Temporal Planning with Preferences and Time-Dependent Continuous Costs

J. Benton  
j.benton@asu.edu  
Arizona State University

Amanda Coles  
amanda.coles@kcl.ac.uk  
King’s College London

Andrew Coles  
andrew.coles@kcl.ac.uk
Makespan != Plan Utility

The Dilemma of the Perishable Food

<table>
<thead>
<tr>
<th>Plan</th>
<th>Makespan</th>
<th>Time-on-shelf</th>
</tr>
</thead>
<tbody>
<tr>
<td>α→β→γ</td>
<td>15</td>
<td>13 + 0 + 0 = 13</td>
</tr>
<tr>
<td>β→γ→α</td>
<td>16</td>
<td>4 + 6 + 4 = 14</td>
</tr>
</tbody>
</table>

Apples last ~20 days
Oranges last ~15 days
Blueberries last ~10 days
Satisficing Temporal Planning

Optimization Metrics

- Monotonic Continuous Cost
- Discrete Cost (Preferences)
- Shortest Makespan
- Any Feasible

Problem Dynamics

- Temporally Simple
- Temporally Expressive

IPC 2002/2004
IPC 2006
IPC 2006/2008/2011
IPC 2011
Satisficing Temporal Planning

Optimizing Preference and Time-dependent Costs based on POPF2

Optimization Metrics
- Monotonic Continuous Cost
- Discrete Cost (Preferences)
- Shortest Makespan
- Any Feasible

Optimization Metrics
- Temporally Simple
- Temporally Expressive

Problem Dynamics

OPTIC

OPTIC

OPTIC

OPTIC

OPTIC

OPTIC
Outline

- Handling Discrete Temporal Costs (PDDL3 Preferences)
- Handling Continuous Costs
- Conclusion
Representation Challenges

- PDDL 3 -- established language from the IPC 2006
  - Offers discrete-cost using linear temporal logic subset

```prolog
(preference p1 (within 45 (at package1 loc1)))
(preference p2 (sometime-after (at package1 loc1)
                              (at package2 loc1)))

(:metric minimize (+ (* (is-violated p1) 10)
                   (* (is-violated p2) 5)))
```
Automata to Represent PDDL3 Preferences

- As in HPlanP, Mips-XXL, LPRPG-P: encode preferences as **automata**
- Update positions as actions are applied

![Automata Diagram]

- Clear interpretation in sequential plans; in temporal plans...?
sometime (and (P) (Q))

Simple Temporal Network in each node:
Built as we add actions
Scheduling Challenge

- Challenge: Efficiently handle temporal preferences
  - e.g., ending A before B

How do we avoid backtracking on this scheduling choice?
Solution: Build a mixed integer program (MIP) for the simple temporal network + dummy steps

- For each dummy step, add “soft constraints” in the MIP
- Each preference gets a [0,1] integer variable
- From these variables and constraints, determine the status of the preference. **Objective: minimize cost.**
sometime (and (P) (Q))

v, w, x, s in [0, 1], N is a large constant

(v + w + x > 0) ⇔ (s = 1)
PDDL3 Results

- Compared to existing Temporal Preference Planners: MIPS-XXL, SGPlan
  - Also SGPlan-W: SGPlan with a dummy never-applicable action added to the domain
- PDDL3 domains from IPC 2006: Pipesworld, Storage, Trucks, TPP, Pathways
- 4GB memory limit, 30 minutes CPU
**OPTIC/SGPlan have similar performance**
- OPTIC struggles on p11, p13

