Lookahead Pathologies & Meta Reasoning in Real-time Decision Making

Vadim Bulitko
University of Alberta
http://ircl.cs.ualberta.ca
Audience Poll

Suppose you are deliberating on a problem:

“To Buy the House or Not To Buy?”

Suppose you have to tell the realtor your decision in one day

Should you deliberate for the entire hour or not?

See what you think at the end of the talk...
Purpose of the talk

- Presents work in progress:
- To start a discussion...
- And hear your opinions
  - Especially on applicability of this research and similar efforts
Talk Map

Motivation

Real-time HS & Pathologies

What to do about them
Real-time decision-making

- Limited time per move
- Limited information at any time
- Lack of effective goal test
- Very large space

Examples
- DCA
- MR ADORE
- RTS games
- NxM-1 puzzles
Real-time Heuristic Search

- Optimal solutions may be impossible/difficult to get
  - NxN-1 puzzle is NP-hard [Ratner86]
  - Solving the 24 puzzle [Korf96]

- Give up optimality…

- Then what?
  - Incomplete real-time heuristic search?
RTA* vs. IDA*

- 8 puzzle --- 1,000 random starting states
- Manhattan distance
- Average number of nodes generated before the first move can be made:
  - IDA*: 7,736
  - RTA* (lookahead of 0): 3
  - RTA* (lookahead of 1): 6
- Average number of nodes generated before a solution is found:
  - IDA*: 7,736
  - RTA* (lookahead of 0): 579
  - RTA* (lookahead of 1): 1,038
**RTA* vs. IDA**

- 24-puzzle

- **RTA***: Manhattan distance
- **IDA***: Manhattan distance with: linear conflict, last-two-moves, corner-tile heuristic + FSA-based pruning

**Per problem instance:**
- **IDA***: 8 billion - 8 trillion nodes (up to 3 months in 1996)
- **RTA*** (ply=2): 2.5 million nodes (~2.5 seconds in 1990)
A Hypothetical Argument

Optimist:
- “Still working on optimal search?... When will you see the light?...”

Pessimist:
- “Your solution quality may be pretty bad!”

Optimist:
- “Increase the lookahead depth!”

Pessimist:
- “By how much?”

Optimist:
- “As much as your resources allow you!”

Pessimist:
- ...silently goes away for a while...
Pessimist Strikes Back

- **Pessimist** (smirking):
  - “Hey you! Suppose you have a choice between $\text{RTA}^*_5$ and $\text{RTA}^*_10$ and you have the resources to do either in real time... Which one would you go for?”

- **Optimist** (immediately):
  - “$\text{RTA}^*_10$ of course!”

- **Pessimist** (with a satisfied smile):
  - “Here we go: $\text{RTA}^*_10$ will spend $\sim27$ times MORE time than $\text{RTA}^*_5$... ...and it will find 1.7 times WORSE solutions!”
The Data

- 8 puzzle
- ANN-learned heuristic
- 1,000 random states

### Average number of steps taken (per problem instance)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>3312</td>
<td>758</td>
<td>336</td>
<td>126</td>
<td>115</td>
<td>68</td>
<td>88</td>
<td>87</td>
<td>95</td>
<td>96</td>
<td>115</td>
</tr>
</tbody>
</table>

### Average number of nodes generated (per problem instance)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>9557</td>
<td>4530</td>
<td>3752</td>
<td>2493</td>
<td>4125</td>
<td>4244</td>
<td>9706</td>
<td>16054</td>
<td>31328</td>
<td>53173</td>
<td>114269</td>
</tr>
</tbody>
</table>
The Take-Home Slide

“More thinking with incomplete information may lead to worse solutions”

Examples:

- Deeper mini-max in games can lead to more frequent losses [Nau]
- Deeper lookahead in real-time heuristic search can lead to worse solutions [Bulitko]
- Larger pattern databases can lead to longer A* searches [Holte]
Talk Map

