Comparison of Approximation Algorithms for Unrelated Parallel Machines to Minimize the Weighted Makespan

1Tariq S. Abdul-Razaq and 2Najwa Raheem Mustafa
1Dept. of Mathematics, College of Science, Al-Mustansiriyah University
2Dept. of Mathematics, College of Science for Women, Baghdad University

ABSTRACT

The problem of scheduling of unrelated parallel machines is considered. In this environment, a set of n jobs has to be scheduled on m unrelated parallel machines. Each job is available for processing at time zero and each machine can process at most one job at a time and a job can be processed by at most one machine at a time. A case study is considered to schedule jobs in a cutting workshop to minimize the weighted makespan. Five algorithms are proposed and their performance is studied.

INTRODUCTION

The problem considered in this paper is the scheduling of unrelated parallel machines. In parallel machines environment there are multiple machines. In identical parallel machines environment all machines operate at the same speed, while in uniform parallel machines environment each machine has its own speed. For unrelated parallel machines; there are multiple machines with different job-related speeds, that is the processing times are unrelated. In this research environment, a set N of n jobs, 1, ..., n, each of which has to be scheduled on one of m machines, M1,..., Mm is given. Each job j has a processing requirement pj, weight wj and is available for processing at time zero. Each machine can process at most one job at a time and a job can be processed by at most one machine at a time. All machines start working at time zero and process their jobs sequentially. For the unrelated parallel machines, the speed of machine Mi on job j, vij, depends on both the machine and the job; job j requires \( p_j / v_{ij} \) processing time on a machine \( M_i \). We define \( p_{ij} = p_j / v_{ij} \) (1). The notation \( N_i \) denotes the set of jobs assigned to machine \( M_i \). Let \( C_j \) denotes the completion time of job j and \( CM_i \) denotes the completion time of the last job on machine \( M_i \), that is:

\[
CM_i = \sum_{j \in N_i} p_{ij} \quad i=1, \ldots, m, \ j=1, \ldots, n
\]
The maximum completion time (makespan) $C_{\text{max}}$ is defined by (2):

$$C_{\text{max}} = \max\{CM_1, \ldots, CM_m\},$$

we can define $C^w_{\text{max}}$ by:

$$C^w_{\text{max}} = \max\{CM_1, \ldots, CM_m\}$$

with respect to some priority rules that depend on job weight; the objective is to minimize the weighted makespan $C^w_{\text{max}}$.

There is extensive literature describing approximation algorithms for unrelated parallel machine scheduling problems. Horwitz and Sahni (1976) proposed several exact and approximate algorithms for some special cases of the $R/C_{\text{max}}$ problem. Also, Ibarra and Kim (3), Davis and Jaffe (4) proposed several algorithms to solve the problem $R/C_{\text{max}}$. There are many papers that present an experimental comparison of approximation algorithms, some of which are based on linear programming and others based on local search heuristics. For example, Hariri and Potts (5) solve the $R/C_{\text{max}}$ problem. The Branch and bound method can also be used, as was the case with Salem, Anagnostopoulos and Rabadi [2] when solving the problem $R/C_{\text{max}}$.

Bruno, Coffman and Sethi (1974) and Lenstra et al.(1977) (6) showed that the problem of minimizing total weighted completion time on two identical parallel machines is NP-hard, thus $R\sum w_j C_j$ is NP-hard in the strong sense (7).

If there is only one machine the problem is solved to optimality by ordering the jobs in a non-decreasing order of the ratio $p_j/w_j$ (the SWPT rule) (Smith 1956). Hence, the problem reduces assigning the jobs appropriately to the machines and then sequencing the jobs on each machine by the SWPT rule.

Karp (1972) show that the problem of scheduling two identical parallel machines to minimize the maximum completion time is NP-hard. Clearly, the more general unrelated parallel machine problem is also NP-hard (5). Thus for the $R/C^w_{\text{max}}$ problem it seems unlikely that a polynomial time algorithm that always produce optimal solution exists.

The Problem $R/C^w_{\text{max}}$

The problem $R/C^w_{\text{max}}$ can be formulated as a linear program as follows:

Minimize $Z = C^w_{\text{max}}$.

Subject to

$$\sum_{j=1}^{n} x_{ij} w_j p_{ij} \leq C^w_{\text{max}} \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, \ldots, n$$
\[ x_{ij} \in \{0,1\} \quad i=1,\ldots,m; j=1,\ldots,n \]

Where \( x_{ij} \) is an assignment variable that is equal to 1 if job \( j \) is assigned to machine \( M_i \) and 0 otherwise.

The best time of the \( j \)th job, denoted by \( b_j \), equals \( \min_{1 \leq i \leq m} p_{ij} \). The efficiency of the \( i \)th machine on the \( j \)th job, denoted by \( e_{ij} \), equals to \( b_j / p_{ij} \) (4). The maximum efficiency is one. Using these concepts, the following algorithm is a modification of the algorithm proposed by Davis and Jaffe (4).

