The Shifting Bottleneck Heuristic

- successful heuristic to solve makespan minimization for job shop
- iterative heuristic
- determines in each iteration the schedule for one additional machine
- uses reoptimization to change already scheduled machines
- can be adapted to more general job shop problems
  - other objective functions
  - workcenters instead of machines
  - set-up times on machines
  - ...
**Basic Idea**

**Notation:** \( M \) set of all machines

**Given:**
fixed schedules for a subset \( M_0 \subset M \) of machines
(i.e. a selection of disjunctive arcs for cliques corresponding to these machines)

**Actions in one iteration:**
- select a machine \( k \) which has not been fixed (i.e. a machine from \( M \setminus M_0 \))
- determine a schedule (selection) for machine \( k \) on the base of the fixed schedules for the machines in \( M_0 \)
- reschedule the machines from \( M_0 \) based on the other fixed schedules
Selection of a machine

Idea: Choose unscheduled machine which causes the most problems (bottleneck machine)

Realization:

- Calculate for each operation on an unscheduled machine the earliest possible starting time and the minimal delay between the end of the operation and the end of the complete schedule based on the fixed schedules on the machines in $M_0$ and the job orders.

- Calculate for each unscheduled machine a schedule respecting these earliest release times and delays.

- Choose a machine with maximal completion time and fix the schedule on this machine.
**Technical realization**

- Define graph $G' = (N, A')$:
  - $N$ same node set as for the disjunctive graph
  - $A'$ contains all conjunctive arcs and the disjunctive arcs corresponding to the selections on the machines in $M_0$

- $C_{\text{max}}(M_0)$ is the length of a critical path in $G'$

**Comments:**

- with respect to $G'$ operations on machines from $M \setminus M_0$ may be processed in parallel
- $C_{\text{max}}(M_0)$ is the makespan of a corresponding schedule
Technical realization - cont. 1

for an operation \((i, j); \ i \in M \setminus M_0\) let

- \(r_{ij}\) be the length of the longest path from \(U\) to \((i, j)\) in \(G'\)
- \(\Delta_{ij}\) be the length of the longest path from \((i, j)\) to \(V\) in \(G'\)
- \(d_{ij} = C_{\text{max}}(M_0) - \Delta_{ij} + p_{ij}\)

Comments:

- \(r_{ij}\) is the release time of \((i, j)\) w.r.t. \(G'\)
- \(d_{ij}\) is the due date of \((i, j)\) for a schedule with makespan \(\leq C_{\text{max}}(M_0)\)
**Technical realization - cont. 2**

For each machine from $M \backslash M_0$ solve the nonpreemptive one-machine problem with release times and due dates and objective to minimize the maximum lateness

**Result:** values $L_{\text{max}}(i)$ for all $i \in M \backslash M_0$

**Action:**

- Chose machine $k$ as the machine with the largest maximum lateness
- Schedule machine $k$ according to the optimal schedule of the one-machine problem
- Add $k$ to $M_0$ and the corresponding disj. arcs to $G'$

**Remark:**

$$C_{\text{max}}(M_0 \cup k) \geq C_{\text{max}}(M_0) + L_{\text{max}}(k)$$
Technical realization - Example 1

Given:

- $M_0 = \{M3\}$
- on $M3$: $(3, 2) \rightarrow (3, 1) \rightarrow (3, 3)$

Graph $G'$:

$C_{max}(M_0) = 13$
Technical realization - Example 2

**Machine M1:**

<table>
<thead>
<tr>
<th>$(i, j)$</th>
<th>(1, 1)</th>
<th>(1, 2)</th>
<th>(1, 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{ij}$</td>
<td>12</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>$\Delta_{ij}$</td>
<td>1</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>$d_{ij}$</td>
<td>13</td>
<td>3</td>
<td>12</td>
</tr>
</tbody>
</table>

$L_{max}(M1) = 0$

**Machine M2:**

<table>
<thead>
<tr>
<th>$(i, j)$</th>
<th>(2, 1)</th>
<th>(2, 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{ij}$</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>$\Delta_{ij}$</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>$d_{ij}$</td>
<td>12</td>
<td>8</td>
</tr>
</tbody>
</table>

