CS540 Introduction to Artificial Intelligence
Lecture 16

Young Wu
Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

July 9, 2020

Uniformed vs. Informed Search

Motivation

- Uninformed search means only the goal G and the successor functions s' are given.
- Informed search means which non-goal states are better is also known.

Heuristic Diagram

Motivation

$g + h \approx$ total cost.

heuristic approximation

$$\underbrace{I \longrightarrow S}_{g(s)} \; \underbrace{\xrightarrow{\approx h(s)} G}_{h^*(s)}$$

cost so far $= g(s)$

? $h^*(s) \rightarrow$ remaining cost.

Uniform Cost Search

Description

- Expand the vertices with the lowest current path cost $g(s)$ first.
- It is BFS with a priority queue based on $g(s)$.
- It is equivalent to BFS if $c = 1$ is constant on all edges.
- It is also called Dijkstra's Algorithm.

577

UCS Example 1

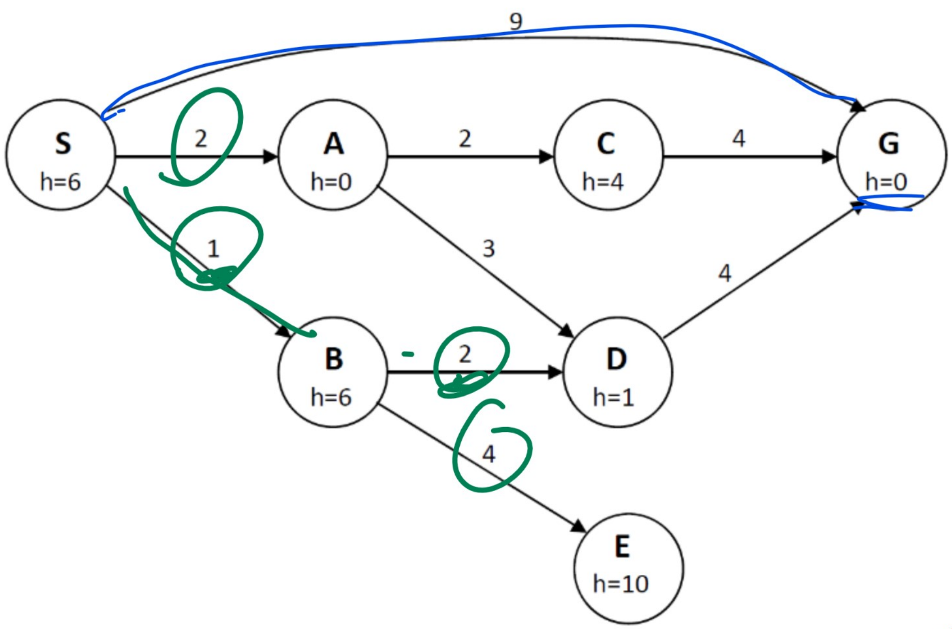
Quiz

- Spring 2017 Midterm Q1
- Given the following adjacency matrix. Find UCS expansion path.

–	S	A	B	C	D	E	G
S	$h = 6$	2	1	–	–	–	9
A	–	$h = 0$	–	2	3	–	–
B	–	–	$h = 6$	–	2	4	–
C	–	–	–	$h = 4$	–	–	4
D	–	–	–	–	$h = 1$	–	4
E	–	–	–	–	–	$h = 10$	–
G	–	–	–	–	–	–	$h = 0$

UCS Example 1 Diagram

Quiz



S B D G solution

opt.

Q ~~S~~ ~~B_S~~ ~~A_S~~, ~~D_B~~, ~~C_A~~ ~~D_A~~ ~~E_B~~, ~~G_D~~, G_C G_S G_D

g 0 1 2 1+2 2+2 2+3 1+4 3+4 4+4 9 5+4

expansion path S, B, A, D, C, D, E, G.

