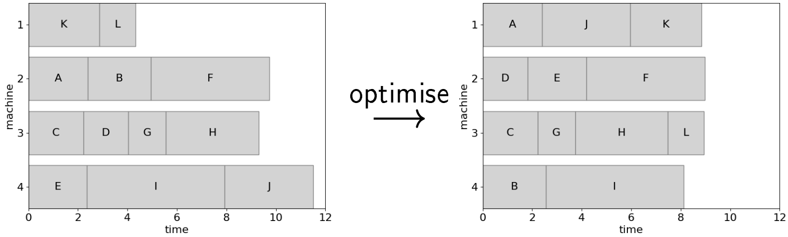
Explaining Makespan Schedules

Myles Lee

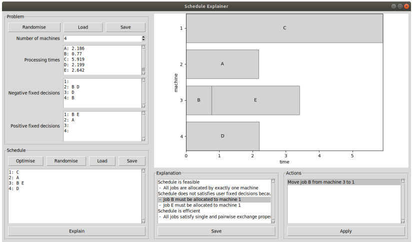
25^{th} June 2019

Introduction

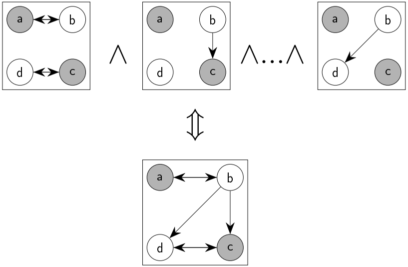


- Solvers and schedules are difficult to understand
- Apply argumentation to makespan schedules to generate explanations

Contributions

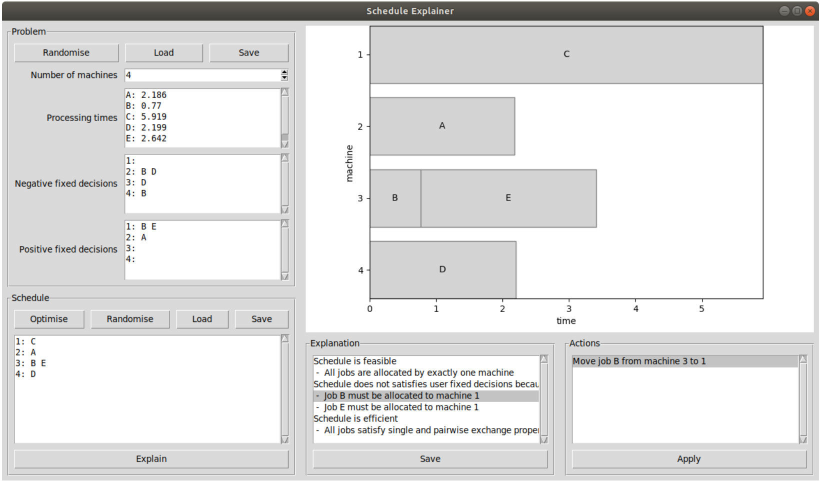


- Interactive tool
- Algorithms



- Theoretical extensions
- Discussion

Demonstration



Abstract Argumentation

- An abstract argumentation framework is a directed graph $\langle Args, \rightsquigarrow \rangle$.
- Extension E is a subset of $Args$.

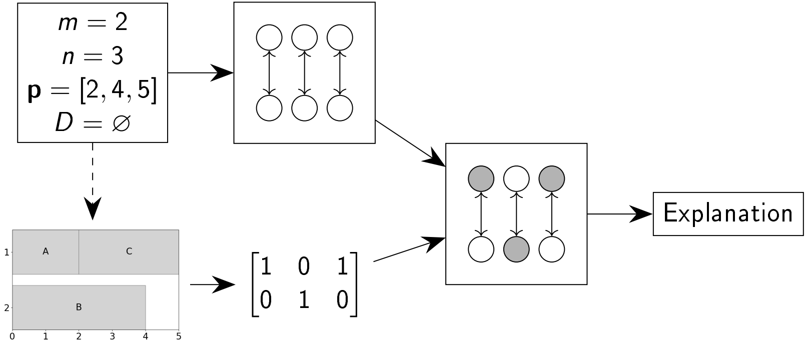
Definition

E is conflict-free on $\langle Args, \rightsquigarrow \rangle$ iff $\forall a, b \in E . a \not\rightsquigarrow b$.

