CS 170 HW 10

Due **2021-11-08, at 10:00 pm**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write none.

In addition, we would like to share correct student solutions that are well-written with the class after each homework. Are you okay with your correct solutions being used for this purpose? Answer “Yes”, “Yes but anonymously”, or “No”

2 How to Gamble With Little Regret

Suppose that you are gambling at a casino. Every day you play at a slot machine, and your goal is to minimize your losses. We model this as the experts problem. Every day you must take the advice of one of n experts (i.e. play at a slot machine). At the end of each day t, if you take advice from expert i, the advice costs you some c_i^t in $[0, 1]$. You want to minimize the regret R, defined as:

$$R = \frac{1}{T}\left(\sum_{t=1}^{T} c_{i(t)}^t - \min_{1\le i\le n}\sum_{t=1}^{T} c_i^t\right)$$

where $i(t)$ is the expert you choose on day t. Notice that in this definition, you are comparing your losses to the best expert, rather than the best overall strategy.

Your strategy will be probabilities where p_i^t denotes the probability with which you choose expert i on day t. Assume an all powerful adversary (i.e. the casino) can look at your strategy ahead of time and decide the costs associated with each expert on each day. Let C denote the set of costs for all experts and all days. Compute $\max_C(\mathbb{E}[R])$, or the maximum possible (expected) regret that the adversary can guarantee, for each of the following strategies.

(a) Any deterministic strategy, i.e. for each t, there exists some i such that $p_i^t = 1$.

(b) Always choose an expert according to some fixed probability distribution at every time step. That is, fix some $p_1, \ldots, p_n$, and for all t, set $p_i^t = p_i$.

What distribution minimizes the regret of this strategy? In other words, what is $\operatorname{argmin}_{p_1,\ldots,p_n} \max_C(\mathbb{E}[R])$?

This analysis should conclude that a good strategy for the problem must necessarily be randomized and adaptive.

3 Zero-Sum Battle

This is a solo question.

Two Pokemon trainers are about to engage in battle! Each trainer has 3 Pokemon, each of a single, unique type. They each must choose which Pokemon to send out first. Of course

1. 3037534676 Shenghan Zheng

2.

(a) To maximize the regret, set the cost of chosen expert for that timestep to be 1 and others to be 0. Thus, the best expert should be the least chosen expert. Denote the number of times for best expert be s

$R = \frac{1}{T}(T-s)$ $\quad \because ns \le T$

$\therefore R = \frac{1}{T}(T-s) \ge \frac{1}{T}(T - \frac{T}{n}) = \frac{n-1}{n}$

equation holds when we choose all experts with the same number of times

(b)

$s = \arg\min_i p_i$ Let s be the expert with least cost. To max the regret, Set $c_s^t = 0$ for all t and $c_i^t = 1$ for $i \neq s$, all t

Since the distribution is fixed, so we know which expert is most unlikely to be chosen.

We want him to be the best expert

because $E[\sum_{t=1}^{T} c_{i(t)}^t] = T(p_1 + \cdots + p_{s-1} + p_{s+1} + \cdots) = T(1-p_s)$

and $\min_i \sum_{t=1}^{T} c_i^t = 0$

$\Rightarrow$ expected regret $= \frac{1}{T} \cdot (T(1-p_s) - 0) = 1 - p_s$

each trainer's advantage in battle depends not only on their own Pokemon, but on which Pokemon their opponent sends out.

The table below indicates the competitive advantage (payoff) Trainer A would gain (and Trainer B would lose). For example, if Trainer B chooses the fire Pokemon and Trainer A chooses the rock Pokemon, Trainer A would have payoff 2.

		Trainer B:		
		ice	water	fire
Trainer A:	dragon	-10	3	3
	steel	4	-1	-3
	rock	6	-9	2

Feel free to use an online LP solver to solve your LPs in this problem.
Here is an example of an online solver that you can use: `https://online-optimizer.appspot.com/`.

