Exercise Set X, Advanced Algorithms 2022

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 Consider a submodular function $f : 2^N \to \mathbb{R}$ over the ground set $N = \{1, 2, 3, 4\}$. What is the value of the Lovàsz extension $\hat{f}(0.75, 0.3, 0.2, 0.3)$ as a function of f?

2 **Online ad allocation.** Alice and Bob started companies selling hand sanitizer, and are now advertising their products online to potential customers $c_1, c_2, \ldots, c_n$, where $c_i \in \mathcal{C}$ for all $i = 1, \ldots, n$. When a customer c_i arrives, they can be shown advertisement for either Alice's or Bob's hand sanitizer – we say that the customer is *allocated* to either Alice or Bob in that case. If S_1 and S_2 are the sets of customers allocated to Alice and Bob respectively at the end of the sequence, Alice will pay $v_1(S_1)$ Francs to the online advertisement engine and Bob will pay $v_2(S_2)$. Here $v_1 : 2^{\mathcal{C}} \to \mathbb{R}_+$ and $v_2 : 2^{\mathcal{C}} \to \mathbb{R}_+$ are non-negative monotone submodular functions. The goal in the online ad allocation problem is to design an allocation rule that maximizes $v_1(S_1) + v_2(S_2)$. In this problem you will analyze the competitive ratio of the greedy algorithm, stated below:

Algorithm 1 Greedy algorithm for online ad allocation

```
S_1 ← ∅, S_2 ← ∅
for i = 1, ..., n do
    if v_1(c_i|S_1) ≥ v_2(c_i|S_2) then
        S_1 ← S_1 ∪ {c_i}                ▷ Allocate i-th customer c_i to Alice
    else
        S_2 ← S_2 ∪ {c_i}                ▷ Allocate i-th customer c_i to Bob
    end if
end for
```

You will prove that greedy achieves a competitive ratio of $1/2$ by induction on n, the number of customers. We now describe the inductive step. Suppose that the first customer is allocated to Alice (the other case is analogous). Define for $S \subseteq \mathcal{C}$ the functions $v'_1(S) = v_1(S|\{c_1\}) = v_1(S \cup \{c_1\}) - v_1(S)$ and $v'_2(S) = v_2(S)$, and let $p = v_1(\{c_1\})$. Let $\mathcal{I} = (v_1, v_2; c_1, \ldots, c_n)$ denote the input instance of the ad allocation problem, and let $\mathcal{I}' = (v'_1, v'_2; c_2, \ldots, c_n)$ denote the instance $\mathcal{I}$ with the first customer removed and the functions v_1, v_2 replaced with v'_1, v'_2. Let ALG denote the value achieved by greedy on $\mathcal{I}$, and let OPT denote the optimal offline solution on $\mathcal{I}$. Similarly, let ALG' denote the value achieved by greedy on $\mathcal{I}'$, and let OPT' denote the optimal offline solution on $\mathcal{I}'$.

2a Prove that $v'_j, j = 1, 2,$ are non-negative monotone submodular functions.

2b Prove that $OPT \leq OPT' + 2p$.

2c Show how to complete the proof using **(b)**.

3 Hypergraph cuts. Let $G = (V, E)$ be a hypergraph with vertex set V and hyperedge set E (every hyperedge $e \in E$ is a subset of V; see Fig. 1 for an illustration). For $S \subseteq V$ the set of hyperedges crossing the cut $(S, V \setminus S)$ is defined as

$$E(S, V \setminus S) = \{e \in E : e \cap S \neq \emptyset \text{ and } e \cap V \setminus S \neq \emptyset\},$$

and the size of the cut $(S, V \setminus S)$ as $|E(S, V \setminus S)|$.

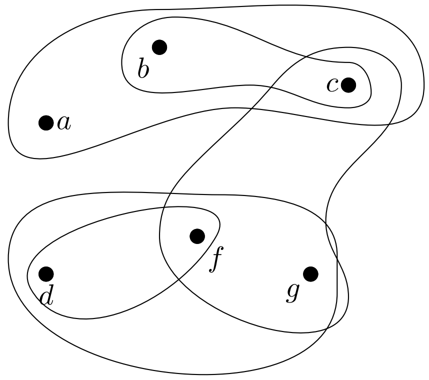

Figure 1. A hypergraph $G = (V, E)$ with $V = \{a, b, c, d, f, g\}$ and hyperedge set $E = \{e_1, e_2, e_3, e_4, e_5\}$, where $e_1 = \{a, b, c\}$, $e_2 = \{b, c\}$, $e_3 = \{d, f\}$, $e_4 = \{c, f, g\}$ and $e_5 = \{d, f, g\}$.

3a Give an algorithm that finds the size of the minimum cut in a given hypergraph G, i.e. outputs

$$\min_{S \subset V, S \neq \emptyset} |E(S, V \setminus S)|.$$

For example, the size of the minimum cut in the hypergraph G in Fig. 1 is 1. There are two minimum cuts: $(\{a\}, \{b, c, d, f, g\})$ and $(\{a, b, c\}, \{d, f, g\})$.

Your algorithm should run in time polynomial in the number of vertices and hyperedges in G.

3b Give a randomized polynomial time algorithm that outputs, given a hypergraph G where every hyperedge contains three vertices, a cut $(S, V \setminus S)$ such that

$$\mathbb{E}[|E(S, V \setminus S)|] \geq (3/4) OPT, \qquad (*)$$

where

$$OPT = \max_{S \subseteq V} |E(S, V \setminus S)|.$$

Note that unlike **3a**, here we are interested in the **maximum** cut. Your algorithm should run in time polynomial in the number of vertices and hyperedges in G, and you should prove that the expected size of the cut that it outputs satisfies $(*)$.