University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, G. Schmid

Algorithm Theory
Exercise Sheet 13

Due: Wednesday, 8th of February, 2023, 11:59 pm

Exercise 1: A Good Approximate MVC *(11 Points)*

Let $G = (A \cup B, E)$ be a bipartite graph, and let $k \geq 1$ be an integer parameter. Assume that M is a matching of G s.t. there exists no augmenting paths of length at most $2k - 1$ w.r.t. M in G. The goal is now to adapt exercise 1 in sheet 10 to compute a $(1 + \frac{1}{k})$-approximate minimum vertex cover of G. To do so we are going to partition the sets A and B as follows: first let $S \subseteq V$, $N(S) := \{u \in V \mid \{u, v\} \in E$ for some $v \in S\}$, and $M(S) = \{u \in V \mid \{u, v\} \in M$ for some $v \in S\}$. Next, let A_0 be the set of unmatched nodes in A and $B_0 := \emptyset$. Thus for $i \in \{1, 2, ..., k\}$, we define the sets $B_i := N(A_{i-1}) \backslash B_{i-1}$ and $A_i := M(B_i)$. Finally, let $B_{k+1} := B \backslash \bigcup_{i=0}^{k} B_i$ and $A_{k+1} := A \backslash \bigcup_{i=0}^{k} A_i$. *NB: you can w.l.o.g assume your bipartite graph to be connected.*

(a) Prove that for every $i \in \{1, 2, ..., k\}$, $C_i := \left(\bigcup_{j=i}^{k+1} A_j\right) \cup \left(\bigcup_{j=1}^{i} B_j\right)$ is a vertex cover. *(2 Points)*

Consider i^* such that $|B_{i^*}| \leq |B_i|$ for all $i \in \{1, 2, ..., k\}$. We will now show that

$$C_{i^*} := \left(\bigcup_{j=i^*}^{k+1} A_j\right) \cup \left(\bigcup_{j=1}^{i^*} B_j\right)$$

is a vertex cover of size at most $(1 + \frac{1}{k})$OPT, where OPT is the size of the minimum vertex cover of G.

(b) Show that there cannot exist an unmatched node in $\bigcup_{i=1}^{k} B_i$. *(2 Points)*

(c) For all $i \in \{1, 2, ..., k\}$ what can you say about the size of A_i and B_i? *(2 Points)*

(d) Show that $|C_{i^*}| \leq (1 + \frac{1}{k}) \cdot |M|$ and deduce that C_{i^*} is our desired vertex cover. *(5 Points)*

Exercise 2: The Densest Subgraph *(9 Points)*

Let $G = (V, E)$ be a graph, $S \subseteq V$, and $E(S) := \{\{u, v\} \in E \mid u, v \in S\}$. We define the density of S to be $den(S) := \frac{|E(S)|}{|S|}$. In the *densest subgraph problem*, the goal is to find a subset $S^* \subseteq V$ that maximizes the $den(S)$ i.e. $den(S^*) := \max_{S \subseteq V} den(S)$. In this exercise, we will study a greedy algorithm that gives a $\frac{1}{2}$-approximation to the problem.

Algorithm 1 Greedy Densest Subgraph ▷ input graph $G = (V, E)$

Let $S := V$, $S' = V$, and $den(S') = den(V)$
while $S \neq \phi$ **do**
    Find $i_{min} \in S$, the vertex of minimum degree in $G(S)$. Delete it from S.
    if $den(S) > den(S')$ **then** $S' = S$
return S'

Consider the following: for each edge $\{i, j\} \in E$, we assign this edge to either i or j arbitrarily. Let $d(i)$ be the edges assigned to $i \in V$. Define $d_{max} := \max_{i \in V} d(i)$.

(a) Show that $\max_{S \subseteq V} den(S) \leq d_{max}$ for any assignement of edges. *(4 Points)*

(b) Consider the following edge assignement where each edge assigns itself to the first incident vertex deleted by the algorithm. Show that $d_{max} \leq 2\alpha$, where $\alpha := den(S')$ and S' is the subset returned by the greedy algorithm. Deduce that the greedy algorithm outputs a $\frac{1}{2}$-approximation to the problem. *(5 Points)*

Hint: use an average argument with respect to the degrees of nodes and the fact that the average degree of a node in graph G is $\frac{\sum_{v \in V} degree(v)}{|V|}$.