Independent Set and Vertex Cover

Hanan Ayad

1 Independent Set Problem

For a graph $G = (V, E)$, a set of nodes $S \subseteq V$ is called *independent* if no two nodes in S are connected by an edge $e \in E$. The Independent Set problem is to find the largest independent set in a graph. It is not hard to find small independent sets, e.g. a trivial independent set is any single node, but it is hard to find large independent sets.

A simple example of a graph is shown in Figure 1, where the following are two independent sets, $\{A, E, G\}$ and $\{B, C, E, G\}$. The second is the largest possible set.

The decision version of the independent set problem is stated as follows: Given a graph G and a number k, does G contain an independent set of size at least k (i.e., $|S| \geq k$)? The largest k for which the answer is `YES` is the size of the largest independent set in G. Finding such k represents the optimization version of Independent Set. It is evident that the decision version of the Independent Set problem is polynomially reducible to its optimization version. In fact, using binary search, it is possible to solve the decision problem for $O(\log n)$ values of k.

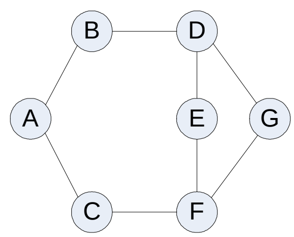

Figure 1: A graph with largest independent set of size 4 and smallest vertex cover of size 3.

2 Vertex Cover Problem

Given a graph $G = (V, E)$, a set of nodes $S \subseteq V$ is called a *vertex cover* if every edge $e \in E$ has at least one end in S. It is not hard to find large vertex covers, e.g., a trivial vertex cover is the set $S = V$. However, it is hard to find small vertex covers. The decision version is stated as follows. Given a graph G and a number k, does G contain a vertex cover of size at most k (i.e., $|S| \leq k$)? The same observations mentioned above on the optimization version of Independent Set apply on Vertex Cover. The different is that Vertex Cover is a minimization problem whereas Independent Set is a maximization problem.

For the graph shown in Figure 1, the following are vertex covers where the second is the smallest possible set $\{B, C, D, F\}$ and $\{A, D, F\}$.

3 Relative Difficulty

Independent set and Vertex Cover were proved to be equally hard, each being polynomially reducible to the other. This is due to the fact that for a given graph G, S is an independent set if and only if the set $V - S$ (called the complement of S) is a vertex cover. In what follows is a proof of this fact.

Proof

If S is an independent set for a given graph $G = (V, E)$, then for any edge $e = (u, v)$ where $e \in E$, at most one of the vertices u and v is in S. Hence, at least one of them must be in $V - S$. Thus, every edge has at least one end in $V - S$. So, $V - S$ must be a vertex cover.

Conversely, if $V - S$ is a vertex cover, if any two nodes u and v in S were connected by an edge e, then neither u nor v would be in $V - S$, which contradicts the initial assumption that $V - S$ is a vertex cover. That is, no two nodes in S can be adjacent and hence S is an independent set. Q.E.D.

Given the fact presented above (S being an independent set if and only if the set $V - S$ is a vertex cover), we can conclude that each of the two problems is polynomially reducible to the other. Below is are formal arguments of this conclusion.

- Independent Set $\leq_p$ Vertex Cover.

  Suppose that we have an efficient algorithm for solving Vertex Cover, it can simply be used to decide whether G has an independent set of size at least k by asking it to determine whether G has a vertex cover of size at most $n - k$.

- Vertex Cover is $\leq_p$ Independent Set.

  Suppose that we have an efficient algorithm for solving Independent Set, it can simply be used to decide whether G has a vertex cover of size at most k, by asking it to determine whether G has an independent set of size at least $n - k$.

Evidently, if any of the Independent Set or Vertex Cover problems is NP-complete, the other must be NP-complete. In fact, no efficient algorithm have been discovered for Independent Set or Vertex Cover. They are known NP-complete problems. The optimization version of each problem is NP-hard. [1]

References

[1] Jon Kleinberg and Éva Tardos. *Algorithm Design*. Addison Wesley, 2006.