A branch-and-bound algorithm to minimize total weighted completion time on identical parallel machines with job release date

Computers & Operations research 2008
Rabia Nassah, Farouk Yalaoui, Chengbin Chu
Contents

- Introduction
- Dominance property
- Lower bounds
- Heuristic
- Branch-and-bound algorithm
- Computational results
- Conclusion
Problem review

Identical parallel machine scheduling

- With job release date, weight
- Objective with sum of weighted flow times
  - no preemption, no breakdowns
Introduction

- **History**
  - $1\|\sum w_iC_i$: SWPT rule
  - $R\|\sum C_i$: matching technique
  - Bruno et al, $P2\|\sum w_iC_i$ is NP-Hard
  - Lenstra et al, $1|r_i|\sum w_iC_i$ is NP-hard in the strong sense $\Rightarrow Pm|r_i|\sum w_iC_i$ is more general
  - $Pm|r_i, split|\sum C_i$: can be solved with the shortest remaining processing time (SRPT) rule

- **Preview**
  - 3 dominance rules
  - Lower bounds (2 kinds of)
Dominance property

- **Notation**
  - $F(\sigma) = \sum_{i=1}^{n} w_i C_i(\sigma)$: weighted completion time of jobs in the partial schedule $\sigma$
  - $\phi_u(\sigma)$: completion time of the last job in $\sigma$ on machine $u$
  - $\Omega(\sigma)$: the schedule obtained by appending to $\sigma$ the optimal partial schedule of the remaining jobs
  - $\sigma \circ i$: the new schedule obtained by adding job $i$ after $\sigma$ on the machine available the earliest
  - $R_i(t) = \max(t, r_i)$ and $E_i(t) = R_i(t) + p_i$, $r_i$ is the release date of job $i$ at time $t$
  - $J(\sigma), J_j(\sigma)$: the set of jobs (in machine $j$) in schedule $\sigma$
  - $\Delta_i(\sigma)$ is the completion of the job immediately preceding job $i$ in schedule $\sigma$
    - If job $i$ is the first in $\sigma$, $\Delta_i(\sigma) = 0$
  - $F(\Omega(\sigma_1)) \leq F(\Omega(\sigma_2))$
    - $\sigma_1$ dominates $\sigma_2$
Remark 1.

In Yalaoui and Chu[6], any optimal schedule resulting from a partial schedule $K$, the number of adding jobs on any machine $j$ is at most $U_j(K) = \min(n - m + 1 - |J_j|, n - |J(K)|)$.

Theorem 1.

Let $\sigma_1$ and $\sigma_1$ be two partial schedules for the same set of jobs $K$. If $F(\sigma_1) \geq F(\sigma_2)$ and $F(\sigma_1) - F(\sigma_2) \geq W\Psi$, then $\sigma_2$ dominates $\sigma_1$, where $W = \sum_{i \in N - k} w_i$ and $\Psi = \max_{k \in [1, n]} (\phi_k(\sigma_2) - \phi_k(\sigma_1))$

$\phi_k(\sigma_1)$ and $\phi_k(\sigma_2)$ are the $k$th smallest machine finishing time in partial schedules $\sigma_1$ and $\sigma_2$, respectively.

Proof.
Theorem 1.

Proof.

\[ F(\sigma_1) \geq F(\sigma_2), \quad F(\sigma_1) - F(\sigma_2) \geq W\Psi (W = \sum_{i \in [N]} w_i, \quad \Psi = \max_{k \in [1..n]} (\phi_k(\sigma_2) - \phi_k(\sigma_1))) \]

Case 1: If \( \Psi \leq 0 \)

1) \( F(\sigma_1) \geq F(\sigma_2) \)
2) \( \Psi \leq 0 \) then \( \phi_i(\sigma_2) \leq \phi_i(\sigma_1) \)
   each job \( i \) .. \( C_i(S) \leq C_i(\Omega (\sigma_1)) \)
   \( \Rightarrow F(S) - F(\Omega (\sigma_1)) \leq F(\sigma_2) - F(\sigma_1) \leq 0 \)
   thus, \( F(\Omega (\sigma_2)) - F(\Omega (\sigma_1)) \leq F(S) - F(\Omega (\sigma_1)) \leq 0. \)
Theorem 1.