$L_{max}(M2) = 0$
Technical realization - Example 3

Choose machine $M1$ as the machine to fix the schedule:

$$\text{add } (1, 2) \rightarrow (1, 3) \rightarrow (1, 1) \text{ to } G'$$

$$M_0 = \{M1, M3\}$$

$C_{max}(M_0) = 13$
Reschedule Machines

Idea: try to reduce the makespan of the schedule for the machines in $M_0$

Realization:

- consider the machines from $M_0$ one by one
- remove the schedule of the chosen machine and calculate a new schedule based on the earliest starting times and delays resulting from the other machines of $M_0$ and the job orders
Technical realization

For a chosen machine \( l \in M_0 \setminus \{k\} \) do:

- remove the arcs corresponding to the selection on machine \( l \) from \( G' \)
- call new graph \( G'' \)
- calculate values \( r_{ij}, \Delta_{ij} \) and \( d_{ij} \) in graph \( G''' \)
- reschedule machine \( l \) according to the optimal schedule of the single machine maximum lateness problem with release and due dates
Technical realization - Example 1

\[ M_0 \setminus \{ k \} = \{ M3 \}, \text{ thus } l = M3 \]

Graph \( G''' \):

\[
\begin{array}{c}
\begin{array}{c}
(3, 1) \quad (3, 2) \quad (3, 3)
\end{array}
\begin{array}{|c|c|c|c|}
\hline
(i, j) & (3, 1) & (3, 2) & (3, 3) \\
\hline
r_{ij} & 0 & 3 & 7 \\
\hline
\Delta_{ij} & 7 & 3 & 1 \\
\hline
d_{ij} & 5 & 8 & 8 \\
\hline
\end{array}
\end{array}
\]

\[ C_{max}(G''') = 8; \]

\[ L_{max}(M3) = 0 \]
Technical realization - Example 2

add \((3,1) \rightarrow (3,2) \rightarrow (3,3)\) to \(G''\)

New graph:

\[
\begin{array}{c}
3,1 \quad 2,1 \quad 1,1 \\
4 \quad 2 \quad 1
\end{array}
\]

Selection: M1, M3

Conjunctive arcs

\[C_{\text{max}}^{\text{new}}(M_0) = 8\]
Shifting Bottleneck Heuristic - 1

**Step 1:** (Initialization)

\[ M_0 := \emptyset; \]
\[ G := \text{graph with all conjunctive arcs}; \]
\[ C_{\text{max}}(M_0) := \text{length longest path in } G; \]

**Step 2:** (Analyze unscheduled machines)

FOR ALL \( i \in M \setminus M_0 \) DO

FOR ALL operation \( (i, j) \) DO

\[ r_{ij} := \text{length longest path from } U \text{ to } (i, j) \text{ in } G; \]

\[ d_{ij} := C_{\text{max}}(M_0) - \text{length longest path from } (i, j) \text{ to } V \text{ in } G + p_{ij}; \]

minimize \( L_{\text{max}} \) for single machine problem on machine \( i \) subject to release dates \( r_{ij} \), due dates \( d_{ij} \);

\[ L_{\text{max}}(i) := \text{minimum lateness on } i; \]
Shifting Bottleneck Heuristic - 2

Step 3: (Bottleneck selection)

determine \( k \) such that

\[
L_{\text{max}}(k) = \max_{i \in M \setminus M_0} L_{\text{max}}(i);
\]

schedule machine \( k \) according to the optimal solution in Step 2;

add corr. disjunctive arcs to \( G \);

\( M_0 := M_0 \cup \{k\} \);
Shifting Bottleneck Heuristic - 3

Step 4: (Resequencing of machines)

FOR ALL $i \in M_0 \setminus \{k\}$ DO

delete disjunctive arcs corresponding to machine $k$ from $G$;

FOR ALL operation $(i, j)$ DO

$r_{ij} := \text{length longest path from } U \text{ to } (i, j) \text{ in } G$;