1. Write an LP to find the optimal strategy for Trainer A. What is the optimal strategy and expected payoff?
2. Now do the same for Trainer B. What is the optimal strategy and expected payoff? How does the expected payoff compare to the answer you get in part (a)?

4 Minimum ∞-Norm Cut

In the Minimum Infinity-Norm Cut problem, you are given a connected undirected graph $G = (V, E)$ with positive edge weights w_e, and you are asked to find a cut in the graph where the largest edge in the cut is as small as possible (note that there is no notion of source or target; any cut with at least one node on each side is valid).

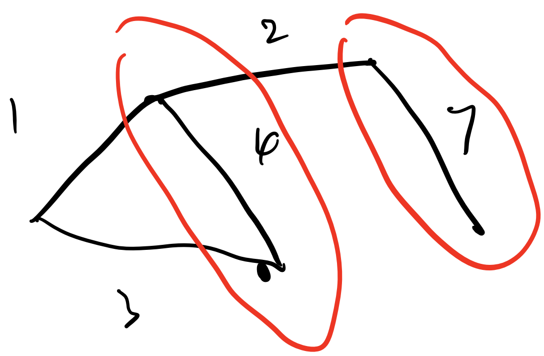
Show how to reduce Minimum Infinity-Norm Cut to Minimum Spanning Tree in linear time. **Give a 3-part solution.**

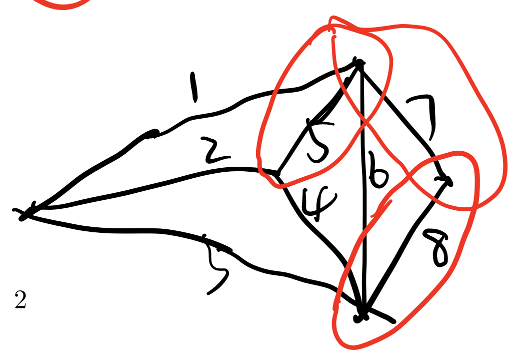
Hint: Minimum Spanning Tree does not require edge weights to be positive.

Kruskal:

keeping adding

max edge





3\.

1) Suppose A shows his strategy $[x_1, x_2, x_3]$

$$z = \min\{-10x_1 + 6x_2 + 6x_3,\ 3x_1 - x_2 - 9x_3,\ 3x_1 - 3x_2 + 2x_3\}$$

$$\max z$$

$$z \le -10x_1 + 4x_2 + 6x_3$$

$$z \le 3x_1 - x_2 - 9x_3$$

$$z \le 3x_1 - 3x_2 + 2x_3$$

$$x_1 + x_2 + x_3 = 1$$

$$x_1, x_2, x_3 \ge 0$$

Model Run Examples Help — Solving finished. An optimal solution was found. — Draft saved just now

Documentation
Model
Solution
Model overview
Variables
Constraints
Output
Log messages