Proof.

$$F(\sigma_1) \geq F(\sigma_2), \quad F(\sigma_1) - F(\sigma_2) \geq W\Psi \quad (W = \sum_{i=1}^{K} w_i, \quad \Psi = \max_{k \in [1, m]} (\phi_k(\sigma_2) - \phi_k(\sigma_1)))$$

Case 2: If $\Psi \geq 0$

1) $\alpha_k \leq 0 \Rightarrow$ the same as case 1.

2) if $\alpha_k > 0$ then

put $L = \{ k \text{ such as } \alpha_k > 0 \}$

$$\sum_{i \in L} \alpha_i \sum_{i \in L} w_i \leq \Psi \sum_{i \in L} \sum_{i \in L} w_i \leq \Psi W$$

$$\Rightarrow F(\Omega(\sigma_2)) - F(\Omega(\sigma_1)) \leq F(S) - F(\Omega(\sigma_1))$$

$$\leq F(\sigma_2) - F(\sigma_1) + \sum_{i \in L} \alpha_i \sum_{i \in L} w_i$$

$$\leq F(\sigma_2) - F(\sigma_1) + \Psi W \leq 0.$$
Theorem 2.

For any partial schedule $\sigma$ and any job $i$ not scheduled in $\sigma$, schedule $\Omega(\sigma \circ i)$ is dominated if there is another unscheduled job $j$ such that

- $w_i \leq w_j$
- $E_j(\phi_h) \leq E_i(\phi_h)$ and
- $w_j E_j(\phi_h) - w_i E_i(\phi_h) \leq w_j p_j - w_i p_i + (p_j - p_i) \max_{x \in \{N-J(\sigma)-(i,j)\}} w_x U(\sigma)$

where $h$ is the machine available the earliest in $\sigma$, $U_j(K) = \min(n - m + 1 - |J|, n - |J(K)|)$. (remark.1)

Proof.

1. If job $i$ and $j$ are both scheduled on machine $h$ in schedule $\Omega(\sigma \circ i)$.
   - Construct another schedule $S$ identical to $\Omega(\sigma \circ i)$ except that jobs $i$ and $j$ are interchanged.

\[ \begin{align*}
1) \quad p_i < p_j & \implies C_k(S) \leq C_k(\Omega(\sigma \circ i)) \quad \text{for all } k \in B \\
F(S) - F(\Omega(\sigma \circ i)) & \leq w_j E_j(\phi_h) - w_i E_i(\phi_h) + w_j (t + p_j) - w_i (t + p_i) \\
& \leq w_j E_j(\phi_h) - w_i E_i(\phi_h) - t(w_i - w_j) - (w_j p_j - w_i p_i) \\
& \leq (p_j - p_i) \max_{x \in \{N-J(\sigma)-(i,j)\}} w_x U(\sigma) \leq 0 \\
\end{align*} \]

2) \[ \begin{align*}
& \quad p_i > p_j \implies C_k(S) \leq C_k(\Omega(\sigma \circ i)) + (p_i - p_j) \quad \text{for all } k \in B \\
& F(S) - F(\Omega(\sigma \circ i)) \leq w_j E_j(\phi_h) - w_i E_i(\phi_h) + w_j (t + p_j) - w_i (t + p_i) + (p_i - p_j) \sum_{k \in B} w_k \\
& \leq w_j E_j(\phi_h) - w_i E_i(\phi_h) - (w_j p_j - w_i p_i) + (p_i - p_j) \max_{x \in \{N-J(\sigma)-(i,j)\}} w_x U(\sigma) \\
& \leq 0
\end{align*} \]
Dominance property

Theorem 2.

Proof.

- case 2. If job $j$ is scheduled on another machine, say machine $g$, two subcases need to be examined

  - If $C_j(\Omega(\sigma \circ i)) \leq R_i(\phi_h) + p_j$ then $R_j(\phi_g) \leq R_i(\phi_h)$
Dominance property

Theorem 2.

Proof.

