A* (BF*) with Non-Additive Cost & Evaluation Functions

Andrey Balabokhin
Neema Kanapala
A* ALGORITHM

- Additive Evaluation function, \( f = g + h \)
- Additive cost function, \( g = g(n') + c(n, n') \)

A* always finds the shortest path to the goal node, if \( h \) is admissible.
TYPES OF COST FUNCTION

- Additive Cost Function
- Non-Additive Costs
## COST FUNCTIONS

### Additive costs

\[ \text{sum} \left[ c(p_i, p_{i+1}) \right] \]

### Non-Additive Costs

<table>
<thead>
<tr>
<th>Multiplicative costs</th>
<th>product [ c(p_i, p_{i+1}) ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>\text{avg} \left[ c(p_i, p_{i+1}) \right]</td>
</tr>
<tr>
<td>Median</td>
<td>\text{middle value in a sorted list of costs}</td>
</tr>
<tr>
<td>Mode</td>
<td>\text{most frequent} \left[ c(p_i, p_{i+1}) \right]</td>
</tr>
<tr>
<td>Range</td>
<td>\text{max} \left[ c(p_i, p_{i+1}) \right] - \text{min} \left[ c(p_i, p_{i+1}) \right]</td>
</tr>
<tr>
<td>Max-cost</td>
<td>\text{max} \left[ c(p_i, p_{i+1}) \right]</td>
</tr>
<tr>
<td>Last edge cost</td>
<td>\text{c}(p_{n-1}, p_n)</td>
</tr>
</tbody>
</table>

### Last edge cost

\[ \text{c}(p_{n-1}, p_n) \]
RANGE

$$g(P) = \max [c(p_i, p_{i+1})] - \min [c(p_i, p_{i+1})]$$

Arrows indicate the parent nodes

$g(P_1) = 4-2 = 2$  \hspace{1cm} P_1 - (A, B, S)$

Arrows indicate the parent nodes
\[ g(P_1) = 5 - 4 = 1 \]  
\[ g(P_2) = 4 - 2 = 2 \]
\[ g(P_2) = 4 - 1 = 3 \]
\[ P_2 - (D, A, C, S) \]

\[ g(P_1) = 5 - 1 = 4 \]
\[ P_1 - (D, A, B, S) \]

but \( 4 > 3 \)!
ORDER PRESERVING COST ORDER FUNCTION

Let $P_1$ and $P_2$ be any two paths from “s” node to “n” node and $P_1P_3$ and $P_2P_3$ paths after concatenation of a new path $P_3$, if $g(P_1) \geq g(P_2) \implies g(P_1P_3) \geq g(P_2P_3)$, $g$ is said to be order preserving.
RANGE IS NON-ORDER PRESERVING

g(P_1) = 4 - 2 = 2
\quad g(P_2) = 5 - 4 = 1
\quad g(P_1) > g(P_2)

\quad g(P_1 P_3) = 4 - 1 = 3
\quad g(P_2 P_3) = 5 - 1 = 4
\quad g(P_1 P_3) < g(P_2 P_3)
Order-preserving property for the mean and mode cost functions

Legend:

<table>
<thead>
<tr>
<th>g(P) - Mean edge cost</th>
<th>g(P) - Mode (most frequent ([c(p_i, p_{i+1})]))</th>
</tr>
</thead>
<tbody>
<tr>
<td>- (P_1), (g(P_1) = 5)</td>
<td>- (P_1), (g(P_1) = 3)</td>
</tr>
<tr>
<td>- (P_2), (g(P_2) = 5.25)</td>
<td>- (P_2), (g(P_2) = 2)</td>
</tr>
<tr>
<td>- (g(P_1P_3) = 6.7)</td>
<td>- (P_3), (g(P_1P_3) = 3), (g(P_2P_3) = 6)</td>
</tr>
<tr>
<td>- (g(P_2P_3) = 6.2)</td>
<td>(g(P_1) &gt; g(P_2)) but (g(P_1P_3) &lt; g(P_2P_3))</td>
</tr>
</tbody>
</table>