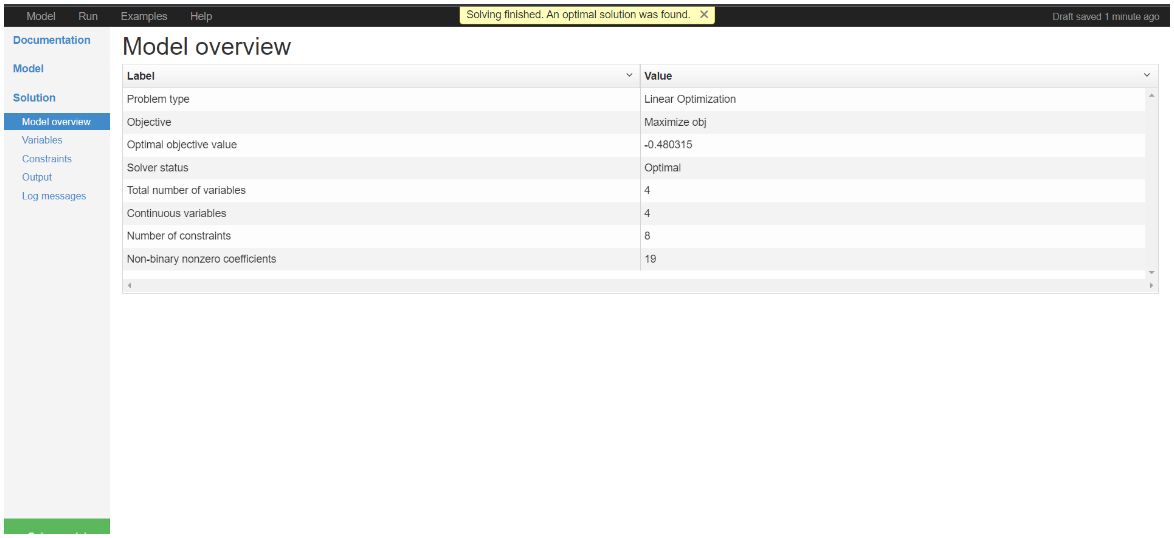
Variables

Variable	Type	Value	Value bounds	Status	Reduced obj coef	Obj coef tol interval
x1	Real	0.3346457	[0, Inf]	Basic	0	[-1, Inf]
x2	Real	0.5629921	[0, Inf]	Basic	0	
x3	Real	0.1023622	[0, Inf]	Basic	0	
x4	Real	-0.480315	[-Inf, Inf]	Basic	0	

x_4 : expected payoff

$[x_1\ x_2\ x_3]$: optimal strategy

Solve model



Model overview

Label	Value
Problem type	Linear Optimization
Objective	Maximize obj
Optimal objective value	-0.480315
Solver status	Optimal
Total number of variables	4
Continuous variables	4
Number of constraints	8
Non-binary nonzero coefficients	19


```
var x1 >= 0;
var x2 >= 0;
var x3 >= 0;
var x4;
maximize obj:x4;

subject to c11:   10*x1 -   4*x2 -6*x3 + x4 <=  0;
subject to c12: -3*x1 +   x2 +9*x3 + x4 <=  0;
subject to c13:   -3*x1 +   3*x2 -2*x3 + x4 <=  0;
subject to c14:     x1 +     x2 +x3 =  1;
subject to c15:  x1>=0;
subject to c16:  x2>=0;
subject to c17:  x3>=0;
end;
```

2) Suppose B shows the strategy $[x_1, x_2\ x_3]$

$$Z = \max\{-10x_1 + 3x_2 + 3x_3,\ 4x_1 - x_2 - x_3,\ 6x_1 - 9x_2 + 2x_3\}$$

$$\min z$$

$$z \geq -10x_1 + 3x_2 + 3x_3$$

$$z \geq 4x_1 - x_2 - x_3$$

$$z \geq 6x_1 - 9x_2 + 2x_3$$

$$x_1 + x_2 + x_3 = 1$$

$$x_1, x_2, x_3 \geq 0$$

Documentation
Model
Solution
- Model overview
- Variables
- Constraints
- Output
- Log messages

Model overview

Label	Value
Problem type	Linear Optimization
Objective	Minimize obj
Optimal objective value	-0.480315
Solver status	Optimal
Total number of variables	4
Continuous variables	4
Number of constraints	8
Non-binary nonzero coefficients	19

Model Run Examples Help
Solving finished. An optimal solution was found. ×
Draft saved just now

