Lecture Outline (Week 9)
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: Color coding, and perfect hash functions.

Color Coding (Alon, Yuster and Zwick 1995)

N. Alon, R. Yuster, and U. Zwick, Color-coding, J. ACM 42(4):844-856, 1995.

The basic idea of this novel method is to use k colors to color elements randomly and then try to find a colorful k-solution, i.e., a k-solution whose elements are in distinct colors. If we can find a colorful k-solution in FPT time, then we can find a k-solution in FPT time with probability $k!/k^k > e^{-k}$. The algorithm can be derandomized by a family of perfect hash functions. The method is particularly useful for parameterized problems whose k-solutions have good structures e.g., linear, cyclic, or treelike.

1. (n, k)**-family of perfect hash functions**: A family $\mathcal{F}$ of functions mapping a domain of size n into a range of size k such that for every k-subset S from the domain, there is a function in $\mathcal{F}$ that is 1-to-1 on S. Based on a construction of Schmidt and Siegel (1990), Naor (1995) gave a construction of an (n, k)-family of perfect hash functions of size $2^{O(k)} \log n$ in time $2^{O(k)} n \log n$.

2. k-Path: $2^{O(k)} m \log n$ time (Alon, Yuster and Zwick 1995)

   Does G contain a path with k vertices?

   Randomly color vertices with colors $\{1, \ldots, k\}$, and call a path *colorful* if colors of its vertices are distinct. Given a coloring $c : V \to \{1, \ldots, k\}$, we can use dynamic programming to determine whether there is a colorful k-path.

   Construct G' from G by adding a new vertex v_0 of color 0 and connect it with every vertex in G. Now the problem is to determine whether there is a colorful $(k+1)$-path in G' that starts at v_0.

   For each vertex v, define $C^i(v) = \{c(P) : P \text{ is a colorful } (v_0, v)\text{-path of length } i\}$, where $c(P)$ is the set of colors of vertices in P. Then G has a colorful k-path iff G' has a vertex v with $C^k(v) \neq \emptyset$. Note $|C^i(v)| \leq \binom{k}{i}$.

Initially, $C^1(v) = \{c(v)\}$ for each vertex v. For $2 \le i < k$,

$$C^{i+1}(v) = \{C \cup \{c(v)\} : u \in N(v), C \in C^i(u) \text{ and } c(v) \notin C\}.$$

Note that we keep track of color sets but not paths. Time: $O(\sum_{i=1}^{k} i\binom{k}{i}m) = O(k2^k m)$.

We can derandomize the algorithm using an (n, k)-family of perfect hash functions to obtain an FPT algorithm with running time $2^{O(k)} m \log n$.

3. Edge-Disjoint Triangle Packing

   Does G contain k edge-disjoint triangles?

   Randomly color edges in k colors, and then find k triangles T_i such that the three edges of T_i all have color i. Let G_i be the graph formed by edges of color i. We only need to find a triangle in each G_i.

   Question: How to derandomize this algorithm?

4. k-Edge Eulerian Subgraph: $2^{O(k)} mn \log n$ time (Cai and Yang 2009)

   Does G contain an Eulerian subgraph consisting of k edges?

   Similar to k-Path but we color edges in k colors. We can find, for any pair (u, v) of vertices, a colorful (u, v)-trail of length k (if it exists) in $O(k2^k m)$ time. To find a colorful k-edge Eulerian subgraph, we consider a colorful (v, v)-trail of length k for each vertex.

5. Maximum k-Vertex Cover: FPT $(1+\epsilon)$-approximation (Marx 2008)

   Let $v_1, \ldots, v_n$ be vertices such that $d(v_i) \ge \ldots \ge d(v_n)$. Set $D = 2\binom{k}{2}/\epsilon$ and consider two cases.

