10. Randomized Algorithms: color coding and monotone local search
COMP6741: Parameterized and Exact Computation

Edward Lee
Semester 2, 2017

Contents
1 Introduction 1
2 Vertex Cover 2
3 Feedback Vertex Set 2
4 Color Coding 3
5 Monotone Local Search 3

1 Introduction

Random Algorithms

• Turing machines do not inherently have access to randomness.
• Assume algorithm is also given access apart to a stream of random bits.
• With $r$ random bits, the probability space is the set of all $2^r$ possible strings of random bits (with uniform distribution).

Monte Carlo algorithms

Definition 1. A Monte Carlo algorithm is an algorithm whose output is incorrect with probability at most $p$.

• A one sided error means that an algorithm’s input is incorrect only on true outputs, or false outputs but not both.

• A false negative Monte Carlo algorithm is always correct when it returns false.

Suppose we have an algorithm $A$ for a decision problem which:

• If no-instance: returns “no”.

• If yes-instance: returns “yes” with probability $p$.

Algorithm $A$ is a one-sided Monte Carlo algorithm with false negatives.

Problem
Suppose $A$ is a one-sided Monte Carlo algorithm with false negatives, that with probability $p$ returns “yes” when the input is a yes-instance. How can we use $A$ and design an a new algorithm which ensures a new success probability of a constant $C$?
Amplification

**Theorem 2.** If a one-sided error Monte Carlo Algorithm has success probability at least $p$, then repeating it independently $\lceil \frac{1}{p} \rceil$ times gives constant success probability. In particular if $p = \frac{1}{f(k)}$ for some computable function $f$, then we get an FPT one-sided error Monte Carlo Algorithm with additional $f(k)$ overhead in the running time bound.

## 2 Vertex Cover

For a graph $G = (V,E)$ a vertex cover $X \subseteq V$ is a set of vertices such that every edge is adjacent to a vertex in $X$.

<table>
<thead>
<tr>
<th>VERTEX COVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input:</td>
</tr>
<tr>
<td>Parameter:</td>
</tr>
<tr>
<td>Question:</td>
</tr>
</tbody>
</table>

**Theorem 3.** There exists a randomized algorithm that, given a Vertex Cover instance $(G,k)$, in time $2^k n^{O(1)}$ either reports a failure or finds a vertex cover on $k$ vertices in $G$. Moreover, if the algorithm is given a yes-instance, it returns a solution with constant probability.

## 3 Feedback Vertex Set

A feedback vertex set of a multigraph $G = (V,E)$ is a set of vertices $S \subseteq V$ such that $G - S$ is acyclic.

<table>
<thead>
<tr>
<th>FEEDBACK VERTEX SET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input:</td>
</tr>
<tr>
<td>Parameter:</td>
</tr>
<tr>
<td>Question:</td>
</tr>
</tbody>
</table>

- Recall 5 simplification rules for Feedback Vertex Set.

**Lemma 4.** Let $G$ be a multigraph on $n$ vertices, with minimum degree at least 3. Then, for every feedback vertex set $X$ of $G$, at least 1/3 of the edges have at least one endpoint in $X$.

**Random Algorithm**

**Theorem 5.** There is a randomized algorithm that, given a Feedback Vertex Set instance $(G,k)$, in time $6^k n^{O(1)}$ either reports a failure or finds a feedback vertex set in $G$ of at most $k$. Moreover, if the algorithm is given a yes-instance, it returns a solution with constant probability.

**Lemma 6.** Let $G$ be a multigraph on $n$ vertices, with minimum degree 3. For every feedback vertex set $X$, then at least $\frac{1}{2}$ of the edges of $G$ have at least one endpoint in $X$.

**Hint:** Let $H = G - X$ be a forest. The statement is equivalent to:

$$|E(G) \setminus E(H)| > |E(G)| > |V(H)|$$

Let $J \subseteq E(G)$ denote edges with one endpoint in $X$, and the other in $V(H)$. Show:

$$|J| > |V(H)|$$

**Random Algorithm 2**

**Lemma 7.** There exists a randomized algorithm that, given a Feedback Vertex Set instance $(G,k)$, in time $4^k n^{O(1)}$ either reports a failure or finds a path on $k$ vertices in $G$. Moreover, if the algorithm is given a yes-instance, it returns a solution with constant probability.

**Corollary 8.** Given a Feedback Vertex Set instance $(G,k)$, in time $4^k n^{O(1)}$ there is an algorithm that either reports a failure or if given a yes-instance finds a feedback vertex set in $G$ of size at most $k$ with constant probability.
4 Color Coding

Longest Path

A simple path is a sequence of edges which connect a sequence of distinct vertices.

<table>
<thead>
<tr>
<th>LONGEST PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: Graph G, integer k</td>
</tr>
<tr>
<td>Parameter: k</td>
</tr>
<tr>
<td>Question: Does G have a simple path of size k?</td>
</tr>
</tbody>
</table>

Problem

- Show that Longest Path is NP-hard.

