CS540 Introduction to Artificial Intelligence
Lecture 19

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

July 28, 2021

Course Evaluation

Admin

- Midterm grades individual adjustment (please email or private Piazza me before the final exam):

1. Incorrect ID: Wisc email ID, not 10-digit campus ID, not CS email ID.
2. Typos that cause the expressions to be not evaluatable.
3. Personal issues: everyone is allowed to use P6 grade (out of 8, but plus 2 for really good projects) to replace one part of the midterm (and final) (out of 10).

- M9 is ready, M10 not ready, P4 is ready (note the difference of maze format from last year, if you submitted before yesterday, you need to resubmit).

Remind Me to Start Recording

Admin

- The messages you send in chat will be recorded: you can change your Zoom name now before I start recording.
- We will play the coordination game to post exam questions again: no communication in chat after I start recording.

Coordination Game

Admin

Q2

- You are not allowed to discuss anything about this question in the public chat. There will be around 10 new questions on the final exam. I will post n of them before the exam (probably next Tuesday):
- A: $n = 0$.
- B: $n = 1$ if more than 50 percent of you choose B.
- C: $n = 2$ if more than 75 percent of you choose C.
- D: $n = 3$ if more than 98 percent of you choose D.
- E: $n = 0$.
- I will repeat this question a second time. If you fail to coordinate both times, I will not post any of the new questions.

Coordination Game Repeat

Admin

Q3

- You are not allowed to discuss anything about this question in the public chat. There will be around 5 new questions on the final exam. I will post n of them before the exam (probably next Tuesday):
- A: $n = 0$.
- B: $n = 1$ if more than 50 percent of you choose B.
- C: $n = 2$ if more than 75 percent of you choose C.
- D: $n = 3$ if more than 98 (95) percent of you choose D.
- E: $n = 0$.

A Search Example 1

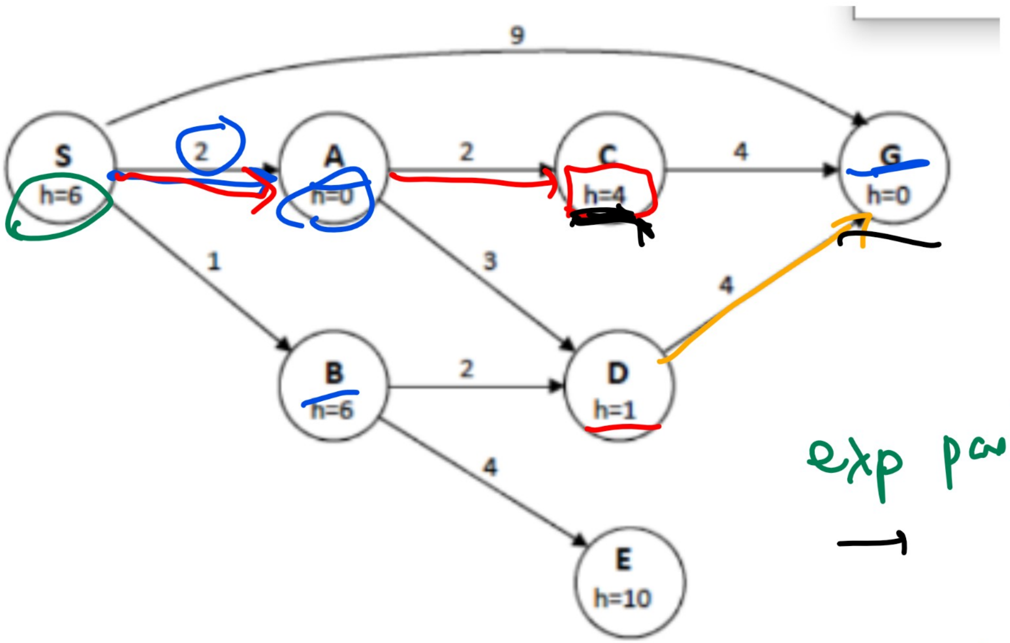
Quiz

- 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 Search not optimal

g cost so far

h estimated future cost

exp path → S, A, D_A, B, D_B, G_D

drop G_D

remove

PQ	A	D_B	S	D_A	B	G_D	C_A	G_S	E
g	2	3	0	5	1	7	4	9	5
h	0	1	6	1	6	0	4	0	10
g+h	2	4	6	6	7	7	8	9	15