**Algorithm 1: Algorithm WCM1**

**Step (1):** \( N = \{1,\ldots,n\}, N^i = \phi \). For \( j=1,\ldots,n \), find \( b_j = \min_{1 \leq i \leq m} p_{ij} \);

for \( j=1,\ldots,n; i=1,\ldots,m \), find \( e_{ij} = b_j / p_{ij} \);

for \( i=1,\ldots,m \), create a list of the jobs \( j=1,\ldots,n \) sorted in a non-increasing order of \( w_{jef_{ij}} \). Set \( \text{sum}_i = 0 \), \( CM_i = 0 \) for \( i=1,\ldots,m \).

Designate all machines as ‘available’ and all jobs as ‘unassigned’;

**Step (2):** If \( N = \phi \), go to step (6), else find a machine \( i \) such that \( \text{sum}_i \) is minimal among all available machines;

**Step (3):** Find the next unassigned job \( j \) on \( i \)’s list;

**Step (4):** If \( j \) does not exist or if \( e_{ij} < \frac{1}{\sqrt{m}} \) then mark \( i \) as unavailable;

**Step (5):** Otherwise assign job \( j \) to machine \( i \); \( N = N - \{j\} \), \( N^i = N^i \cup \{j\} \). Set \( \text{sum}_i = \text{sum}_i + p_{ij} \), \( CM_i = CM_i + p_{ij} \), return to step (2);

**Step (6):** \( C_{max}^w = \max_{1 \leq i \leq m} \{CM_i\} \).

Figure -1: The WCM1 algorithm

The ratio \( e_{ij} \) can be used to propose another algorithm that is called WCM2.

**Algorithm 2: Algorithm WCM2**

Algorithm WCM2 is similar to algorithm WCM1 except that in step (1) the jobs are sorted in a non-decreasing order of \( e_{ij} / w_j \).

Figure -2: The WCM2 algorithm
Ibarra and Kim (3) proposed several algorithms for the problem $\max \{C_i/\sum p_j \}$ Some of these algorithms are modified to fit the problem $\max \{C_i/\sum p_j \}$ as follows.

**Algorithm 3: Algorithm WCM3**

**Step (1):** $N = \{1,\ldots, n\}$, $N^i = \phi$ and $\sum_i = 0$ for $i=1,\ldots, m$;

**Step (2):** If $N = \phi$, go to step (4);

**Step (3):** Otherwise find a job $j \in N$ such that

\[
\min \{\sum_i + p_{ij}\} \leq \min \{\sum_i + p_{ij'}\} \text{ for all } j' \in N;
\]

let $i$ be such that $\sum_i + p_{ij}$ is minimum;

$N^i = N^i \cup \{j\}, N = N - \{j\}$, return to step 2;

**Step (4):** $C^w_{\max} = \max_{1 \leq i \leq m} \{CM_i\}$.

**Algorithm 4: Algorithm WCM4**

**Step (1):** $N = \{1,\ldots, n\}, N^i = \phi, \sum_i = 0, i=1,\ldots, m$;

**Step (2):** For each job $j$, find $p_{\min}^i(j) = \min_{1 \leq i \leq m} p_{ij}/w_j$

**Step (3):** Order the jobs in $N$ according to non-decreasing order of $p_{\min}^i(j)$;

**Step (4):** If $N = \phi$, go to step 6;

**Step (5):** Else, find the machine $i$ such that $\sum_i + w_j p_{ij} \leq \sum_k + w_j p_{kj}$ for all $k = 1,\ldots, m$; set $\sum_i = \sum_i + w_j p_{ij}$; $N = N - \{j\}$;

$CM_i = CM_i + w_j p_{ij}$; $N^i = N^i \cup \{j\}$; return to step 4;

**Step (6):** $C^w_{\max} = \max_{1 \leq i \leq m} \{CM_i\}$.

Figure 3: The WCM3 algorithm

Figure 4: The WCM4 algorithm

Also the following algorithm is a modification of the LRPT-FM (Longest Remaining Processing Time on the Fastest Machine) rule.
Algorithm 5: Algorithm WCM5

Step (1): \( N = \{1,\ldots,n\}, N^i = \phi, \sum_i = 0, i=1,\ldots,m; \)

Step (2): For each job \( j \), find \( p'_{\text{max}}(j) = \max_{1 \leq i \leq m} p_{ij} / w_j \)

Step (3): Order the jobs in \( N \) according to non-increasing order of \( p'_{\text{max}}(j) \);

Step (4): If \( N = \phi \), go to step (6);

Step (5): Else, find the machine \( i \) such that \( \sum_i + p_{ij} \leq \sum_k + p_{kj} \)
for all \( k=1,\ldots,m \); set \( \sum_i = \sum_i + p_{ij}; CM_i = CM_i + p_{ij}, \)
\( N = N - \{j\}; N^i = N^i \cup \{j\}; \) return to step (4);

Step (6): \( C_{\text{max}}^w = \max_{1 \leq i \leq m} \{CM_i\}. \)

Figure -5: The WCM5 algorithm

Case Study

The Five algorithms: WCM1, WCM2, WCM3, WCM4 and WCM5 are applied in the cutting workshop in Al-Karama General State Company. The company has a complete engineering department for design and technology that works jointly with the planning and follow-up department and different factories to accomplish production operations within the annual plan, in addition to some specified orders for special projects. One of the divisions of the planning and follow-up department is the cutting workshop, which carries out a huge and basic part in preparing and providing production work orders the raw materials for all factories in primary measurements specified by technological procedure of these parts production.

