Planning: STRIPS and A* GOAP

CMPS 146, Fall 2013

Josh McCoy
How do decision making methods compare?

Decision Trees, FSMs, Behavior Trees

They encode decision making.

All possible behaviors are static at design-time.

They react; they do not think ahead.
In comparison, planning algorithms...

They search over possible action sequences (i.e. plans).

Possibility space defined at design-time and searched from at run-time.

Look at the possible futures created by changing the world.

Goal-driven (mostly).
Planners To Be Discussed

- STRIPS
- A* GOAP
- Hierarchical Task Networks
STRIPS

Operators
  Add List
  Delete List

World State

Goal State
STRIPS

Operators
Add List
Delete List

GrabAmmo
+inventory(ammo)
-floor(ammo)

World State

Goal State

World State
floor(ammo)

Goal State
inventory(ammo)
STRIPS Algorithm

STRIPS($A$, $s$, $g$)
$p$ = empty plan
loop...
if $s$ satisfies $g$ then return $p$
a = [an applicable action in $A$, relevant for $g$]
if $a$ = null, then return failure
$p'$ = STRIPS($A$, $s$, precond($a$))
if $p'$ = failure, then return failure
$s$ = apply $p'$ to $s$
$s$ = apply $a$ to $s$
p = $p$ + $p'$ + $a$

STRIPS Algorithm: The Original

Resolvents are metric of how relevant the next possible actions are.

Compute a successor node: adds a node to the internal A*.
That's right: A*!

Both developed at SRI AI center.

STRIPS uses it to expand plans based on:  
g(n)+h(n) cost evaluation.

What could you use as the heuristic?

In a sense, this gives forward and backward search to the goal.
STRIPS Example

; Invert stack (good goal ordering)
> (gps '(a on b)(b on c) (c on table) (space on a) (space on table))
  '(b on a) (c on b))
Goal: (B ON A)
Consider: (MOVE B FROM C TO A)
  Goal: (SPACE ON B)
  Goal: (SPACE ON A)
  Goal: (SPACE ON TABLE)
  Goal: (A ON B)
Action: (MOVE A FROM B TO TABLE)
  Goal: (SPACE ON A)
  Goal: (B ON C)
  Goal: (C ON B)
Action: (MOVE B FROM C TO A)
  Goal: (C ON B)
  Goal: (SPACE ON C)
Consider: (MOVE C FROM TABLE TO B)
  Goal: (SPACE ON C)
  Goal: (SPACE ON B)
  Goal: (C ON TABLE)
Action: (MOVE C FROM TABLE TO B)
  ((START)
   (EXECUTING (MOVE A FROM B TO TABLE))
   (EXECUTING (MOVE B FROM C TO A))
   (EXECUTING (MOVE C FROM TABLE TO B)))
Games that use STRIPS-style planning

http://aigamedev.com/open/review/planning-in-games/
In this example, each element in the state vector takes on Boolean values:

```plaintext
{
    Recipe: true if character has a recipe
    Pizza Phone #: true if character has a phone number for ordering pizza
    Food: true if something to eat is in possession
    Ingredients: true if the ingredients to make the recipe are in possession
    Silverware: true if the silverware is in possession
    Hungry: true if the character is hungry
    Requires Silverware: true if the food requires silverware to eat
}
```
A* GOAP Example – Actions

Look up recipe (lur)
- P: \{N, _, _, _, _, _, _\}
- E: \{Y, _, _, _, _, _, _\}

Look up phone number (luph)
- P: \{_, N, _, _, _, _, _\}
- E: \{_, Y, _, _, _, _, _\}

Shop for ingredients (sfi)
- P: \{Y, _, _, N, _, _, _\}
- E: \{_, _, _, Y, _, _, _\}

Call Pizza (cp)
- P: \{_, Y, N, _, _, _, _\}
- E: \{_, _, Y, N, _, _, _\}

Cook (c)
- P: \{Y, _, N, Y, _, _, _\}
- E: \{_, _, Y, N, _, _, Y\}

Get silverware (gs)
- P: \{_, _, _, _, N, _, _\}
- E: \{_, _, _, _, Y, _, _\}

Eat with silverware (ews)
- P: \{_, _, Y, _, Y, Y, Y\}
- E: \{_, _, N, _, _, N, _\}

