# Assignment 3 - Problem 5.3

Kousshik Raj (17CS30022)

02-10-2020

## 1 Problem 5.3

Consider the following problem: given an undirected graph $G$ and positive integers $k$ and $q$, find a set $X$ of at most k vertices such that $G - X$ has at least two components of size at least $q$. Show that this problem can be solved in time $2^{O(q+k)}n^{O(1)}$

### 1.1 Solution

#### 1.1.1 Randomized Algorithm

Let us look at an one-sided error Monte Carlo algorithm with false negatives for the given instance $(G, k, q)$. Define a uniform random coloring $\chi : V(G) \to \{1, 2\}$ and $V_i = \chi^{-1}(i), \forall i \in \{1, 2\}$. Now, consider the following algorithm.

Now, if there are at least 2 connected components of size at least $q$ in $G[V_1]$, we proceed further. Otherwise, we say that the given instance is a NO-instance. Let $C_1, C_2, ..., C_r$ be the connected components of size at least $q$ in $G[V_1]$. Now, consider $C_i, C_j, \forall 1 \le i < j \le r$. We want to find vertex set $X \subseteq V(G)$ and $|X| \le k$ such that $G - X$ disconnects $C_i$ and $C_j$. This is a standard case of max-flow min-cut problem and can be checked by running $k$ iterations of Ford-Fulkerson algorithm which runs in $n^{O(1)}$ time. If there exists any such $X$, we return that this is a YES-instance with $X$ as a solution. Otherwise, we say that the given instance is a NO-instance.

Let $G$ be a YES-instance, $X$ be one of its solution and $C_1, C_2$ be any two arbitrary components of size at least $q$ in $G - X$. The probability that our algorithm returns YES-instance is at least $2^{-k} \cdot 2^{-q} \cdot 2^{-q} = 2^{-(k+2q)}$ as, if the vertices in $X$ and $q$ vertices in $C_1, C_2$ are colored 2 and 1 respectively, we will arrive at a solution. So, if we perform the above algorithm for $2^{(k+2q)}$ iterations, we will have an algorithm with constant error probability which runs in $2^{O(q+k)}n^{O(1)}$ time.

#### 1.1.2 Deterministic Algorithm (Derandomization)

Now, based on the randomized algorithm seen above, we will develop a deterministic algorithm. Let the given instance be $(G, k, q)$. Assume that the given instance is a YES-instance, $X$ be one of its solution, and $C_1, C_2$ be any two arbitrary components of size at least $q$ in $G-X$. Furthermore, let $C_1^{'} \subseteq C_1, C_2^{'} \subseteq C_2$ such that $|C_1^{'}| = |C_2^{'}| = q$. We will also define a coloring $\chi : V(G) \to \{1, 2\}$ such that $\chi(v) = 1, \forall v \in C_1^{'} \cup C_2^{'}$ and $\chi(v) = 2, \forall v \in X$ and arbitrary assignment for the rest of the vertices.

If we can find such a coloring $\chi$ in a deterministic way, we can follow the previous algorithm to determine the vertex set $Y$ (which may or may not be equal to $X$), such that $|Y| \leq k$ and $C_1$ and $C_2$ are disjoint in $G - Y$. To that extent, we will use the $(n, k + 2q)$-*universal set* $\mathcal{U}$ (Theorem 5.20 in book). Define a coloring $\chi_A$ such that $\chi_A(v) = 1$ if $v \in A$, else $\chi_A(v) = 2$ and let $\mathcal{Z} = \{\chi_A : A \in \mathcal{U}\}$. According to the definition of *universal sets*, $\chi \in \mathcal{Z}$. Thus, we try each coloring in $\mathcal{Z}$ and perform the previous algorithm to determine $Y$. If there is no such $Y$ for any of the coloring, then we can say that $(G, k, q)$ is a NO-instance.

As, $|\mathcal{U}| = O(2^{O(k+q)} logn)$ and $\mathcal{U}$ can be calculated in $O(2^{O(k+q)} nlogn)$, combining with the polynomial algorithm to find the max-flow min-cut, we have a deterministic algorithm which runs in $O(2^{O(k+q)} n^{O(1)})$ and solves the problem instance $(G, k, q)$.