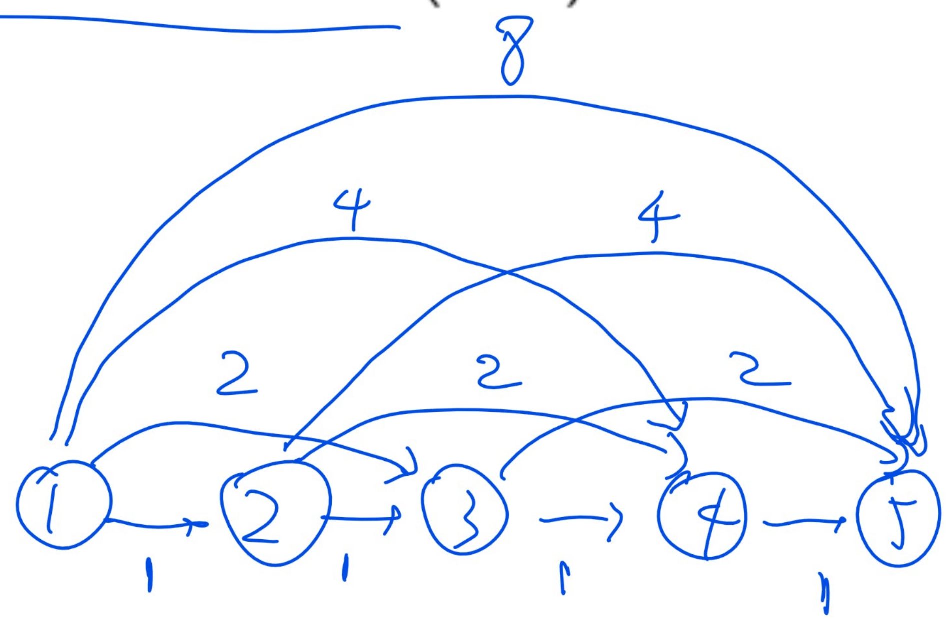
UCS Example 2

Quiz

Q5 (last)

- Given that cost from state i to j is 2^{j-i-1} for $j > i$. The initial state is 1 and goal state is 5. What is a vertex expansion sequence if Uniform Cost Search (UCS) is used?
- A: 1, 5
- B: 1, 2, 3, 4, 5
- C: 1, 2, 3, 4, 4, 5
- D: 1, 2, 3, 3, 4, 4, 5
- E: 1, 2, 3, 3, 4, 4, 4, 5

4



1, 2

UCS Example 2 Diagram

Quiz

Q	$\cancel{1}$	$\cancel{2}$	$\cancel{3_1}$	$\cancel{4_1}$	5_1	$\cancel{3_2}$	$\cancel{4_2}$	5_2	,
g	0	1	2	4	8	1+1	1+2	1+4	

$\cancel{4_3}$	$\cancel{5_3}$	$\cancel{4_3}$	5_3	,	5_4	,	5_4
2+1	2+2	2+1	2+2		3+1		4+1

record a video.

Uniform Cost Search

Algorithm

- Input: a weighted digraph (V, E, c), initial states I and goal states G.
- Output: a path from I to G.
- EnQueue initial states into a priority queue Q. Here, Q is ordered by $g(s)$ for $s \in Q$.

$$Q = I$$

- While Q is not empty and goal is not deQueued, deQueue Q and enQueue its successors.

$$s = Q_{(0)} = \arg\min_{s \in Q} g(s)$$

$$Q = Q + s'(s)$$

Uniform Cost Search Performance

Discussion

- UCS is complete.
- UCS is optimal with any c.

MQ, MP

PS → bugs.

↳ wait for Sat.

Best First Greedy Search

Description

- Expand the vertices with the lowest heuristic cost $h(s)$ first.
- Use a priority queue based on $h(s)$.

A $g(s) + h(s)$

$\approx \boxed{h(s)}$

I → S → G

$\boxed{g(s)}$

UCS

$h^*(s)$

Greedy.

