

# CS170–Spring 2019 — Homework 8 Solutions

Ran Liao, SID 3034504227

March 20, 2019

Collaborators:Jingyi Xu, Renee Pu

## 1 Study Group

Name	SID
Ran Liao	3034504227
Jingyi Xu	3032003885
Renee Pu	3032083302

## 2 Modeling: Tricks of the Trade

(a)

$$\begin{aligned} \min \quad & z_1 + z_2 + \cdots + z_n \\ & z_i \geq y_i - (a + bx_i) \text{ for } i = 1, \cdots, n \\ & z_i \geq -(y_i - (a + bx_i)) \text{ for } i = 1, \cdots, n \end{aligned}$$

(b)

$$\begin{aligned} \min \quad & z \\ & z \geq y_i - (a + bx_i) \text{ for } i = 1, \cdots, n \\ & z \geq -(y_i - (a + bx_i)) \text{ for } i = 1, \cdots, n \end{aligned}$$

### 3 Zero Sum Games

- (a)  $x_1$  is the probability that Alice will play strategy 1.  
 $x_2$  is the probability that Alice will play strategy 2.  
 $p$  is Alice's payoff

(b)

$$\begin{aligned} \max p \\ p &\leq 4x_1 + 2x_2 \\ p &\leq x_1 + 5x_2 \\ x_1 + x_2 &= 1 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \\ p &\geq 1 \end{aligned}$$

(c)

$$\begin{aligned} \max p \\ p &\leq 4x_1 + 2(1 - x_1) = 2x_1 + 2 \\ p &\leq x_1 + 5(1 - x_1) = -4x_1 + 5 \\ 0 &\leq x_1 \leq 1 \\ p &\geq 1 \end{aligned}$$

(d)



Figure 1: Feasible Region

- (e) The optimal solution is  $(\frac{1}{2}, \frac{1}{2})$  and the value of game is 3.

## 4 Repairing a Flow

### (a) Main Idea

Build the residual network based on maximum flow  $f$ . Then find a path from  $t$  to  $s$  that go through edge  $(v, u)$  in residual network. Push 1 unit flow into this path and fix the wrong capacity from  $c_{uv}$  to  $c_{uv} - 1$ . Then run the original max-flow algorithm starting from this residual network.

### (b) Proof of Correctness

Pushing 1 unit back from  $t$  to  $s$  that go through edge  $(v, u)$  will make the flow go through edge  $(u, v)$  be valid. Then run the original max-flow algorithm again from this is sure will give an optimal solution.

### (c) Runtime Analysis

Build residual network will cost  $O(|E|)$  time. Find a path from  $t$  to  $s$  that go through edge  $(v, u)$  by DFS or BFS will cost  $O(|V| + |E|)$  time. Then push 1 unit into this path will cost at most  $|E|$  time. Since the max-flow solution after repairing capacity of edge  $(u, v)$  cannot be larger than the original one. At most 1 iteration is needed if run max-flow algorithm starting from this. So total runtime is still linear, which is  $O(|V| + |E|)$ .

## 5 Generalized Max Flow

### (a) Constraints

#### (i) Capacity Constraints

$$\sum_{i=1}^k f_e^{(i)} \leq c_e \quad \text{for } \forall e$$

#### (ii) Flow Conservation

$$\sum_{i=1}^k \sum_{u:(u,v) \in E} f_{(u,v)}^{(i)} = \sum_{i=1}^k \sum_{u:(v,w) \in E} f_{(v,w)}^{(i)} \quad \text{for } \forall v \text{ except } s_1, \dots, s_k, t_1, \dots, t_k$$

#### (iii) Nonnegativity

$$f_e^{(i)} \geq 0 \quad \text{for } \forall e, i = 1, \dots, k$$

#### (iv) Demand Constraints

$$\sum_{u:(u,t_i) \in E} f_{(u,t_i)}^{(i)} \geq d_i \quad \text{for } i = 1, \dots, k$$

### (b) Objective

$$\max \sum_{i=1}^k \sum_{u:(u,t_i) \in E} f_{(u,t_i)}^{(i)}$$

The sum of all flows go to destination.

## 6 Reductions Among Flows

- (a) Separate each vertex into two new vertices and denote them as  $v_{in}$  and  $v_{out}$ . Then link all incoming edges in original graph  $G$  to  $v_{in}$  and all outgoing edges to  $v_{out}$ . More formally, create a new edge  $(u, v_{in})$  in new graph if edge  $(u, v)$  is in original graph and create a new edge  $(v_{out}, u)$  in new graph if edge  $(v, u)$  is in original graph. Finally, create a new edge  $(v_{in}, v_{out})$  with capacity  $c_v$ .

Proof:

If  $F$  is a flow in  $G$  satisfying the vertex capacity constraint, set  $f_{(v_{in}, v_{out})}$  to be  $\sum_{u:(u,v) \in E} f_{uv}$  and let all other flows remain same. This is a valid flow because  $f_{(v_{in}, v_{out})} = \sum_{u:(u,v) \in E} f_{uv} \leq c_v$ . And since all other flows are unchanged, it's flow with same size.

If  $F'$  is a flow in  $G'$ , ignore all  $f_{(v_{in}, v_{out})}$  will give a valid flow  $F$  in  $G$  with same size.

- (b) Create an artificial vertex  $s$  and create edges  $(s, s_1), \dots, (s, s_k)$  with  $\infty$  capacity.

Proof:

If  $F$  is a flow in  $G$  satisfying the vertex capacity constraint, there's always a possible solution in  $G'$  with same size. Since the capacity for edges  $(s, s_1), \dots, (s, s_k)$  is infinity. It can be arbitrary large. So just assign  $f_{(s, s_i)}$  with  $\sum_{w:(s_i, w) \in E} f_{(s_i, w)}$ .

If  $F'$  is a flow in  $G'$ , ignore all  $f_{(s, s_i)}$  will give a valid flow  $F$  in  $G$  with same size.

## 7 A Flowy Metric

- (a) According to **Max-flow min-cut theorem**, the size of the maximum flow in a network equals the capacity of the smallest  $(s, t)$ -cut. Since any cut in  $G$  has capacity at least 1, the max flow from  $s$  to  $t$  is at least 1.
- (b)