Eat with hands (ewh)
- P: \{_, _, Y, _, N, Y, N\}
- E: \{_, _, N, _, _, N, _\}

{Recipe, Phone #, Food, Ingredients, Silverware, Hungry, Need Silverware}
A* GOAP Example – Graph
DIY: find the plan.
A* GOAP Example – Trace

Initial State: \{N, N, N, N, N, Y, N\}
Goal State: \{_, _, _, _, _, N, _\}
Heuristic: number of state elements different from goal

Open list: [\{N, N, N, N, N, Y, N\} cost 1]
Valid actions: lur, luph, gs
Open list:
  \{Y, N, N, N, N, Y, N\} lur cost: 2,
  \{N, Y, N, N, N, Y, N\} luph cost: 2,
  \{N, N, N, N, Y, Y, N\} gs cost: 2

Closed list: [\{N, N, N, N, N, Y, N\} cost: 1]
A* GOAP Example – Trace

{Y, N, N, N, N, Y, N} lur cost: 2
Valid actions: luph, sfi, gs
Open list:
{N, Y, N, N, N, Y, N} luph cost: 2,
{N, N, N, N, Y, Y, N} gs cost: 2
{Y, Y, N, N, N, Y, N} luph cost: 3
{Y, N, N, Y, N, Y, N} sfi cost: 3
{Y, N, N, N, Y, Y, N} gs cost: 3
Closed list:
{N, N, N, N, N, Y, N} cost: 1
{Y, N, N, N, N, Y, N} lur cost: 2
A* GOAP Example – Trace

{N, Y, N, N, N, Y, N}  luph cost: 2,
Valid actions: lur (duplicate state), cp, gs

Open list:
{N, N, N, N, Y, Y, N}  gs cost: 2
{Y, Y, N, N, N, Y, N}  luph cost: 3
{Y, N, N, Y, N, Y, N}  sfi cost: 3
{Y, N, N, N, Y, Y, N}  gs cost: 3
{N, Y, Y, N, N, Y, N}  cp cost: 3
{N, Y, N, N, Y, Y, N}  gs cost: 3

Closed list:
{N, N, N, N, N, Y, N}  cost: 1
{Y, N, N, N, N, Y, N}  lur cost: 2
{N, Y, N, N, N, Y, N}  luph cost: 2
A* GOAP Example – Trace

\{N, N, N, N, Y, Y, N\} gs cost: 2
Valid actions: lur (duplicate), luph (duplicate)

Open list:
\{Y, Y, N, N, N, Y, N\} luph cost: 3
\{Y, N, N, Y, N, Y, N\} sfi cost: 3
\{Y, N, N, N, Y, Y, N\} gs cost: 3
\{N, Y, Y, N, N, Y, N\} cp cost: 3
\{N, Y, N, N, Y, Y, N\} gs cost: 3

Closed list:
\{N, N, N, N, N, N, Y, N\} cost: 1
\{Y, N, N, N, N, N, Y, N\} lur cost: 2
\{N, Y, N, N, N, Y, N\} luph cost: 2
\{N, N, N, N, N, Y, Y, N\} gs cost: 2
A* GOAP Example – Trace

\{N, Y, N, N, N, Y, N\}  luph cost: 2,
Valid actions: lur (duplicate state), cp, gs

Open list:

\{N, N, N, N, Y, Y, N\}  gs cost: 2
\{Y, Y, N, N, N, Y, N\}  luph cost: 3
\{Y, N, N, Y, N, Y, N\}  sfi cost: 3
\{Y, N, N, N, Y, Y, N\}  gs cost: 3
\{N, Y, Y, N, N, Y, N\}  cp cost: 3
\{N, Y, N, N, Y, Y, N\}  gs cost: 3

Closed list:

\{N, N, N, N, N, Y, N\}  cost: 1
\{Y, N, N, N, N, Y, N\}  lur cost: 2
\{N, Y, N, N, N, Y, N\}  luph cost: 2
A* GOAP Example – Trace

{N, N, N, N, Y, Y, N} gs cost: 2
Valid actions: lur (duplicate), luph (duplicate)

Open list:
{Y, Y, N, N, N, Y, N} luph cost: 3
{Y, N, N, Y, N, Y, N} sfi cost: 3
{Y, N, N, N, Y, Y, N} gs cost: 3
{N, Y, Y, N, N, Y, N} cp cost: 3
{N, Y, N, N, Y, Y, N} gs cost: 3

Closed list:
{N, N, N, N, N, Y, N} cost: 1
{Y, N, N, N, N, Y, N} lur cost: 2
{N, Y, N, N, N, Y, N} luph cost: 2
{N, N, N, N, Y, Y, N} gs cost: 2
DIY: 6 Steps Left
A* GOAP Example – Trace

{Y, Y, N, N, N, Y, N} lurph cost: 3
Valid actions: sfi, cp, gs
Open list:
{Y, N, N, Y, N, Y, N} sfi cost: 3
{Y, N, N, N, Y, Y, N} gs cost: 3
{N, Y, Y, N, N, Y, N} cp cost: 3
{N, Y, N, N, Y, Y, N} gs cost: 3
{Y, Y, N, Y, N, Y, N} lurph cost: 4
{Y, Y, Y, N, N, Y, N} cp cost: 4
{Y, Y, N, N, Y, Y, N} gs cost: 4

Closed list:
{N, N, N, N, N, Y, N} cost: 1
{Y, N, N, N, N, Y, N} lur cost: 2
{N, Y, N, N, N, Y, N} lurph cost: 2
{N, N, N, N, Y, Y, N} gs cost: 2
{Y, Y, N, N, N, Y, N} lurph cost: 3
# A* GOAP Example – Trace

**Valid actions:** luph (duplicate), c, gs

**Open list:**

<table>
<thead>
<tr>
<th>State</th>
<th>sfi cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Y, N, N, Y, N, Y, N}</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>cp cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{N, Y, Y, N, N, Y, N}</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>gs cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{N, Y, N, N, Y, N}</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>sfi cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Y, Y, N, Y, N, Y, N}</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>cp cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Y, Y, Y, N, N, Y, N}</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>gs cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Y, Y, N, Y, N, Y, N}</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>c cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Y, N, Y, Y, N, Y, N}</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>gs cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Y, N, N, Y, Y, N}</td>
<td>4</td>
</tr>
</tbody>
</table>

**Closed list:**

<table>
<thead>
<tr>
<th>State</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{N, N, N, N, N, N, Y, N}</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>lur cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Y, N, N, N, N, Y, N}</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>luph cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{N, Y, N, N, N, Y, N}</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>gs cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Y, Y, N, N, Y, N}</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>luph cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Y, Y, N, N, Y, N}</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>sfi cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Y, Y, N, N, Y, Y, N}</td>
<td>3</td>
</tr>
</tbody>
</table>
A* GOAP Example – Trace

{Y, N, N, N, Y, Y, N} gs cost: 3
Valid actions: luph (dup), sfi (dup)
Open list:
{N, Y, Y, N, N, Y, N} cp cost: 3
{N, Y, N, N, Y, Y, N} gs cost: 3
{Y, Y, N, Y, Y, N} sfi cost: 4
{Y, Y, Y, N, N, Y, N} cp cost: 4
{Y, Y, N, N, Y, Y, N} gs cost: 4
{Y, N, Y, Y, N, Y, Y} c cost: 4
{Y, N, N, Y, Y, Y, N} gs cost: 4

Closed list:
{N, N, N, N, N, Y, N} cost: 1
{Y, N, N, N, N, Y, N} lur cost: 2
{N, Y, N, N, N, Y, N} luph cost: 2
{N, N, N, N, Y, Y, N} gs cost: 2
{Y, Y, N, N, N, Y, N} luph cost: 3
{Y, N, N, Y, N, Y, N} sfi cost: 3
{Y, N, N, N, Y, Y, N} gs cost: 3
**A* GOAP Example – Trace**

