Lecture Outline (Week 8)
Topics in Graph Algorithms (CSCI5320-20S)

CAI Leizhen
Department of Computer Science and Engineering
The Chinese University of Hong Kong
lcai@cse.cuhk.edu.hk

April 1, 2020

Keywords: Iterative compression, and iterative expansion.

1. **Iterative compression method** (Reed, Smith and Vetta 2004)

   The main idea is to repeatedly compress a solution to get a smaller one, and a key component of the method is a compression subroutine to construct a k-solution from a $(k+1)$-solution in FPT time.

2. k-Vertex Cover

   Let X be a set of vertices in G. For a subset $I \subseteq X$, the *outside neighbourhood* of I (w.r.t. X), denoted $N^*(I)$, consists of vertices in $V - X$ that are adjacent to some vertices in I, i.e., $N^*(I) = \bigcup_{v \in I} N(v) - X$. The following characterization of a minimum vertex cover can well be a very old folklore in graph theory, and was observed by the author in 2004 when he was desperately looking for simple examples to teach the iterative compression method.

   Lemma 0.1 *A vertex cover X of a graph $G = (V, E)$ is a minimum vertex cover iff for every independent set I of $G[X]$, $|N^*(I)| \geq |I|$.*

   Proof. If $G[X]$ contains an independent set I with $|N^*(I)| < |I|$, then $(X - I) \cup N^*(I)$ is a vertex cover of G smaller than X as all edges covered by I are covered by $X - I$ and $N^*(I)$. Conversely, suppose that G has a smaller vertex cover X'. Let $I = X - X'$. Then I is an independent set of $G[X]$, and $N^*(I) \subseteq X' - X$. Since $|X'| < |X|$, we have $|X' - X| < |X - X'|$ and hence $|N^*(I)| < |I|$. ∎

   The above lemma enables us to determine in FPT-time whether a $(k+1)$-vertex cover of G is a minimum vertex cover, and obtain a smaller one when it is not. The following compression routine takes $O(2^k kn)$ time as at most 2^{k+1} subsets of X need to be examined in the worst case:

Consider each independent set I in a $(k+1)$-vertex cover X, and check if $\lvert N^*(I)\rvert < \lvert I\rvert$. If so we get a smaller vertex cover $(X-I)\cup N^*(I)$, otherwise G has no k-vertex cover.

3. **Three ways to obtain a $(k+1)$-vertex cover**

   (a) *Recursively*: Arbitrarily choose a vertex v in G and recursively solve the problem for $G - v$. If $G - v$ has no solution then neither has G. Otherwise, we obtain a k-vertex cover S, which yields a $(k+1)$-vertex cover $S \cup \{v\}$ for G. We need to compress $O(n)$ times, which results in an $O(2^k kn^2)$-time algorithm.

   (b) *Iteratively*: Order vertices as $v_1, \ldots, v_n$ and let $G_i = G[\{v_1, \ldots, v_i\}]$. Then a k-vertex cover V' of G_i yields a $(k+1)$-vertex cover $V' \cup \{v_{i+1}\}$ of G_{i+1}, which is then compressed into a k-vertex cover. Again, we get an $O(2^k kn^2)$-time algorithm.

   (c) *Approximation*: Use a 2-approximation algorithm for vertex covers to find a k'-vertex cover of G first. If $k' > 2k$ then the answer is no, otherwise we compress solutions at most k times, which gives us an $O(4^k k^2 n)$-time algorithm.

4. **Faster algorithm for k-VERTEX COVER**

   Algorithmically, it is not efficient to consider all possible independent sets I inside X, and in fact it is sufficient to consider maximal independent sets only.

   Lemma 0.2 [Cai 2017]*A vertex cover X of a graph $G = (V, E)$ is a minimum vertex cover iff for every maximal independent set I in X, I is a minimum vertex cover of the induced subgraph $G[I \cup N^*(I)]$.*

   Proof. If there is a maximal independent set $I \subseteq X$ such that the induced subgraph $G[I \cup N^*(I)]$ admits a vertex cover X' of size smaller than I, then $(X - I) \cup X'$ is a smaller vertex cover of G as $X - I$ covers all edges outside $G[I \cup N^*(I)]$.