- case 2. If job $j$ is scheduled on another machine, say machine $g$, two subcases need to be examined

  - If $C_j(\Omega \circ i) > R_i(\phi_h) + p_j$, then $R_j(\phi_g) > R_i(\phi_h)$

The earliest start time of job $i$ and the completion time of machine $h$ are used in the calculations.

$$F(S) - F(\Omega(\sigma \circ i)) \leq w_jE_j(\phi_h) - w_iE_i(\phi_h) + t(w_i - w_j) + w_ip_i - w_jp_j + (p_i - p_j)\sum_{k \in B}w_k$$

$$\leq w_jE_j(\phi_h) - w_iE_i(\phi_h) - t(w_i - w_j) - (w_ip_i - w_jp_j) + (p_i - p_j)(\max_{x \in N - J(\sigma) - \{i,j\}}w_x)U(\sigma)$$

$$w_jE_j(\phi_h) - w_iE_i(\phi_h) \leq w_jp_j - w_iP_i + (p_j - p_i)(\max_{x \in N - J(\sigma) - \{i,j\}}w_x)U(\sigma)$$
Theorem 3.

For any partial schedule $\sigma$ and any job $i$ not scheduled in $\sigma$, schedule $\Omega(\sigma \circ i)$ is dominated if there is another unscheduled job $j$ such that

- $w_i \leq w_j$
- $E_i(\Delta_j) \leq E_j(\Delta_j)$ and
- $w_jE_j(\Delta_h) - w_iE_i(\Delta_h) \leq w_jp_j - w_ip_i + (p_j - p_i)(\max_{x \in \{N - J(\sigma) - \{i\}} w_x)U(\sigma)$
Lower bounds

- Two-lower bound scheme suggested
  - Based on lower bound of $1|r_i|\sum w_i C_i$ (Beloudah et al.) $\Rightarrow$ (1)
  - Based on $P_m|r_i$, split$|\sum C_i$ $\Rightarrow$ (2)

- Lower bound (1)

**Theorem 4 (Belouadah et al. [21])**. Let $\sigma^*$ be an optimal schedule of the problem $P$ and $\sigma^*_1$ be an optimal schedule of $P_1$. If $\sigma^*_2$ is an optimal schedule of problem $P_2$, then $\sum_{i \in N_1} w_i C_i(\sigma^*_2) + CBRK \leq \sum_{i \in N_1} w_i C_i(\sigma^*_1) + CBRK \leq \sum_{i \in N} w_i C_i(\sigma^*)$ where $CBRK = \sum_{x=1}^{k-1} w_{ix} \sum_{j=x+1}^{k} p_{ij}$.

- a job $i$ is split or broken into $k$ pieces (the set of $N_1$)
  - $\sum p_{ix} = p_i (x=1,2,...,k)$
  - $\sum w_{ix} = w_i, r_{ix} = r_i$
- $P : 1|r_i|\sum w_i C_i$
- $P1 :$ job $i$ is split into $k$ pieces, contiguously.
- $P2 :$ relaxation of $P_i$, not required to be scheduled contiguously.  
  (in the literature..)
Lower bounds

Lower bound (1, cont)

- A job $i$ is split or broken into $k$ pieces (the set of $N_1$)
  - $\sum p_{ix} = p_i (x=1,2,...,k)$
  - $\sum w_{ix} = w_i$, $r_{ix} = r_i$

- $P1 : Pm|r_i| \sum w_i C_i$
- $P2 : 1|r_i, p' = p/m| \sum w_i C_i$
- $P3 : 1|r_i'=mr_i, p_i| (1/m)\sum w_i C_i$

- $P1$ is relaxed to $P2$ ($P2$ and $P3$ are equivalent)

Theorem 5.

- $LB_{GS}$ applied to the problem $P3$ is a lower bound of problem $P1$
Lower bounds

- Lower bound (2)
  - basic idea
    - job $i$ is split into $k_i = \lfloor w_i \rfloor$
    - $p_{ih} = p_i / w_i, (h=1,...,k_i - 1)$ and $p_{i_0} = p_i (w_i - \lfloor w_i \rfloor + 1) / w_i$
    - $r_{ih} = r_i + (h-1) p_i / w_i$
    - $Q : Pm|r_i|\sum w_i C_i$, $Q1$ : corresponding problem in which job is split and scheduled contiguously
    - $Q2$ : relaxation of $Q1$ (not required to be contiguously)

