CS 671: Graph Streaming Algorithms and Lower Bounds **Rutgers: Fall 2020**

Problem set 9

(Optional)

Problem 1. For any integer $b \geq 1$, a *fractional b-matching* in a bipartite graph $G = (L \cup R, E)$ is any feasible point in the following linear program on variables $\{x_e \mid e \in E\}$:

$$\forall v \in L, R: \qquad \sum_{e \ni v} x_e \leq b,$$
$$\forall e \in E: \qquad x_e \in [0, 1].$$

Notice that for $b = 1$, this becomes the linear program for bipartite matching we studied in Lecture 9.

The goal of this problem set is to design a streaming algorithm for this problem. For simplicity, we assume that $|L| = |R| = n$ and there is an *integral* b-matching of value $n \cdot b$ in G, i.e., one can fully match all vertices by picking $x_e \in \{0, 1\}$ (not fractionally). Also, we pick $\varepsilon > 0$ to be an *absolute constant*.

(*i*) Specify the feasibility problem we need to solve using MWU framework. What is the oracle?

(*ii*) Design an $\widetilde{O}(n \cdot b)$-space streaming algorithm for implementing a width poly(b) oracle in $O(1)$ passes.

(*iii*) Use the previous part to design an $\widetilde{O}(n \cdot b)$-space streaming algorithm for finding a $(1 + \varepsilon)$-approximate fractional matching (assuming optimum has value $n \cdot b$) in poly$(b, \log n)$ passes.

A harder question. Can you design a poly$(\log n)$ pass algorithm for this problem?