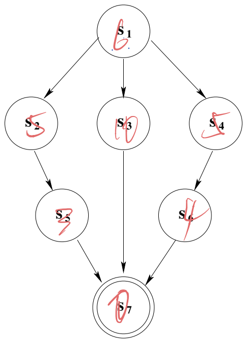
6. search Node - represents states
Σ. set of all possible nodes

AI Planning for Autonomy

Problem Set II: Heuristic Search Continued

1. Consider the following state space S, where $s_0 = s_1$ and $S_G = \{s_7\}$



where actions changing a state s into another state s' are given by the edges. The cost to transition from state s to s' is given by the following table:

s	s'	$c(s, s')$	s	s'	$c(s, s')$
s_1	s_2	2	s_3	s_7	10
s_1	s_3	2	s_4	s_6	1
s_1	s_4	1	s_5	s_7	3
s_2	s_5	2	s_6	s_7	4

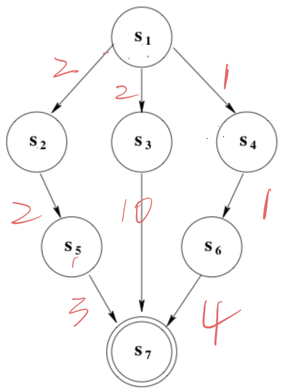
and heuristic estimates for each state:

s	$h_1(s)$	$h_2(s)$	$h_3(s)$
s_1	4	6	6
s_2	3	5	1
s_3	5	10	1
s_4	3	5	5
s_5	2	3	3
s_6	2	4	4
s_7	0	0	0

- Which heuristics are admissible?
- Which are consistent?
- Does any heuristic *dominate* any other?

Domination

Describe the execution of one of the following algorithms in this problem using one of the heuristics above. Fill in a table like the one below, showing the contents of the OPEN and CLOSED lists at the end of each iteration.



- Which heuristics are admissible?

[illegible]
的实际值. for example $h_2(S)$

$h^* = 6$ ∴ As $h(s') \leq h^*(s)$, $s' \in S$

, $h_1(s)$ and $h_3(s)$ are admissible.

$h_2(S)$

all three

$h(S_1) = 6 < h^*(S_1) = cost(S_1, S_3) + cost(S_3, S_7) = 12 \ldots$

- Which are consistent?

consistent $h(s) \leq h^*(s') + c(s')$

$h_1(s) > h_2(S)$

- Does any heuristic *dominate* any other?

~~if $h_2(n) \geq h_1(n)$ for every state n. (both admissible) then h_2 dominates h_1~~

$h_2 = h^*$ and therefore dominates all other admissible heuristics.

$h_1(S_1) < h_3(S_1)$ and $h_3(S_2) < h_1(S_2)$ therefore neither of h_1 and h_3 dominates each other.

Choose one of: A*, WA* ($w = 5$), or Greedy Best-First Search.

	Iteration 1	Iteration 2
OPEN	$n_1 = \langle s_1, 6, 0, nil \rangle$ *	$n_2 = \langle s_2, 5, 2, n_1 \rangle$ $n_3 =$ $n_4 =$
CLOSED		n_1

$n_k = \langle s_j, f(s_j), g(s_j), parent(n_k) \rangle$

- Which is the path returned as a solution?
- Is this the optimal plan? Has the algorithm proved this?

2. Consider an $m \times m$ manhattan grid, and a set of coordinates G to visit in any order.
   - Formulate a state-based search problem to find a tour of all the desired points (i.e. define a state space, applicable actions, transition and cost functions).
   - What is the branching factor of the search?
   - What is the size of the state space in terms of m and G.
   - Define an admissible heuristic function.

