CS3243 Tutorial 5

Eric Han

March 3, 2022

Annoucements

1. Assignment 3 scores are now on Gradebook, please check.
2. Assignment 4 is *currently* being marked, check back on turnitin to check its status.
3. Assignment scores on Gradebook will have attendance taken into account.

Good thinking question to further understand search vs local search

From student

If the *h* function is the same for local search steepest ascent (Hill-Climbing) as well as greedy best first are the two are actly pretty much the same thing?

Good thinking question to further understand search vs local search

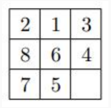
From student
If the *h* function is the same for local search steepest ascent (Hill-Climbing) as well as greedy best first are the two are actly pretty much the same thing?

- There are subtle differences:
  - BFS
    - Finding the goal from an inital state.
    - Stops when shallowest Goal Node is found.
  - Hill-Climbing
    - Finding the best $f(s_i)$ value.
    - Stops when no better $f(s_{i+1})$ can be found (can differ) - May/not reach Goal Node.

Ask more questions and everyone will benefit from better learning!

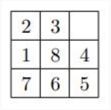
Previously from T04, Q3

Suppose you are given a 3×3 board with 8 tiles, where each tile has a distinct number between 1 to 8, and one empty space, as shown. Your goal is to reach the goal state.



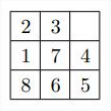
2	1	3
8	6	4
7	5	

Figure 1: Inital



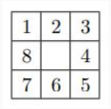
2	3	
1	8	4
7	6	5

Figure 2: 3a Inital



2	3	
1	7	4
8	6	5

Figure 3: 3b Inital



1	2	3
8		4
7	6	5

Figure 4: Goal

$f(s) =$ number of mismatched tiles compared to the goal state

Previously from T04, Q3

Suppose you are given a 3×3 board with 8 tiles, where each tile has a distinct number between 1 to 8, and one empty space, as shown. Your goal is to reach the goal state.

2	1	3
8	6	4
7	5	

Figure 1: Inital

2	3	
1	8	4
7	6	5

Figure 2: 3a Inital

2	3	
1	7	4
8	6	5

Figure 3: 3b Inital

1	2	3
8		4
7	6	5

Figure 4: Goal

$$f(s) = \text{number of mismatched tiles compared to the goal state}$$

Question 3a - hill-climbing (steepest descent)
From 3a inital state, Left - $f(s_{0,L}) = 3$, Down - $f(s_{0,D}) = 5$; Sequence: `L-L-D-R`

Question 3b - hill-climbing (steepest descent)
Similarly, sequence: `L-L-D-D`, local Minima.

Question 1

Consider the 4-queens problem on a 4×4 chess board. Suppose the leftmost column is column 1, and the topmost row is row 1. Let Q_i denote the row number of the queen in column i, $i = 1, 2, 3, 4$. Assume that variables are assigned in the order Q_1, Q_2, Q_3, Q_4, and the domain values of Q_i are tried in the order $1, 2, 3, 4$. Show a trace of the backtracking algorithm with forward checking to solve the 4-queens problem.

Recap

- What is backtracking?
  - When do we backtrack? - on domain wipeout!!
- How does forward checking improves backtracking?

Prelude

How does it backtracking look like *without* forward checking?

Backtracking without forward checking

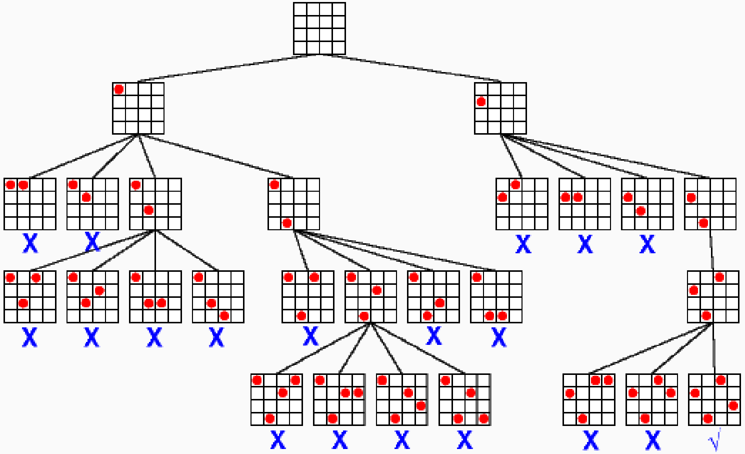


Figure 5: 4-queens problem and BT (Taken from Constraint Propagation)

Answer

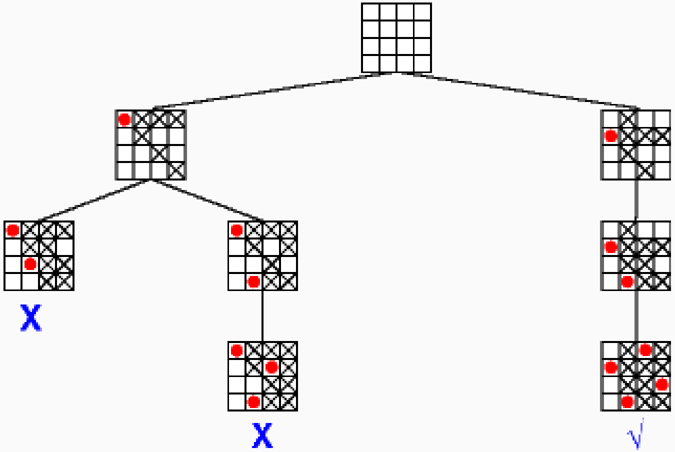


Figure 6: 4-queens problem and FC (Taken from Constraint Propagation)

Question 2

The classes are:

- C_1 - Programming Methodology: 8.00am to 9.00am
- C_2 - Discrete Structures: 8.30am to 9.30am
- C_3 - Data Structures and Algorithms: 9.00am to 10.00am
- C_4 - Introduction to Artificial Intelligence: 9.00am to 10.00am
- C_5 - Machine Learning: 9.30am to 10.30am

The professors are as follows, they can teach 1 class at a time:

- Professor Tess, who is available to teach classes C_3 and C_4.
- Professor Jill, who is available to teach classes C_2 , C_3 , C_4 , and C_5.
- Professor Bell, who is available to teach classes C_1 , C_2 , C_3 , C_4 , and C_5.

