Finding and Exploiting LTL Trajectory Constraints in Heuristic Search

Salomé Simon  Gabriele Röger

University of Basel, Switzerland

SoCS 2015
Motivation

Goal

framework for describing information about the search space

• combining information from different sources
  \(\rightarrow\) creating stronger heuristics

• decoupling the derivation and exploitation of information
  \(\rightarrow\) split work between different experts
Linear Temporal Logic on Finite Traces (LTLₙ): evaluated over a linear sequence of worlds (= variable assignments) and extends propositional logic with:

- **Always** ($\square \varphi$): $w_0$ always holds through $w_n$.
- **Eventually** ($\diamond \varphi$): $\varphi$ holds at some point in $w_0$ through $w_n$.
- **Next** ($\circ \varphi$): $\varphi$ holds in the next world $w_1$.
- **Until** ($\varphi U \psi$): $\varphi$ holds through $w_i$ and $\psi$ holds from $w_{i+1}$ through $w_n$.
- **Release** ($\varphi R \psi$): $\psi$ holds through $w_i$ and $\psi \land \varphi$ holds from $w_{i+1}$ through $w_n$.
- **Last world** ($last$): $w_0$ through $w_n$ with $w_n$ as the last world.
**LTL\textsubscript{f} Formulas in the Search Space**

<table>
<thead>
<tr>
<th>variable</th>
<th>\iff</th>
<th>state variable or action</th>
</tr>
</thead>
<tbody>
<tr>
<td>world</td>
<td>\iff</td>
<td>node in search space (with incoming action)</td>
</tr>
<tr>
<td>world sequence</td>
<td>\iff</td>
<td>path to a goal node</td>
</tr>
</tbody>
</table>

LTL\textsubscript{f} formulas associated to nodes

\[ \varphi \rightarrow \text{express conditions all optimal paths to a goal need to fulfill} \]
Motivation

LTL<sub>f</sub> in Heuristic Search

Finding Information

Exploiting Information

Experiments

Conclusion

Feasibility for Nodes

Definition (Feasibility for nodes)

An LTL<sub>f</sub> formula φ is **feasible for n** if for all paths ρ such that

- ρ is applicable in n,
- the application of ρ leads to a goal state, and
- \( g(n) + c(\rho) = h^* \)

it holds that \( w^s_\rho \models \phi \).

(where \( w^s_\rho = \langle \{a_1\} \cup s[a_1], \{a_2\} \cup s[\langle a_1, a_2 \rangle], \ldots, \{a_n\} \cup s[\rho], s[\rho] \rangle \))
Adding and Propagating Information during the Search

1. new information during the search
directly added to the corresponding node with conjunction
Adding and Propagating Information during the Search

- new information during the search
directly added to the corresponding node with conjunction
new information during the search

directly added to the corresponding node with conjunction

Example

\[ a \mathcal{U} b \land \Diamond d \land \Box \neg c \land g=5 \]

\[ g=4 \]

\[ a, c \]

\[ a, d \]

\[ a, b \]
formulas can be propagated with progression to successor nodes

Theorem

Let $\varphi$ be a feasible formula for a node $n$, and let $n'$ be the successor node reached from $n$ with action $a$. Then $\text{progress}(\varphi, \{a\} \cup s(n'))$ is feasible for $n'$. 
Formulas can be propagated with progression to successor nodes.

**Theorem**

Let $\varphi$ be a feasible formula for a node $n$, and let $n'$ be the successor node reached from $n$ with action $a$. Then $\text{progress}(\varphi, \{a\} \cup s(n'))$ is feasible for $n'$.

**Example**

\[
\begin{align*}
\& a \mathcal{U} b \land \lozenge d \land \Box \neg c \\
& g = 4 \quad a \\
& g = 5 \quad a, c \\
& a, d \\
& a, b
\end{align*}
\]
Adding and Propagating Information during the Search

Theorem

Let $\varphi$ be a feasible formula for a node $n$, and let $n'$ be the successor node reached from $n$ with action $a$. Then $\text{progress}(\varphi, \{a\} \cup s(n'))$ is feasible for $n'$.

