

# CSCI 270 Homework 8

Qilin Ye

December 9, 2022

## 1. Class Selection with Prerequisite Penalty

*Solution.* This question is vastly similar to the one presented in KT §7.11, and we will adopt a similar approach. Let  $C^+ := \{\text{class } i : r_i > 0\}$  and  $C^- := \{\text{class } i : r_i < 0\}$ . We begin by some set-theoretic manipulations. For any  $S \subset \{1, \dots, n\}$ ,  $S = (S \cap C^+) \cup (S \cap C^-) = (C^+ \setminus (C^+ \setminus S)) \cup (S \cap C^-) = (C^+ \setminus (S^c \cap C^+)) \cup (S \cap C^-)$ , so

$$R(S) = \sum_{i \in S} r_i - \sum_{j \notin S, i \in S} p_{j,i} = \sum_{C^+} r_i - \sum_{S^c \cap C^+} r_i + \sum_{S \cap C^-} r_i - \sum_{j \notin S, i \in S} p_{j,i}.$$

Since  $\sum_{C^+} r_i$  is fixed given knowledge of  $\{x_i\}_{i=1}^n$ , we attempt to construct a graph whose cuts have capacities corresponding to the other three terms. A canonical example is by defining a graph  $G = (V, E)$ , where

$$V = \{\text{a source } s, \text{ a sink } t\} \cup \{x_i\}_{i=1}^n \quad (*)$$

and

$$E = \{(s, x_i) \text{ with weight } r_i \mid i \in C^+\} \cup \{(x_i, t) \text{ with weight } -r_i \mid i \in C^-\} \\ \cup \{(x_i, x_j) \text{ with weight } p_{j,i} \mid \text{class } j \text{ is a prerequisite of class } i\}. \quad (**)$$

We now show that given any  $S \subset \{1, \dots, n\}$ , the cut  $(S \cup \{s\}, S^c \cup \{t\})$  in  $G$  has capacity  $\sum_{S^c \cap C^+} r_i - \sum_{S \cap C^-} r_i + \sum_{j \notin S, i \in S} p_{j,i}$ . To this end, we let  $e$  be an edge with one endpoint in  $S$  and the other  $S^c$ . By construction, one of the following three scenarios take place:

- (1)  $e$  starts with  $s$  and ends at some  $x_i$ . Since  $s \in S \cup \{s\}$ , we have  $x_i \in S^c$ . That  $(s, x_i)$  is an edge implies  $i \in C^+$ , so  $i \in S^c \cap C^+$ . The total capacity of all such edges is  $\sum_{S^c \cap C^+} r_i$ .
- (2)  $e$  ends with  $t$  and starts from some  $x_i$ . Since  $t \in S^c \cup \{t\}$ , we have  $x_i \in S$ . Similar to (1), we now must have  $i \in C^-$ , so  $i \in S \cap C^-$ . The total capacity of all such edges is  $\sum_{S \cap C^-} (-r_i) = -\sum_{S \cap C^-} r_i$ .
- (3) Finally,  $e$  can be of form  $(x_i, x_j)$ , with  $x_i \in S$ ,  $x_j \in S^c$ . The total capacity of all such edges is  $\sum_{j \notin S, i \in S} p_{j,i}$ .

Having shown that on  $G$ , the cut capacities satisfy

$$c(S \cup \{s\}, S^c \cup \{t\}) = \sum_{S^c \cap C^+} r_i - \sum_{S \cap C^-} r_i + \sum_{j \notin S, i \in S} p_{j,i},$$

maximizing  $R(S)$  is equivalent to finding a minimum cut on  $G$ , so all that remains is to apply Edmonds-Karp.

---

**Algorithm 1:** Class selection with prerequisite penalty

---

- 1 **Inputs:** classes  $\{x_i\}_{i=1}^n$  with rewards  $\{r_i\}_{i=1}^n$ ; prerequisite relations  $\{p_{j,i}\}$ , not necessarily for all  $1 \leq i, j \leq n$ .
  - 2 **Initialization:** a directed, weighted graph  $G = (V, E)$  defined by (\*) and (\*\*).
  - 3 Run Edmonds-Karp on  $G$  and obtain a min cut  $(S \cup \{s\}, S^c \cup \{t\})$ .
  - 4 **Output:**  $S$ ; namely, take class  $x_i$  if and only if  $x_i \in S$ .
- 