Theorem 6. If $\sigma$ is a feasible schedule for $Q$ and $\sigma_1$ is the corresponding schedule for $Q_1$, then

$$\sum_{i \in N} w_i C_i(\sigma) = \sum_{i \in N_1} C_i(\sigma_1) + \sum_{i \in N} \delta_i C_i(\sigma) + CBRK,$$

where $\delta_i = w_i - \lfloor w_i \rfloor$ and $CBRK = \sum_{i \in N} (p_i / 2 w_i) (\lfloor w_i \rfloor - 1) (2w_i - \lfloor w_i \rfloor)$.

$$w_i C_i(\sigma) = k_i C_i(\sigma) + \delta_i C_i(\sigma)$$

$$= \sum_{i=1}^{k_i} (t_i + p_i) + \delta_i C_i(\sigma)$$

$$= \sum_{i=1}^{k_i-1} (t_i + xp_i / w_i) + (t_i + p_i) + \sum_{i=1}^{k_i-1} (p_i - xp_i / w_i) + \delta_i C_i(\sigma)$$

$$= \sum_{i=1}^{k_i} C_i(\sigma_1) + \frac{p_i}{2 w_i} (\lfloor w_i \rfloor - 1) (2w_i - \lfloor w_i \rfloor) + \delta_i C_i(\sigma).$$
Lower bounds

- Lower bound(2) cont..
  - Lemma 1.

**Lemma 1.** Let two series of numbers \((x_1, x_2, \ldots, x_n), (y_1, y_2, \ldots, y_n)\) such that \(x_1 \leq x_2 \leq \cdots \leq x_n\). If \((y'_1, y'_2, \ldots, y'_n)\)

is the series obtained by sorting the series \((y_1, y_2, \ldots, y_n)\) in nondecreasing order, the following relation holds:

\[
\sum_{i=1}^{n} \max(x_i, y'_i) \leq \sum_{i=1}^{n} \max(x_i, y_i).
\]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>(\max(x, y))</th>
<th>(\max(x, y'))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>22</td>
<td>30</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>39</td>
<td>27</td>
<td>39</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>42</td>
<td>28</td>
<td>42</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>42</td>
<td>28</td>
<td>42</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>39</td>
<td>30</td>
<td>39</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>28</td>
<td>39</td>
<td>28</td>
<td>39</td>
</tr>
<tr>
<td>7</td>
<td>27</td>
<td>39</td>
<td>27</td>
<td>39</td>
</tr>
<tr>
<td>8</td>
<td>28</td>
<td>42</td>
<td>28</td>
<td>42</td>
</tr>
<tr>
<td>9</td>
<td>22</td>
<td>42</td>
<td>22</td>
<td>42</td>
</tr>
<tr>
<td>sum</td>
<td></td>
<td></td>
<td>297</td>
<td>297</td>
</tr>
</tbody>
</table>
Lower bounds

- Lower bound(2) cont..
  - Theorem 7.

  \[ \sum_{i \in N} w_i C_i(\sigma) = \sum_{i \in N_1} C_i(\sigma_1) + \sum_{i \in N} \delta_i C_i(\sigma) + CBRK, \]

**Theorem 7.** Let \(a_{ij} = r_{ij} + p_{ij}, \ i = 1, \ldots, n, \ j = 1, \ldots, [w_i].\) Let \((a_{11}, a_{22}, \ldots, a_{WW})\) be the series obtained by sorting the series \((a_1, a_2, \ldots, a_W)\) in nondecreasing order. Then \(\sum_{i=1}^{W} \max\{C_i(\sigma_2), a_{[i]}\}\) is a lower bound on problem \(Q_1\), where \(\sigma_2\) is the schedule obtained by the SRPT rule applied to the problem \(Q_2\).

- Proof.
  - let \(\pi\) be the optimal schedule of the problem \(Q_2\).
  \[ \sum_{j=1}^{W} \max(C_i(\sigma_2), a_{[j]}) \leq \sum_{j=1}^{W} \max(C_\pi, a_j) = \sum_{j=1}^{W} C_\pi \]
Lower bounds

- Lower bound(2) cont..
  - Lemma 2.

  **Lemma 2.** Let \((x_1, \ldots, x_q)\) and \((y_1, \ldots, y_q)\) be two series of numbers such that \(x_1 \leq \cdots \leq x_q\). The series \((y_1', \ldots, y_q')\) is obtained by sorting the series \((y_1, \ldots, y_q)\) in decreasing order. The following relation holds:

  \[
  \sum_{i=1}^{q} x_i y_i \geq \sum_{i=1}^{q} x_i y_i'.
  \]

- Lemma 3.