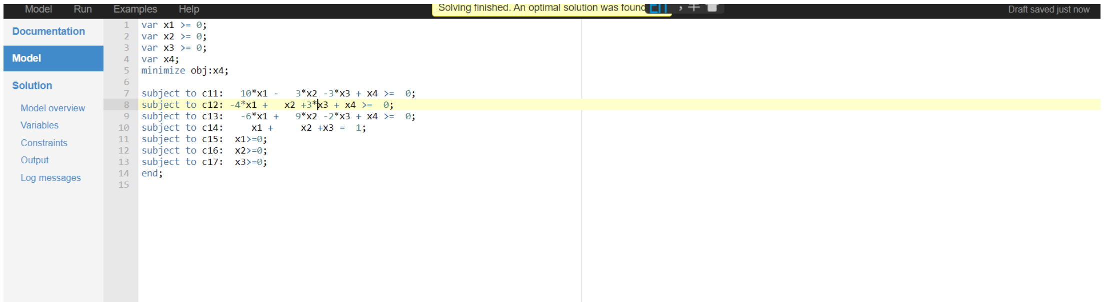
Documentation
Model
Solution
- Model overview
- Variables
- Constraints
- Output
- Log messages

Variables

Variable	Type	Value	Value bounds	Status	Reduced obj coef	Obj coef tol interval
x1	Real	0.2677165	[0, Inf]	Basic	0	[-1, Inf]
x2	Real	0.3228346	[0, Inf]	Basic	0	
x3	Real	0.4094488	[0, Inf]	Basic	0	
x4	Real	-0.480315	[-Inf, Inf]	Basic	0	