   Conversely, suppose that G has a vertex cover X' smaller than X. By Lemma 0.1, there is an independent set $I' \subseteq X$ satisfying $|N^*(I')| < |I'|$. Let $I \subseteq X$ be a maximal independent set containing I'. Then $G[I \cup N^*(I)]$ is a bipartite graph as $N^*(I) \subseteq V - X$ is an independent set, and hence admits $(I - I') \cup N^*(I')$ as a vertex cover, which is smaller than I as $|N^*(I')| < |I'|$. ∎

   With Lemma 0.2 in hand, we obtain a different compression routine which examines fewer subsets of X:

Consider each maximal independent set I in a $(k+1)$-vertex cover X, and check if $G[I \cup N^*(I)]$ has a vertex cover X' smaller than I. If so we obtain a smaller vertex cover $(X - I) \cup X'$, otherwise G has no k-vertex cover.

   The above compression routine takes $O(3^{k/3} kn^{1.5})$ time as $G[X]$ can contain $\Omega(3^{k/3})$ maximal independent sets, which can be listed in time $O(3^{k/3}(m+n))$ and it takes $O(m\sqrt{n})$ time to find a minimum vertex cover in a bipartite graph. Recall that $m = O(kn)$ for graphs with k-vertex covers.

   Cai, Vertex Covers Revisited: Indirected Certificates and FPT Algorithms, arXiv preprint arXiv:1807.11339 (2018).

5. FEEDBACK VERTEX SET

   Determine whether a graph $G = (V, E)$ contains at most k vertices whose deletion results in an acyclic graph, i.e., a forest.

Key step in the compression subroutine: Given a $(k+1)$-fvs X, determine whether it is possible to obtain a k-fvs $X' \subseteq V - X$ in FPT time.

By definition, $G - X$ is a forest. In fact, $G[X]$ is also a forest consisting of at most k trees as it is an induced subgraph of $G - X'$, which is a forest.

Task: Find $\leq k$ vertices X' from $G - X$ to form a fvs. For each vertex $v \in V - X$, we either put it into X' or move it to X side without introducing a cycle.

Let v be a vertex in $V - X$ that has at least two neighbors in X.

Case 1. If at least two neighbors of v are from the same tree of $G[X]$, then we put v into X', i.e., delete v and reduce k by 1.

Case 2. If Case 1 doesn't apply, then neighbors of v in X are from different trees. We use bounded search tree idea: either put v into X' (i.e., delete v and reduce k by 1) or move v into X which merges trees in $G[X]$ containing neighbors of v into a single tree.

Case 3. If Cases 1 and 2 don't apply, then every vertex in $V - X$ has at most one neighbor in $G[X]$. Since $G - X$ is a forest, every leaf of $G - X$ has degree at most 2. Delete every degree-1 leaf and for a degree-2 leaf v, delete v and add an edge between the two neighbors of v.

For details, see Chen et. al., Improved algorithms for feedback vertex set problems, J. of Comp. and System Sci. 74 (2008) 1188-1198.

6. k-Vertex Bipartization (Reed, Smith and Vetta 2004)

Remove at most k vertices (called *odd cycle transversal*) from graph G to form a bipartite graph, or equivalently, to destroy all odd cycles in G.

To obtain an FPT-time compression subroutine, we consider the following task for an odd cycle transversal O of G: determine if O is a minimum odd cycle transversal, and find a smaller one if it is not. For this purpose, we construct a bipartite graph G' from G and O as follows. Note that $G - O$ is a bipartite graph with bipartition (X, Y).

(a) Replace each vertex $v \in O$ by two new vertices x_v and y_v.