Definition

E is stable on $\langle Args, \rightsquigarrow \rangle$ iff E is conflict-free on $\langle Args, \rightsquigarrow \rangle$ and $\forall a \in Args \setminus E . \exists e \in E . e \rightsquigarrow a$.

Pipeline



$m = 2$
$n = 3$
$\mathbf{p} = [2, 4, 5]$
$D = \varnothing$
A
C
B
1
2
0 1 2 3 4 5
$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
Explanation

Feasibility Framework Construction

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\forall j \in \mathcal{J} .\, \sum_{i \in \mathcal{M}} x_{i,j} = 1$$

$$\begin{tikzcd}
\langle 1, \mathtt{a} \rangle \arrow[d, leftrightarrow] & \langle 1, \mathtt{b} \rangle \arrow[d, leftrightarrow] & \langle 1, \mathtt{c} \rangle \arrow[d, leftrightarrow] \\
\langle 2, \mathtt{a} \rangle & \langle 2, \mathtt{b} \rangle & \langle 2, \mathtt{c} \rangle
\end{tikzcd}$$

$$Args_F = \mathcal{M} \times \mathcal{J}$$

$$\langle i, j \rangle \rightsquigarrow_F \langle i', j' \rangle \text{ iff } i \neq i' \land j = j'$$

Framework Modelling

Definition

Framework $\langle Args, \rightsquigarrow \rangle$ stability-models a property P iff $\forall E \subseteq Args . E$ is stable on $\langle Args, \rightsquigarrow \rangle \Leftrightarrow P$

$$P_F \equiv \forall j \in \mathcal{J} . \sum_{i \in \mathcal{M}} x_{i,j} = 1$$

$$\langle Args_F , \rightsquigarrow_F \rangle \quad \begin{tikzcd} \langle 1, \mathrm{a} \rangle \arrow[d, leftrightarrow] & \langle 1, \mathrm{b} \rangle \arrow[d, leftrightarrow] & \langle 1, \mathrm{c} \rangle \arrow[d, leftrightarrow] \\ \langle 2, \mathrm{a} \rangle & \langle 2, \mathrm{b} \rangle & \langle 2, \mathrm{c} \rangle \end{tikzcd}$$

Theorem

$\langle Args_F , \rightsquigarrow_F \rangle$ stability-models P_F

Algorithm Notation

$$\mathbf{0}^{2\times 2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\ominus \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \owedge \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \ovee \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

Construct-Feasibility Algorithm

Algorithm

function Construct-Feasibility(m, n)
  $\twoheadrightarrow_F \leftarrow \mathbf{0}^{(m \times n)^2}$
  for $i_1, i_2 \in \mathcal{M}, j \in \mathcal{J}$ **do**
    if $i_1 \neq i_2$ **then**
      $\twoheadrightarrow_{F\, i_1,j,i_2,j} \leftarrow 1$
    end if
  end for
  return $\twoheadrightarrow_F$
end function

