Lecture Outline 13
Topics in Graph Algorithms (CSCI5320-19S)

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

March 27, 2019

Keywords: Random separation.

1. Covering Exactly k Edges (Cai, Chan and Chan 2006)

   Find a set of vertices in a graph G to cover exactly k edges.

   W.l.o.g., we may assume that G has no isolated vertices. Color each vertex either red or green each with probability 1/2 to form a random partition (V_g, V_r) of V. A solution S is "well-colored" if all vertices of S are green and all vertices in $N(S)$ are red. Since S covers exactly k edges, we see that $|S| \leq k$ and $|N(S)| \leq k$. Therefore a solution has probability at least 2^{-2k} to be well-colored.

   The problem of finding a well-colored solution is equivalent to the problem of finding a collection $\mathcal{H}'$ of green components such that the total number of edges in G covered by vertices in $\mathcal{H}'$ is exactly k. To find such a collection $\mathcal{H}'$, we compute, for each green component H_i, the number e_i of edges in G covered by vertices in H_i. Since for any two green components H_i and H_j, the number of edges covered by vertices in $H_i \cup H_j$ equals $e_i + e_j$, we can obtain such a collection $\mathcal{H}'$ in $O(kn)$ time by using the standard dynamic programming algorithm for the Subset Sum problem. Therefore we can solve the problem in $O(4^k(m+n))$ expected time when G contains such a vertex cover and, using $(n, 2k)$-universal sets for derandomization, $O(4^k(2k)^{O(\log k)}(m+n)\log n)$ worst-case time, which is $O((m+n)\log n)$ for each fixed k.

2. Maximum Weight Independent k-Set for planar graphs (Cai, Chan and Chan 2006)

   Let $G = (V, E; w)$ be a weighted planar graph with $w : V \to Z^+$, and consider the problem of finding a maximum weight independent set of size k in G.

   Since a planar graph always contains a vertex of degree at most 5, we can orient edges of G in $O(n)$ time so that the outdegree of each vertex is at most 5. Color each vertex either red or green each with probability 1/2 to form a random partition (V_g, V_r) of V. For a maximum weight independent k-set V', there is at least 2^{-6k} chance that all vertices of V' are green and all out-neighbors of V' are red. Therefore, with probability at least 2^{-6k}, V' consists of sinks of $G[V_g]$ and thus k sinks of largest weights in $G[V_g]$. We can easily find such V' in $O(kn)$ time, and thus, with probability at least 2^{-6k}, we can find a maximum weight independent k-set in $O(kn)$ time. We can derandomize the algorithm by using $(n, 6k)$-universal sets to obtain a deterministic algorithm that runs in $O(n \log n)$ time for each fixed k.

3. Three methods for Maximum k-Vertex Domination on degree-bounded graphs $G = (V, E)$ (Cai, Chan and Chan 2006). For a subgraph H of G, let $\rho(H)$ be the number of vertices dominated by vertices of H. An optimal k-solution S is "well-colored" if all vertices of S are green and all vertices in $N(S)$ are red. However, two green components H and H' may not satisfy $\rho(H \cup H') = \rho(H) + \rho(H')$, and it seems not easy to find a well-colored solution. However, we may redefine "well-colored solutions" to obtain the following three algorithms.

   Algorithm 1: *$N[S]$ green and $N^2(S)$ red.* For each green component H_i, let $h_i = |V(H_i)|$, and find minimum number k_i of vertices that dominate all vertices in H_i and have no red neighbors. Then for any two green components H_i and H_j, we have $\rho(H_i \cup H_j) = h_i + h_j$, and we can use DP for (0,1)-knapsack to find a well-colored solution.

   Algorithm 2: Use three colors – *S green, $N(S)$ blue and $N^2(S)$ red.* For each green-blue component H_i, let k_i be the number of green vertices and $h_i = |V(H_i)|$. Then for any two green-blue components H_i and H_j, we have $\rho(H_i \cup H_j) = h_i + h_j$.

   Algorithm 3: *S green, and both $N(S)$ and $N^2(S)$ red.* Then for any pair $\{H_i, H_j\}$ of green components whose distance $d_G(H_i, H_j)$ in G is at most two, we have $V(H_i) \subseteq S$ iff $V(H_j) \subseteq S$. This fact allows us to merge green components into clusters of green components, called *2-cgc*s. Construct a graph G_H by taking each green component H_i as a vertex and $H_i H_j$ as an edge if $d_G(H_i, H_j) \leq 2$. Then each connected component of G_H corresponds to a 2-cgc of G, and S consists of a collection of 2-cgcs. For each 2-cgc C_i, let $k_i = V(C_i)$ and $h_i = \rho(C_i)$. Then for any two 2-cgcs C_i and C_j, $\rho(C_i \cup C_j) = h_i + h_j$.

4. The idea of **Algorithm 3** can be generalized to require that a "well-colored" solution S satisfies S green and all $N^t(S)$ red for $1 \leq t \leq i$, which forms an *i-separating partition* for a solution. We can then merge green components into *i-cgc*s, and a "well-colored" solution will consists of a collection of i-cgcs. This idea leads to the following general theorem.

   Theorem (Cai, Chan and Chan 2006) Let $G = (V, E)$ be a graph of maximum degree d, where d is a constant. Let $\phi : 2^V \to R \cup \{-\infty, +\infty\}$ be an objective function to be optimized. Then it takes FPT time to find k vertices V' in G that optimizes $\phi(V')$ if

(a) For all $V' \subseteq V$ with $|V'| \leq k$, $\phi(V')$ can be computed in FPT time, and

(b) There is an integer i computable in FPT time such that for all $V_1, V_2 \subseteq V$ with $|V_1| + |V_2| \leq k$, if $d(V_1, V_2) > i$ then $\phi(V_1 \cup V_2) = \phi(V_1) + \phi(V_2)$.

5. INDUCE k-PATH for d-degenerate graphs (Cai, Chan and Chan 2006).

   G is d-degenerate if it admits an acyclic orientation G' such that $d^+_{G'}(v) \leq d$ for each vertex v.

   We use random separation to separate an induced k-path from its out-neighbors and then use color coding for the green subgraph to find an induced k-path. For this purpose, we use $k+1$ colors $\{0, 1, \ldots, k\}$ with color 0 for red and colors $\{1, \ldots, k\}$ for an induced k-path.

   (a) To generate a $(k+1)$-coloring, we produce a random partition $\{V_g, V_r\}$ of V, color red vertices V_r by 0 and randomly color green vertices V_g by colors in $\{1, \ldots, k\}$.

   An induced k-path of G is "well-colored" if its vertices are colored from 1 to k along the path, and $N^+_{G'}(P)$ has color 0.

   (b) To find a well-colored induced k-path, we mark vertices as follows.

   For each vertex v with color 1, mark it if all vertices in $N^+_{G'}(v)$ have color 0, except perhaps one vertex of color 2.

   For $i = 2$ to k process vertices v of color i: mark v if there is a marked vertex in $N_G(v)$ of color $i-1$, and all vertices in $N^+_{G'}(v)$ have color 0, except perhaps one vertex of color $i+1$.

   It can be shown that G has a well-colored induced k-path iff there is a marked vertex of color k.