(b) Replace each edge xv, where $x \in X$, by edge xy_v, and each edge yv, where $y \in Y$, by edge yx_v.

(c) Replace each edge uv, where $u, v \in O$, by edges $x_u y_v$ and $x_v y_u$.

Note that G' is a bipartite graph with bipartition $X' = X' \cup \{x_v : v \in O\}$ and $Y' = Y \cup \{y_v : v \in O\}$. For a subset O' of O, a partition of $\{x_v, y_v : v \in O'\}$ into S and $\overline{S}$ is *valid* if for every $v \in O'$, S contains exactly one of $\{x_v, y_v\}$.

Theorem *An odd cycle transversal O of G is a minimum one iff for every valid partition $(S, \overline{S})$ of any subset O' of O, there are $|O'|$ vertex disjoint paths in $G' - \{x_v, y_v : v \in O - O'\}$ between S and $\overline{S}$.*

We can use the above lemma to test whether an odd cycle transversal O with $k+1$ vertices is a minimum one by considering all possible valid partition $(S, \overline{S})$ ($O(2^k)$ in total) of each possible subset O' ($O(2^k)$ in total) of O. By solving a maximum flow problem for each $(S, \overline{S})$, we can determine in $O(km)$ time whether there are $|O'|$ vertex disjoint paths between S and $\overline{S}$ in $G' - \{x_v, y_v : v \in O - O'\}$, and find a minimum vertex cutset C to separate S and $\overline{S}$ when the answer is no. Therefore we can get a smaller odd cycle transversal $(O - O') \cup C$ in $O(4^k km)$ time when it exists. We need $O(n)$ iterations of the compression routine, and therefore we have an $O(4^k kmn)$-time algorithm.

For details, see Reed, Smith, and Vetta, Finding odd cycle transversals, Oper. Res. Lett. 32(4) (2004) 299-301.

7. **Iterative expansion** (Cai and Leung 2004)

   In a similar spirit, we can use expansion instead of compression to solve some fixed-cardinality optimization problems. The idea is to iteratively extend an optimal i-solution to an optimal $(i+1)$-solution until we get a k-solution.

8. Maximum k-Vertex Cover: Find k vertices in a graph to cover as many edges as possible.

   For a subset V' of vertices, let $e(V')$ denote the number of edges covered by V'. A set of k vertices S is called a *maximum k-vertex cover* if it covers the maximum number of edges. A maximum k-vertex cover S is *extendible* if we can add one new vertex to S to obtain a maximum $(k+1)$-vertex cover S'.

   Lemma *Let S be an arbitrary maximum k-vertex cover of a graph G. If S is not extendible, then for any maximum $(k+1)$-vertex cover S' of G, every vertex in $S' - S$ is adjacent to some vertices of $S - S'$.*

   The above lemma indicates that in order to extend a maximum k-vertex cover S to a maximum $(k+1)$-vertex cover, we can essentially focus on the open neighborhood $N(S)$ of S, which enables us to solve Maximum k-Vertex Cover for degree-bounded graphs and planar graphs.

9. Degree-bounded graphs

   Let G be a graph of maximum degree d, where d is a constant. Initially, S contains a single vertex v of maximum degree. To extend a maximum i-vertex cover S to a maximum $(i+1)$-vertex cover S', we either add a vertex of maximum degree in $G - S$ to S or replace a t-subset T, $1 \le t \le k-1$, of S by a $(t+1)$-subset T' of $N(S)$.

   If S is not extendible, then there are at most $2^i - 1$ subsets of S need to be considered. For each t-subset T, $1 \le t \le k-1$, of S, we need to find a $(t+1)$-subset T' of vertices in $N(S-T)$ to maximize $e(T \cup T')$. Since $N(S-T)$ contains at most dk vertices, there are at most 2^{dk} possibilities for such a T'. Therefore it takes $O(2^{(d+1)k} km)$ time to find a maximum k-vertex cover in a graph of maximum degree d.