$n_i = \langle s, g(n_i) + h(s), g(n_i), n_p \rangle$

A*

I1

OPEN

$n_1 = \langle s_1, 6, 0, nil \rangle$*

CLOSED

I2

OPEN

$n_2 = \langle s_2, \cancel{5}\ 7, 2, n_1 \rangle$

$n_3 = \langle s_3, \cancel{10}\ 12, 2, n_1 \rangle$

$n_4 = \langle s_4, \cancel{5}\ 6, 1, n_1 \rangle$

CLOSED

$n_1 = \langle s_1, 6, 0, nil \rangle$*

I3

OPEN

$n_2 = \langle s_2, \cancel{5}\ 7, 2, n_1 \rangle$

$n_3 = \langle s_3, \cancel{10}\ 12, 2, n_1 \rangle$

$n_5 = \langle s_6, \cancel{4}\ 6, 2, n_4 \rangle$

CLOSED

$n_1 = \langle s_1, 6, 0, nil \rangle$*

$n_4 = \langle s_4, \cancel{5}\ 6, 1, n_1 \rangle$

I_4

OPEN

$n_3 = \langle s_3, \cancel{10}\ 12, 2, n_1 \rangle$

$n_2 = \langle s_2, \cancel{5}\ 7, 2, n_1 \rangle$

$n_6 = \langle s_7, 6, 6, n_5 \rangle$

CLOSED

$n_1 = \langle s_1, 6, 0, nil \rangle^*$

$n_4 = \langle s_4, 5, 1, n_1 \rangle$

$n_5 = \langle s_6, \cancel{4}\ 6, \cancel{1}\ 2, n_4 \rangle$

I_5

OPEN

$n_3 = \langle s_3, 10, 2, n_1 \rangle$

$n_2 = \langle s_2, 5, 2, n_1 \rangle$

CLOSED

$n_1 = \langle s_1, 6, 0, nil \rangle^*$

$n_4 = \langle s_4, 5, 1, n_1 \rangle$

$n_6 = \langle s_7, 6, 6, n_5 \rangle$

$n_5 = \langle s_6, \cancel{4}\ 6, \cancel{1}\ 2, n_4 \rangle$

- Which is the path returned as a solution?

return solution $s_1 \to s_4 \to s_6 \to s_7$

- Is this the optimal plan? Has the algorithm proved this?

Yes, yes.

WA*

I_1

OPEN

$n_1 = \langle s_1, 6, 0, nil \rangle^*$

CLOSED

I_2

$n_2 = \langle s_2, 5, 2, n_1 \rangle$

$n_3 = \langle s_3, 10, 2, n_1 \rangle$

$n_4 = \langle s_4, 5, 1, n_1 \rangle$

$n_1 = \langle s_1, 6, 0, nil \rangle^*$

I_3

I_4

OPEN

$n_2 = \langle S_2, 5, 2, n_1 \rangle$

$n_3 = \langle S_3, 10, 2, n_1 \rangle$

$n_5 = \langle S_6, 4, 1, n_4 \rangle$

CLOSED

$n_1 = \langle S_1, 6, 0, nil \rangle^*$

$n_4 = \langle S_4, 5, 1, n_1 \rangle$

$n_3 = \langle S_3, 10, 2, n_1 \rangle$

$n_2 = \langle S_2, 5, 2, n_1 \rangle$

$n_6 = \langle S_7, 0, 4, n_5 \rangle$

$n_1 = \langle S_1, 6, 0, nil \rangle^*$

$n_4 = \langle S_4, 5, 1, n_1 \rangle$

$n_5 = \langle S_6, 4, 1, n_4 \rangle$

I_5

OPEN

$n_3 = \langle S_3, 10, 2, n_1 \rangle$

$n_2 = \langle S_2, 5, 2, n_1 \rangle$

CLOSED $n_1 = \langle S_1, 6, 0, nil \rangle^*$

$n_4 = \langle S_4, 5, 1, n_1 \rangle$

$n_6 = \langle S_7, 0, 4, n_5 \rangle$

$n_5 = \langle S_6, 4, 1, n_4 \rangle$

- Which is the path returned as a solution?

return solution $S_1 \to S_4 \to S_6 \to S_7$

- Is this the optimal plan? Has the algorithm proved this?

