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

- Homework extension Monday at midnight (HW questions)
- Section Lyman 425
- Laptop policy
- name tags...

3 board 0.5

- mistake n depth $n + 1$
- Depth m

4 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:

5 Board 2

Redraw table:

6 Board 3

Summary

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

7 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:

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

8 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.

9 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.

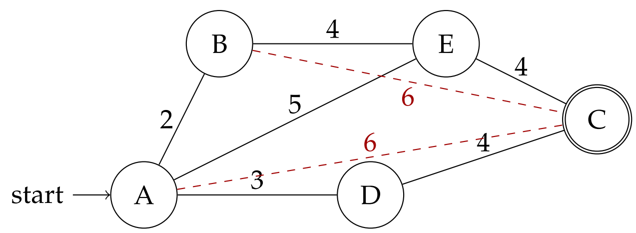

Figure 1: An example heuristic: straight-line distance.

10 Board 7

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

11 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.

12 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.

---

13 Board 10

14 Board 11

Greedy Best-First Search

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

Same setup as UCS but with

$$f(p) = h(p_{\text{Last}})$$

15 Board 12

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

16 Board 13

- Combine UCS and Greedy Best-first

$$f(p) \triangleq g(p) + h(p_{\text{Last}})$$

- A* search.

17 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	-	-

18 Board 14

- Completeness: Yes.
- Optimality: ?.

19 Board 14.5

Let p be $(s_0, a_0), \ldots, (s_{n-1}, a_{n-1})$
Define a (strict) subpath of $p \in \mathcal{P}$ as $(s_0, a_0), \ldots, (s_{n'-1}, a_{n'-1})$ for $n' \leq n$.
By our definition of cost $g(p') \leq g(p)$.

20 Board 15

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

$$g(p) + h(p_{\text{Last}}) \leq g(\hat{p})$$

where $\hat{p} \in \mathcal{Q}$ is any solution with p as a sub-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 $a \in \textsc{Act}(s)$.

Consistency $\Rightarrow$ admissibility. However, there are cases where just admissibility is sufficient so we define both.

21 Board 16

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

Proof:

The values of expanded paths are non-decreasing.
Assume expansion of path p leads to p' with action a:

$$\begin{aligned} f(p') &= g(p') + h(p'_{\text{Last}}) = g(p) + c(p_{\text{last}}, a) + h(p'_{\text{last}}) \\ &\geq g(p) + h(p_{\text{Last}}) = f(p), \end{aligned}$$

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

22 Board 17

After A^* expands a path, the optimal path to that state has already been expanded.
By contradiction.
Assume we have expand path $p \in \mathcal{P}$ before the optimal path to this state $q \in \mathcal{P}$.

Implies $p_{\text{last}} = q_{\text{last}}$ and $g(q) < g(p)$, this implies $f(q) < f(p)$.

Since we expanded p first, this also mean there is some sub-path of q (called q') in frontier.

Also we know that $f(p) \leq f(q')$ (or else it would have been expanded first).

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

Together this gives a contradiction:

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

□

23 Board 18

- Completeness: Yes. (graph-search requires both, tree-search just need admissable)
- 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)

24 Board 19

24.1 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:

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

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:

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

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

25 Board 20

25.1 Comparing Heuristic Functions

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

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

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