(Only domain where SGPlan-W is effective)
## Storage, Trucks

<table>
<thead>
<tr>
<th>Problem</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>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OPTIC</strong></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>13</td>
<td>127</td>
<td>208</td>
<td>397</td>
<td>477</td>
<td>737</td>
<td>914</td>
</tr>
<tr>
<td><strong>MIPSXXL</strong></td>
<td>3</td>
<td>4</td>
<td>38</td>
<td>81</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>SGPLAN</strong></td>
<td>18</td>
<td>37</td>
<td>86</td>
<td>113</td>
<td>227</td>
<td>322</td>
<td>505</td>
<td>591</td>
<td>887</td>
<td>1082</td>
</tr>
<tr>
<td><strong>SGPLAN-W</strong></td>
<td>13</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OPTIC</strong></td>
<td>1334</td>
<td>1472</td>
<td>2063</td>
<td>2366</td>
<td>3206</td>
<td>3383</td>
</tr>
<tr>
<td><strong>MIPSXXL</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>SGPLAN</strong></td>
<td>1532</td>
<td>1670</td>
<td>2248</td>
<td>2622</td>
<td>3290</td>
<td>3571</td>
</tr>
<tr>
<td><strong>SGPLAN-W</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem</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>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OPTIC</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td><strong>MIPSXXL</strong></td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>SGPLAN</strong></td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td><strong>SGPLAN-W</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OPTIC</strong></td>
<td>13</td>
<td>0</td>
<td>6</td>
<td>21</td>
<td>0</td>
<td>36</td>
<td>54</td>
<td>45</td>
<td>19</td>
<td>73</td>
</tr>
<tr>
<td><strong>MIPSXXL</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>SGPLAN</strong></td>
<td>8</td>
<td>6</td>
<td>12</td>
<td>11</td>
<td>8</td>
<td>23</td>
<td>50</td>
<td>29</td>
<td>32</td>
<td>46</td>
</tr>
<tr>
<td><strong>SGPLAN-W</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
## TPP, Pathways

<table>
<thead>
<tr>
<th>Problem</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>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTIC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>14</td>
<td>22</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MIPSXXL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SGPLAN</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>16</td>
<td>7</td>
<td>29</td>
<td>42</td>
</tr>
<tr>
<td>SGPLAN-W</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTIC</td>
<td>-</td>
<td>26</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MIPSXXL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SGPLAN</td>
<td>41</td>
<td>40</td>
<td>50</td>
<td>48</td>
<td>36</td>
<td>55</td>
<td>70</td>
<td>65</td>
<td>87</td>
<td>79</td>
</tr>
<tr>
<td>SGPLAN-W</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

OPTIC has difficulty finding its first solution to p9 onwards

<table>
<thead>
<tr>
<th>Problem</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>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTIC</td>
<td>3.0</td>
<td>13.6</td>
<td><strong>10</strong></td>
<td>18</td>
<td><strong>23.442</strong></td>
<td>20</td>
<td><strong>27.3</strong></td>
<td>33</td>
<td>29</td>
<td>33.9</td>
</tr>
<tr>
<td>MIPSXXL</td>
<td>2</td>
<td>8.2</td>
<td>11</td>
<td><strong>16.6</strong></td>
<td>25</td>
<td>21</td>
<td><strong>28.3</strong></td>
<td>34</td>
<td>30</td>
<td>34.9</td>
</tr>
<tr>
<td>SGPLAN</td>
<td>6.0</td>
<td>12.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SGPLAN-W</td>
<td>6.0</td>
<td>12.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Consistently the best performer from p5 upwards
Continuous Case

The Dilemma of the Perishable Food

Apples last ~20 days
Oranges last ~15 days
Blueberries last ~10 days
Solving for the Continuous Case

- Handling continuous costs
  - Directly model continuous costs
  - Compile into discretized cost functions (PDDL3 preferences)
Handling Continuous Costs

Model passing time as a PDDL+ process

Use “Collect Cost” Action for Goal

Conditional effects
- \( \text{precondition} \): \( t_g < d : 0 \)
- \( \text{effect} \):
  - \( d < t_g < d + c : f(t,g) \)
  - \( t_g \geq d + c : \text{cost}(g) \)

New goal
\( \text{collected_at(apples, } \alpha) \)
“Anytime” Search Procedure

- Enforced hill-climbing search for an incumbent solution $P$

- Restart using best-first branch-and-bound:
  - Prune using cost($P$)
    - Use admissible heuristic for pruning
Compile to Discretized Cost

\[
\text{Cost} = d + c
\]

\[
f(t, g)
\]

\[
\text{cost}(g)
\]

0 \quad d \quad d + c

Time
Discretized Compilation

Cost

\( f_1(t, g) \)

\[ \text{Time} \]

\( d_1 \)

Cost

\( f_2(t, g) \)

\[ \text{Time} \]

\( d_2 \)