The work of the cutting workshop is of great importance in preparing and providing raw materials in correct dimensions and required quantities within accurate times and specified types to all factories by best utilization of available self capabilities. Thus the work of the cutting workshop is a bottleneck to progress of production operation in factories and the company.

The cutting workshop environment includes the following components:

a) Machines: There are different types of machines used in this unit. The speed of each machine depends on its own specifications and the raw materials.

b) Raw materials: There is a wide range of metal ores and materials used in the workshop works which are divided into five kinds according to their cutting speed, these types are: Aluminium, Stainless steel, Other types of steel, Teflon and different plastics and Cooper and brass.
c) **Work Style:** The cutting workshop receives work orders to prepare materials weekly. The company runs an annual production plan and the planning department has a monthly plan for the factories production and the cutting workshop prepares materials at least one month ahead.

d) **Constraints:** Jobs in the cutting workshop include cutting shafts, blocks and plates, each of these materials need certain time as a loading cost. Some machines have certain constraints, for example, they cannot handle some raw materials. Raw materials are available in standard measurements.

The algorithms WCM1, WCM2, WCM3, WCM4 and WCM5 are implemented on a case study of 10 assemblies. These assemblies consist of different numbers of jobs and machines. The jobs vary in their types and raw materials, also the machines are of different types. The efficiency of the proposed algorithms is tested using programs coded in Microsoft FORTRAN Power Station version 4.0; the sketches were drawn using MATLAB 6.5. Both codes are executed on a Pentium III 1GHz personal computer with 256 MB memory.

The schedules yielded from these algorithms are compared. Table (1) presents the results obtained by these algorithms and the results are compared to the best of them using the percentage relative deviation from the best value (PRD), calculated as $\frac{H - B}{B} \times 100\%$, where H and B represents the heuristic and best values, respectively. Also, they are compared using the deviation (DEV) of each value w.r.t. arithmetic mean of all values. The best result is presented in bold.

<table>
<thead>
<tr>
<th>ASSEMBLY</th>
<th>NO. OF JOBS</th>
<th>NO. OF MACHINES</th>
<th>ALGORITHM</th>
<th>VALUE</th>
<th>PRD</th>
<th>DEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21</td>
<td>6</td>
<td>WCM1</td>
<td>750</td>
<td>79.42583</td>
<td>165.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WCM2</td>
<td>522</td>
<td>24.88038</td>
<td>-62.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WCM3</td>
<td>733</td>
<td>75.35885</td>
<td>148.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WCM4</td>
<td>499</td>
<td>19.37799</td>
<td>-85.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WCM5</td>
<td>418</td>
<td>0.000</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>18</td>
<td>6</td>
<td>WCM1</td>
<td>836</td>
<td>12.66846</td>
<td>32.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WCM2</td>
<td>742</td>
<td>0.000</td>
<td>-61.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WCM3</td>
<td>865</td>
<td>16.57682</td>
<td>61.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WCM4</td>
<td>811</td>
<td>9.29919</td>
<td>7.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WCM5</td>
<td>762</td>
<td>2.69542</td>
<td>-41.2</td>
</tr>
</tbody>
</table>
It can be seen from table (1) that algorithm WCM5 is the best one. It generates the best solution almost for all the assemblies given (nine out of 10).
The performance of other algorithms varies radically from an assembly to another. It is interesting to see that the five algorithms perform identically on assembly (4). Except algorithm WCM5, other algorithms perform badly on some assemblies and well on others. The running time for the program of 10 assemblies is 1.500000E – 01 seconds. The results are presented in the figure (6).

![Figure-6: The performance of algorithms WCM1, WCM2, WCM3, WCM4 and WCM5](image)

Conclusions

In this work an applied problem was studied, which is the problem of scheduling n jobs on m unrelated parallel machines. Several algorithms are proposed to minimize the maximum weighted completion time. They are applied on the work of a cutting workshop in Al-Karama general state company and their performance is analyzed hoping to suit the company demands. Algorithm WCM5 is the best one for the maximum weighted completion time problem of unrelated parallel machines. The performance of other algorithms for the problems (minimizing the maximum weighted completion time) varies between an assembly and another one. To improve the performance, one can study the preemption of jobs. Also, on-line scheduling, i.e. scheduling when jobs arrive over time, can be studied.
REFERENCES