Greedy Example 1

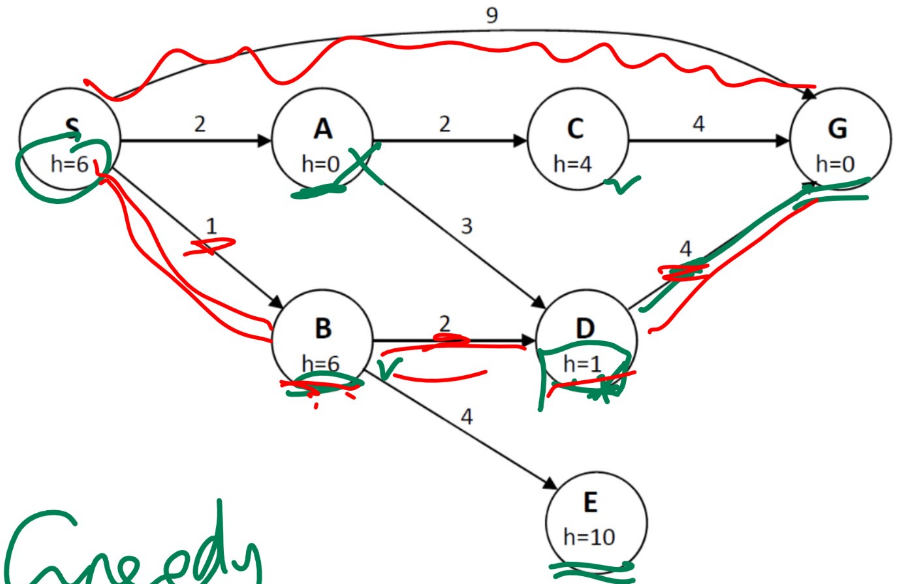
Quiz

- Given the following adjacency matrix. Find Greedy Search expansion path.

–	S	A	B	C	D	E	G
S	$h = 6$	2	1	–	–	–	9
A	–	$h = 0$	–	2	3	–	–
B	–	–	$h = 6$	–	2	4	–
C	–	–	–	$h = 4$	–	–	4
D	–	–	–	–	$h = 1$	–	4
E	–	–	–	–	–	$h = 10$	–
G	–	–	–	–	–	–	$h = 0$

Greedy Example 1 Diagram

Quiz



UCS

SBDG lowest cost

7

Greedy

Q: ~~A~~$_S$ ~~G~~$_S$ D_A C_A ~~S~~ B_S

h: 0 0 1 4 6 6

expansion. S A G

→ not a solution

solution ~~SG~~

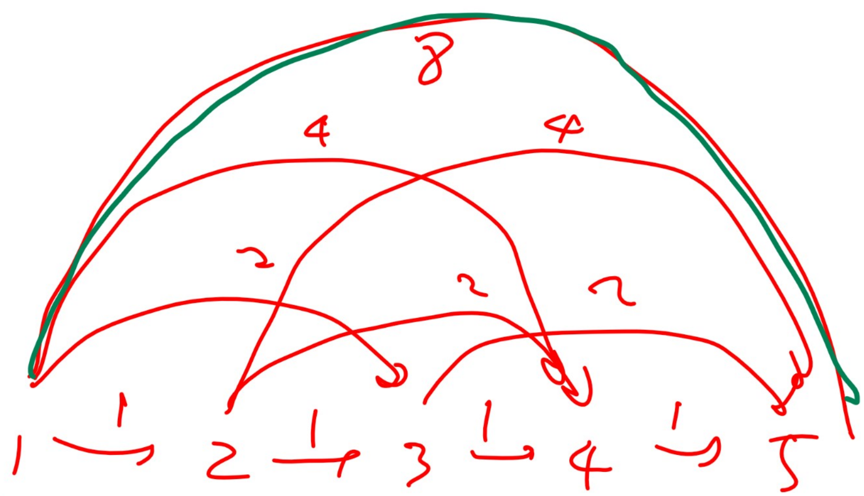
Greedy Example 2

Quiz

Q1

- Given that cost from state i to j is 2^{j-i-1} for $j > i$. The heuristic is $h(i) = 5 - i$, The initial state is 1 and goal state is 5. What is a vertex expansion sequence if Best First Greedy Search is used?
- A: 1, 5
- B: 1, 2, 3, 4, 5
- C: 1, 2, 3, 4, 4, 5
- D: 1, 2, 3, 3, 4, 4, 5
- ~~E~~: 1, 2, 3, 3, 4, 4, 4, 5

UCS



h = 4 = 16   3   2   1   0

Q: ~~1~~ ~~5~~ 4 3 2

h: 4 0 1 2 3

$\boxed{1, 5}$

Greedy Example 2 Diagram

Quiz

Best First Greedy Search

Algorithm

- Input: a weighted digraph (V, E, c), initial states I and goal states G, and the heuristic function $h(s), s \in V$.
- Output: a path from I to G.
- EnQueue initial states into a priority queue Q. Here, Q is ordered by $h(s)$ for $s \in Q$.

$$Q = I$$

- While Q is not empty and goal is not deQueued, deQueue Q and enQueue its successors.

$$s = Q_{(0)} = \arg\min_{s \in Q} h(s)$$

$$Q = Q + s'(s)$$

Best First Greedy Search Performance

Discussion

- Greedy is incomplete.
- Greedy is not optimal.

A Search

Description

expansion path = list of deQed → always ends with Goal

popped

- Expand the vertices with the lowest total cost $g(s) + h(s)$ first.
- Use a priority queue based on $g(s) + h(s)$.
- A stands for Always be optimistic?