Recap

- What are the parts needed to model a CSP?
    - Variables and its corresponding domain
    - Constraints
- What are the parts needed for local search?
- What are the parts needed for search?

Question 2a

Formulate this as a CSP with each class being a variable, stating the effective domains and constraints.

Answer

Variables and its corresponding domain

Variables	C_1	C_2	C_3	C_4	C_5
Domains	$\{B\}$	$\{J, B\}$	$\{T, J, B\}$	$\{T, J, B\}$	$\{J, B\}$

Constraints

$$Q = \{C_1 \neq C_2, \quad C_2 \neq C_3, \quad C_3 \neq C_4, \quad C_4 \neq C_5, \quad C_2 \neq C_4, \quad C_3 \neq C_5\}$$

Question 2b

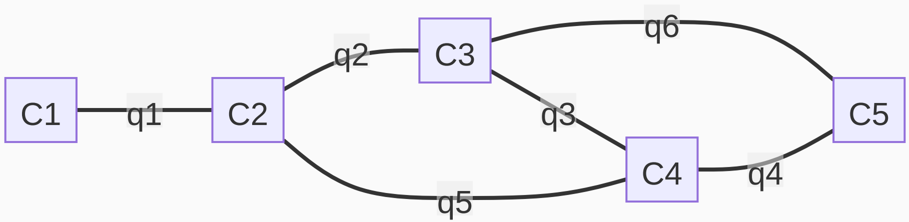


Figure 7: Sketching a Graph for the CSP (This looks familiar...)

Specify one solution to this CSP.

Answer

$$s_0 = (C_1, C_2, C_3, C_4, C_5) = (B, J, B, T, J),$$
$$s_1 = (B, J, T, B, J)$$

Question 4

Consider the item allocation problem:

- Group of people $N = \{1, \cdots, n\}$
- Group of items $G = \{g_1, \cdots, g_m\}$
- Each person $i \in N$ has a utility function $u_i : G \to \mathbb{R}_+$

The constraint is that every person is assigned at most one item, and each item is assigned to at most one person. An allocation simply says which person gets which item.

Question 4a

Write out the constraints: ‘each person receives no more than one item’ and ‘each item goes to at most one person’, using only the $x_{i,j}$ variables.

Question 4a

Write out the constraints: ‘each person receives no more than one item’ and ‘each item goes to at most one person’, using only the $x_{i,j}$ variables.

Answer

Each person receives no more than one item

$$\forall i \in N : \sum_{g_j \in G} x_{i,j} \leq 1$$

Each item goes to at most one person

$$\forall g_j \in G : \sum_{i \in N} x_{i,j} \leq 1$$

Question 4b

Suppose that people are divided into disjoint types $N_1, \cdots, N_k$ (think of, say, genders or ethnicities), and items are divided into disjoint blocks $G_1, \cdots, G_\ell$. We further require that each N_p only be allowed to take no more than λ_{pq} items from block G_q . Write out this constraint using the $x_{i,j}$ variables. (Note that each N_i corresponds to the set of people who are of that person type.)

Question 4b

Suppose that people are divided into disjoint types $N_1, \cdots, N_k$ (think of, say, genders or ethnicities), and items are divided into disjoint blocks $G_1, \cdots, G_\ell$. We further require that each N_p only be allowed to take no more than λ_{pq} items from block G_q . Write out this constraint using the $x_{i,j}$ variables. (Note that each N_i corresponds to the set of people who are of that person type.)

Answer

$$\forall N_p \in \{N_1, \cdots, N_k\}, G_q \in \{G_1, \cdots, G_\ell\} : \sum_{i \in N_p} \sum_{g_j \in G_q} x_{i,j} \leq \lambda_{pq}$$

Question 4c

We say that player i envies player i' if the utility that player i has from **their assigned item** is strictly lower than the **utility that player** i has from the **item assigned to player** i'. Write out the constraints that ensure that in the allocation, no player envies any other player. You may assume that the validity constraints from (a) hold.

Question 4c

We say that player i envies player i' if the utility that player i has from **their assigned item** is strictly lower than the **utility that player** i has from the **item assigned to player** i'. Write out the constraints that ensure that in the allocation, no player envies any other player. You may assume that the validity constraints from (a) hold.

Answer

$$\forall_{i,i' \in N} \ \ \forall_{g_j, g_{j'} \in G} : (x_{i,j} \wedge x_{i',j'}) \Rightarrow u_i(g_j) \geq u_i(g_{j'})$$

Another Answer, first we define for person i:

- $G_i = \{g_j \in G | x_{i,j}\}$ to be the set of items assigned to i and
- $G_i' = \{g_j \in G/G_i | \vee_{i' \in N/\{i\}} x_{i',j}\}$ to be the set of *assigned items* not assigned to i

$$\forall_{i \in N} \ \ \forall_{g_j \in G_i} \ \ \forall_{g_{j'} \in G_i'} : u_i(g_j) \geq u_i(g_{j'})$$