The SoftCumulative Constraint with Quadratic Penalty

Yanick Ouellet  Claude-Guy Quimper

Department of Computer Science and Software Engineering
Université Laval, Quebec, QC, Canada

Abstract

The cumulative constraint greatly contributes to the success of constraint programming in solving scheduling problems. SoftCumulative, a version of the cumulative constraint where overloading the resource incurs a penalty, is, however, less studied. We introduce a checker and adapting algorithm for SoftCumulative, which are inspired by the energetic reasoning rule for the cumulative. Both algorithms can be used with a classic linear penalty function, but also with a quadratic penalty function, where the penalty of overloading the resource increases quadratically with the amount of the overload. We show that these algorithms are more general than existing algorithms and outperform a decomposition of SoftCumulative in practice.

Motivation

Tasks
• 5 surgeries of 4 hours each to complete in the day
• Possible to call additional surgeons if required
• We want to call as few additional surgeons as possible

Task definition

<table>
<thead>
<tr>
<th>Task</th>
<th>ΔS</th>
<th>𝐬(1)</th>
<th>𝐬(2)</th>
<th>𝐬(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Exemple

Linear cost: $Z \geq \sum_{i=1}^{n} p_i = 2 + 1 + 1 = 4$

Quadratic cost: $Z \geq \sum_{i=1}^{n} p_i^2 = 4 + 1 + 1 = 6$

Objective function
Minimize the overcost $Z$

SoftCumulative constraint

SoftCumulative($S, p, C, f(x), Z$) def \[ Z \geq \sum_{t \in T} f(\max(0, Z S_t + p_i - C)) \]

Additional parameters
• $f(x)$: Cost function
• $Z$: Overcost variable

Example

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z$</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

Minimum intersection

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Energetic reasoning

• $M(1, x) = \min(Z(1, x); R(1, x))$

Experiments

- Based on classical RCPSP instances
- Reduced resource capacities to force overload
- Comparison against a decomposition of the SoftCumulative

Benchmark

• Graph with time points as nodes
• Each edge represents an interval
• We find the longest path

Notation

<table>
<thead>
<tr>
<th>Task</th>
<th>$p$</th>
<th>$c$</th>
<th>$h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Task 2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Task 3</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Task 4</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Task 5</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Experiments with linear cost function

Experiments with quadratic cost function

Energetic reasoning

$M(1, x) = \min(Z(1, x); R(1, x))$

Figures

1. Motivation
2. Time-Tabling and Edge-Finding [De Clerc et al. 2010]
3. Minimum intersection
4. SoftCumulative constraint
5. Example
6. Our contribution

Figures

1. Motivation
2. Time-Tabling and Edge-Finding [De Clerc et al. 2010]
3. Minimum intersection
4. SoftCumulative constraint
5. Example
6. Our contribution