Constraint-based Scheduling: Introduction

Hana Rudová*, Rong Qu’

*Fakulty of Informatics, Masaryk University
  Brno, Czech Republic
  http://www.fi.muni.cz/~hanka

’School of Computer Science
  University of Nottingham, UK
  http://www.cs.nott.ac.uk/~rxq
Outline

1. Scheduling
2. CSP model
3. Resources
4. Optimization
5. Problem: Basic Class Timetabling
Scheduling

optimal resource allocation of a given set of activities in time

resource or machine

activity or task

Machine $M_j, j = 1, \ldots, 3$

Task $T_i, i = 1, \ldots, 9$

Machine-oriented Gantt chart

Constraint-based Scheduling: Introduction 3/22
Example: Bicycle Assembly

- 3 workers who can perform tasks
- 10 tasks with its own duration
- Precedence constraints ($T_i \preceq T_j$): activity must be processed before other activity
- No preemption: activity cannot be interrupted during processing

**Schedule**

<table>
<thead>
<tr>
<th>T4</th>
<th>T6</th>
<th>T8</th>
<th>T10</th>
</tr>
</thead>
</table>

| T5 | T3 | T9 |   |

Diagram showing task dependencies and durations.
Example: Bicycle Assembly

- 3 workers who can perform tasks
- 10 tasks with its own duration
- Precedence constraints ($T_i \prec T_j$): activity must be processed before other activity
- No preemption: activity cannot be interrupted during processing

Schedule:

<table>
<thead>
<tr>
<th>T4</th>
<th>T6</th>
<th>T8</th>
<th>T10</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T5</th>
<th>T3</th>
<th>T9</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>

Optimal schedule:

<table>
<thead>
<tr>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T3</th>
<th>T8</th>
<th>T9</th>
<th>T10</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>7</td>
<td>14</td>
<td>16</td>
<td>24</td>
<td>32</td>
</tr>
</tbody>
</table>
Example: Classroom allocation

- One day seminar with several courses to be presented in several available rooms
- **14 courses** (A, B, ... N) each course has several meetings with pre-assigned time periods
- **5 rooms** (1, 2, 3, 4, 5) ... resources
- **Find suitable room for each meeting**

<table>
<thead>
<tr>
<th>Course</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periods</td>
<td>2</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td>8</td>
<td>3</td>
<td>2</td>
<td></td>
<td>7</td>
<td>4</td>
<td></td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

**Solution = Schedule/Timetable:**

<table>
<thead>
<tr>
<th>Periods</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Room 1</td>
<td>D</td>
<td>D</td>
<td>C</td>
<td>C</td>
<td>F</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Room 2</td>
<td>I</td>
<td>I</td>
<td>E</td>
<td>E</td>
<td>G</td>
<td>G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Room 3</td>
<td>H</td>
<td>H</td>
<td>J</td>
<td>K</td>
<td>K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Room 4</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Room 5</td>
<td>A</td>
<td>L</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Constraint-based Scheduling: Introduction**
Scheduling problems

- Educational timetabling
- Schedule for NHL
- Sports scheduling
- Transport scheduling
- Nurse rostering
- Employees scheduling
- Car production
- Schedule of flexible assembly systems
- Allocation of jobs to computational resources
- Machine scheduling
- Software project planning
- Project planning and scheduling
Activity A is an entity occupying some space (resources) and time

Variables and their domains for each activity for **time assignment**

- **start(A)**: start time of the activity
  - activity cannot start before its release date
  - \( \text{est}(A) = \min(\text{start}(A)) \), earliest start time

- **end(A)**: completion time of the activity
  - activity must finish before the deadline
  - \( \text{lct}(A) = \max(\text{end}(A)) \), latest completion time

- **p(A)**: processing time (duration) of the activity
  - \( \text{start}(A) = \{\text{est}(A), \ldots, (\text{lct}(A)-p(A))\} \)
  - \( \text{end}(A) = \{(\text{est}(A)+p(A)), \ldots, \text{lct}(A)\} \)
Scheduling as a CSP: basic constraints I.

- Non-preemptive activity: no interruption during processing
  - \( \text{start}(A) + p(A) = \text{end}(A) \)

- Preemptible activity: can be interrupted during its processing
  - \( \text{start}(A) + p(A) \leq \text{end}(A) \)
Scheduling as a CSP: basic constraints I.