A Search Example 1

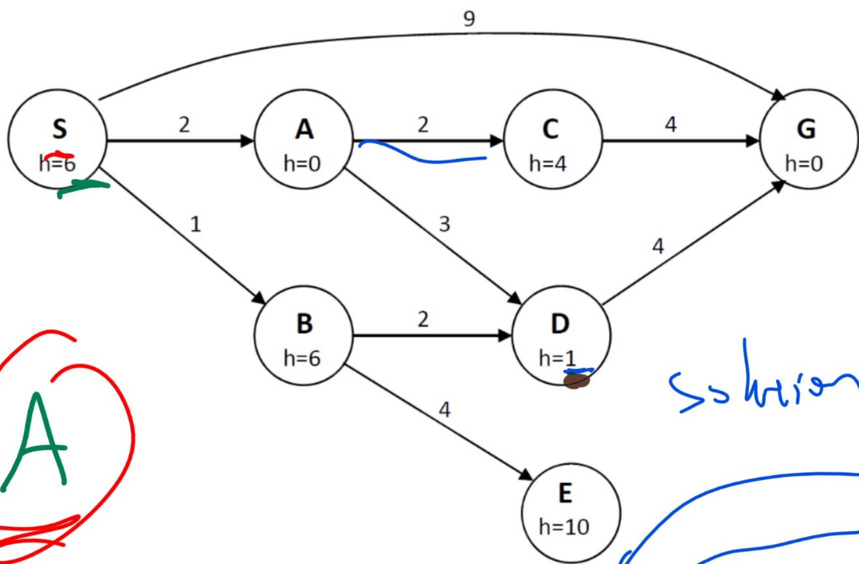
Quiz

- Given the following adjacency matrix. Find A Search expansion path.

–	S	A	B	C	D	E	G
S	$h = 6$	2	1	–	–	–	9
A	–	$h = 0$	–	2	3	–	–
B	–	–	$h = 6$	–	2	4	–
C	–	–	–	$h = 4$	–	–	4
D	–	–	–	–	$h = 1$	–	4
E	–	–	–	–	–	$h = 10$	–
G	–	–	–	–	–	–	$h = 0$

A Search Example 1 Diagram

Quiz



A

expansion path: S, A, D, B, D, G

solution SBDG

Q	A_S	~~D_S~~	~~D_A~~	~~S_B~~	~~B_S~~	~~C_D~~	C_A	G_S	G_D	E_B
g	$\boxed{2}$	1+2	2+3	0	1	3+4	2+2	9	5+4	1+4
h	0	1	1	6	6	0	4	0	0	10
g+h	2	4	6	6	7	7	8	9	9	15

back tracker

	S	A	B	C	D	G
Initial	S		S	A	~~A~~ B	~~S~~ D

A Search

Algorithm

- Input: a weighted digraph (V, E, c), initial states I and goal states G, and the heuristic function $h(s), s \in V$.
- Output: a path from I to G.
- EnQueue initial states into a priority queue Q. Here, Q is ordered by $g(s) + h(s)$ for $s \in Q$.

$$Q = I$$

- While Q is not empty and goal is not deQueued, deQueue Q and enQueue its successors.

$$s = Q_{(0)} = \arg\min_{s \in Q} g(s) + h(s)$$

$$Q = Q + s'(s)$$

A Search Performance

Discussion

- A is complete.
- A is not optimal.

A Star Search

Description

h

- $A^{\star}$ search is A search with an admissible heuristic.

Admissible Heuristic

Definition

- A heuristic is admissible if it never over estimates the true cost.

true

$$\boxed{0 \leqslant h(s) \leqslant h^{\star}(s)}$$

A search h $\longrightarrow$ A^{*} optimal

Dominated Heuristic

Definition

- One heuristic, h_1, is dominated by another, h_2, if:

$$h_1(s) \leqslant h_2(s) \leqslant h^\star(s), \ \forall s \in S$$

h2 better h1

- If h_2 dominates h_1, then h_2 is better than h_1 since $A^\star$ using h_1 expands at least as many states (or more) than $A^\star$ using h_2.
- If h_2 dominated h_1, $A^\star$ with h_2 is better informed than $A^\star$ with h_1.

Admissible Heuristic 8 Puzzle Example

Quiz

C = 1

- Which ones (select multiple) of the following are admissible heuristic function for the 8 Puzzle?

h(goal) = 0

- A: $h(s)$ = number of tiles in the wrong position.

admissible not good

- $\boxed{\text{B: } h(s) = 0.}$ ✓ UCS

worst possible admissible h.