\(g(P_1) < g(P_2)\) but \(g(P_1P_3) > g(P_2P_3)\)
Order-preserving property for the median and max cost functions

\[ g(P) - \text{Median edge cost} \]
- \( P_1, g(P_1) = 5.5 \)
- \( P_2, g(P_2) = 5 \)
- \( g(P_1P_3) = 4 \)
- \( g(P_2P_3) = 5 \)
\[ g(P_1) > g(P_2) \text{ but } g(P_1P_3) < g(P_2P_3) \]

\[ g(P) - \text{Max} \]
- \( P_1, g(P_1) \geq g(P_2) \)
- \( P_2 \)
- \( P_3 \)
Let \( g(P_1) \geq g(P_2) \)
if \( g(P_3) \geq g(P_1) \Rightarrow g(P_1P_3) = g(P_2P_3) = g(P_3) \)
if \( g(P_3) < g(P_1) \Rightarrow g(P_1P_3) = g(P_1) \geq \max (g(P_2), g(P_3)) = g(P_2P_3) \)

Legend:
- Red for \( P_1, g(P_1) = 5.5 \)
- Green for \( P_2, g(P_2) = 5 \)
- Blue for \( P_3 \)
- Black for \( g(P_1P_3) = 4 \)
- Green for \( g(P_2P_3) = 5 \)
If P is a path with nodes $p_1, p_2, \ldots, p_{n-1}, p_n$ and $c(p_i, p_{i+1})$ is an cost value for the edge between nodes $p_i$ and $p_{i+1}$. Cost measure function $g(P)$ can be any function:

<table>
<thead>
<tr>
<th>Cost Function</th>
<th>Is $g(P)$ order-preserving?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive costs</td>
<td>sum $[c(p_i, p_{i+1})]$</td>
</tr>
<tr>
<td>Multiplicative costs</td>
<td>product $[c(p_i, p_{i+1})]$</td>
</tr>
<tr>
<td>Mean</td>
<td>avg $[c(p_i, p_{i+1})]$</td>
</tr>
<tr>
<td>Median</td>
<td>middle value in a sorted list of costs</td>
</tr>
<tr>
<td>Mode</td>
<td>most frequent $[c(p_i, p_{i+1})]$</td>
</tr>
<tr>
<td>Range</td>
<td>max $[c(p_i, p_{i+1})]$ - min $[c(p_i, p_{i+1})]$</td>
</tr>
<tr>
<td>Max-cost</td>
<td>max $[c(p_i, p_{i+1})]$</td>
</tr>
<tr>
<td>Last edge cost</td>
<td>$c(p_{n-1}, p_n)$</td>
</tr>
</tbody>
</table>
BEST FIRST* (BF*) ALGORITHM

● “f” is an arbitrary function
eg: \( f = \sqrt{g^2 + h^2} \)
\( f = \max(g, h) \)

● Builds path along the least values of \( f(P_i) \)

● A* is a special case of BF*
Admissibility of BF*

If in every graph searched by BF* there exists at least one optimal solution path along which $f$ attains its maximum value on the goal node, then BF* is admissible.

- optimal path $P_1$
- not optimal path $P_2$
- not optimal path $P_3$

$g(P_1) \leq g(P_2), \ g(P_1) \leq g(P_3)$

$n_1, n_2, ..., n_k, ..., \Gamma$ - nodes along $P_1$

$BF^*$ is admissible if $f(\Gamma) = \max(f(n_i))$ along optimal path $P_1$
BF* ALGORITHM REQUIREMENT

- “f” must be Order-preserving for all the paths from start node “s” to goal node “Г”.
- “f” must be admissible.
BF* EXAMPLE WITH MAX-COST FUNCTION

\[ g(P_i) = \max \{c(p_i, p_{i+1})\} \]
\[ f(P_i) = g(P_i) \]

h - trivial

f - order-preserving

f - admissible, as it is not decreasing with increasing number of nodes

\[ \Rightarrow f(\text{goal node}) = \max(f_i) \]
THANK YOU!
QUESTIONS ?
1. Judea Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving (The Addison-Wesley series in artificial intelligence) 1984; Section 3.3 ("Some Extensions to Nonadditive Evaluation Functions")