DFS
BFS
Depth Limited
Iterative Deepening

Artificial Intelligence **(Harvard CS182, Fall 2015)**

Lectures Notes on Search

Alexander Rush

1 Board 0

- Start with XKCD (have it on when they come in)
  - Show pacman searches

2 Board 0

Start with XKCD

3 Board 1

- Completeness; Is the algorithm guaranteed to find a solution? (infinite paths)
- Optimality; Is the algorithm guaranteed to find the optimal solution?
- Time:
- Space:

4 Board 2

Redraw table:

5 Board 3

Summary

- Algorithms
  - Uniform-Cost Search
- Heuristics
- Informed Search
  - Greedy Best-First Search
  - A* Search

6 Board 4

Uniform-cost search

Idea: Fix BFS optimality

- Be conservative based on path-cost (as opposed to depth)
- Expand in path cost order
- Utilize a priority queue

Priority function:

$$f : \mathcal{P} \mapsto \mathbb{R}^+$$

7 Board 5

Recall

c is step cost

g is path cost

For UCS, we set this function to be $f(p) \triangleq g(p)$, i.e. the cost of the partial path.

8 Board 6

iter	p	f	s	frontier (p)	explored
0	-		-	[A:0]	{}
1	A	0	A	[A:B:2, A:D:3, A:E:5]	{A}
2	A:B	2	B	[A:D:3, A:E:5]	{A, B}
3	A:D	3	D	[A:E:5, A:D:C:7]	{A, B, D}
4	A:E	5	E	[A:D:C:8]	{A, B, D, E}
5	A:D:C	C	8	-	-

Because UCS expands paths in cost order and costs are non-negative, each path it expands must be at least as costly as all previous paths. This means that if UCS finds a goal node it must be optimal. With a few further assumptions (see AIMA) we can also show it is complete. AIMA also includes a description of the time- and space-complexity that uses the optimal score and smallest path cost ϵ.

9 Board 7

- Completeness: Yes.
- Optimality: Yes.
- Time-Complexity: $O(b^{1+\lfloor C^*/\delta \rfloor})$ (smallest step cost)
- Space-Complexity: $O(b^{1+\lfloor C^*/\delta \rfloor})$

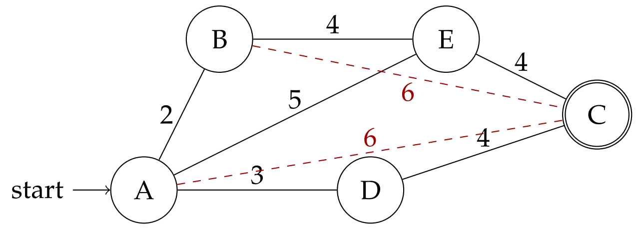

Figure 1: An example heuristic: straight-line distance.

10 Board 8

Uninformed vs. Informed

Up to this point we assumed that we have no further knowledge into the nature of the search model. With this requirement there is no hope to find an optimal solution without first expanding all path with cost $< C*$ (as in UCS). However it in practice we often have more insight into the structure of the problem, for instance roughly how close we are to a goal state. When we have this information we can instead run informed search.

11 Board 9

Heuristic: Guess at future cost.

- number of dots left - best translation for each word - shortest line cost

---

Heuristic	$h : \mathcal{S} \mapsto \mathbb{R}$	Estimate of cost from state s to a goal state.

---

12 Board 10

13 Board 11

Greedy Best-First Search

- Be greedy on heuristic!
- Expand in heuristic cost order
-

Same setup as UCS but with

$$f(p) = h(p_s)$$

14 Board 12

- Completeness: No.
- Optimality: No.
- Time-Complexity: $O(b^m)$
- Space-Complexity: $O(b^m)$

15 Board 13

- Combine UCS and Greedy Best-first

$$f(p) \triangleq g(p) + h(p_s)$$

- A* search.

