Lab week 5: OPT and CSPs **Print Last Name:** **Winter**

Part A: Optimization

1. In lecture, we presented a hill climbing algorithm for a discrete state space. Discuss how you might extend hill climbing for a continuous state space. (Give two ways, one of which should not use calculus!)

   Two ways in which we could extend the hill climbing optimization algorithm for a continuous space would be to either:

   A. Adding a calculation to consider the slope from each position reached, with respect to the current parameters at the position. This should point / add more weight to the steepest path, maximizing and optimizing the objective.

   B. Dividing the continuous space into a finite number of set spaces, such as sampling, in order to break the problem into smaller, more manageable sections and computations.

2. Consider searching for an answer to the N-Queens problem using hill climbing. Let us define the states as a configuration of all the queens (possibly conflicting) and let us define the neighbors as the states you can reach by moving one queen. What is the eval function for this problem? In this case what do *plateau* and *local maxima* represent in terms of the eval function?

   The eval function for this problem is the total pairs of queens which are currently conflicting. In this case, ‘plateau’ refers to all the neighbors having the exact same eval value (there is no single move that would improve the current state, resulting in a multi-way tie) and ‘local maxima’ refers to when all the neighbors have a lower eval value than the current state (there is no move that would avoid increasing conflicts, resulting in a backtrack).

3. Your boss woke up in the middle of the night with the bright idea of using local search algorithms for sorting (increasing order) N distinct integers, where N is a fixed number. You’ve been asked to test this idea. The state representation you have decided to use is simply a sequence of N integers; a successor of a state is obtained by swapping any two adjacent elements in the sequence. For example, two successors of (13, 12, 19, 2, 5) are (12, 13, 19, 2, 5) and (13, 19, 12, 2, 5).
   a. Formulate an objective function for this problem.

   An objective function for this problem would be breaking the sequence into pairs (x,y), where the value of x is less than the value of y to ensure ascending order of elements.

b. Does your objective function have any local maxima or just a global maximum? Explain your answer.

My objective function has a global maximum value, representing the sequence perfectly sorted in increasing order. In this function, there is a global maximum because a successor state is not guaranteed to be perfectly sorted in ascending order when swapping a single pair. Therefore, a series of swaps must occur, the maximum value representing the highest number of swaps that can occur before the final state must be reached.

In our example, where N= 5, the maximum number of pairs / switches is 10, found from ((N * (N-1)) / 2).

c. Let N = 5 and let the initial state be (5, 9, 3, 7, 2). Show a possible next state and a possible state after that next state, along with their objective function values, when using:Hill climbing

A. Possible Next State: (5, 3, 9, 7, 2)

Objective Function Value: 4

→ pairs where x < y: (5, 9), (5, 7), (3, 9), (3, 7)

B. Possible State After Next State: (3, 5, 9, 7, 2)

Objective Function Value: 5

→ pairs where x < y: (3, 5), (3, 9), (3, 7), (5, 9), (5, 7)

d. Let N = 5 and let the initial state be (5, 9, 3, 7, 2). Show a possible next state and a possible state after that next state, along with their objective function values, when using: Simulated annealing (you get to choose a value for T)

T = 1500 (high temperature for randomness)