- Motivation
- Real-time HS & Pathologies
- What to do about them
Real Time Search

- RTA* [Korf 1990]
- SRTA* [Russell, Wefald 1991]
- RTDP [Barto 1995]
- DTA* [Russell, Wefald 1991]

For more details:
- http://www.cs.ualberta.ca/ircl/applications/dmai/
Framework (RTA*)

For the rest of the paper we assume this type of decision-making [RTA* adapted from Russell/Wefald 1991]:

Agent(state $s_{\text{start}}$)

- $s = s_{\text{start}}$
- **While not terminate($s$)**
  - Generate immediate children $s_1, \ldots, s_n$ of $s$
  - Evaluate each of them:
    - $F(s_i) = \text{cost}(a_i) + \text{TABLE}(s_i)$ if $s_i$ has been visited before
    - $F(s_i) = \text{cost}(a_i) + \text{evaluate}(s_i,\text{ply})$ if $s_i$ is novel
  - $s^* = \arg \min F(s_i)$ // break ties randomly
  - $\text{TABLE}(s) = \text{cost}(a^*) + F(s_{\text{second_best}})$
  - Execute $a^*$ : $s = s^*$
Node Evaluation (mini-min)

Function `evaluate(s,ply)` is the lookahead-enhanced static evaluator.

**Evaluate(s,ply)**
- Build lookahead tree rooted in s of depth ply
  - Can use alpha-pruning if h is admissible
- Consider all frontier and goal nodes (if the goal has been seen while looking ahead)
- Compute their heuristic values h
- Mini-min the f=g+h values of such nodes up to level 1
- Return the minimum f value
RTA* In Action

![RTA* In Action Diagram]

- RTA* In Action Diagram with nodes and edges labeled with values.
**Caveats with RTHS**

- Real-time heuristic search operates with limited information
- H values are shown in blue
- Mini-min value of c1 is 6+2=8
- Mini-min value of c2 is 4+2=6
- So RTA* will head down to c2 even though c1 is better
Extensions to RTA*

**Path-max:**
- Along every path in the lookahead tree:
  \[ f'(\text{child}) = \max(f(\text{child}), f(\text{parent})) \]

**Full-pruning:**
- A hash table is used so that the nodes generated at the frontier of each of the lookahead trees cannot be reached via any shorter route
T1(k) Pathologies

For each state $s$ the expected solution length vector can be defined:

$$(\|s,1\|,\ldots,\|s,m\|)$$

Where $m$ is the maximum lookahead depth used with RTA*

A collection of such vectors for all starting states is called expected solution length matrix
Example

Consider the 1D domain of size 6:

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & 5 \\
\end{array}
\]

Heuristic: \([0, 1, 0, 0, 1, 1]\)

Here is what happens with RTA*
T1(k) Pathologies

A state is called T1(k) pathological iff its expected solution length vector has exactly $k$ consecutive increases in it.

Heuristic is called T1(k) pathological if its solution length vector has exactly $k$ consecutive increases in it.

Example: $h=[0,1,0,0,1,1]$ with its solution length vector $(4.2, 3.3, 3.4, 3.0)$ is T1(1) pathological.
Empirical Domains

- **1D Navigation**

  
  ![1D Navigation Diagram](image)

- **NxM-1 sliding tile puzzles**

  
  ![8-puzzle Diagram](image)

  \[
  \text{8-puzzle} \\
  181,440 \text{ nodes}
  \]
Hypotheses

- Pathological heuristics/states are frequent
- The proportion of pathological heuristics/states increases with domain size
- Pathological heuristics can be effectively detected:
  - Off-line
  - On-line
- Optimal lookahead depth can be effectively computed on-line
Hypothesis 1a

"Pathological heuristics are frequent"