16 14

iter	p	s	frontier (p)	explored
0	-	-	[A=0+6]	{}
1	A	A	[A:D=3+4, A:B=2+6, A:E=5+4]	{A}
2	A:D	D	[A:D:C:7+0, A:B=2+6, A:E=5+4]	{A, D}
3	A:D:C	C	-	-

17 Board 14

- Completeness: Yes.
- Optimality: ?.

18 Board 15

Definition 1 *An **admissible** heuristic never overestimates the cost to a goal state, i.e.*

$$g(p) + h(p_s) \leq g(\hat{p})$$

where $\hat{p} \in \mathcal{Q}$ is any solution path with p as a partial path.

Definition 2 *A **consistent** heuristic obeys the property that for any state s,*

$$h(s) \leq c(s, a) + h(\textsc{Res}(s, a))$$

for all actions $a \in \textsc{Act}(s, a)$.

Consistency is a stronger condition and implies admissibility. However, there are cases where just admissibility is sufficient so we define both.

19 Board 16

Theorem 1 *A^* with consistent heuristic is optimal.*

Proof:

We first prove that the A^* estimated cost increase at each expansion, and then use this to show that the first expansion of any state is optimal. This implies that the first time we expand a goal state is optimal.

-The values of expanded paths are non-decreasing.

Assume we have expanded a path p to a path p' with action a, we have:

$$\begin{aligned} f(p') &= g(p') + h(p'_s) = g(p) + c(p_s, a) + h(p'_s) \\ &\geq g(p) + h(p_s) = f(p), \end{aligned}$$

Where we have directly used the definition of consistency for the inequality.

20 Board 17

- Whenever A^* expands a path, the optimal path to that node has already been found.

We will prove this by contradiction.

Say we have expand path p before the optimal path to this state q. This means that $p_{\text{last}} = q_{\text{last}}$ and $g(q) < g(p)$, this implies $f(q) < f(p)$.

Since we expanded p first, this also mean there is some partial path of q called q' in the frontier that currently has $f(p) \leq f(q')$ (or else it would have been expanded first).

However by Property (1) we know that as a partial path of q, $f(q') \leq f(q)$.

Together this gives a contradiction:

$$f(q) < f(p) \leq f(q') \leq f(q)$$

$\square$

21 Board 18

- Completeness: Yes.
- Optimality: Yes.
- Time-Complexity: $O(b^{\Delta})$ (depends on the absolute error of h, see AIMA3e, p.98)
- Memory-Complexity: $O(b^{\Delta})$ (graph search)

22 Board 19

22.1 Heuristics for Path-Finding

Now let's return to the heuristics we discussed above. How can we check whether they satisfy the properties necessary for A* search? First consider straight-line distance. We would like to show that for any state s and valid action a:

$$h(s) \leq c(s,a) + h(\textsc{Res}(s,a))$$

and let's call the resulting state $s' = \textsc{Res}(s,a)$.

We have defined our problem such that the cost of the action is the distance between s and s', i.e. $d(s,s')$. And we have defined our heuristic $h(s)$ as the distance to our goal $h(s) = d(s,\mathrm{C})$. So for consistency we need to show that:

$$d(s,\mathrm{C}) \leq d(s,s') + d(s',\mathrm{C})$$

However this is just the triangle inequality! Since this holds for Euclidean distance, we have consistency for this problem.

23 Board 20

23.1 Comparing Heuristic Functions

$$\Delta(p,s) = g(\hat{p}^*) - (g(p) + h(p_s))$$

Definition 3 *Given two consistent heuristics h and h', we say h **dominates** h' if for all $s \in \mathcal{S}$, $h(s) \geq h'(s)$.*

24 Board 21

- Write down the explicit constraints that are required for the problem.
- Select a subset of these constraints to “relax”.
- Calculate the optimal solution without these constraints and use as heuristic.

taking $h(s) = \max\{h_1(s), h_2(s)\}$.