Explain-Stability Algorithm

Algorithm

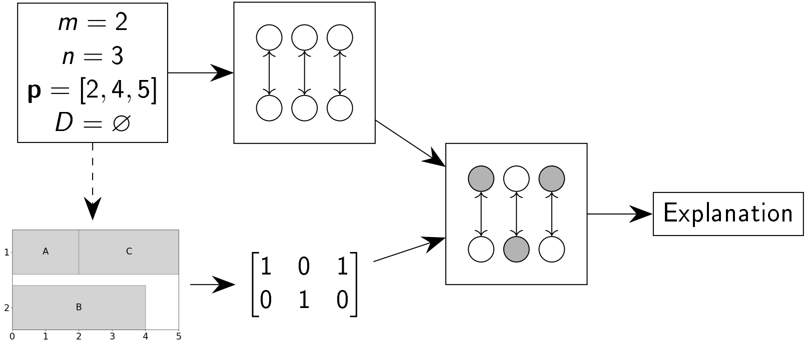
function Explain-Stability($\mathbf{x}$, $\twoheadrightarrow$, $\bar{\mathbf{u}}$, $\bar{\mathbf{c}}$)
    $\mathbf{u} \leftarrow \textcircled{$\neg$}\ \mathbf{x}$
    $\mathbf{c} \leftarrow \mathbf{0}^{(m \times n)^2}$
    for $i \in \mathcal{M}, j \in \mathcal{J}$ **do**
        if $x_{i,j} = 1$ **then**
            $\mathbf{u} \leftarrow \mathbf{u}\ \textcircled{$\wedge$}\ \textcircled{$\neg$} \twoheadrightarrow_{i,j}$
            $c_{i,j} \leftarrow \mathbf{x}\ \textcircled{$\wedge$} \twoheadrightarrow_{i,j}$
        end if
    end for
    $\mathbf{u} \leftarrow \mathbf{u}\ \textcircled{$\wedge$}\ \textcircled{$\neg$}\ \bar{\mathbf{u}}$
    $\mathbf{c} \leftarrow \mathbf{c}\ \textcircled{$\wedge$}\ \textcircled{$\neg$}\ \bar{\mathbf{c}}$
    return $\langle \mathbf{u}, \mathbf{c} \rangle$
end function

Explain-Feasibility Algorithm

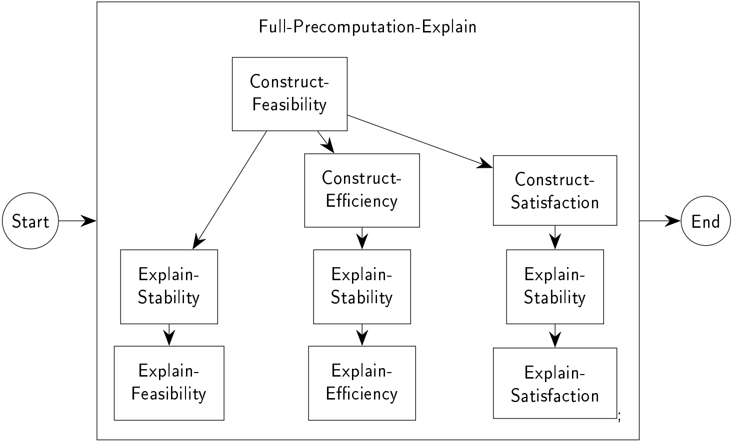
Algorithm

function Explain-Feasibility($\mathbf{u}$, $\mathbf{c}$)
    if $m = 0$ **then**
        if $n = 0$ **then**
            There are no jobs, so the schedule is trivially feasible.
        else
            There are no machines to allocate to jobs.
        end if
    else
        if $\mathbf{u} = \mathbf{0} \wedge \mathbf{c} = \mathbf{0}$ **then**
            All jobs are allocated to exactly one machine.
        else
            for $j \in \mathcal{J}$ **do**
                if $u_{:,j} = 1$ **then**
                    Job j is not allocated to any machine.
                end if
                if $c_{:,j,:,j} \neq \mathbf{0}$ **then**
                    Job j is over-allocated to machines $\{i \mid i \in \mathcal{M}, x_{i,j} = 1\}$.
                end if
            end for
        end if
    end if
end function