<table>
<thead>
<tr>
<th>TOTAL</th>
<th>6</th>
<th>24</th>
<th>120</th>
<th>720</th>
<th>5,040</th>
<th>40,320</th>
<th>362,880</th>
<th>3,628,800</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 (1)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>5.8%</td>
<td>21.9%</td>
<td>46.9%</td>
<td>64.6%</td>
<td>74.9%</td>
<td>80.4%</td>
</tr>
<tr>
<td>T1 (2)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>1.1%</td>
<td>2.1%</td>
<td>4.5%</td>
<td>6.7%</td>
<td>9.3%</td>
</tr>
<tr>
<td>T1 (3)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.2%</td>
<td>0.2%</td>
<td>0.4%</td>
<td>0.6%</td>
</tr>
<tr>
<td>T1 (4)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T1 (5)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T1 (6)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
Hypothesis 1b

“Pathological states are frequent”

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>MD</th>
<th>NM</th>
<th>Unif</th>
<th>Unif-mon</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 (1)</td>
<td>25.8%</td>
<td>57.2%</td>
<td>53.6%</td>
<td>38.1%</td>
</tr>
<tr>
<td>T1 (2)</td>
<td>54.2%</td>
<td>26.7%</td>
<td>34.4%</td>
<td>12.5%</td>
</tr>
<tr>
<td>T1 (3)</td>
<td>0.0%</td>
<td>4.7%</td>
<td>8.9%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T1 (4)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T1 (5)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T1 (6)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
An Observation

- Remarkably, the good old Manhattan distance is $T_1(1)$-pathological in the 2x3 puzzle:
- This is not the case with the 3x3-puzzle though as Korf demonstrated in 1990
Hypothesis 2a

“The proportion of pathological heuristics goes up with the domain size”

<table>
<thead>
<tr>
<th>TOTAL</th>
<th>6</th>
<th>24</th>
<th>120</th>
<th>720</th>
<th>5,040</th>
<th>40,320</th>
<th>362,880</th>
<th>3,628,800</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 (1)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>5.8%</td>
<td>21.9%</td>
<td>46.9%</td>
<td>64.6%</td>
<td>74.9%</td>
<td>80.4%</td>
</tr>
<tr>
<td>T1 (2)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>1.1%</td>
<td>2.1%</td>
<td>4.5%</td>
<td>6.7%</td>
<td>9.3%</td>
</tr>
<tr>
<td>T1 (3)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.2%</td>
<td>0.2%</td>
<td>0.4%</td>
<td>0.6%</td>
</tr>
<tr>
<td>T1 (4)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T1 (5)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T1 (6)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
Hypothesis 2a

“The proportion of pathological heuristics goes up with the domain size”

Even for consistent heuristics...

<table>
<thead>
<tr>
<th>Consistent</th>
<th>5</th>
<th>13</th>
<th>35</th>
<th>96</th>
<th>267</th>
<th>750</th>
<th>2,123</th>
<th>6,046</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 (1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>44</td>
<td>210</td>
<td>791</td>
<td>2,814</td>
</tr>
<tr>
<td>T1 (2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>T1 (3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (4)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (5)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (6)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Hypothesis 2a

“The proportion of pathological heuristics goes up with the domain size”

Even with full pruning and path-max...

<table>
<thead>
<tr>
<th>Consistent</th>
<th>5</th>
<th>13</th>
<th>35</th>
<th>96</th>
<th>267</th>
<th>750</th>
<th>2,123</th>
<th>6,046</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 (1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>48</td>
<td>264</td>
</tr>
<tr>
<td>T1 (2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (4)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (5)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (6)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inconsistent</th>
<th>1</th>
<th>11</th>
<th>85</th>
<th>624</th>
<th>4,773</th>
<th>39,570</th>
<th>360,757</th>
<th>3,622,754</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 (1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>56</td>
<td>1,200</td>
<td>19,004</td>
</tr>
<tr>
<td>T1 (2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (4)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (5)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T1 (6)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Hypothesis 2b

“The proportion of pathological states goes up with the domain size”

