On the Discovery and Utility of Precedence Constraints in Temporal Planning

Yanmei Hu¹, Minghao Yin¹*, Dunbo Cai²

¹School of Computer Science and Information Technology, Northeast Normal University, Changchun, P. R. China, 130117
²School of Computer Science and Engineering, Wuhan Institute of Technology, Changchun, P. R. China, 430022
huym260@gmail.com, ymh@nenu.edu.cn, dunbocai@gmail.com

Abstract

Temporal planning considers temporal dependencies and numeric resources. As a further step towards real-world applications, it has been attracting many attentions and triggered some great work. Recently, most of those works are based on heuristic search, e.g., in the work of Sapa (Do and Kambhampati 2003), LPG (Gerevini et al. 2008), SGP (Chih-wei et al. 2008) and TFD (Eyerich et al. 2009). In this line, the design of reasonable and informative heuristics shows a great progress, especially in the planner TFD.

TFD proposed an extension of the SAS+ formulism, which is called temporal numeric SAS+. The formulism basically has two advantages: leading to a smaller search space in comparison with the STRIPS representation, and building a convenient base for designing causal graph (Helmer 2006) based heuristics. Specifically, TFD proposed an extension of the heuristic $h^{co}$ (Helmer and Geffner 2008), which is potentially more informative than the causal graph heuristic $h^{cg}$ (Helmer 2006) and can lead to plans of high quality. Noted that Cai et al. (2009) proposed an enhancement of $h^{co}$, which is called $h^{rec}$. $h^{rec}$ considers precedence constraints over both logical fluents and derived comparison variables. Note that Eyerich et al., (2009) only considers dependencies among fluents and the methods of Cai et al. (2009) only suit for fluents. However, in our setting, there may be comparison variables in the precondition of an action. For example, to model $w = (n_0 - h_0) \geq 0$, Eyerich et al. (2009) introduces a new boolean comparison variable $v_c$, where $v_c$ is true iff. $w = (n_0 - h_0) \geq 0$ holds. To handle comparison variables, we proposed methods for accounting precedence constraints over both fluents and comparison variables.

Introduction

Temporal planning considers temporal dependencies and numeric resources. As a further step towards real-world applications, it has been attracting many attentions and triggered some great work. Recently, most of those works are based on heuristic search, e.g., in the work of Sapa (Do and Kambhampati 2003), LPG (Gerevini et al. 2008), SGP (Chih-wei et al. 2008) and TFD (Eyerich et al. 2009). In this line, the design of reasonable and informative heuristics shows a great progress, especially in the planner TFD.

TFD proposed an extension of the SAS+ formulism, which is called temporal numeric SAS+. The formulism basically has two advantages: leading to a smaller search space in comparison with the STRIPS representation, and building a convenient base for designing causal graph (Helmer 2006) based heuristics. Specifically, TFD proposed an extension of the heuristic $h^{co}$ (Helmer and Geffner 2008), which is potentially more informative than the causal graph heuristic $h^{cg}$ (Helmer 2006) and can lead to plans of high quality. Noted that Cai et al. (2009) proposed an extension of $h^{co}$, which is called $h^{rec}$. $h^{rec}$ considers precedence constraints over both logical fluents and derived comparison variables. Note that Eyerich et al., (2009) only considers dependencies among fluents and the methods of Cai et al. (2009) only suit for fluents. However, in our setting, there may be comparison variables in the precondition of an action. For example, to model $w = (n_0 - h_0) \geq 0$, Eyerich et al. (2009) introduces a new boolean comparison variable $v_c$, where $v_c$ is true iff. $w = (n_0 - h_0) \geq 0$ holds. To handle comparison variables, we proposed methods for accounting precedence constraints over both fluents and comparison variables.

Temporal Heuristic with Precedence Constraints

We follow the notation and definition of Eyerich et al. (2009) and Cai et al. (2009). For our purpose, a rule $r$ corresponding to an instant operator (io), (Eyerich et al. 2009) has the form $r: Z_r \rightarrow x_r$ (or $r: Z_r \rightarrow \mathcal{G}_r$) with cost($r$)
\( h^{\text{pcc}}(x|s) = \begin{cases} 0 & \text{if } x \in s \\ \min_{e' \in Z_r \rightarrow \var(x)} \{ c(s') + \sum_{y \in Z_r} h^{\text{pcc}}(y|e^r(y,r',s')) \} & \text{if } x \notin s, \text{ var}(x) \notin V_e \\ \min_{e \in \text{pred}(x,s)} \{ c(x) + \sum_{y \in Z_r} h^{\text{pcc}}(y|e^r(y,r,s)) \} & \text{if } x \notin s, \text{ var}(x) \in V_e \\ \end{cases} \) 

In Eq. (1), we extend \( h^{\text{pcc}} \) (Cai et al. 2009) into a temporal and numeric setting using instant actions (Eyerich et al. 2009), which results the heuristic function \( h^{\text{pcc}}(x|s) \) is the estimated cost of reaching an atom \( x \) from a state \( s \). In Eq. (1), \( s \) is the state corresponding to the rule \( r \), \( s' \) is the state corresponding to the rule \( r' \), and \( e^r(y, r, s) \) is the context state associated with the condition \( y \) of \( r \), with respect to \( s \), which is computed like the equations (5) and (6) in the paper of Cai et al. (2009).

