Building a Heuristic for Greedy Search

Christopher Wilt and Wheeler Ruml

University of New Hampshire
Department of Computer Science

Grateful thanks to NSF for support.
Greedy Best-First Search (GBFS)

1. best-first on $h(n)$, no $g(n)$. maintains open list.
2. very important in applications (eg planning)
3. building block for anytime algorithms
4. heuristic is crucial
Greedy Best-First Search (GBFS)

1. best-first on \( h(n) \), no \( g(n) \). maintains open list.
2. very important in applications (e.g., planning)
3. building block for anytime algorithms
4. heuristic is crucial

Two central points:

1. suboptimal is different!
   - heuristics for optimal search can be inappropriate
2. not hard to do better
   - Goal Distance Rank Correlation (GDRC)
8+4 PDBs dominates 8+0, right?
Example 1/3: 12-Disk 4-Peg Towers of Hanoi

Subopt $\neq$ Optimal
- Overview
- Hanoi
- GBFS on Hanoi
- TopSpin
- Tile Puzzle
- Summary

GDRC

Conclusion

8+4 PDBs dominates 8+0, right? Yes, but...

<table>
<thead>
<tr>
<th>heuristic</th>
<th>node expansions</th>
</tr>
</thead>
<tbody>
<tr>
<td>8+4 PDBs</td>
<td>2,153,558</td>
</tr>
<tr>
<td>8+0 PDB</td>
<td>4,618,913</td>
</tr>
</tbody>
</table>

better for A* $\neq$ better for GBFS
GBFS Behavior on Towers of Hanoi

Subopt ≠ Optimal
- Overview
- Hanoi
- GBFS on Hanoi
- TopSpin
- Tile Puzzle
- Summary

GDRC

Conclusion

- no local minima
- 256 states at each $h$ value

- low $h$ states can be far from goal
- want ‘serialized $h$ values’
Example 2/3: 12-Token 4-Turnstile TopSpin

Subopt ≠ Optimal
- Overview
- Hanoi
- GBFS on Hanoi
- TopSpin
- Tile Puzzle
- Summary

GDRC
Conclusion

4/12 TopSpin with Different PDB's

many disconnected $h = 0$ regions
Example 3/3: $3 \times 4$ Sliding Tile Puzzle

Subopt $\neq$ Optimal

- Overview
- Hanoi
- GBFS on Hanoi
- TopSpin
- Tile Puzzle
- Summary

GDRC

Conclusion

Wheeler Ruml (UNH)

Heuristics for Greedy Search – 6 / 11
### Example 3/3: 3 × 4 Sliding Tile Puzzle

<table>
<thead>
<tr>
<th>outer L</th>
<th>checkered</th>
</tr>
</thead>
<tbody>
<tr>
<td>A A 3</td>
<td>1 A 3</td>
</tr>
<tr>
<td>A A A 7</td>
<td>4 A 6 A</td>
</tr>
<tr>
<td>8 9 10 11</td>
<td>A 9 A 11</td>
</tr>
</tbody>
</table>

- **One** \( h = 0 \) **region**
- **Same size**
- **Node expansions**
  - **GBFS**
    - 258
    - 11,583
    - 17,641
    - 193,849
  - **A***
    - 1,251,260
    - 1,423,378
    - 1,596,041
    - 1,911,566

GBFS is sensitive to \( h \)

GBFS wants serialized \( h \) values
Two central points:

1. **suboptimal is different!**
   
   heuristics for optimal search can be inappropriate

2. **not hard to do better**

   Goal Distance Rank Correlation (GDRC)
Goal Distance Rank Correlation (GDRC)

Subopt ≠ Optimal

- GDRC
- Building $h$
- GDRC $h$ Results

Conclusion

Goal distance: number of steps to the goal
Rank correlation: how well orderings line up

$$GDRC(h) = \tau(h, d^*)$$

can use all nodes or just a sample near the goal
Goal Distance Rank Correlation (GDRC)

- **Goal distance**: number of steps to the goal
- **Rank correlation**: how well orderings line up

\[
GDRC(h) = \tau(h, d^*)
\]

can use all nodes or just a sample near the goal

**Hanoi**

**3 × 4 Puzzle**
example: simple hill-climbing in PDB-space

start by abstracting everything
loop until done:
  consider all refinements
  pick one with highest GDRC
Performance of GDRC-Built Heuristics

TopSpin with unit costs: good for both GBFS and A*

TopSpin with some heavy tokens:
Performance of GDRC-Built Heuristics

TopSpin with unit costs: good for both GBFS and A*

TopSpin with some heavy tokens:

<table>
<thead>
<tr>
<th>PDB</th>
<th>node expansions</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GDRC-built (contiguous)</td>
<td>411</td>
<td>10,607</td>
<td>52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>built for A* (heavy)</td>
<td>961</td>
<td>411</td>
<td>94</td>
<td></td>
<td></td>
</tr>
<tr>
<td>random</td>
<td>2,387</td>
<td>26,017</td>
<td>48</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Subopt = Optimal

GDRC
■ GDRC
■ Building h
■ GDRC h Results

Conclusion
Performance of GDRC-Built Heuristics

TopSpin with unit costs: good for both GBFS and A*

TopSpin with some heavy tokens:

<table>
<thead>
<tr>
<th>PDB</th>
<th>node expansions</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GBFS</td>
<td>A*</td>
<td>avg (h) value</td>
</tr>
<tr>
<td>GDRC-built (contiguous)</td>
<td>411</td>
<td>10,607</td>
<td>52</td>
</tr>
<tr>
<td>built for A* (heavy)</td>
<td>961</td>
<td>411</td>
<td>94</td>
</tr>
<tr>
<td>random</td>
<td>2,387</td>
<td>26,017</td>
<td>48</td>
</tr>
</tbody>
</table>

Sliding Tiles:

<table>
<thead>
<tr>
<th>abstraction</th>
<th>GBFS</th>
<th>A*</th>
</tr>
</thead>
<tbody>
<tr>
<td>outer L</td>
<td>258</td>
<td>1,251,260</td>
</tr>
<tr>
<td>checkered</td>
<td>11,583</td>
<td>1,423,378</td>
</tr>
<tr>
<td>average 6-tile PDB</td>
<td>17,641</td>
<td>1,596,041</td>
</tr>
</tbody>
</table>

instance-specific                  | 8,530           | 480,250  |
| GDRC-built (4th/462)              | 427             | 1,197,789|

GDRC-built heuristics seem well-tuned for GBFS!
suboptimal is different!

- more sensitive to $h$
- better $h$ for optimal can be worse for greedy
- needs its own theory and methods

not hard to do better!

Goal Distance Rank Correlation (GDRC)

- estimates seem effective
- can search space of PDBs

this area is wide open!
Extra Slides

Subopt ≠ Optimal

GDRC

Conclusion

Extra Slides

- More results