Finally, this algorithm runs in polynomial time in  $n$ , since initializing the graph takes at most  $\mathcal{O}(n^2)$  iterations (worse case, though impossible, is when  $p_{j,i}$  is defined for all  $1 \leq i, j \leq n$ ). Edmonds-Karp therefore runs in polynomial time w.r.t. variables that are polynomial in  $n$  and is therefore itself also in polynomial w.r.t.  $n$ .  $\square$

## 2. Interval Selection & Interval Hitting

*Proof of the weak duality lemma for interval selection.* Let  $J, T$  be arbitrarily given as stated. Since intervals in  $J$  are pairwise disjoint, one point in  $T$  can hit at most one interval in  $J$ . Thus, to hit all of  $\{I_j\}_{j=1}^n$ ,  $T$  needs to hit at least all of  $J$ , and in doing so  $T$  needs one point for each interval in  $J$ , so  $|T| \geq |J|$ .  $\square$

*Duality proof of Greedy optimality.* We sort the intervals and relabel them as  $I_1, \dots, I_n$ , in ascending order of their right endpoints. Let  $J_n$  be the greedy output on  $I_1, \dots, I_n$  (i.e., iteratively pick non-overlapping intervals with smallest right endpoint). We define

$$T_n := \{x : x \text{ is a right endpoint of an interval picked by } J_n\}.$$

The intervals in  $J_n$  are pairwise disjoint, so  $(I_k \mapsto I_k\text{'s right endpoint})$  is a well-defined bijection, and hence  $|J_n| = |T_n|$ . It remains to show that  $T_n$  is a valid solution for interval hitting. Pick any  $I_k$ . Either  $I_k \in J_n$  or not. In the former case, the right endpoint of  $I_k$  is by construction contained in  $T_n$ , so  $T_n \cap I_k \neq \emptyset$ . If  $I_k \notin J_n$ , then the greedy algorithm must have excluded  $I_k$  for it overlaps with some other interval  $I_j$  already selected, with  $j < k$ . If  $I_j = [a_j, b_j]$  and  $I_k = [a_k, b_k]$ , then  $b_j < b_k$  and  $I_j \cap I_k \neq \emptyset$  imply  $a_k < b_j < b_k$ . Since  $I_j \in J_n$ , we know  $b_j \in T_n$  and indeed  $T_n \cap I_k \supset \{b_j\}$ . Since  $I_k$  is arbitrarily chosen we conclude that  $T_n$  is a valid interval hitting solution. The weak duality lemma implies  $\max_{\text{valid } J} |J| \leq \min_{\text{valid } T} |T|$ , whereas our  $J_n$  attains equality. Therefore  $J_n$  is the optimal solution for interval selection (and  $T_n$  too for interval hitting).  $\square$

## 3. Chocolate — Attack & Defense Game

*Solution.* We assume the game ends after finitely many rounds. Otherwise, comparing to playing an infinite game, I'd rather give the chocolates up.

*Lemma 1.* If  $c \geq 3$ , the following strategy, if possible to be carried out, is sufficient for us to win the game: for each target  $u$ , if the damage it receives over all rounds is  $d$ , then for each  $1 \leq k \leq \lfloor d \rfloor$ , we repair the shield once when the total damage taken by  $d$  lies in  $[k-1, k)$ .

*Proof of lemma 1.* Fix  $u$ . If  $0 \leq t-2 \leq t \leq \lfloor d \rfloor$ , then we repaired  $u$  twice, once when its damage taken is between  $[t-2, t-1)$  and the other  $[t-1, t)$ . The damage  $u$  takes in-between these two repairs  $< 2$ , so if  $c \geq 3$ , our target  $u$  will have  $> 1$  shield left, making the enemy impossible to destroy it next round.

On the other hand, the total damage  $u$  takes since its final repair is bounded by  $1 + (d - \lfloor d \rfloor) < 2 < c$ . Therefore the enemy is unable to destroy  $u$  throughout the game. END OF PROOF OF LEMMA 1

Notation-wise, let  $d(x_i, t)$  be the total damage  $x_i$  receives after round  $t$ , i.e.,  $d(x_i, t) = \sum_{k=1}^t d_{k,i}$ . Then lemma 1 indicates it is sufficient that we (strategically) repair  $x_i$  a total of  $\lfloor d(x_i, T) \rfloor$  times.