A. Possible Next State: (5, 3, 9, 7, 2)

Objective Function Value: 4

→ Probability : e^((4-3)/150) = 1.0, so we accept the state

B. Possible State After Next State: (3, 5, 9, 7, 2)

Objective Function Value: 5

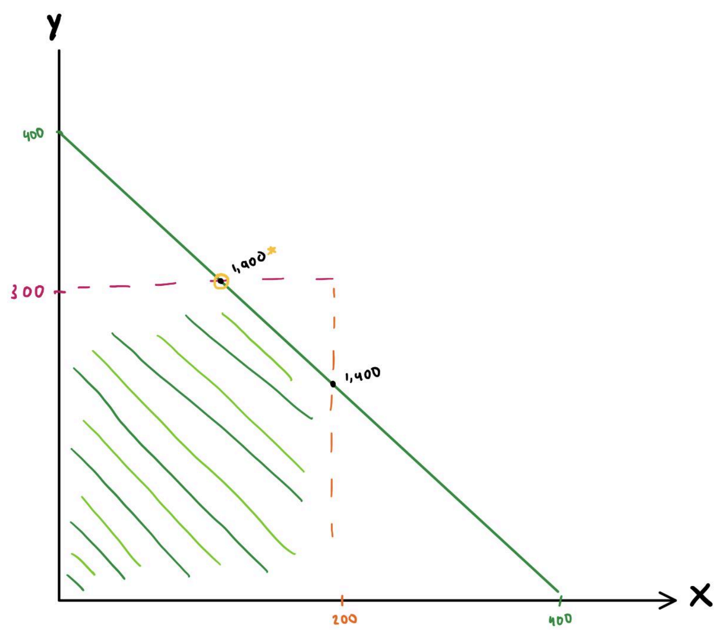
→ Probability : e^((5-4)/150) = 1.0, so we accept the state

4. Linear Programming: **Maximize the profit from a Chocolate Shop.** The Chocolate Shop produces two types of chocolate: Normal which has a profit of \$1 per box and **Deluxe** which has a profit of \$6 per box. **12 points**

- The variables are the number of boxes produced per day.
  - » x is the number of boxes of normal chocolate
  - » y is the number of boxes of deluxe chocolate
- The constraints are the number of boxes produced per day.
  - » $x \leq 200$ *Maximum demand of normal boxes per day*
  - » $y \leq 300$ *Maximum demand of deluxe boxes per day*
  - » $x + y \leq 400$ *Maximum production capacity*
  - » $x, y \geq 0$ *Can't have a negative number of boxes*

▪ The objective is to maximize profits which means to maximize x + 6y. This involves showing the feasible region and the point where the objective intersect it and proves the most profit is \$1900

Draw the graph that proves that the maximum value is \$1900. Briefly explain why the graph is a proof that it is the maximum profit



The graph is a proof that the maximum profit value is $1900 because the highest point on the objective function that lies within the feasible region is 1900, satisfying all of the constraints while maximizing the possible profit.

Part B: CSP

1. **Course Scheduling:** You are in charge of scheduling for computer science classes that meet Mondays, Wednesdays and Fridays. There are 5 classes that meet on these days and 3 professors who will be teaching these classes. You are constrained by the fact that each professor can only teach one class at a time.

The classes are:

1. Class 1 - Intro to Programming: meets from 8:00-9:00 am
2. Class 2 - Intro to Artificial Intelligence: meets from 8:30-9:30 am
3. Class 3 - Natural Language Processing: meets from 9:00-10:00 am

4. Class 4 - Computer Vision: meets from 9:00-10:00 am
5. Class 5 - Machine Learning: meets from 10:30-11:30am

The professors are:

1. Professor A, who is qualified to teach Classes 1, 2, and 5.
2. Professor B, who is qualified to teach Classes 3, 4, and 5.
3. Professor C, who is qualified to teach Classes 1, 3, and 4.

1. Formulate this problem as a CSP problem in which there is **one variable per class**, stating the domains (after enforcing unary constraints), and binary constraints. Constraints should be specified formally and precisely, but may be implicit rather than explicit.

Domains:

D1 - {A, C}, “For Class 1, you can either have Professor A or Professor C”

D2 - {A}, “For Class 2, you can only have Professor A”

D3 - {B, C}, “For Class 3, you can either have Professor B or Professor C”

D4 - {B, C}, “For Class 4, you can either have Professor B or Professor C”

D5 - {A, B}, “For Class 5, you can either have Professor A or Professor B”

Constraint (s):

C1 - D1 != {A}, “Professor A cannot teach Class 1; this is due to timing overlaps with Class 2, which is required to be taught by Professor A qualification wise.”

