E0 225: Homework 4

Submission Deadline: November 7 (Saturday), 5pm

Instructions

- Both the problems carry equal weight.
- Academic dishonesty/plagiarism will be dealt with severely.
- Late submissions are accepted only with prior approval or medical certificate.

1 (Rank Functions) For a matroid matroid $\mathcal{M} = (E, \mathcal{I})$, the rank function $r : 2^E \mapsto \mathbb{Z}_+$ captures,[1] for each subset $X \subseteq [m]$, the size of the largest (cardinality wise) independent subset within X; formally,

$$r(X) \coloneqq \max\{|I| \; : \; I \subseteq X \text{ and } I \in \mathcal{I}\}.$$

Rank functions of matroids provide a combinatorial generalization of linear-algebraic notions of independence and rank. This exercise shows that—for any given weights $w : E \mapsto \mathbb{R}_+$—the weight of an optimal independent set can be expressed in terms of the rank function.

In particular, order and index the n elements in E such that $w(1) \geq w(2) \geq \ldots \geq w(n)$ and write $E_i \coloneqq \{1, 2, \ldots, i\}$, for $1 \leq i \leq n$, along with $E_0 \coloneqq \emptyset$.

Prove that the weight of an optimal (maximum-weight) independent set, say $\textsc{Opt}$, can be expressed as

$$\textsc{Opt} = \sum_{i=1}^{n} w(i) \cdot \left(r(E_i) - r(E_{i-1}) \right).$$

Recall that the weight of the solution computed by the greedy algorithm is equal to $\textsc{Opt}$.

[1] Here, 2^E denotes the power set of E.

2 (Reverse-Delete Algorithm for Matroid Optimization) Given a matroid $\mathcal{M} = (E, \mathcal{I})$ with nonnegative and distinct weights associated with the elements, $w : E \mapsto \mathbb{R}_+$, we consider reverse-delete algorithms for finding a maximum-weight independent set of $\mathcal{M}$.

Assume that the following oracle access to the matroid $\mathcal{M}$ is provided: for any subset $S \subseteq E$ we can query a (blackbox) subroutine to determine whether S is an independent set (i.e., $S \in \mathcal{I}$ or not).

2a. In terms of finding a maximum-weight independent set, prove or disprove the correctness of the following algorithm that orders the elements in increasing order of weight and deletes them greedily till an independent set is obtained:

Input: The set of n elements E and oracle access to independent sets $\mathcal{I}$ of the matroid $\mathcal{M} = (E, \mathcal{I})$
Output: A maximum weight independent subset, $\arg\max_{S \in \mathcal{I}} w(S)$, where for any subset S the weight $w(S) := \sum_{e \in S} w_e$.

Order the elements of E such that $w_1 < w_2 < w_3 < \ldots < w_n$
Initialize $S = E$
for $i = 1$ to n **do**
    if $S \notin \mathcal{I}$ **then**
        Update $S \leftarrow S \setminus \{i\}$
    else if $S \in \mathcal{I}$ **then**
        Exit loop
    end if
end for
return S

2b. For the matroid $\mathcal{M} = (E, \mathcal{I})$, design an algorithm that, for any given subset $A \subseteq E$, computes the rank $r(A)$ in polynomial time.

2c. Prove or disprove the correctness of the following greedy algorithm

---

Input: The set of n elements E and oracle access to independent sets $\mathcal{I}$ of the matroid $\mathcal{M} = (E, \mathcal{I})$
Output: A maximum weight independent subset, $\arg\max_{S \in \mathcal{I}} w(S)$, where for any subset S the weight $w(S) := \sum_{e \in S} w_e$.

Order the elements of E such that $w_1 < w_2 < w_3 < \ldots < w_n$
Initialize $S = E$
for $i = 1$ to n **do**
    if $r(S \setminus \{i\}) = r(E)$ **then**
        Update $S \leftarrow S \setminus \{i\}$
    end if
end for
return S
