

Algorithms & Complexity: Lecture 14: Coping with **NP**-hardness II

Sam Barrett

March 23, 2021

We have previously seen that **IndependentSet** $\leq_P$ **VertexCover** as if S is an independent set, $V \setminus S$ is a vertex cover. I.e. to convert an independent set to a vertex cover, take its complement and vice versa.

We can therefore say that the time complexity of **IndependentSet** and **VertexCover** is the **same**. Any $f(n)$ time algorithm for one problem also works for the other.

We know that **VertexCover** is a **NP**-hard problem but what if we are not concerned with finding the **minimum** sized vertex cover and instead a cover of some arbitrary size 25?

1 Fixed-parameter tractable (FPT) algorithms

We can define the parameterised vertex cover problem to be the same as the standard vertex cover problem but instead of attempting to find the minimum sized cover we attempt to find a vertex cover of size $\leq k$

We say the parameterised vertex cover problem is fixed-parameter tractable if it can be solved in $f(k) \cdot n^{O(1)}$ time.

Problem 1 (*The k -VC problem*) Given an undirected graph $G = (V, E)$ on n vertices and a parameter k , is there a set $X \subseteq V$ of size $\leq k$ s.t. each edge of G has at least one endpoint in X ?

The standard **VertexCover** problem can then be solved using a series of $O(n)$ calls to the k -VC problem.

If we assume **P** $\neq$ **NP**, we cannot then expect to find an algorithm for k -VC which runs in time polynomial in **both** k and n .

We can construct an algorithm for k -VC which uses bounded-depth search trees.

This algorithm is best visualised as:

If at least one of the leaf nodes at the bottom of the graph has no edges remaining then there exists a vertex cover of size $\leq k$, otherwise one does not exist.

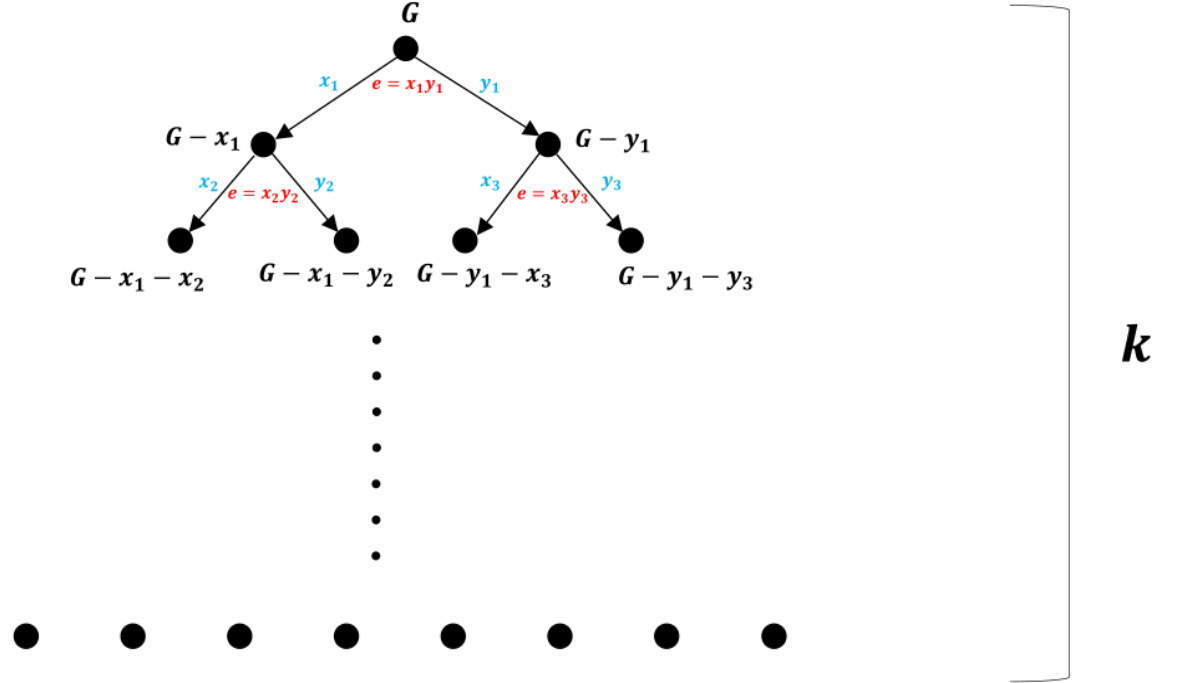


Figure 1: FPT k -VC algorithm

This algorithm can be seen to have a running time in $2^k \cdot n^{O(1)}$. As there are 2^k leaf nodes for a tree of height k and we must spend $n^{O(1)}$ time at each node to determine if any edges remain.