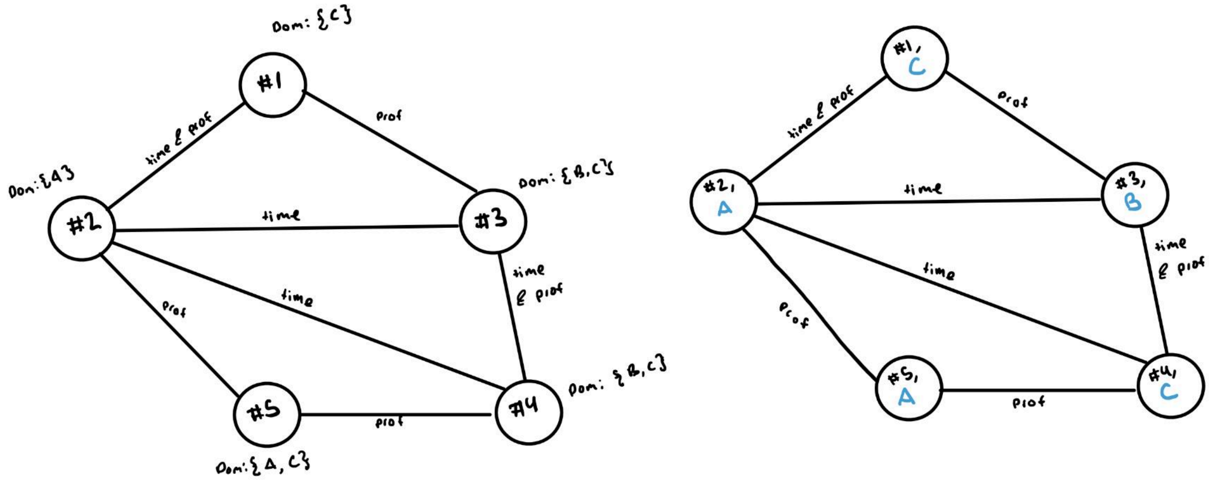
C3 - D3 != {A}, “Professor A cannot teach Class 3; this is due to them being underqualified”

C4 - D4 != {A}, “Professor A cannot teach Class 4; this is due to them being underqualified”

C5 - D5 != {C}, “Professor C cannot teach Class 5; this is due to them being underqualified”

C2 - D2 != {B, C}, “Professor B and Professor C cannot teach Class 2; this is due to them being underqualified”

Draw the constraint graph associated with your CSP.



2. **Air Traffic Control:** We have five planes: A, B, C, D, and E and two runways: international and domestic. We would like to schedule a time slot and runway for each aircraft to either land or take off. We have four time slots: {1; 2; 3; 4] for each runway, during which we can schedule a landing or take off of a plane. We must find an assignment that meets the following constraints:
   - Plane B has lost an engine and must land in time slot 1.
   - Plane D can only arrive at the airport to land during or after time slot 3.
   - Plane A is running low on fuel but can last until at most time slot 2.
   - Plane D must land before plane C takes off, since passengers must transfer from D to C.
   - No two aircraft can reserve the same time slot for the same runway.

A. Complete the formulation of this problem as a CSP in terms of variables, domains, and constraints (both unary and binary). Constraints should be expressed using mathematical or logical notation rather than with words.

Variables: Ptr, where ‘P’ is the plane with given information ‘t’ = time slot and ‘r’ = runway.

Domains:

D1, Atr - {(A 1,international), (A 2,international), (A 3,international), (A 4,international), (A 1,domestic), (A 2,domestic), (A 3,domestic), (A 4,domestic)}

D2, Btr - {(B 1,international), (B 2,international), (B 3,international), (B 4,international), (B 1,domestic), (B 2,domestic), (B 3,domestic), (B 4,domestic)}

D3, Ctr - {(C 1,international), (C 2,international), (C 3,international), (C 4,international), (C 1,domestic), (C 2,domestic), (C 3,domestic), (C 4,domestic)}

D4, Dtr - {(D 1,international), (D 2,international), (D 3,international), (D 4,international), (D 1,domestic), (D 2,domestic), (D 3,domestic), (D 4,domestic)

D4, Etr - {(E 1,international), (E 2,international), (E 3,international), (E 4,international), (E 1,domestic), (E 2,domestic), (E 3,domestic), (E 4,domestic)

Unary Constraints:
C1 - B = {B 1,international, B 1,domestic}

C2 - D = {(D 3,international), (D 3,domestic), (D 4,international), (D 4,domestic)}

C3 - A = {(A 1,international), (A 1,domestic), (A 2,international), (A 2,domestic)}

Binary Constraints:
C4 - {D, C} = Dt < Ct

C5 - {A, B, C, D, E} = Atr != Btr != Ctr != Dtr != Etr

B. For the following subparts, we add the following two constraints:
- Planes A, B, and C cater to international flights and can only use the international runway.
- Planes D and E cater to domestic flights and can only use the domestic runway.

