A Novel Lookahead Strategy for Delete Relaxation Heuristics in Greedy Best-First Search

Maximilian Fickert

ICAPS 2020

SAARLAND UNIVERSITY
SAARBRÜCKEN GRADUATE SCHOOL OF COMPUTER SCIENCE
Variables: at, fuel

Actions:

\[ \text{drive}(x, y): \]
- \( \text{pre} = \{\text{at} = x, \text{fuel} = 1\} \)
- \( \text{eff} = \{\text{at} = y, \text{fuel} = 0\} \)

\[ \text{refuel}: \]
- \( \text{pre} = \emptyset \)
- \( \text{eff} = \{\text{fuel} = 1\} \)

Initial State: \( \{\text{at} = A, \text{fuel} = 1\} \)

Goal: \( \{\text{at} = C\} \)
Variables accumulate values instead of switching between them

Relaxed plan: \( \text{drive}(A, B), \text{drive}(B, C) \)

The \( h^{FF} \) heuristic yields the length of a (non-optimal) relaxed plan
The relaxed plan provides more information than just a heuristic value:

- Preferred operators (Hoffmann 2001, Helmert 2006)
- Generate lookahead state using executable prefix (YAHSP; Vidal 2004, 2011)
- Relaxed subgoal counting (BFWS; Lipovetzky and Geffner, 2017)
Lazy GBFS with Relaxed Subgoal Lookahead (GBFS-RSL)

At each expansion of a state $s$:

1. Initialize relaxed subgoal counting heuristic $h^{rsc}$ with the relaxed plan for $s$
2. Perform a bounded lookahead search with $h^{rsc}$
3. Return the best state $s'$ according to $h^{rsc}$
4. Insert $s'$ at the front of the open list if $h^{FF}(s') < h^{FF}(s)$, otherwise discard it
GBFS-RSL in VisitAll

VisitAll (30x30)
GBFS-RSL in VisitAll

Expansion 1
Relaxed Plan
GBFS-RSL in VisitAll

Expansion 1
Lookahead Search Tree
Expansion 1
Lookahead Path
GBFS-RSL in VisitAll

Expansion 1
State after Lookahead
GBFS-RSL in VisitAll

Expansion 1
Relaxed Plan
Lookahead Search Tree
Expansion 1

State after Lookahead
Expansion 2
Lookahead Path

Expansion 2

Expansion 3
Lookahead Path

Expansion 3

Expansion 4
Lookahead Path

Expansion 4

Expansion 5
Lookahead Path

Expansion 5
Expansion 2
State after Lookahead
GBFS-RSL in VisitAll

Expansion 3
Lookahead Path
GBFS-RSL in VisitAll

Expansion 3
State after Lookahead
GBFS-RSL in VisitAll

Expansion 4
Lookahead Path
GBFS-RSL in VisitAll

Expansion 4
State after Lookahead
GBFS-RSL in VisitAll

Expansion 5
Lookahead Path
GBFS-RSL in VisitAll

Expansion 5
State after Lookahead
Partial Delete Relaxation

Take some delete information into account:

- Red-Black Planning (Katz and Hoffmann 2014, Domshlak et al. 2015) → un-relax fuel variable
- Explicit Conjunctions (Keyder et al. 2014, Fickert et al. 2016) → achieve fuel = 1 and at = B at the same time
Experiments

49 IPC domains, 30min timeout, 4GB memory
Lazy GBFS with dual-queue for preferred operators

<table>
<thead>
<tr>
<th>Lookahead</th>
<th>–</th>
<th>RSL</th>
<th>YAHSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h^{\text{FF}}$</td>
<td>1494</td>
<td>1518</td>
<td>1529</td>
</tr>
<tr>
<td>$h^{\text{rb}}$</td>
<td>1508</td>
<td>1513</td>
<td>1541</td>
</tr>
<tr>
<td>$h^{\text{CFF}}_{\text{offline}}$</td>
<td>1498</td>
<td>1577</td>
<td>1603</td>
</tr>
<tr>
<td>$h^{\text{CFF}}_{\text{online}}$</td>
<td>–</td>
<td>1665</td>
<td>1573</td>
</tr>
</tbody>
</table>
# Experiments

<table>
<thead>
<tr>
<th></th>
<th>GBFS-RSL</th>
<th>LAMA</th>
<th>BFWS($f_5$)</th>
<th>Dual-BFWS</th>
<th>Mercury</th>
<th>MERWIN</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBFS-RSL</td>
<td>–</td>
<td>20</td>
<td>20</td>
<td>16</td>
<td>15</td>
<td>13</td>
<td>1665</td>
</tr>
<tr>
<td>LAMA</td>
<td>6</td>
<td>–</td>
<td>19</td>
<td>10</td>
<td>5</td>
<td>4</td>
<td>1574</td>
</tr>
<tr>
<td>BFWS($f_5$)</td>
<td>8</td>
<td>15</td>
<td>–</td>
<td>5</td>
<td>11</td>
<td>9</td>
<td>1530</td>
</tr>
<tr>
<td>Dual-BFWS</td>
<td>9</td>
<td>18</td>
<td>22</td>
<td>–</td>
<td>12</td>
<td>10</td>
<td>1623</td>
</tr>
<tr>
<td>Mercury</td>
<td>9</td>
<td>14</td>
<td>19</td>
<td>13</td>
<td>–</td>
<td>2</td>
<td>1605</td>
</tr>
<tr>
<td>MERWIN</td>
<td>10</td>
<td>17</td>
<td>20</td>
<td>14</td>
<td>12</td>
<td>–</td>
<td>1634</td>
</tr>
</tbody>
</table>
Conclusion

- New lookahead strategy for GBFS using relaxed subgoals
- Beats state-of-the-art planners with $h^{CFF}$ and online refinement
- Lookahead strategy is fairly general, and could be used with other methods that provide subgoals or abstract plans