Cost

\( f_3(t, g) \)

\[ \text{Time} \]

\( d_3 \)
Final Discretized Compilation

\[ fd(t,g) = f_1(t,g) + f_2(t,g) + f_3(t,g) \]

What’s the best granularity?
The Secret Discretization Advantage

With an admissible heuristic we can do this early enough to reduce the search effort
The Secret Discretization Advantage

You’ll miss this better plan

The cost function!
The Secret Discretization Advantage

Cost

cost(g)

but find this one

fd(t,g)

0
d1
d2
d3
d1 + c

Time
Continuous vs. Discretization

The Contenders

- **Continuous Advantage**
  - More accurate solutions
  - Represents actual cost functions

- **Discretized Advantage**
  - “Faster” search
  - Looks for bigger jumps in quality
Continuous + Discrete-Mimicking Pruning

Tiered Search

- Continuous Representation
  - More accurate solutions
  - Represents actual cost functions

- Mimicking Discrete Pruning
  - “Faster” search
  - Looks for bigger jumps in quality
Tiered Approach

Cost: $d + c$

Cost ($g$)

$f(t,g)$

Cost: 128 (sol)
Tiered Approach

Sequential pruning bounds where we **heuristically prune** from the cost of the best plan so far.

\[ \text{Cost}(s_1): 128 \text{ (sol)} \]
\[ \text{Prune} \geq \text{sol} - s_1/2 \]
Tiered Approach

Cost $d + c$

cost(g)
f(t,g)

Time

Sequential pruning bounds where we **heuristically prune** from the cost of the best plan so far

Cost(s₁): 128 (sol)
Prune $\geq$ sol $- s₁/4$
Tiered Approach

Sequential pruning bounds where we **heuristically prune** from the cost of the best plan so far.

Cost

Cost\(\text{(g)}\)

\(f(t, g)\)

\(d\)

\(d + c\)

heuristically prune

solution value

Cost\((s_1)\): 128 (sol)

Prune \(\geq \text{sol} - s_1/8\)
Tiered Approach

Cost: $d + c$

Cost($g$): $f(t,g)$

Sequential pruning bounds where we **heuristically prune** from the cost of the best plan so far.
Tiered Approach

Sequential pruning bounds where we **heuristically prune** from the cost of the best plan so far

Cost: $d + c$

Cost($s_1$): 128 (sol)

Prune $\geq$ sol
Continuous Cost Results

- **Comparison:**
  - Continuous, discretized, tiered search

- **Generated three continuous cost domains**
  - Elevators: Soft deadlines where each person is solved for independently (w/ random delay)
  - Openstacks: Soft deadlines based on production durations
  - Crew Planning: Soft deadlines on each crew member waking up and payloads

- **4GB memory limit, 30 minutes CPU**
Time-dependent Cost Results

Elevators

Score (IPC 2008 Metric)

Problem Number

Minimize Makespan
Split into 10
Split into 5
Split into 3
Continuous
Continuous, tiered
Time-dependent Cost Results

![Graph showing crew planning performance over problem number]
Time-dependent Cost Results

Openstacks

Score (IPC 2008 Metric)

Problem Number

Minimize Makespan
Split into 10
Split into 5
Split into 3
Continuous
Continuous, tiered

36
Summary

• State-of-the-art planning for PDDL3 preferences + temporal planning

• Handles (monotonic) continuous cost functions on goal achievement

• Efficient search technique for handling monotonic continuous cost functions
Future Work

• Handle continuous costs over time windows (analogous to “always-within”)

• Test search approach on other planning problems (e.g., classical cost-based planning)
Thanks!

**OPTIC**


- Built on the code of POPF2
- Temporally expressive problems
- Problems with continuous numeric effects
- PDDL3 preferences
- Net-benefit planning
- Continuous time-dependent costs
- (ADL: Only in preferences)

Enjoy the match later!
Cost trade-off example

(preference S (sometime (and (P) (Q))))
(preference T (within 15 (R)))
Representation Challenges

- Continuous costs
  - We can express these using a subset of the language PDDL+
  - Should we directly represent continuous costs or discretize them?

![Diagram showing cost and achievement time with functions f(t,g) and fd(t,g)]