

#5 Assignment - CMPT 405

Luiz Fernando Peres de Oliveira - 301288301 - lperesde@sfu.ca

November 29, 2018

#1

Because an edge covering of G is a set $A \subset E$ such that for every node $v \in V$, v is one of the vertices in at least one of the edges in A , we know that there is a polynomial algorithm that describes finding an edge cover of G , for example by simply running *BFS*. However, because finding the smallest possible set of edges $A^* \subset E$ that satisfy an edge covering is an optimization problem, we need to prove that finding such set can be done in polynomial time.

The intuition of our algorithm to minimize the number of edges will be that an edge covering cannot have subpaths of more than two edges. A simple proof would be: let A' be an edge covering of G and let p be a subpath formed by some of the edges of A' , where $|p| > 2$. If p is $a \rightarrow b \rightarrow c \rightarrow d$, then $|A'| > |A^*|$, as there is an extra edge, (b, c) in this case, that could be removed, thus breaking p into two different paths $p_1 = a \rightarrow b$ and $p_2 = c \rightarrow d$ would minimize the edge covering and therefore $|A'|$ cannot equal $|A^*|$.

Because the property above holds true, where for every subpath $p \subset A^*$, $|p| = 1$ or $|p| = 2$, we can solve this problem by first computing a maximum matching on G (therefore covering every p , where $|p| = 1$), and then, for each vertex v left uncovered, covering v by adding an edge connecting to one of the already covered vertices (therefore augmenting the size of some p). Because the second part of the algorithm is trivial ($O(m + n)$, $n = |V|$, $m = |E|$), the total running time of the algorithm is the same as Edmonds algorithm ($O(n^2m)$ or $O(\sqrt{mn})$ with improvements), thus, it can be implemented in polynomial time.

#2

#3

#4

#5