<table>
<thead>
<tr>
<th>Domain size</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 (1)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>4.4%</td>
<td>10.6%</td>
<td>18.9%</td>
<td>26.2%</td>
<td>30.6%</td>
<td>32.2%</td>
</tr>
<tr>
<td>T1 (2)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.9%</td>
<td>3.0%</td>
<td>6.8%</td>
<td>11.6%</td>
</tr>
<tr>
<td>T1 (3)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.2%</td>
<td>1.1%</td>
</tr>
<tr>
<td>T1 (4)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T1 (5)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>T1 (6)</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
Some Insights

- The number of admissible natural-number-valued heuristics in 1D domain goes up as $N!$

- Why is it that the proportion of pathological states and heuristics increases?

- Short answer: any pathological heuristic in the 1D domain of size $N$ can be extended for the 1D domain of size $N+1$ in more than $N+1$ ways and will still be pathological
Talk Map

- Motivation
- Real-time HS & Pathologies
- What to do about them
Heuristics

Here are some ways to get heuristics:

- Manual engineering
- Constraint relaxation [Pearl84]
- Limiting subgoal interaction [Korf96]
- Abstraction [Prieditis93]
- Pattern databases [Culberson, Schaeffer, Holte, et al.]
- Machine learning
  - On-line (reinforcement learning)
  - Off-line (sampling + supervised learning)

All of these techniques can possibly produce pathological heuristics...
Remedies

1. Beforehand:
   - Detect and filter out pathological heuristics off-line

2. If cannot do that then:
   - Detect pathology on-line and adjust the heuristic accordingly on the fly (incrementally)

3. If we are positively stuck with a fixed pathological heuristic then at least:
   - Select the lookahead depth appropriately...
Remedy 1: Off-line Detection

Presently the only way to tell if a heuristic is pathological is to run extensive on-line tests with it.

Example: in the 1D domain of size 8 one can try all heuristics many times and collect the training data of this sort:

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>k in T1(k)</th>
<th>Pathological?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 5 1 2</td>
<td>0</td>
<td>no</td>
</tr>
<tr>
<td>0 0 0 0 0 5 1 3</td>
<td>0</td>
<td>no</td>
</tr>
<tr>
<td>0 0 0 0 0 5 1 4</td>
<td>2</td>
<td>yes</td>
</tr>
<tr>
<td>0 0 0 0 0 5 1 5</td>
<td>1</td>
<td>yes</td>
</tr>
<tr>
<td>0 0 0 0 0 5 1 6</td>
<td>1</td>
<td>yes</td>
</tr>
<tr>
<td>0 0 0 0 0 5 1 7</td>
<td>2</td>
<td>yes</td>
</tr>
</tbody>
</table>
Sounds like a perfect task for the classical supervised learning?

Sure:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>40320</td>
<td>25470</td>
<td>ANN: 8-10-2</td>
<td>20.5%</td>
<td>62.7%</td>
<td>37.4%</td>
<td>10.7%</td>
<td>89.3%</td>
</tr>
<tr>
<td>8</td>
<td>40320</td>
<td>25470</td>
<td>466 decision rules</td>
<td>16.1%</td>
<td>69.5%</td>
<td>30.5%</td>
<td>7.7%</td>
<td>92.3%</td>
</tr>
</tbody>
</table>
Problems

First, in any reasonably sized domain it won’t be possible to plug the entire explicitly specified heuristic as an input to the MLed detector...

Even if we could, the raw $h$ values are not sufficiently meaningful. Indeed, let’s inspect the learned decision rules:
Memorization...

Clearly, the 466 rules simply memorize the specific cases.
Clearly, if we use MLing to detect and filter out pathological heuristics, we need features:

- Compact
- Effectively computable
- Meaningful

Ideas?
Remedy 2: on-line correction

- What does pathology mean?
- It means that the heuristic is inaccurate in a certain way.
- Any detected inaccuracies can (potentially) be used to adjust the heuristic.