Example

\[ a \mathcal{U} b \land \Diamond d \land \Box \neg c \quad g=5 \]
\[ a \mathcal{U} d \]
\[ g=4 \quad a \]
\[ a, c \]
\[ a, d \]
\[ a, b \]
\[ a \mathcal{U} b \land \Box \neg c \]
Adding and Propagating Information during the Search

**Theorem**

Let $n$ and $n'$ be two search nodes such that $g(n) = g(n')$ and $s(n) = s(n')$. Let further $\varphi_n$ and $\varphi_{n'}$ be feasible for the respective node. Then $\varphi_n \land \varphi_{n'}$ is feasible for both $n$ and $n'$.

**duplicate elimination:** conjunction of formulas of “cheapest” nodes is feasible
duplicate elimination: conjunction of formulas of “cheapest” nodes is feasible

**Theorem**

Let $n$ and $n'$ be two search nodes such that $g(n) = g(n')$ and $s(n) = s(n')$. Let further $\varphi_n$ and $\varphi_{n'}$ be feasible for the respective node. Then $\varphi_n \land \varphi_{n'}$ is feasible for both $n$ and $n'$.

**Example**

$\af Ub \land \af d \land \af c$  \hspace{1cm} $g=5$

$\af Ub \land \af c$  \hspace{1cm} $g=4$
duplicate elimination: conjunction of formulas of “cheapest” nodes is feasible

**Theorem**

Let \( n \) and \( n' \) be two search nodes such that \( g(n) = g(n') \) and \( s(n) = s(n') \). Let further \( \varphi_n \) and \( \varphi_{n'} \) be feasible for the respective node. Then \( \varphi_n \land \varphi_{n'} \) is feasible for both \( n \) and \( n' \).

**Example**

\[
\begin{align*}
a \mathcal{U} b \land \Box d \land \Box \neg c & \quad g=5 \\
g=4 & \quad a \\
a \mathcal{U} d & \quad a, c \\
\end{align*}
\]
duplicate elimination: conjunction of formulas of “cheapest” nodes is feasible

**Theorem**

Let $n$ and $n'$ be two search nodes such that $g(n) = g(n')$ and $s(n) = s(n')$. Let further $\varphi_n$ and $\varphi_{n'}$ be feasible for the respective node. Then $\varphi_n \land \varphi_{n'}$ is feasible for both $n$ and $n'$.

**Example**

\[ a \mathcal{U} b \land \Diamond d \land \Box \neg c \quad g=5 \]
\[ a, c \]
\[ g=4 \]
\[ a \]
\[ a \mathcal{U} d \]
\[ a, d \]
\[ a \mathcal{U} b \land \Box \neg c \]
\[ a, b \]
Motivation

LTL$_f$ in Heuristic Search

Finding Information

Exploiting Information

Experiments

Conclusion

Adding and Propagating Information during the Search

**8** duplicate elimination: conjunction of formulas of “cheapest” nodes is feasible

### Theorem

Let $n$ and $n'$ be two search nodes such that $g(n) = g(n')$ and $s(n) = s(n')$. Let further $\varphi_n$ and $\varphi_{n'}$ be feasible for the respective node. Then $\varphi_n \land \varphi_{n'}$ is feasible for both $n$ and $n'$.

### Example

\[ a \mathcal{U} b \land \Box d \land \Box \neg c \quad g=5 \]

\[ a, c \]

\[ a \mathcal{U} d \]

\[ a, d \]

\[ a \mathcal{U} b \land \Box \neg c \]

\[ a \]

\[ g=4 \]

\[ a, b \]

\[ a \mathcal{U} d \land \Box \neg c \]
Encoding Information in LTL$_f$ Formulas

Possible sources of information:

- domain-specific knowledge
- temporally extended goals
- here: information used in specialized heuristics
  - Landmarks and their orderings (Hoffmann et al. 2004, Richter et al. 2008)
  - Unjustified Action Applications (Karpas and Domshlak 2012)
**Fact Landmark**: A fact that must be true at some point in every solution (Hoffmann et al. 2004)