Pipeline

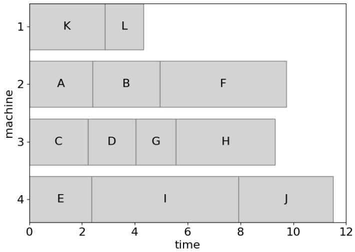

$m = 2$
$n = 3$
$\mathbf{p} = [2, 4, 5]$
$D = \varnothing$
A
C
B
1
2
0 1 2 3 4 5
$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
Explanation

Algorithms Overview



Generating Efficiency Explanations

- Single and pair-wise exchange properties
- Fixed decision awareness
- Explanations are:
  - superfluous
  - local
  - expensive



Generating Fixed Decisions Explanations

Algorithm

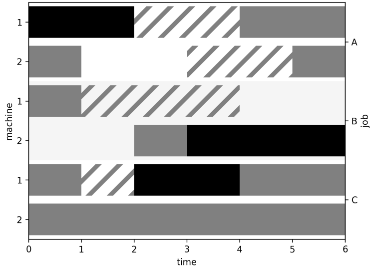
function Explain-Satisfaction(D, $\mathbf{u}$, $\mathbf{c}$)
  for $j \in \mathcal{J}$ **do**
    if $\exists i \in \mathcal{M}\ \langle i, j \rangle \notin D^-$ **then**
      Job j cannot be allocated to any machine.
    end if
    if D^- and D^+ are not disjoint **then**
      Job j has conflicting negative and positive fixed decisions.
    end if
    if $|\{i \in \mathcal{M} \mid \langle i, j \rangle \in D^+\}| > 1$ **then**
      Job j cannot be allocated to multiple machines.
    end if
  end for
  ...
end function

Partial-Precomputation Optimisation

- $\twoheadrightarrow$ construction has $\mathcal{O}(m^2 n^2)$ memory usage.
- Inline stability and framework construction algorithms
- Operate on sub-graphs $\twoheadrightarrow_{i,j}$

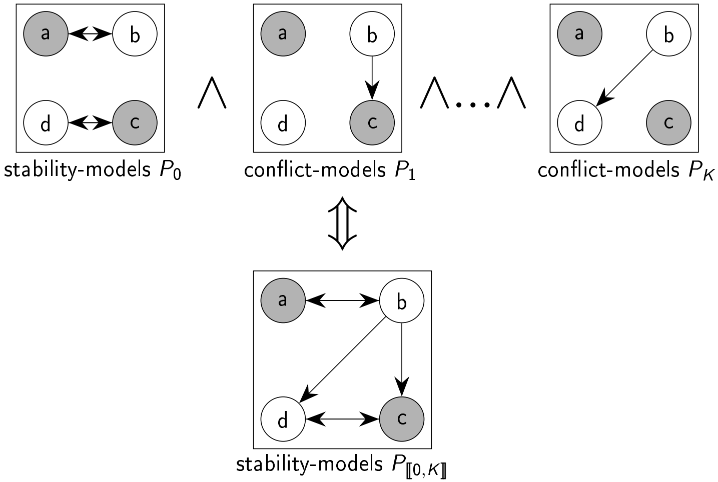
- Full-Precomputation complexity: $\mathcal{O}(m^2 n^2)$
- Partial-Precomputation complexity: $\mathcal{O}(mn^2)$

Time-indexed Interval Scheduling



- Single-allocation of jobs
- No overlapping jobs
- Start and finish times
- Fixed decisions

Union of Modelling Frameworks Theorem



Theoretical Applications

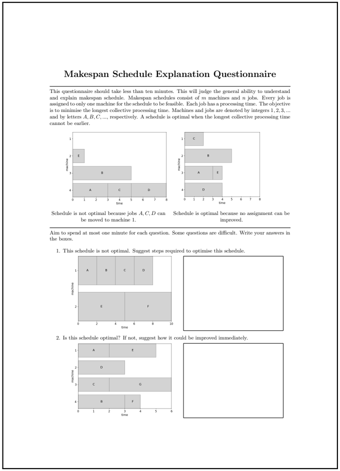
Definition

Property P is stability-modellable iff $\exists\langle Args, \rightsquigarrow\rangle.\langle Args, \rightsquigarrow\rangle$ stability models P.

