(\tilde{s}) \leq \sum_{k=1}^{j} s_k + \frac{1}{m} \sum_{k=1}^{j} p_k + \left(1 - \frac{1}{m}\right) p_j \leq 2 \left(\frac{1}{2} \sum_{k=1}^{j} s_k + \frac{1}{2} \sum_{k=1}^{j} \frac{p_k}{m}\right) + \left(1 - \frac{1}{m}\right) p_j.
\]

From Lemma 4, one can see that inequality

\[
CC^*_j \geq \frac{1}{2} \left(\sum_{i=1}^{j} s_i + \sum_{i=1}^{j} \frac{p_i}{m}\right)
\]

holds. Therefore, we get the inequality

\[
C_j(\tilde{s}) \leq 2CC^*_j + \left(1 - \frac{1}{m}\right) C^*_j.
\]

Now, inequality

\[
\sum_{i} w_i C_i(\tilde{s}) \leq 2 \sum_{i} w_i CC^*_i + \left(1 - \frac{1}{m}\right) \sum_{i} w_i C^*_i
\]

holds. From Lemmas 1 - 3, one can see that

\[
\sum w_i CC^*_i \leq \sum w_i C^*_i,
\]

therefore, we obtain

\[
\sum_{i} w_i C_i(\tilde{s}) \leq 2 \sum w_i C^*_i + \left(1 - \frac{1}{m}\right) \sum w_i C^*_i \leq \left(3 - \frac{1}{m}\right) \sum w_i C^*_i.
\]

\(\square\)

3 Concluding remarks

In this paper, we considered the scheduling problem \( P, S1||\sum w_i C_i \). We proposed an approximation algorithm with a worst-case ratio of \( 3 - \frac{1}{m} \). This result improves an existing \((5 - \frac{1}{m})\) - approximation algorithm given in [4]. However, we conjecture that this result can be improved further, since one can derive a \( 2 \) - approximation algorithm for the problem \( P2, S1||\sum w_i C_i \). So, the derivation of an approximation algorithm with a better worst-case ratio remains an interesting subject for future research.

References