We design a context function (Cai et al. 2009) \( \text{ctx} \) to account the precedence constraints over preconditions of a rule \( r \), where \( \text{ctx}(r, q) = p \) indicates that the context of \( q \in Z_r \) should be the state that results from achieving \( p \in Z_r \). To obtain precedence constraints, we build the following rules:

**Rule 1** For \( p, q \in Z_r \), if \( p \) and \( q \) are landmarks, there are orderings \( q \rightarrow p \), \( q \rightarrow q.p \) or \( q \rightarrow q.m.p \), then \( \text{ctx}(p, r) = q \).

**Rule 2** For \( p, q \in Z_r \), if \( p \) is a comparison atom and \( q \) is a logical atom, \( \exists q' \in \text{pred}(p, s) \wedge \exists q' \in Z_r \wedge \text{var}(x) = \text{var}(q) \), then \( \text{ctx}(q, r) = p \).

Note that **Rule 2** is for accounting the precedence among comparison variables and logical variables. To consider the reasonability of **Rule 2**, we may think cases where the value change of \( p \) involves the value change of \( q \).

**Experimental Results and Conclusions**

To evaluate \( h^{\text{pcc}} \), we implement it on top of the code of Fast Downward (Helmert 2006) and LAMA (Richter et al. 2008), and test it on 12 benchmarks used in the temporal satisficing track of IPC 2008, with \( h^{\text{cea}} \) as a reference. All experiments are done on a PC with a 2.4GHz CPU and 3GB memory. The limit on time is 30 minutes and on memory is 2GB. Table 1 shows, for each heuristic, the number of solved problems on each domain. Table 2 compares the plan quality resulted from using \( h^{\text{pcc}} \) and that resulted from using \( h^{\text{cea}} \) on problems in each domain. In Tab. 2, \( +n/-m \) indicates that \( h^{\text{pcc}} \) results better plans on \( n \) problems while results worse plans on \( m \) problems when compared with \( h^{\text{cea}} \).

From Tab. 1, we can see \( h^{\text{pcc}} \) is worse than \( h^{\text{cea}} \) totally, which is mostly due to our currently very rough implementation. From Tab. 2, we can see that \( h^{\text{pcc}} \) leads to better plans than \( h^{\text{cea}} \) dose on 6 domains. Therefore, we consider \( h^{\text{pcc}} \) as a promising heuristic and will improve our implementation in the future work.

**Table 1: Number of instances solved with \( h^{\text{cea}} \) and \( h^{\text{pcc}} \).**

<table>
<thead>
<tr>
<th>Domain</th>
<th>( h^{\text{pcc}} )</th>
<th>( h^{\text{cea}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevators-numeric</td>
<td>29/23</td>
<td>29/23</td>
</tr>
<tr>
<td>Elevators-strips</td>
<td>27/23</td>
<td>27/23</td>
</tr>
<tr>
<td>Crewplanning-strips</td>
<td>6/29</td>
<td>6/29</td>
</tr>
<tr>
<td>Openstacks-adl</td>
<td>30/30</td>
<td>30/30</td>
</tr>
<tr>
<td>Openstacks-numadic</td>
<td>30/30</td>
<td>30/30</td>
</tr>
<tr>
<td>Openstacks-strips</td>
<td>30/30</td>
<td>30/30</td>
</tr>
<tr>
<td>Parcprinter-strips</td>
<td>13/3</td>
<td>13/3</td>
</tr>
<tr>
<td>Pegsol-strips</td>
<td>29/30</td>
<td>29/30</td>
</tr>
<tr>
<td>Sokoban-strips</td>
<td>14/7</td>
<td>14/7</td>
</tr>
<tr>
<td>Transport-numeric</td>
<td>11/11</td>
<td>11/11</td>
</tr>
<tr>
<td>Woodwork-num</td>
<td>29/27</td>
<td>29/27</td>
</tr>
<tr>
<td>Total</td>
<td>291/259</td>
<td>291/259</td>
</tr>
</tbody>
</table>

**Table 2: Plan quality comparison.**

<table>
<thead>
<tr>
<th>Domain</th>
<th>( h^{\text{pcc}} ) vs. ( h^{\text{cea}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elevators-numeric</td>
<td>+1/-7</td>
</tr>
<tr>
<td>Elevators-strips</td>
<td>+7/-21</td>
</tr>
<tr>
<td>Crewplanning-strips</td>
<td>+0/-23</td>
</tr>
<tr>
<td>Openstacks-adl</td>
<td>+3/-7</td>
</tr>
<tr>
<td>Openstacks-numadic</td>
<td>+7/-0</td>
</tr>
<tr>
<td>Openstacks-strips</td>
<td>+10/-10</td>
</tr>
<tr>
<td>Parcprinter-strips</td>
<td>+1/-0</td>
</tr>
<tr>
<td>Pegsol-strips</td>
<td>+7/-4</td>
</tr>
<tr>
<td>Sokoban-strips</td>
<td>+1/-8</td>
</tr>
<tr>
<td>Transport-numeric</td>
<td>+5/-4</td>
</tr>
<tr>
<td>Woodwork-num</td>
<td>+8/-6</td>
</tr>
</tbody>
</table>

**References**