$d_{ij} := C_{\text{max}}(M_0) - \text{length longest path from } (i, j) \text{ to } V \text{ in } G + p_{ij}$;

minimize $L_{\text{max}}$ for single machine problem on machine $i$ subject to release dates $r_{ij}$, due dates $d_{ij}$;

insert corr. disjunctive arcs to $G$;

Step 5: (Stopping condition)

IF $M_0 = M$ THEN Stop
ELSE go to Step 2;
$SBH - Example - cont. 1$

$M_0 = \{M1, M3\};$ thus $M2$ is bottleneck graph $G$:

\[
\begin{array}{c|c|c}
(i,j) & (2,1) & (2,3) \\
\hline
r_{ij} & 4 & 0 \\
\Delta_{ij} & 3 & 7 \\
d_{ij} & 7 & 3 \\
\end{array}
\]

$C_{max}(G) = 8$; 

$LM_{ax}(M2) = -1$
SBH - Example - cont. 2

\[ M_0 = \{M1, M2, M3\}, \ C_{max}(M_0) = 8 \]

Graph \( G \):

Corresponding Schedule:

Makespan \( C_{max} = 8 \)
An Important Subproblem

Within the SHB the following one-machine problem occurs frequently:

**Given:**
- release dates $r_i$
- due dates $d_i$

**Goal:** Find a nonpreemptive schedule with minimal maximum lateness

**Remarks:**
- this problem was also used within branch and bound to calculate lower bounds
- the problem is NP-hard
- there are efficient solution methods for smaller instances
- the actual one-machine problem is a bit more complicated than stated above (see foll. example)
Example Delayed Precedences 1

Jobs: J1  (1,1)--> (2,1)
J2  (2,2)--> (1,2)
J3  (3,3)
J4  (3,4)

Processing Times: \[ p_{11} = 1, \ p_{21} = 1 \]
\[ p_{22} = 1, \ p_{12} = 1 \]
\[ p_{33} = 4 \]
\[ p_{34} = 4 \]

Initial graph \( G \):

[Diagram showing the initial graph with nodes labeled U, V, 1,1, 2,1, 2,2, 1,2, 3,3, 3,4 and arrows indicating the precedence relationships.]
Example Delayed Precedences 2

After 2 iterations SBH we get:

\[ M_0 = \{M3, M1\} \]

(3, 4) \rightarrow (3, 3) \text{ and } (1, 2) \rightarrow (1, 1)

Resulting graph \( G: (C_{max}(M_0) = 8) \)

\[
\begin{array}{c}
1,1 & 2,1 \\
2,2 & 1,2 \\
3,3 \\
3,4
\end{array}
\]

3. iteration: only \( M2 \) unscheduled

<table>
<thead>
<tr>
<th>((i, j))</th>
<th>(p_{ij})</th>
<th>(r_{ij})</th>
<th>(d_{ij})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2, 1)</td>
<td>1</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>(2, 2)</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>
**Example Delayed Precedences 3**

Possible schedules for $M_2$:

Both schedules are feasible and have $L_{max} \leq 0$

But: 2. schedule leads to

\[
\begin{array}{c}
1,1 & 2,1 \\
2,2 & 1,2 \\
3,3 & 3,4 \\
3,4 & 3,4 \\
\end{array}
\]

which contains a cycle
Delayed Precedences

The example shows:

- not all solutions of the one-machine problem fit to the given selections for machines from $M_0$
- the given selections for machines from $M_0$ may induce precedences for machines from $M \setminus M_0$

Example: scheduling operation $(1, 2)$ before $(1, 1)$ on machine M1 induces a delayed precedence constraint between $(2, 2)$ and $(2, 1)$ with length 3

operation $(2, 1)$ has to start at least 3 time units after $(2, 2)$

this time is needed to process operations $(2, 2), (1, 2)$, and $(1, 1)$
Opgaven voor werkcollege

| Date:    | Thursday, 23 May, 2002  
| Time:    | 13.45-15.30 (5+6)    
| Room:    | BB 3                 
| Exercises: |                    
| College 5 | 5.1, 5.3, 5.6 (a) 
| College 6 | 5.5 (a), 5.7 (b), 5.8 |