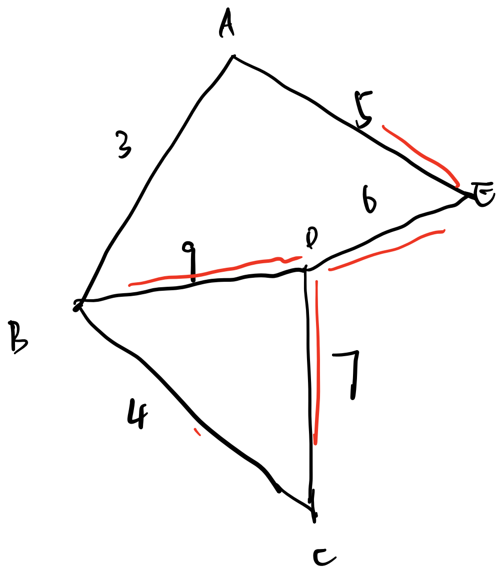
x_4 obj

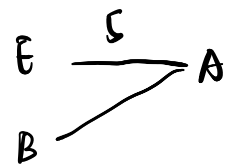
$[x_1, x_2, x_3]$: optimal strategy

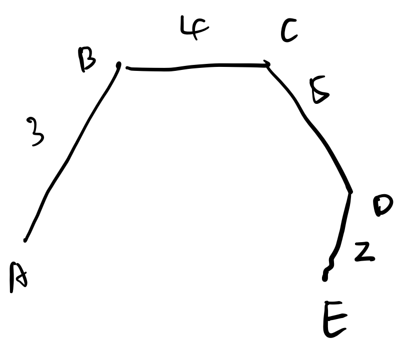


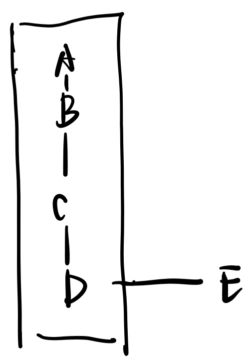
they are the same

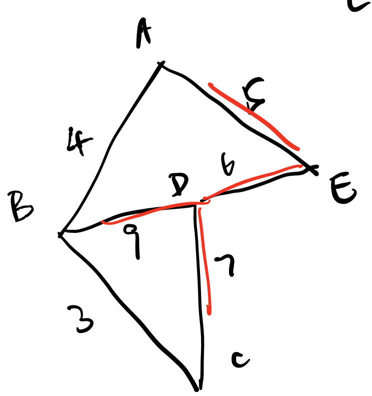
4. find maximum spanning tree

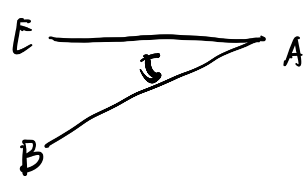


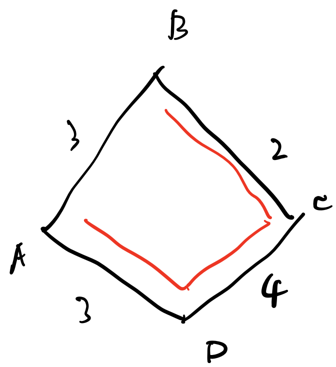




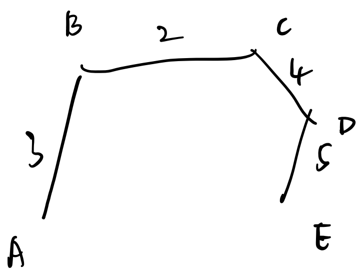


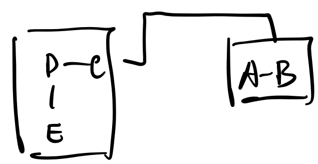





B
3
2
C
A
3
4
D


C-D
1
A
B


B
2
C
4
D
3
5
A
E


D-C
1
E
A-B

4.

1) Main Idea:

Set the cut $S_1 = \phi$ $S_2 = V/S_1$

Run DFS, if we find that there are at least 2 connected component $\bar{S}_1, \dots, \bar{S}_k$ $(k \geq 2)$ Let $S_1 = \bar{S}_1$ $S_2 = V/S_1$ return.

Else:

Make a copy of the graph with -ve weight for each edge, run MST algorithm and find the

MST 

Let (v_i, v_{i+1}) be the max weight edge in the MST. Split V into $S_1 = \{v_1, \dots, v_i\}$, $S_2 = S_2/S_1$

Return (S_1, S_2)

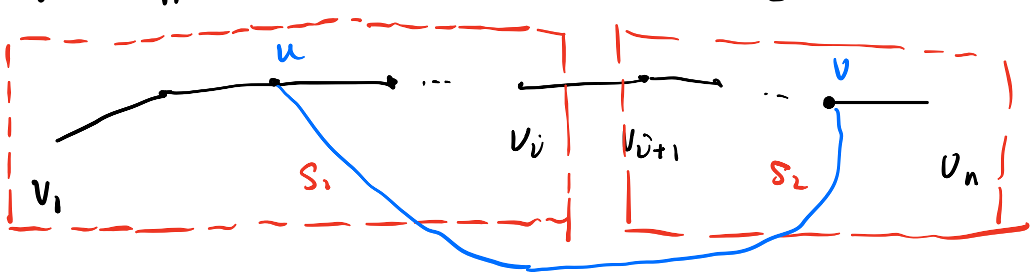
2) proof of correctness (Notice that the MST in negated graph is the maximum spanning tree in original graph)

For the maximum spanning tree, there must be one edge in it that crosses the cut

Thus, the minimum edge in the maximum spanning tree $\leq$ max weight edge cross the cut $\leq$ the maximum edge in the maximum spanning tree

claim: The cut algorithm returns will always make the max weight edge cross the cut be the minimum edge in the maximum spanning tree

proof: suppose (v_i, v_{i+1}) be the minimum edge



If there is an edge $(u,v) \in E$

s.t $u \in S_1$ $v \in S_2$ and $w_{(u,v)} > w_{(v_i, v_{i+1})}$

Thus, delete (v_i, v_{i+1}) but add (u,v)

will form a bigger maximum spanning tree. This means our MST in negated graph isn't the smallest.

Contradiction!

proved!

3) Runtime

DFS : $O(|V|+|E|)$

Create a negated graph : $O(|V|+|E|)$

the space is $O(|E|+|V|)$

Thus, the reduction is linear

MST : $O(|E|\log|V|)$

Thus the total runtime is $O(|E|\log|V|)$