Theorem

Interval scheduling feasibility is stability-modellable.

Questionnaire



Makespan Schedule Explanation Questionnaire

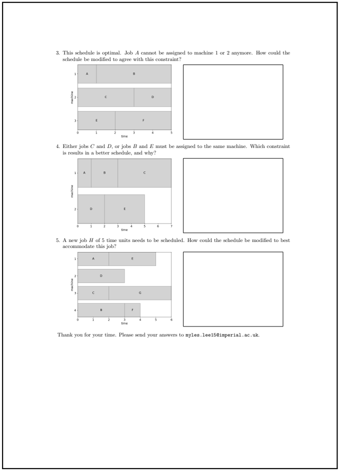
This questionnaire should take less than ten minutes. This will judge the general ability to understand and explain makespan schedule. Makespan schedules consist of m machines and n jobs. Every job is assigned to only one machine for the schedule to be feasible. Each job has a processing time. The objective is to minimise the longest collective processing time. Machines and jobs are denoted by integers $1, 2, 3, ...$ and by letters $A, B, C, ...$, respectively. A schedule is optimal when the longest collective processing time cannot be earlier.

Schedule is not optimal because jobs A, C, D can be moved to machine 1.

Schedule is optimal because no assignment can be improved.

Aim to spend at most one minute for each question. Some questions are difficult. Write your answers in the boxes.

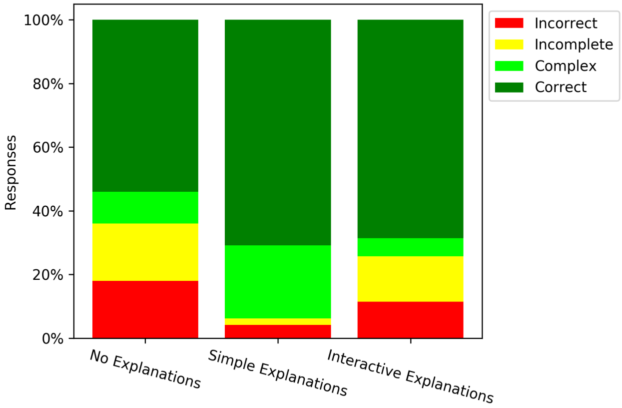
1. This schedule is not optimal. Suggest steps required to optimise this schedule.
2. Is this schedule optimal? If not, suggest how it could be improved immediately.



3. This schedule is optimal. Job A cannot be assigned to machine 1 or 2 anymore. How could the schedule be modified to agree with this constraint?
4. Either jobs C and D, or jobs B and E must be assigned to the same machine. Which constraint is results in a better schedule, and why?
5. A new job H of 5 time units needs to be scheduled. How could the schedule be modified to best accommodate this job?

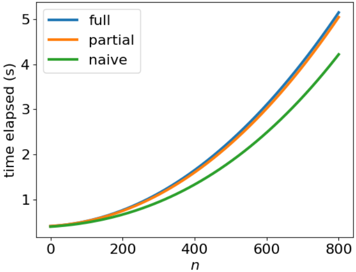
Thank you for your time. Please send your answers to `myles.lee15@imperial.ac.uk`.

Questionnaire Results

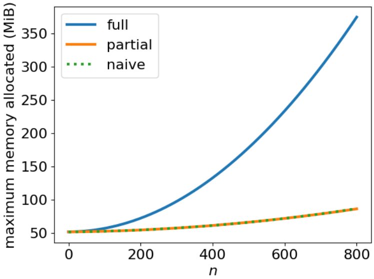


Profiling

- $m = 10$



CPU Performance



Memory Performance

blah