  **Lemma 3.** Let \((\alpha_1, \ldots, \alpha_q)\) be a series of positive numbers in decreasing order. \((x_1, \ldots, x_q)\) be a series of numbers in nondecreasing order and \((y_1, \ldots, y_q)\) be a series of numbers. The series \((y_1', \ldots, y_q')\) is obtained by sorting the series \((y_1, \ldots, y_q)\) in nondecreasing order. The following relation holds:

  \[
  \sum_{i=1}^{q} \alpha_i \max(x_i, y_i) \geq \sum_{i=1}^{q} \alpha_i \max(x_i, y_i').
  \]
Theorem 8. Let $b_i = r_i + p_i, i = 1, \ldots, n$ and $(b_{[1]}, \ldots, b_{[n]})$ be the series obtained by sorting the series $(b_1, \ldots, b_n)$ in nondecreasing order. If $(\delta_{[1]}, \ldots, \delta_{[n]})$ is the series obtained by sorting the series $(\delta_1, \ldots, \delta_n)$ in decreasing order, then $\sum_{i=1}^{n} \delta_{[i]} \max(C_{[i]}(\xi), b_{[i]})$ is a lower bound on $\sum_{i \in N} \delta_i C_i(\sigma)$.

where, $\xi$ is the schedule obtained by SRPT rule (Splitted).
Heuristic

Suggested algorithm

- based on LB_{GS} (LB1) and LB_{NYC} (LB2)
- 2steps - WSPT, Improve

Let $S = \{1, 2, ..., n\}$, $M = \{1, ..., m\}$, $\phi_j = -\infty$, $\sigma_j = 0$, $\forall j \in M$, $\mu = 1$ and $WC = 0$

while $S \neq \emptyset$
  Select $i \in S$ with $p_i/w_i = \min\{p_j/w_j, r_x \text{ is minimum}, x \in S\}$, $\phi_i := \max(r_i, \phi_r) + p_i$, $\sigma_i := \sigma_r \cup \{i\}$, $\mu := \arg\min_{j \in S} \phi_j$, $S := S - \{i\}$
end while

Let $\phi_j = -\infty$, $\forall j \in M$

for $\mu = 1$ to $m$
  while $\sigma_\mu \neq \emptyset$
    Select $i_1 \in \sigma_\mu$ with $p_i/w_i = \min\{p_j/w_j \mid j \in \sigma_\mu\}$
    Select $i_2 \in \sigma_\mu$ with $p_i/w_i = \min\{p_j/w_j \mid j \in \sigma_\mu\}$
    if $PRTWC(i_1, i_2, t) < 0$ then $i := i_1$ else $i := i_2$ end if
    Select $j \in \sigma_\mu$ with $C_j(R_j(t)) = \min_{x \in \sigma_\mu, r_x \leq t} C_x(R_x(t))$ and $h \in \sigma_\mu - \{i, j\}$ with $r_h = \min_{x \in \sigma_\mu - \{i, j\}} r_x$
    if $F(\sigma \circ x) - F(\sigma \circ j) \geq (R_a(\max(r_i, \max(t, r_j) + p_j) + p_i) - R_a(\max(r_i + t) + p_j) + p_i) \sum_{x \in \sigma_\mu} w_x$ then $x := j$ else $x := i$ end if
    $t = R_x(t) + p_x$, $\sigma = \sigma \cup x$, $WC := WC + w_x$, $\sigma_\mu = \sigma_\mu - \{x\}$
  end while
end for

WSPT order

$WC_{ji}(t) = (w_i + w_j) \max(t, r_i) + w_j p_i$

$PRTWC(i, j, t) = WC_{ji}(t) - WC_{ij}(t)$
Branch-and-bound algorithm

- Branching
  - Job by Job branching
  - Job assigned to the earliest start machine

- Bounding
  - LB\textsubscript{NYC}

- Dominance rule
  - Theorem 1, 2, 3
Computational results

- **Configuration**
  - Coded in C
  - Biprocessor G5000 HP Workstation, 440 MHz, 1Giga Ram