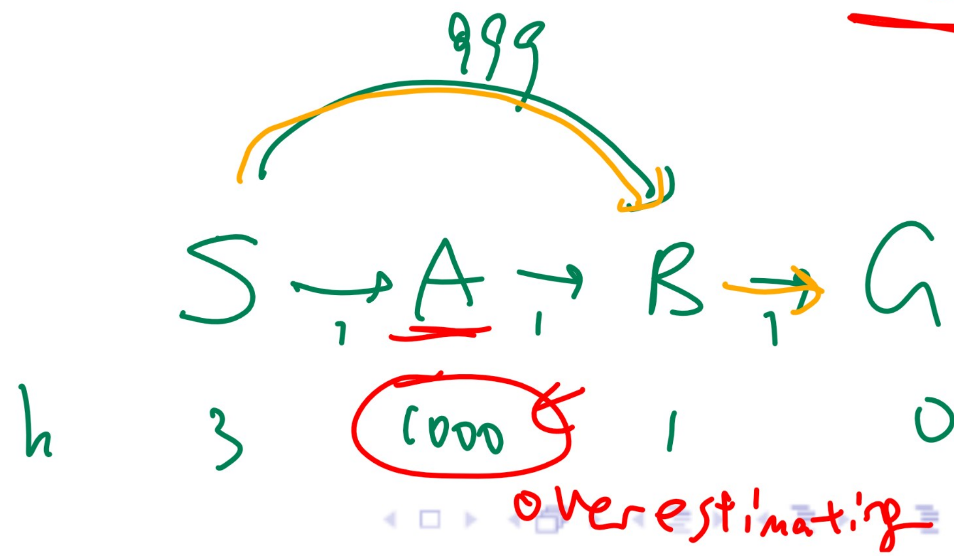
A Search Example 2

Quiz

- Find A search expansion path

–	S	A	B	G
S	$h = 3$	1	999	–
A	–	$h = 1000$	1	–
B	–	–	$h = 1$	1
G	–	–	–	$h = 0$

- A: S, A, B, G
- B: S, B, G
- C: S, B, A, G
- D: S, B, A, B, G



A Search Example 2 Diagram

Quiz

A Search Performance

Discussion

- A is complete.
- A is not optimal.

Admissible Heuristic

Definition

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

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

Dominated Heuristic

Definition

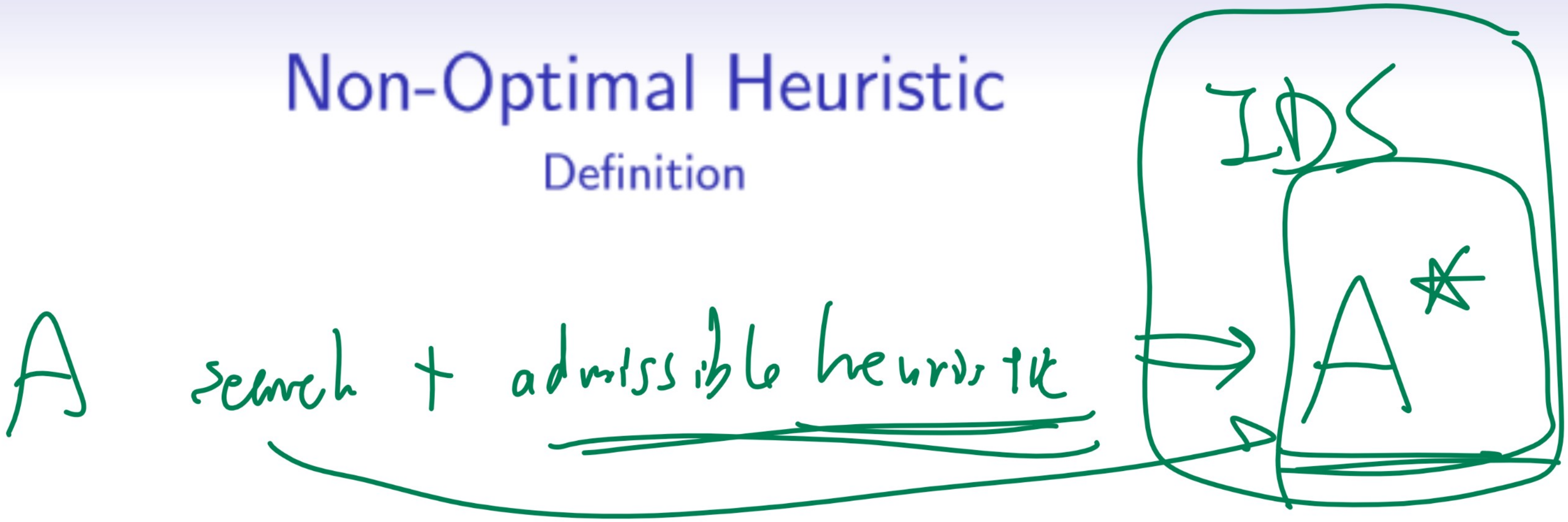
- One heuristic, h_1, is dominated by another, h_2, if:

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

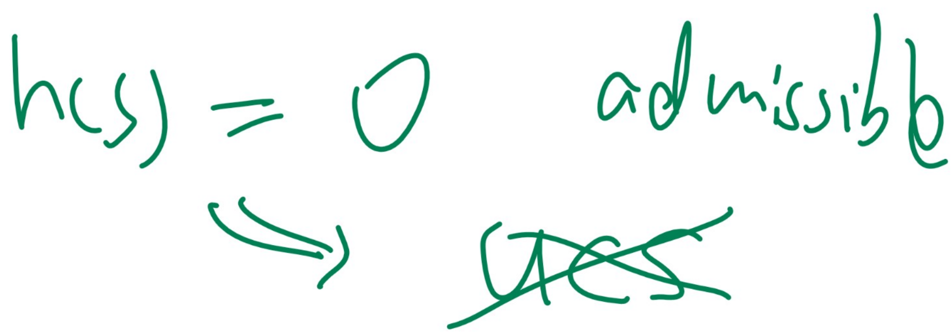
- If h_2 dominates h_1, then h_2 is better than h_1 since $A^{\star}$ using h_1 expands at least as many states (or more) than $A^{\star}$ using h_2.
- If h_2 dominated h_1, $A^{\star}$ with h_2 is better informed than $A^{\star}$ with h_1.

Non-Optimal Heuristic

Definition



- If optimality is not required and a satisfying solution is acceptable, then the heuristic should be as close as possible, either under or over, to the actual cost.
- This results in fewer states being expanded compared to using poor but admissible heuristics.



Admissible Heuristic General Example 1

Quiz

- Which ones (select multiple) of the following are admissible heuristic function?
- A: $h(s) = h^{\star}(s) \cdot 2$.
- B: $h(s) = \sqrt{h^{\star}(s)}$.
- C: $h(s) = h^{\star}(s) + 1$.
- D: $h(s) = \min\{1, h^{\star}(s)\}$.
- E: $h(s) = h^{\star}(s) \cdot \frac{1}{2}$.
- F: $h(s) = h^{\star}(s)^{2}$.
- G: $h(s) = \max\{1, h^{\star}(s)\}$.
- H: $h(s) = h^{\star}(s) - 1$.

Admissible Heuristic General Example 1 Again

Quiz

- Which ones (select multiple) of the following are admissible heuristic function?
- A: $h(s) = h^{\star}(s) \cdot 2$.
- B: $h(s) = \sqrt{h^{\star}(s)}$.
- C: $h(s) = h^{\star}(s) + 1$.
- D: $h(s) = \min\{1, h^{\star}(s)\}$.
- E: $h(s) = h^{\star}(s) \cdot \dfrac{1}{2}$.
- F: $h(s) = h^{\star}(s)^{2}$.
- G: $h(s) = \max\{1, h^{\star}(s)\}$.
- H: $h(s) = h^{\star}(s) - 1$.

A Search Performance

Discussion

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

Iterative Deepening A Star Search

Discussion

IDS

- $A^{\star}$ 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 $g(s) + h(s) > d$.

IDA

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

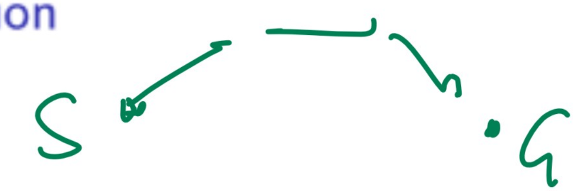
- Beam is incomplete.
- Beam is not optimal.