- ~~C: $h(s) = 1$.~~ h(Goal) = 0 ≠ 1
- D: $h(s)$ = sum of Manhattan distance between each tile and its goal location. h^* ✓ best
- E: $h(s)$ = sum of Euclidean distance between each tile and its goal location. ✓

Admissible Heuristic General Example 1

Quiz

$s \longrightarrow$ any config of the game. puzzle.

- Which ones (select multiple) of the following are admissible heuristic function?
- $0 \leq$ A: $h(s) = h^\star(s)$. $\leq h^*$
- B: $h(s) = \cancel{\max\{2, h^\star(s)\}}$. $h(\text{goal}) = 2 \neq 0$
- C: $h(s) = \min\{2, h^\star(s)\}$. $\leq h^*$
- D: $\cancel{h(s) = h^\star(s)} - 2$. $= -2$
- E: $\cancel{h(s)} = \sqrt{h^\star(s)}$.

$\sqrt{h^* > 1} < h^*$

$h^* > 1$

$h^* = 0.25$

$\boxed{\sqrt{h^*} = h = 0.5 > h^*}$

A Star Search Example 2

Quiz

Q2

- Given that cost from state i to j is 2^{j-i-1} for $j > i$. How many of the following heuristic functions are admissible? For $i = 1, 2, 3, 4, 5$:

$$h(i) = 5 - i$$

$$h(i) = \sqrt{5 - i}$$

$$h(i) = \log_2(6 - i)$$

$$h(i) = 1 - \mathbb{1}_{i=5}$$

$$h(i) = 0$$

$0 \le h \le h^*$

$\begin{cases} 1 & \text{if } i = 5 \\ 0 & \text{if } i \ne 5 \end{cases}$

- A: 1, B: 2, C: 3, D: 4, E: 5

A Star Search Example 2

Quiz

Q3 (last), A

- Given that cost from state i to j is 2^{j-i-1} for $j > i$. Which one of the following heuristic functions is not dominated (among the admissible ones)? For $i = 1, 2, 3, 4, 5$:
- A: $h_a(i) = 5 - i$ = h^*
- B: $h(i) = \sqrt{5 - i}$
- C: $h(i) = \log_2(6 - i)$
- D: $h(i) = 1 - \mathbb{1}_{i=5}$
- E: $h(i) = 0$

E is dom by D.

$0 \le h_1 \le h_2 \le h^*$

$h_1 = h^*$

h_1 dominated by h_2

h_2 dominates h_1

A Star Search with Revisit, Part I

Algorithm

- Input: a weighted digraph (V, E, c), initial states I and goal states G, and the heuristic function $h(s), s \in V$.
- Output: a path with minimum cost from I to G.
- EnQueue initial states into a priority queue Q. Here, Q is ordered by $g(s) + h(s)$ for $s \in Q$.

$$Q = I$$
$$g(I) = 0$$
$$g(s) = \infty, \text{ for } s \notin I$$

- Initialize the list of visited vertices, P.

$$P = \varnothing$$

A Star Search with Revisit, Part II

Algorithm

- While Q is not empty and goal is not deQueued, deQueue Q, put it on P and enQueue its successors to Q, and update the cost functions.

$$s = Q_{(0)} = \arg\min_{s \in Q} g(s) + h(s)$$

$$P = P + s$$

$$Q = Q + s'(s), \text{ update } g(s') = \min\left\{g(s'), g(s) + c(s, s')\right\}$$

A Search Performance

Discussion

- $A^{\star}$ is complete.
- $A^{\star}$ is optimal.

Iterative Deepening A Star Search

Discussion

- $A^{\star}$ can use a lot of memory.
- Do path checking without expanding any vertex with $g(s) + h(s) > 1$.
- Do path checking without expanding any vertex with $g(s) + h(s) > 2$.
- ...
- Do path checking without expanding any vertex with $g(s) + h(s) > d$.

Iterative Deepening A Star Search Performance

Discussion

- IDA* is complete.
- IDA* is optimal.
- IDA* is more costly than $A^\star$.

Beam Search

Discussion

- Version 1: Keep a priority queue with fixed size k. Only keep the top k vertices and discard the rest.
- Version 2: Only keep the vertices that are at most ε worse than the best vertex in the queue. ε is called the beam width.

Beam Search Performance

Discussion

- Beam is incomplete.
- Beam is not optimal.