# CS3243 Tutorial 5

Eric Han

Sep 26, 2022

## Annoucements

1. Assignment 3 scores and comments are out on Luminus.
2. Midterm will be in Week 8 (not this week):
   2.1 You should always refer to the schedule on LumiNUS.

### Hot-tips for Midterms

1. Get your traces right; This is a basic understanding of the algorithms in class.
   - Go through and understand all algorithms (and different modes) covered in our class.
   - Use software to help you check the traces.
   - Everyone should be able to do this; Students in my class have no execuse.
2. Formulation is doable, remember all the various formulation parts.
   - Minimally remember the formulation parts.
   - Read wide and practice your creativity by trying to formulate various problems.
3. Proofs, this is hard.
   - Remember the proof structure.
   - Try to apply the same structure with different parts.

**Further understand search vs local search**

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? 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.

**Construction vs Mutation Style**

Do we have to use a 'mutation' style for local search? Nope, you can use any style you want, as long as the formulation is correct. Both can work and have tradeoffs.

*Ask more questions and everyone will benefit from better learning!*

3

## Previously from T04, Q3

Suppose you are given a $3 \times 3$ 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.



**Figure 1:** Inital

**Figure 2:** 3a Inital

**Figure 3:** 3b Inital

**Figure 4:** Goal

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

**Question 3a - hill-climbing (steepest descent)**

From inital, L - $f(s_{0,L}) = 3$, D - $f(s_{0,D}) = 5$; Sequence: L-L-D-R, global minima.

**Question 3b - hill-climbing (steepest descent)**

Similarly, sequence: L-L-D-D, local minima.

4

## Question 1

Consider the 4-queens problem on a $4 \times 4$ 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.
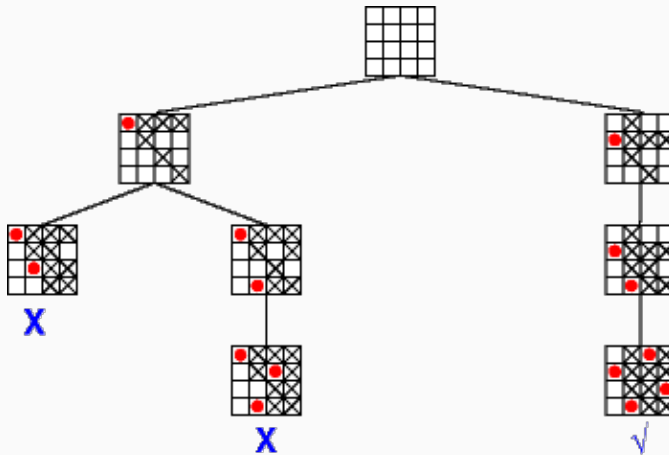
### Recap

- What is backtracking?
  - When do we backtrack? - on domain wipeout!!
- What is forward checking? 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.

## Question 2a

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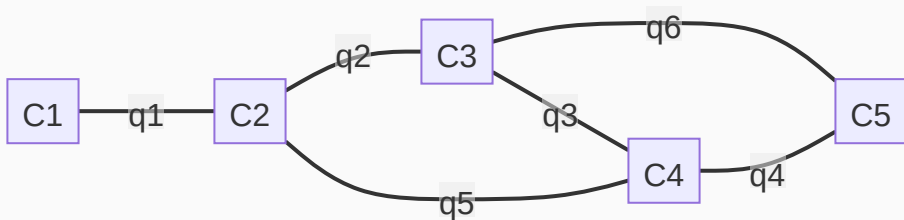
**Answer**

**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**

$$Q = \{C_1 \neq C_2, \ \ C_2 \neq C_3, \ \ C_3 \neq C_4, \ \ C_4 \neq C_5, \ \ C_2 \neq C_4, \ \ C_3 \neq C_5\}$$
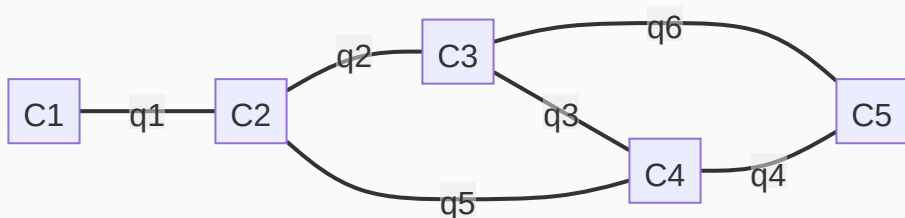
**Figure 7:** Sketching a Graph for the CSP (This looks familiar...)

Specify one solution to this CSP.

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

Specify one solution to this CSP.

Looks like Graph Coloring. . .

**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 3

Assignment Question; we will go through this question next week.

## 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.

$x_{i,j} \in \{0, 1\}$ where $x_{i,j} = 1$ if person $i$ receives good $g_j$ else 0.

## 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 $\lambda_{pq}$ 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 $\lambda_{pq}$ 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**

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

Lazy, more succinct version:

$$\forall p, q : \sum_{i \in N_p} \sum_{g_j \in G_q} x_{i,j} \leq \lambda_{pq}$$

**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**

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

*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 $i$.

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