(Updated) Unary Constraints:
C1 - B = {B 1,international}

C2 - D = {(D 3,domestic), (D 4,domestic)}

C3 - A = {(A 1,international), (A 2,international))}

(Updated) Binary Constraints:
C4 - {D, C} = Dt < Ct

C5 - {A, B, C, D, E} = Atr != Btr != Ctr != Dtr != Etr

C6 - {A, B, C} = {Ati, Bti, Cti}

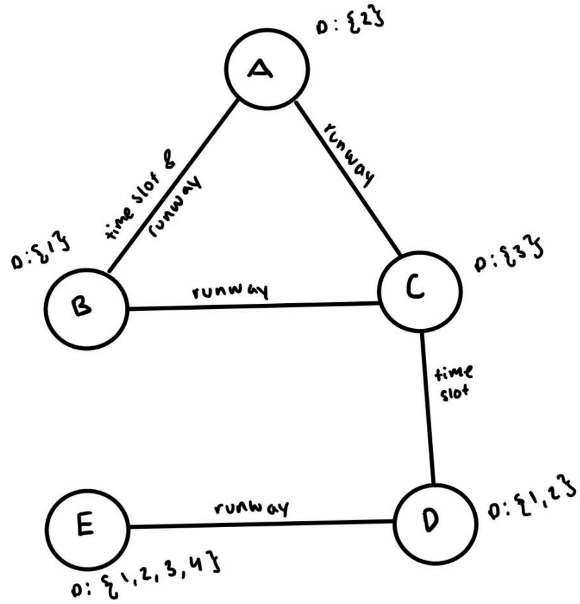
C7 - {D, E} = {Dtd, Etd}

With the addition of the two constraints above, we completely reformulate the CSP. You are given the variables and domains of the new formulation.

New Variables: A, B, C, D, E for each plane.

New Domains: {1,2,3,4}

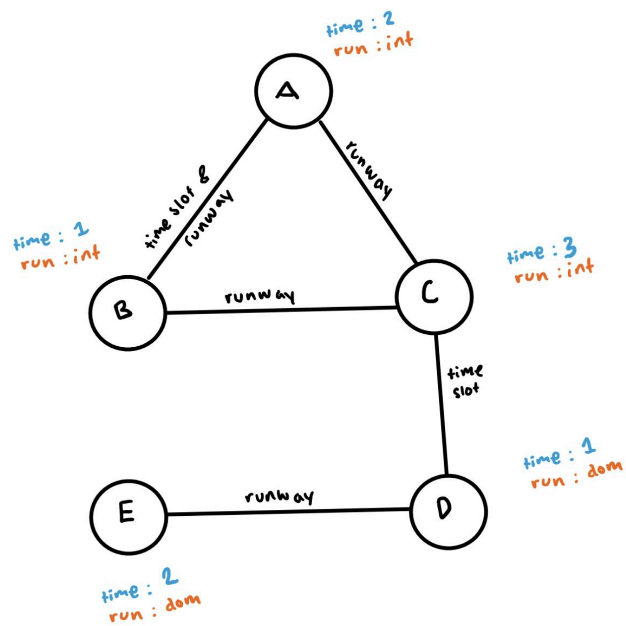
Complete the constraint graph for this problem given the original constraints and the two added ones.



What are the domains of the variables after enforcing arc-consistency? Begin by enforcing unary constraints. (Cross out values that are no longer in the domain.)

A	1	2	3	4
B	1	2	3	4
C	1	2	3	4
D	1	2	3	4
E	1	2	3	4

Arc-consistency can be rather expensive to enforce, and we believe that we can obtain faster solutions using only forward-checking on our variable assignments. Perform backtracking search on the graph with forward checking. First, list the (variable; assignment) pairs in the order they occur (including the assignments that are reverted upon reaching a dead end). Enforce unary constraints before starting the search.


time : 2
run : int
A
runway
time slot &
runway
time : 1
run : int
B
runway
C
time : 3
run : int
time
slot
time : 1
run : dom
E
runway
D
time : 2
run : dom