→ In LTL\textsubscript{f}: $\lozenge l$

Further information about landmarks:

- **First achievers**: $l \lor \bigvee_{a \in FA_l} \lozenge a$
- **Required again**: $(\lozenge l) \mathcal{U} l'$
- **Goal**: $\bigwedge_{g \in G} ((\lozenge g) \mathcal{U} \bigwedge_{g' \in G} g')$
Unjustified Action Applications

If an action is applied, its effects must be of some use (Karpas and Domshlak 2012)

1. one of its effects is necessary for applying another action
2. one of its effects is a goal variable (that is not made false again)

\[
\varphi_a = \bigvee_{e \in \text{add}(a) \setminus \text{G}} ((e \land \bigwedge_{a' \in A \text{ with } e \in \text{add}(a')} \neg a') \cup \bigvee_{a' \in A \text{ with } e \in \text{pre}(a')} a') \\
\bigvee_{e \in \text{add}(a) \cap \text{G}} ((e \land \bigwedge_{a' \in A \text{ with } e \in \text{add}(a')} \neg a') \cup (\text{last} \lor \bigvee_{a' \in A \text{ with } e \in \text{pre}(a')} a'))
\]
**Heuristics**

- Very rich temporal information possible
  - heuristics can use different levels of relaxation
- Proof of concept heuristic extracts landmarks from node-associated formulas
  - loses temporal information between landmarks
Experiment Setup

Configurations:

1. $h_{LA}$: standard admissible landmark heuristic (Karpas and Domshlak 2009)
2. $h_{LM}^{AL}$: LTL landmark extraction heuristic with sources:
   - Landmarks (First achievers, Required again, Goal)
3. $h_{LM+UAA}^{AL}$: LTL landmark extraction heuristic with sources:
   - Landmarks (First achievers, Required again, Goal)
   - Unjustified Action Applications

- all heuristics use BJOLP landmark extraction and optimal cost partitioning
- search algorithm: $h_{LA}$ uses LM-A*, the others a slight variant we call LTL-A*
## Coverage

<table>
<thead>
<tr>
<th></th>
<th>$h_{LA}$</th>
<th>$h_{LM}$</th>
<th>$h_{LM+UAA}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>airport (50)</td>
<td>31</td>
<td>28</td>
<td>26</td>
</tr>
<tr>
<td>elevators-08 (30)</td>
<td>14</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>floortile (20)</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>freecell (80)</td>
<td>52</td>
<td>51</td>
<td>50</td>
</tr>
<tr>
<td>mprime (35)</td>
<td>19</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>nomystery (20)</td>
<td>18</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>openstacks-08 (30)</td>
<td>14</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>openstacks-11 (20)</td>
<td>9</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>parcprinter-08 (30)</td>
<td>15</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>parcprinter-11 (20)</td>
<td>11</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>pipesworld-tan (50)</td>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>scanalyzer-08 (30)</td>
<td>10</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>sokoban-08 (30)</td>
<td>22</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>tidybot (20)</td>
<td>14</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>other domains (931)</td>
<td>483</td>
<td>483</td>
<td>483</td>
</tr>
<tr>
<td><strong>Sum (1396)</strong></td>
<td><strong>723</strong></td>
<td><strong>711</strong></td>
<td><strong>709</strong></td>
</tr>
</tbody>
</table>
$h_{LA}$ looses no task due to memory limit, but $h^{LM}_{AL}$ 11 in total

- airport: over 300% of memory usage compared to $h_{LA}$
- average: 120%
- approx. half the domains < 100%
Impact of Unjustified Action Applications

Comparison of expansions between $h_{LM}^{AL}$ and $h_{LM+UAA}^{AL}$.
associate nodes in the search space with LTL$_f$ formulas → conditions for optimal plan

separation between finding information and exploiting information

allows to easily combine information from different sources

concrete examples in this paper:
  - finding information: landmarks and unjustified action applications
  - exploiting information: extracting landmarks
Future Work

- better informed heuristics (less relaxation)
- encodings for other kinds of information
- strengthening other heuristics with the information of LTL\(_f\) trajectory constraints