- **Non-preemptive activity:** no interruption during processing
  \[ \text{start}(A) + p(A) = \text{end}(A) \]

- **Preemptible activity:** can be interrupted during its processing
  \[ \text{start}(A) + p(A) \leq \text{end}(A) \]

Scheduling as a CSP: basic constraints II.

- **Sequencing** $A \ll B$ of activities $A,B$
  (also: precedence constraint between activities $A,B$)
  - $\text{end}(A) \leq \text{start}(B)$

- **Disjunctive constraint**: non-overlapping of activities $A, B$
  - non-preemptive activities
  - $A \ll B$ or $B \ll A$
  - $\text{end}(A) \leq \text{start}(B)$ or $\text{end}(B) \leq \text{start}(A)$
  - related with the idea of unary resource
Domain variables for resources
- \text{cap}(A): \text{requested capacity of the resource}
- unary resources
- cumulative resources
- producible/consumable resources
- \text{resource}(A): \text{alternative resources for A}
Unary (disjunctive) resources

- Each activity requests unary capacity of the resource: \( \text{cap}(A) = 1 \)
- Single activity can be processed at given time
- Any two non-preemptive activities are related by the disjunctive constraint \( A \ll B \) or \( B \ll A \)

Example: one machine with jobs running on it
Cumulative (discrete) resources

- Each activity uses some capacity of the resource $\text{cap}(A)$
- Several activities can be processed in parallel if a resource capacity is not exceeded

Example: multi-processor computer with parallel jobs
Producible/consumable resources

- Resource = reservoir
- Activity consumes some quantity of the resource $\text{cap}(A)<0$ or activity produces some quantity of the resource $\text{cap}(A)>0$
- Minimal capacity is requested (consumption) and maximal capacity cannot be exceeded (production)

Example: inventory for some products, activities producing them and activities using them in other production
Alternative resources

- Activity can be processed on a set of alternative resources defined by the domain variable \textit{resource(A)}
- One of them is selected for the activity
- Alternative unary resources
  - activity can be processed on any of the unary resources
  - can be modeled as one cumulative resource with resource capacity corresponding to the number of alternative unary resources
    - suitable for symmetric unary resources
- Example: any of the persons can process set of tasks
Optimization

Various criteria and objective function

Common criteria: makespan

- completion time of the last activity
- modeling
  - introduced a new additional activity L, $p(L)=0$
  - added precedence constraint for each activity $T$ with no successor: $T \ll L$
Various criteria and objective function

Common criteria: *makespan*

- completion time of the last activity
- modeling
  - introduced a new additional activity $L$, $p(L)=0$
  - added precedence constraint for each activity $T$ with no successor: $T \prec L$
Various criteria and objective function

Common criteria: **makespan**

- completion time of the last activity
- modeling
  - introduced a new additional activity L, $p(L)=0$
  - added precedence constraint for each activity T with no successor: $T << L$

- makespan = start(L)
Problem: Basic Class Timetabling

Create a schedule for one hour classes with given earliest and latest starting time. All classes are taught in one classroom. Physical training and drawing should be taught as latest as possible.

<table>
<thead>
<tr>
<th>Class</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drawing</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Physical T.</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Chemistry</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Math</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Biology</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Computers</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>
CSP Model: Basic Class Timetabling

- **Variables:**
  - for start time of each class: Drawing, PhysicalT, Chemistry, ...

- **Domain:**
  - earliest and latest starting time
    - Drawing={3...6}, PhysicalT={3...4}, Chemistry={2...5}
    - Math={2...4}, Biology={3...4}, Computers={1...6}

- **Classroom = unary resource**
  - with all start times for classes
  - with unit processing time

  or better with: all-different constraint over start time variables

- **Optimization:** maximize (Drawing+PhysicalT)

- **Solutions:**
  - Drawing=6, PhysicalT=3, Chemistry=5, Math=2, Biology=4, Computers=1
  - Drawing=6, PhysicalT=4, Chemistry=5, Math=2, Biology=3, Computers=1

- **Optimal solution:**
  - Drawing=6, PhysicalT=4, Chemistry=5, Math=2, Biology=3, Computers=1
Some topics are described in


- Roman Barták: Filtering Techniques in Planning and Scheduling, ICAPS 2006, June 6-10, 2006, Cumbria, England
  http://www.plg.inf.uc3m.es/icaps06/preprints/i06-tu2-allpapers.pdf