- **Instances**
  - Processing time ~ DU(1,100)
  - Weight ~ DU(1,10)
  - Release dates ~ DU(0, 50.5 * n * α/m), α = {0.2, 0.4, 0.6, 0.8, 1, 1.25, 1.50, 1.75, 2.0, 3.0}
  - # of jobs = {20, 30, 40, 60, 80, 100, 120, 140, 160, 180, 200, 300, 400 and 500}
  - 10 instances by each level
Computational results

- Computational results of heuristics

<table>
<thead>
<tr>
<th>n</th>
<th>$g(M_1)/%$</th>
<th>gap$\left(\text{LB}<em>{\text{NYC}}/\text{LB}</em>{\text{GS}}\right)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.34</td>
<td>34.34</td>
</tr>
<tr>
<td>5</td>
<td>2.23</td>
<td>158.30</td>
</tr>
<tr>
<td>9</td>
<td>2.81</td>
<td>333.24</td>
</tr>
</tbody>
</table>

- Gap
  
  $\left(\text{LB}_{\text{NYC}}/\text{LB}_{\text{GS}}\right) = 100 \times (\text{NYC} - \text{LB}_{\text{GS}}) / \text{LB}_{\text{GS}}$

- Computational results of branch and bound

<table>
<thead>
<tr>
<th>n</th>
<th>m</th>
<th>Time</th>
<th>NoRs</th>
<th>Ngenes</th>
<th>LB$_{NYC}$</th>
<th>Opt</th>
<th>Heur</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>2</td>
<td>44.47</td>
<td>0</td>
<td>28.8</td>
<td>7963.23</td>
<td>8921.98</td>
<td>9174.17</td>
</tr>
<tr>
<td>3</td>
<td>36.90</td>
<td>0</td>
<td>24.9</td>
<td>6585.21</td>
<td>7663.90</td>
<td>6579.36</td>
<td>6625.39</td>
</tr>
<tr>
<td>5</td>
<td>26.30</td>
<td>0</td>
<td>22.0</td>
<td>5696.61</td>
<td>18948.28</td>
<td>19268.15</td>
<td>15907.95</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>613.74</td>
<td>0</td>
<td>683.3</td>
<td>21250.34</td>
<td>23391.89</td>
<td>23925.49</td>
</tr>
<tr>
<td>3</td>
<td>709.5</td>
<td>2</td>
<td>811.1</td>
<td>16384.68</td>
<td>18948.28</td>
<td>19268.15</td>
<td>15907.95</td>
</tr>
<tr>
<td>5</td>
<td>346</td>
<td>9</td>
<td>414.2</td>
<td>13856.94</td>
<td>15139</td>
<td>15907.95</td>
<td>15907.95</td>
</tr>
<tr>
<td>60</td>
<td>2</td>
<td>7739.60</td>
<td>0</td>
<td>8095.6</td>
<td>37782.67</td>
<td>40346.65</td>
<td>41870.05</td>
</tr>
<tr>
<td>3</td>
<td>2528.82</td>
<td>11</td>
<td>3099.7</td>
<td>28642.10</td>
<td>31770.57</td>
<td>33100.76</td>
<td>33100.76</td>
</tr>
<tr>
<td>5</td>
<td>1744.54</td>
<td>17</td>
<td>1680.50</td>
<td>23119.45</td>
<td>25039.67</td>
<td>25926.08</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>2</td>
<td>60775.37</td>
<td>2</td>
<td>70611.58</td>
<td>67961.26</td>
<td>68984.86</td>
<td>69977.42</td>
</tr>
<tr>
<td>3</td>
<td>8510.01</td>
<td>23</td>
<td>8495.10</td>
<td>44513.28</td>
<td>49418.27</td>
<td>50809.39</td>
<td>50809.39</td>
</tr>
<tr>
<td>5</td>
<td>4117.10</td>
<td>32</td>
<td>6184.50</td>
<td>34870.09</td>
<td>37581.17</td>
<td>38808.89</td>
<td>38808.89</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
<td>982856.14</td>
<td>8</td>
<td>1206232.01</td>
<td>83170.53</td>
<td>89521.53</td>
<td>90919.89</td>
</tr>
</tbody>
</table>

NoRs : Not solved problems
Ngenes : generated nodes
Conclusions

- Identical parallel machine scheduling
  - with ready time
  - weighted flow times

- Branch and bound algorithm
  - Dominance rules
  - Lower Bounds

- Adv&DisAdv