Some more notations. We let  $x_{i,j}$  denote the **state** where the damage  $x_i$  has taken lies in  $[j-1, j)$ . Note that there are at most  $T$  states:

$$\sum_{i=1}^n \lfloor d(x_i, T) \rfloor \leq \sum_{i=1}^n d(x_i, T) = \sum_{i=1}^n \sum_{t=1}^T d_{t,i} = T.$$

We say time  $t$  is **associated** with state  $x_{i,j}$  if  $(d(x_i, t-1), d(x_i, t)) \cap [j-1, j) \neq \emptyset$ . Finally, we construct a graph and a flow on it as follows:

- $G = (V, E)$ :
  - $V$  includes a source  $a$ , a sink  $b$  (sadly letter  $t$  is taken), “state” nodes  $x_{i,j}$ , and “round” nodes  $r_1, \dots, r_T$ .
  - $E$  includes all edges of forms  $(a, x_{i,j})$  and  $(r_t, b)$ , as well as all edges of form  $(x_{i,j}, r_t)$ , where  $t$  is associated with  $x_{i,j}$ . All edges have capacities 1.
- Flow  $f$ :
  - $f((a, x_{i,j})) = 1$  for all “state” nodes  $x_{i,j}$ .
  - For each edge of form  $(x_{i,j}, r_t)$ :
    - (i) If  $t$  is the only time associated with  $x_{i,j}$ , set  $f((x_{i,j}, r_t)) = d_{t,i}$ .
    - (ii) If  $t$  and  $t+1$  are both associated with  $x_{i,j}$ , set  $f((x_{i,j}, r_t)) = j - b(x_i, t)$  and  $f((x_{i,j+1}, r_{t+1})) = b(x_i, t+1) - j$ .
  - Assign  $f((r_t, b))$  accordingly (i.e. based on flow in).

*Lemma 2.  $f$  is a valid flow on  $G$ .*

*Proof of lemma 2.* Some ambiguity issues first: note that (i) and (ii) above are mutually exclusive. This is because  $d(x_i, t-1), d(x_i, t)$  are at most 1 apart, so all  $t$ 's associated with  $x_{i,j}$  must also be.

In addition, (ii) essentially breaks the  $(t+1)^{\text{th}}$  attack into two separate ones, both of which satisfy (i). Converting all states corresponding to (ii) to (i), we can safely assume that each integer in  $1, \dots, \lfloor d(x_i, T) \rfloor$  is the total damage taken after some round. In particular, the flow out of  $x_{i,j}$  is simply the length of  $[j-1, j)$ , which is 1. On the other hand, the flow into  $r_t$  is  $\sum_{i=1}^n d_{t,i} = 1$ . Therefore, flow conservations as well as edge capacities are both satisfied, and  $f$  is a valid flow. END OF PROOF OF LEMMA 2

It is clear from construction that  $\nu(f) =$  the number of states and that it is already an upper bound of any flow on  $G$ , since it saturates all edges leaving  $a$ . MaxFlow-MinCut theorem therefore states that there exists an *integer* max flow  $f'$  forced to have  $\nu(f') = \nu(f)$ . The matching corresponding to  $f'$  is thus the answer we seek. For each state  $x_{i,j}$ , there exists precisely one outgoing edge to some  $r_t$  with flow 1, and the winning strategy is to repair the  $i^{\text{th}}$  component after round  $t$ .

We are almost done but with one caveat left — (ii) leaves us the concern of missing our repair on  $x_i$  on some damage interval  $[j-1, j)$ . This happens when the following both happen:

- In  $f'$ , the flow leaves  $x_{i,j}$  and goes to  $r_{t_0}$ , where  $t_0 = \operatorname{argmax}_t (r_t \text{ is incident to } x_{i,j})$ , and
- Both  $t$  and  $t+1$  are associated with  $x_{i,j}$ .

While we would like to repair  $x_i$  right when the total damage taken reaches  $j$ , we can't. Instead, the total damage taken jumps from  $d(x_i, t) < j$  to  $d(x_i, t+1) > j$ . Luckily, this will not cost us the game. The repair for damage interval  $[j-2, j-1)$  happens at  $j-2$  the earliest. Since  $d(x_i, t-1) = j - \epsilon < j$  for some  $\epsilon > 0$  and the  $t^{\text{th}}$  attack deals at most 1 damage on  $x_i$ ,  $d(x_i, t) \leq d(x_i, t-1) + 1 < j + 1 - \epsilon$ . This means after the last repair,  $x_i$  has taken at most  $(j + 1 - \epsilon) - (j - 2) = 3 - \epsilon$  damage — indeed we have shield left, so it is perfectly safe to (1) skip repair for damage interval  $[j-1, j)$  due to the condition above but (2) immediately repair afterwards. This concludes the proof!  $\square$