<table>
<thead>
<tr>
<th>Open list:</th>
<th>Closed list:</th>
</tr>
</thead>
<tbody>
<tr>
<td>{N, Y, N, N, Y, N} cp cost: 3</td>
<td>{N, N, N, N, Y, N} cost: 1</td>
</tr>
<tr>
<td>{Y, Y, N, Y, N} gs cost: 3</td>
<td>{N, Y, N, N, N, Y, N} lur cost: 2</td>
</tr>
<tr>
<td>{Y, Y, Y, N, Y, N} cp cost: 4</td>
<td>{N, N, N, Y, Y, Y, N} cp cost: 3</td>
</tr>
<tr>
<td>{Y, Y, Y, N, N, Y} gs cost: 4</td>
<td>{N, Y, Y, N, N, Y, Y, N} cp cost: 3</td>
</tr>
<tr>
<td>{Y, N, Y, Y, N, Y, Y} c cost: 4</td>
<td>{N, Y, N, Y, N, Y, N} gs cost: 3</td>
</tr>
<tr>
<td>{Y, N, N, Y, Y, Y} gs cost: 4</td>
<td>{N, Y, N, N, N, Y, Y, N} gs cost: 3</td>
</tr>
<tr>
<td>{N, Y, Y, N, Y, Y} gs cost: 4</td>
<td>{N, N, N, N, Y, Y, N} gs cost: 2</td>
</tr>
<tr>
<td>{N, Y, N, N, N, N} ewh cost: 3</td>
<td>{N, Y, N, N, N, Y, N} lur cost: 2</td>
</tr>
</tbody>
</table>

Valid actions: lur (dup), gs, ewh
A* GOAP Example – Trace

{N, Y, N, N, Y, Y, N} gs cost: 3
Valid actions: lur (dup), cp (dup)
Open list:
{Y, Y, N, Y, N, Y, N} sfi cost: 4
{Y, Y, Y, N, N, Y, N} cp cost: 4
{Y, Y, N, N, Y, Y, N} gs cost: 4
{Y, N, Y, Y, N, Y, Y} c cost: 4
{Y, N, N, Y, Y, Y, N} gs cost: 4
{N, Y, Y, N, Y, Y, N} gs cost: 4
{N, Y, N, N, N, N, N} ewh cost: 3

Closed list:
{N, N, N, N, N, Y, N} cost: 1
{N, Y, N, N, N, Y, N} lur cost: 2
{N, Y, N, N, N, Y, N} lumph cost: 2
{N, N, N, N, Y, Y, N} gs cost: 2
{Y, Y, N, N, N, Y, N} lumph cost: 3
{Y, N, N, Y, N, Y, N} sfi cost: 3
{Y, N, N, N, N, Y, N} gs cost: 3
{N, Y, Y, N, N, Y, N} cp cost: 3
{N, Y, N, N, N, Y, N} gs cost: 3
A* GOAP Example – Trace

{N, Y, N, N, N, N, N, N} ewh cost: 3

Popped Goal!

Closed list:

{N, N, N, N, N, Y, N} cost: 1
{Y, N, N, N, N, Y, N} lur cost: 2
{N, Y, N, N, N, Y, N} luhp cost: 2
{N, Y, N, N, N, Y, N} gs cost: 2
{Y, Y, N, N, N, Y, N} luhp cost: 3
{Y, N, N, Y, N, Y, N} sfi cost: 3
{Y, N, N, N, Y, Y, N} gs cost: 3
{N, Y, Y, N, N, Y, N} cp cost: 3
{N, Y, N, N, Y, Y, N} gs cost: 3

Plan: look up phone number, call pizza, eat with hands
Iterative deepening

- Combine the memory efficiency of depth-first search with the completeness of breadth-first search

- First search for the goal to depth limit of 1
- If no goal found, search again from the beginning to a depth limit of 2
- Continue searching from beginning, increasing depth limit, until goal found or there are no more nodes to search