Related research:
- Reinforcement learning (e.g., TD(λ)-learning) [Sutton & Barto, et al.]
- Real-time dynamic programming [Barto]
- Incremental trap elimination [VKB, 2003]
Remedy 3 : Meta-level Control

- If we cannot discard or adjust the heuristic then at least we can be smart about using it
- How?

- We need to control our search intelligently
  - ==> meta-level control
Meta Level Control

- We need to control the search adaptively

Previous research
- Hand-engineered control
  - search extensions in alpha-beta
- Machine-learned control:
  - DTA* and MGSS* [Russell/Wefald 1991]
  - BPIP [Baum/Smith 1995]
Control Input: Features

- **Global features:**
  - “Temperature” (a la simulated annealing)

- **Markov features (history-free):**
  - Manhattan distance of the current state
  - Hamming distance of the current state
  - “fragility” [Schaeffer]
  - Variations in $f(a)$ from ply to ply (a la in two-player games)

- **History-based features:**
  - Number of state revisits in a moving window
  - Number of min $f$ increases in a moving window
Output : Control Actions

- Random jumps
- Reversing direction (max instead of min)
- Selecting search horizon
Adaptive Lookahead Depth Selection

- Average solution length is 22
  - IDA* with Manhattan takes 7,736 nodes to find it

Notes:
- MD becomes nearly optimal (26)
- Tree becomes optimal (22)

### Average number of steps taken (per problem instance)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Adapt</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>3312</td>
<td>758</td>
<td>336</td>
<td>126</td>
<td>115</td>
<td>68</td>
<td>88</td>
<td>87</td>
<td>95</td>
<td>96</td>
<td>115</td>
<td>32</td>
</tr>
<tr>
<td>Unif</td>
<td>608</td>
<td>916</td>
<td>968</td>
<td>868</td>
<td>1410</td>
<td>1742</td>
<td>2031</td>
<td>2526</td>
<td>2775</td>
<td>1768</td>
<td>1274</td>
<td>145</td>
</tr>
<tr>
<td>Tree</td>
<td>9811</td>
<td>3293</td>
<td>45</td>
<td>29</td>
<td>27</td>
<td>27</td>
<td>26</td>
<td>27</td>
<td>26</td>
<td>26</td>
<td>25</td>
<td>22</td>
</tr>
<tr>
<td>NM</td>
<td>1630</td>
<td>916</td>
<td>929</td>
<td>626</td>
<td>351</td>
<td>164</td>
<td>115</td>
<td>97</td>
<td>79</td>
<td>69</td>
<td>57</td>
<td>34</td>
</tr>
<tr>
<td>&quot;0&quot;</td>
<td>127237</td>
<td>80456</td>
<td>49482</td>
<td>27051</td>
<td>22958</td>
<td>12389</td>
<td>8248</td>
<td>4399</td>
<td>3306</td>
<td>1884</td>
<td>1313</td>
<td>482</td>
</tr>
<tr>
<td>kNN</td>
<td>3482</td>
<td>1674</td>
<td>1021</td>
<td>843</td>
<td>696</td>
<td>568</td>
<td>544</td>
<td>514</td>
<td>576</td>
<td>449</td>
<td>402</td>
<td>92</td>
</tr>
<tr>
<td>MD</td>
<td>205</td>
<td>172</td>
<td>171</td>
<td>101</td>
<td>94</td>
<td>73</td>
<td>58</td>
<td>53</td>
<td>52</td>
<td>43</td>
<td>35</td>
<td>26</td>
</tr>
</tbody>
</table>
Meta-level Control Policy?

- Hand-engineered

- Machine learned:
  - On-line
  - Off-line

- Current research...
Future Research

- Different types of search
  - SRTA*
  - RTDP
- Different types of heuristics
  - Pattern database heuristics
- Pathology in real life domains:
  - Real-time path-planning in RTS games?
  - MR ADORE
- Pathology in two-player games:
  - MLed functions
  - Particular positions (traps)
Discussion

http://ircl.cs.ualberta.ca