Yes, since n_6 is at a goal state and has the lowest admissible cost estimate of any node in the open list in iteration 4, all

Greedy Best-First Search

other paths from any open node must be longer, given h_2 is both admissible and consistent

	I_1	I_2
OPEN	$n_1 = \langle S_1, 6, 0, nil \rangle^*$	$n_2 = \langle S_2, 5, 2, n_1 \rangle$ $n_3 = \langle S_3, 10, 2, n_1 \rangle$ $n_4 = \langle S_4, 5, 1, n_1 \rangle$
CLOSED		$n_1 = \langle S_1, 6, 0, nil \rangle^*$

	I_3	I_4
OPEN	$n_3 = \langle S_3, 10, 2, n_1 \rangle$ $n_4 = \langle S_4, 5, 1, n_1 \rangle$ $n_5 = \langle S_5, 3, 2, n_2 \rangle$	$n_3 = \langle S_3, 10, 2, n_1 \rangle$ $n_4 = \langle S_4, 5, 1, n_1 \rangle$ $n_6 = \langle S_7, 0, 3, n_5 \rangle$
CLOSED	$n_1 = \langle S_1, 6, 0, nil \rangle^*$ $n_2 = \langle S_2, 5, 2, n_1 \rangle$	$n_1 = \langle S_1, 6, 0, nil \rangle^*$ $n_2 = \langle S_2, 5, 2, n_1 \rangle$ $n_5 = \langle S_5, 3, 2, n_2 \rangle$

- Which is the path returned as a solution?

$S_1 \rightarrow S_2 \rightarrow S_5 \rightarrow S_7$

- Is this the optimal plan? Has the algorithm proved this?

No.

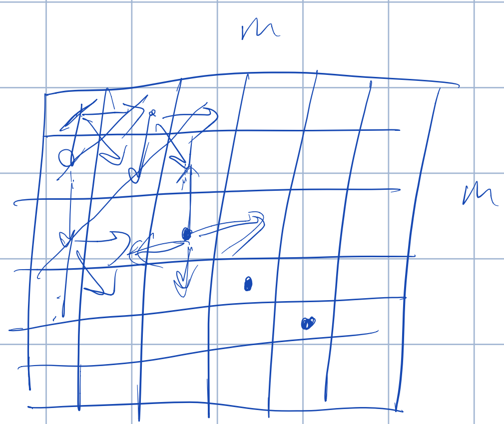
2. Consider an $m \times m$ manhattan grid, and a set of coordinates G to visit in any order.

- Formulate a state-based search problem to find a tour of all the desired points (i.e. define a state space, applicable actions, transition and cost functions).

- What is the branching factor of the search?

- What is the size of the state space in terms of m and G. ?

- Define an admissible heuristic function. $h(\langle x, y, V\rangle) = \max_{(x_g, y_g) \in V} (|x - x_g| + |y - y_g|)$



- Formulate a state-based search problem to find a tour of all the desired points (i.e. define a state space, applicable actions, transition and cost functions).

We have a set of coordinates G to visit in any order
a starting location is Vstart, and a set of Edges

$Pos = \{\langle x, y\rangle \mid x \in [0, m] \wedge y \in [0, m]\}$

$S = \{\langle current_{pos}, pos', flag\rangle \mid current_{pos} \in Pos \wedge pos' \in Pos$ and if $pos' \in G$ $flag = True$, or $False\}$

$S_0 = \langle V_{start}, \langle V_{start}\rangle\}$

$A(\langle Current_{pos}, Pos'\rangle) = \{\langle current_{pos}, Pos'\rangle \mid |current_{pos} pos'| \leq \sqrt{2}\}$

f

$C(a, s) = cost(edge)$

$S_G = \{ \langle current_v, G \rangle \}$

- What is the branching factor of the search?

$b = 4$

- What is the size of the state space in terms of m and G.

$\cancel{m^2}$ $\quad |S| = m^2 \times 2^{|G|}$

S whole state space $= \{ \langle x, y, V \rangle \mid x \in \underbrace{\{0, \ldots, m-1\}}_{M},\ y \in \{0, \ldots, m-1\},\ V \subseteq G \}$

$A(s) = \{ (\Delta x, \Delta y) \mid x + \Delta x \in M,\ y + \Delta y \in M,\ |\Delta x| + |\Delta y| \leq 1 \}$

$T(s, a) = \langle x + \Delta x, y + \Delta y, V$ if $s' \notin G$

$V \cup \{s'\}$ if $s' \in G \}$

$c(a, s) = 1$