Traveling Salesperson Example

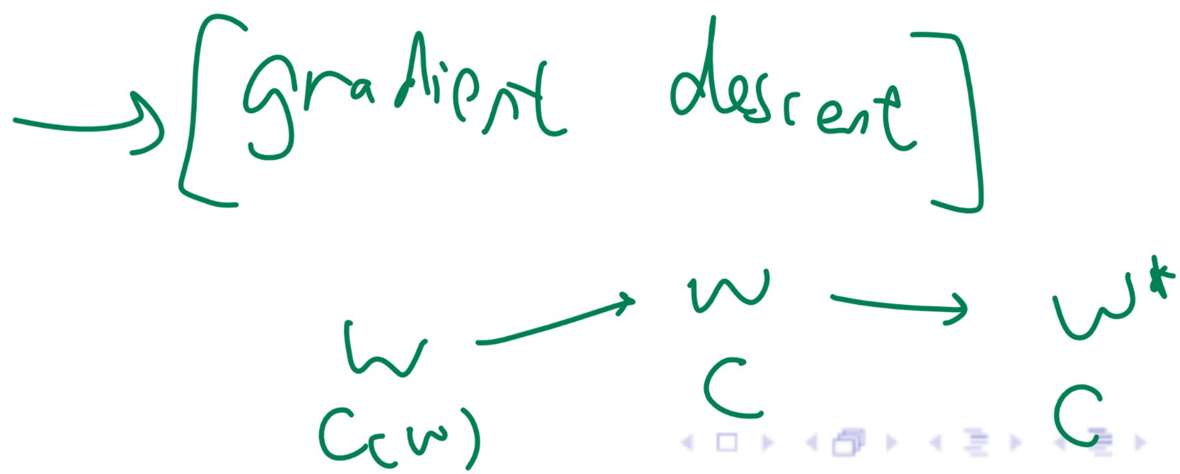
Motivation

Local Search

Motivation



- Local search is about searching through a state space by iteratively improving the cost to find an optimal or near-optimal state.
- The successor states are called the neighbors (sometimes move set).
- The assumption is that similar (nearby) solutions have similar costs.



Hill Climbing (Valley Finding)

Description

- Start at a random state.
- Move to the best neighbor state (one of the successors).
- Stop when all neighbors are worse than the current state.
- The idea is similar to gradient descent.

Boolean Satisfiability Example 1

Quiz

- Assume all variables A, B, C, D, E are set to True. How many of the following clauses are satisfied?
- $A \vee \neg B \vee C$
- $\neg A \vee C \vee D$
- $B \vee D \vee \neg E$
- $\neg C \vee \neg D \vee \neg E$
- $\neg A \vee \neg C \vee E$

Boolean Satisfiability Example 2

Quiz

- Assume all variables A, B, C, D, E are set to True. Which one of the variables should be changed to False to maximize the number of clauses satisfied?
- $A \vee \neg B \vee C$
- $\neg A \vee C \vee D$
- $B \vee D \vee \neg E$
- $\neg C \vee \neg D \vee \neg E$
- $\neg A \vee \neg C \vee E$

Boolean Satisfiability Example 3

Quiz

- Assume all variables A, B, C, D, E are set to True. Which one of the variables should be changed to False to maximize the number of clauses satisfied?
- $\neg A \vee \neg B \vee \neg E$
- $\neg A \vee \neg B \vee \neg D$
- $\neg A \vee \neg C \vee \neg D$
- $\neg B \vee \neg C \vee \neg D$
- $\neg C \vee \neg D \vee \neg E$

Hill Climbing

Algorithm

- Input: state space S and cost function f.
- Output: $s^{\star} \in S$ that minimizes $f(s)$.
- Start at a random state s_0.
- At iteration t, find the neighbor that minimizes f.

$$s_{t+1} = \arg\min_{s \in s'(s_t)} f(s)$$

- Stop when none of the neighbors have a lower cost.

$$\text{stop if } f(s_{t+1}) \leqslant f(s_t)$$

Random Restarts

Discussion

- A simple modification is picking random initial states multiple times and finding the best among the local minima.

First Choice Hill Climbing

Discussion

- If there are too many neighbors, randomly generate neighbors until a better neighbor is found.
- This method is called first choice hill climbing.

Simulated Annealing

Description

- Each time, a random neighbor is generated.
- If the neighbor has a lower cost, move to the neighbor.
- If the neighbor has a higher cost, move to the neighbor with a small probability.
- Stop until bored.
- It is a version of Metropolis-Hastings Algorithm.