Color Coding

Lemma 9. Let U be a set of size n, and let X ⊆ U be a subset of size k. Let χ : U → [k] be a coloring of the elements of U, chosen uniformly at random. Then the probability that the elements of X are colored with pairwise distinct colors is at least e^{-k}.

Colorful Path

A path is colorful if all vertices of the path are colored with pairwise distinct colors.

Lemma 10. Let G be an undirected graph, and let χ : V(G) → [k] be a coloring of its vertices with k colors. There exists a deterministic algorithm that checks in time 2^k n^{O(1)} whether G contains a colorful path on k vertices and, if this is the case, returns one such path.

Longest Path

Theorem 11. There exists a randomized algorithm that, given a Longest Path instance (G, k), in time (2e^k n^{O(1)}) either reports a failure or finds a path on k vertices in G. Moreover, if the algorithm is given a yes-instance, it returns a solution with constant probability.

5 Monotone Local Search

Exact Exponential Algorithms vs Parameterized Algorithms

<table>
<thead>
<tr>
<th>Exact Exponential Algorithms</th>
<th>Parameterized Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find exact solutions with respect to parameter n, the input size.</td>
<td>Include parameter k, commonly the solution size.</td>
</tr>
<tr>
<td>Feedback Vertex set O(1.7347^n) [Fomin, Todonca and Villanger 2015]</td>
<td>Feedback Vertex Set: O(3.592^k) [Kociumaka and Pilipczuk 2013]</td>
</tr>
<tr>
<td>Running Time: O(α^n n^{O(1)})</td>
<td>Running Time: O(f(k) · n^{O(1)})</td>
</tr>
</tbody>
</table>

Can we use Parameterized Algorithms to design fast Exact Exponential Algorithms?

Subset Problems

An implicit set system is a function Φ with:

- Input: instance I ∈ {0, 1}^*, |I| = N
- Output: set system (U_I, F_I):
  - universe U_I, |U_I| = n
  - family F_I of subsets of U_I
**Φ-SUBSET**
Input: Instance I
Question: Is $|\mathcal{F}_I| > 0$?

**Φ-EXTENSION**
Input: Instance $I$, a set $X \subseteq U_I$, and an integer $k$
Question: Does there exist a subset $S \subseteq (U_I \setminus X)$ such that $S \cup X \in \mathcal{F}_I$ and $|S| \leq k$?

**Algorithm**
Suppose Φ-EXTENSION has a $O^*(c^k)$ time algorithm $B$.

**Algorithm for checking whether contains a set of size $k$**
- Set $t = \max \left(0, \frac{ck-n}{c-1}\right)$
- Uniformly at random select a subset $X \subseteq U_I$ of size $t$
- Run $B(I, X, k-t)$

Running time: [Fomin, Gaspers, Lokshtanov & Saurabh 2016]

$$O^* \left( \binom{n}{t} \cdot c^{k-t} \right) = O^* \left( 2 - \frac{1}{c} \right)^n$$

**Intuition**

**Brute-force randomized algorithm**
- Pick $k$ elements of the universe one-by-one.
- Suppose $\mathcal{F}_I$ contains a set of size $k$.

Success probability:

$$\frac{k}{n} \cdot \frac{k-1}{n-1} \cdot \frac{k-t}{n-t} \cdot \frac{2}{n-(k-2)} \cdot \frac{2}{n-(k-1)} = \frac{1}{\binom{n}{k}} \|rac{1}{c}$$

**Theorem 12.** If there exists an algorithm for Φ-EXTENSION with running time $c^k n^{O(1)}$ then there exists a randomized algorithm for Φ-SUBSET with running time $(2 - \frac{1}{c})^n \cdot n^{O(1)}$

- Can be derandomized at the expense of a multiplicative $2^{n(1)}$ factor in the running time.

**Theorem 13.** For a graph $G$ there exists a randomized algorithm which finds a smallest feedback vertex set in time $(2 - \frac{1}{3.592})^n \cdot n^{O(1)} = 1.7217^n \cdot n^{O(1)}$.

**References**

**Bonus Slides 1**

**1-Regular Deletion**

<table>
<thead>
<tr>
<th>Input:</th>
<th>Graph $G = (V, E)$, integer $k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter:</td>
<td>$k$</td>
</tr>
<tr>
<td>Question:</td>
<td>Does there exist $X \subseteq V$ with $</td>
</tr>
</tbody>
</table>

- Design a randomized FPT algorithm with running time $O^*(4^k)$

**Bonus Slides 2**

**Triangle Packing**

<table>
<thead>
<tr>
<th>Input:</th>
<th>Graph $G$, integer $k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter:</td>
<td>$k$</td>
</tr>
<tr>
<td>Question:</td>
<td>Does $G$ have $k$-vertex disjoint triangles?</td>
</tr>
</tbody>
</table>

- Design a randomized FPT algorithm for **Triangle Packing**.