   Case 1. $d(v_1) > D$. Take $v_1, \ldots, v_k$ as our solution. Then the approximation ratio is at least

   $$\frac{\sum_{i=1}^{k} d(v_i) - \binom{k}{2}}{\sum_{i=1}^{k} d(v_i)} \ge 1 - \frac{\binom{k}{2}}{D} = 1 - \frac{\epsilon}{2} > \frac{1}{1+\epsilon}.$$

   Case 2. $d(v_1) \le D$. In this case, we use color coding to derive an FPT algorithm. For each $1 \le l \le kD$, we check whether there are at most k vertices that cover at least l edges. For a given l, we color edges in l colors and consider each possible partition of l colors into k sets $\{C_1, \ldots, C_k\}$. For each such partition, we check whether there is a vertex v_i that covers at least one edge of each color in C_i.

6. Vertex Cover

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

   Let $G = (V, E; f)$ be a vertex coloured graph with $f : V \to \{1, \ldots, k\}$. We use V_i to denote the set of vertices with colour i, and call each V_i a *colour class*. A vertex cover X of G is *colourful* if all vertices in X have distinct colours, i.e., X contains at most one vertex from each colour class.

COLOURFUL VERTEX COVER
INSTANCE: Vertex coloured graph G with colours in $\{1, \ldots, k\}$.
QUESTION: Does G contain a colourful vertex cover?

First we note that the problem is no easier than 2SAT as we can reduce 2SAT to our problem in linear time: For an arbitrary instance (U, C) of 2SAT, we construct a vertex coloured graph G by creating, for each Boolean variable $u_i \in U$, two vertices u_i and $\overline{u}_i$ with colour i and edge $u_i\overline{u}_i$; and adding, for each binary clause $\{x, y\} \in C$, an edge between vertices x and y.

Inspired by the above connection with 2SAT, we reduce COLOURFUL VERTEX COVER to 2SAT to obtain a quadratic algorithm. For this purpose, we construct a Boolean formula $\Phi(G)$ for G as follows:

(a) For each vertex v, introduce a Boolean variable x_v.

(b) For edge set E, let $\Phi(E) = \bigwedge_{uv \in E} x_u \vee x_v$.

(c) For each colour class V_i, let $\Phi(V_i) = \bigwedge_{u,v \in V_i \text{ and } u \neq v} \overline{x_u \wedge x_v}$.

(d) Set $\Phi(G) = \Phi(E) \bigwedge_{i=1}^{k} \Phi(V_i)$.

Theorem 0.1 *A vertex coloured graph G admits a colourful vertex cover iff its corresponding formula $\Phi(G)$ is satisfiable.*

Proof. Clearly COLOURFUL VERTEX COVER is equivalent to the following integer linear programming: For all $v \in V$, find $x_v \in \{0, 1\}$ to satisfy

$$x_u + x_v \geq 1 \text{ for each edge } uv \text{ of } G, \text{ and}$$

$$\sum_{v \in V_i} x_v \leq 1 \text{ for each colour class } V_i \text{ of } G.$$

Note that a vertex v belongs to a colourful vertex cover iff $x_v = 1$.

Now for any $x, y \in \{0, 1\}$, $x + y \geq 1$ is equivalent to $x \vee y$ when we also interpret x and y as Boolean variables. Similarly, $x+y \leq 1$ is equivalent to $\overline{x \wedge y}$. Furthermore, for each colour class V_i, $\sum_{v \in V_i} x_v \leq 1$ is equivalent to $x_u + x_v \leq 1$ for all distinct $u, v \in V_i$. It follows that $\Phi(G)$ is satisfiable iff the above integer linear programming has a solution, and equivalently G admits a colourful vertex cover. ∎

The above theorem enables us to solve COLOURFUL VERTEX COVER in $O(n^2)$ time as $\Phi(G)$ contains $O(n^2)$ binary clauses (note that $\overline{x \wedge y} = \overline{x} \vee \overline{y}$), and 2SAT is solvable in linear time. In fact, we can solve COLOURFUL VERTEX COVER in linear time by a limited backtracking approach.