Finding and Exploiting LTL Trajectory Constraints in Heuristic Search

Salomé Simon   Gabriele Röger

University of Basel, Switzerland

HSDIP 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 (LTL\(_f\))

- evaluated over a linear sequence of worlds
  (= variable assignments)
- extends propositional logic with:
  - □\(\varphi\) Always
  - ◊\(\varphi\) Eventually
  - ○\(\varphi\) Next
  - \(\varphi U \psi\) Until
  - \(\varphi R \psi\) Release
  - last Last world
Progression

What if we only know the beginning of the sequence?

Definition (Progression)

For an LTL$_f$ formula $\varphi$ and a world sequence $\langle w_0, \ldots, w_n \rangle$ with $n > 0$ it holds that $\langle w_1, \ldots, w_n \rangle \models \text{progress}(\varphi, w_0)$ iff $\langle w_0, \ldots, w_n \rangle \models \varphi$.

Example

progress$(a \land c \land e \land \Box(c \lor d) \land (b \, U \, d), \{a, d\}) = e \land \Box(c \lor d)$
### LTL$_f$ Formulas in the Search Space

<table>
<thead>
<tr>
<th>Variable</th>
<th>↔</th>
<th>STRIPS variable or action</th>
</tr>
</thead>
<tbody>
<tr>
<td>World</td>
<td>↔</td>
<td>node in search space (with incoming action)</td>
</tr>
<tr>
<td>World sequence</td>
<td>↔</td>
<td>path to a goal node</td>
</tr>
</tbody>
</table>

LTL$_f$ formulas associated to nodes

→ express conditions all optimal paths to a goal need to fulfill
Feasibility for Nodes

Definition (Feasibility for nodes)

An LTL$_f$ formula $\varphi$ is feasible for $n$ if for all paths $\rho$ such that
- $\rho$ is applicable in $n$,
- the application of $\rho$ leads to a goal state ($G \subseteq s[\rho]$), and
- $g(n) + c(\rho) = h^*$

it holds that $w_\rho^s \models \varphi$.

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

How can we add/propagate information while preserving feasibility?

1. **new information during the search**
   directly added to the corresponding node with conjunction

2. **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

\[
\text{progress}(\varphi, \{a\} \cup s(n')) \text{ is feasible for } n'.
\]
Adding and Propagating Information during the Search

How can we add/propagate information while preserving feasibility?

- **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

\[ \Diamond a \land (b \cup d) \land (c \lor e) \]
Example

\[ \Diamond a \land (b \cup d) \land (c \lor e) \]

\[ \Diamond a \land (b \cup d) \]

\[ b, c \]

\[ c, e \]

\[ d, c \]

\[ b, e \]

\[ d, e \]

\[ a, e \]
Example

\[ \Diamond a \land (b \mathcal{U} d) \land (c \lor e) \]

\[ \Diamond a \land (b \mathcal{U} d) \]

\[ b, c \]

\[ d, c \]

\[ c, e \]

\[ b, e \]

\[ d, e \]

\[ a, e \]

\[ \Diamond a \]

\[ \Box (\neg c) \]
Example

\[ \Diamond a \land (b \cup d) \land (c \lor e) \]

\[ \Diamond a \land (b \cup d) \]

\[ b, c \]

\[ \Diamond a \]

\[ c, e \]

\[ b \]

\[ b, e \]

\[ \Diamond a \land (b \cup d) \land \Box (\neg c) \]

\[ d, c \]

\[ d, e \]

\[ a, e \]
Example

\[ \Diamond a \land (b \cup d) \land (c \lor e) \]

\[ \Diamond a \land (b \cup d) \]

\[ \bot \]

\[ \Diamond a \land (b \cup d) \land \Box (\neg c) \]

\[ \Diamond a \]

\[ (c, e) \]

\[ (d, e) \]

\[ (a, e) \]

\[ (d, c) \]

\[ (b, c) \]

\[ (b, e) \]
Example

\[ \Diamond a \land (b \bigcup d) \land (c \lor e) \]

\[ \Diamond a \land (b \bigcup d) \quad b, c \]

\[ \Diamond a \quad c, e \]

\[ \perp \]

\[ \Diamond a \land (b \bigcup d) \quad b, e \]

\[ \Diamond a \quad d, c \]

\[ \Diamond a \land (b \bigcup d) \land \Box (\neg c) \]

\[ d, e \]

\[ a, e \]
Example:

\[\Diamond a \land (b \cup d) \land (c \lor e)\]

\[\Diamond a \land (b \cup d)\]

\[\perp\]

\[\Diamond a \land (b \cup d) \land \Box(\neg c)\]

\[\Box(\neg c)\]
Example

\[ \Diamond a \land (b \cup d) \land (c \lor e) \]

\[ \Diamond a \land (b \cup d) \]

\[ \perp \]

\[ \Diamond a \land \square (\neg c) \]

\[ \Diamond a \land \square (\neg c) \land (c \lor e) \]
Example

\[ \Diamond a \land (b \cup \neg d) \land (c \lor e) \]

\[ \Diamond a \land (b \cup d) \land (c \lor e) \]

\[ \Diamond a \land (b \cup d) \land \Box (\neg c) \]

\[ \Diamond a \land (b \cup d) \land \Box (\neg c) \]

\[ \perp \]
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)
Landmarks

**Fact Landmark:** A fact that must be true at some point in every plan (Hoffmann et al. 2004)
→ In $LTL_f$: $\Diamond l$

Further information about landmarks:

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

\[
\varphi_a = \bigvee_{e \in \text{add}(a) \setminus G} \left( (e \land \bigwedge_{a' \in A \text{ with } e \in \text{add}(a')} \neg a') \cup \bigvee_{a' \in A \text{ with } e \in \text{pre}(a')} a' \right) \\
\bigvee_{e \in \text{add}(a) \cap G} \left( (e \land \bigwedge_{a' \in A \text{ with } e \in \text{add}(a')} \neg a') \cup (\text{last} \lor \bigvee_{a' \in A \text{ with } e \in \text{pre}(a')} a') \right)
\]
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
Extracting Landmarks from the Formula

1. Convert formula into **positive normal form** ("¬" only before atoms)
   - can be computed efficiently
   - progression preserves positive normal form

2. Transform formula into implied formula where ♦ in front of every literal, no other temporal operators

3. Transform formula into **CNF**

4. Dismiss clauses which are **true already in current state**

5. Extract **disjunctive action landmarks** from individual clauses
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>$h_{LM}^{AL}$</th>
<th>$h_{LM+UAA}^{AL}$</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>
Memory Consumption

\( h_{LA} \) looses no task due to memory limit, but \( h_{LM} \) 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)
- describe other kinds of information
  - PDDL 3 trajectory constraints
  - flow-based heuristics (van den Briel et al. 2007; Bonet 2013; Pommerening et al. 2014)
  - mutex information
- strengthening other heuristics with the information of $\text{LTL}_f$ trajectory constraints