Testing cycle-freeness in bounded-degree graphs

Oded Goldreich and Dana Ron

Algorithmica **32** (2002) 302–343.

Speaker: Joseph, Chuang-Chieh Lin
Supervisor: Professor Maw-Shang Chang

Computation Theory Laboratory
Department of Computer Science and Information Engineering
National Chung Cheng University, Taiwan

September 28, 2010

Outline

1 Background on property testing

2 Cycle-freeness

3 A two-sided-error property tester for cycle-freeness

Outline

1 Background on property testing

2 Cycle-freeness

3 A two-sided-error property tester for cycle-freeness

- Try to answer "yes" or "no" for the following *relaxed* decision problems by observing only a small fraction of the input.
  - Does the input **satisfy a designated property**, or
  - is ϵ**-far from satisfying the property**?

Background on property testing

- Try to answer "yes" or "no" for the following *relaxed* decision problems by observing only a small fraction of the input.
  - Does the input **satisfy a designated property**, or
  - is **ϵ-far from satisfying the property**?

Background on property testing (contd.)

- In property testing, we use ϵ-far to say that the input is far from a certain property.
- ϵ: the least fraction of the input needs to be modified.
- For example:
  - A sequence of integers $L = (0, 2, 3, 4, 1)$.
  - Allowed operations: integer deletions
  - L is 0.2-far from being monotonically nondecreasing.

Background on property testing (contd.)

- In property testing, we use ϵ-far to say that the input is far from a certain property.
- ϵ: the least fraction of the input needs to be modified.
- For example:
  - A sequence of integers $L = (0, 2, 3, 4, 1)$.
  - Allowed operations: integer deletions
  - L is 0.2-far from being monotonically nondecreasing.

The model for bounded-degree graphs

- Graph model: adjacency **list** for graphs with vertex-degree bounded by d.
  - It takes $O(1)$ time to access to a function $f_G : [n] \times [d] \mapsto [n] \cup \{*\}$.
    - The value $f_G(v, i)$ is the ith neighbor of v or a special symbol '$*$' if v has less than i neighbors.
- ϵ-far from satisfying a graph property $\mathbb{P}$:
  - one has to modify $> \epsilon dn$ entries in f_G (i.e., $> \epsilon dn/2$ edges) to make the input graph satisfy $\mathbb{P}$.

The model for bounded-degree graphs

- Graph model: adjacency **list** for graphs with vertex-degree bounded by d.
  - It takes $O(1)$ time to access to a function $f_G : [n] \times [d] \mapsto [n] \cup \{*\}$.
    - The value $f_G(v, i)$ is the ith neighbor of v or a special symbol '$*$' if v has less than i neighbors.
- ϵ-far from satisfying a graph property $\mathbb{P}$:
  - one has to modify $> \epsilon dn$ entries in f_G (i.e., $> \epsilon dn/2$ edges) to make the input graph satisfy $\mathbb{P}$.

Background on property testing (contd.)

- The complexity measure: queries & running time.
- The complexity (say $q(n, d, \epsilon)$) is asked to be sublinear in $|V| = n$.
  - $q(n, d, \epsilon) = o(f(n))$ if $\lim_{n \to \infty} \frac{q(n,d,\epsilon)}{f(n)} \to 0$, where ϵ and d are viewed as constants.

Background on property testing (contd.)

- The complexity measure: queries & running time.
- The complexity (say $q(n, d, \epsilon)$) is asked to be sublinear in $|V| = n$.
  - $q(n, d, \epsilon) = o(f(n))$ if $\lim_{n \to \infty} \frac{q(n,d,\epsilon)}{f(n)} \to 0$, where ϵ and d are viewed as constants.

Property testers

- A property tester for $\mathbb{P}$ is an algorithm utilizing sublinear queries such that:
  - if the input satisfies $\mathbb{P}$: answers "yes" with probability $\geq 2/3$ ($1 \rightarrow$ one-sided error);
  - if the input is ϵ-far from satisfying $\mathbb{P}$: answers "no" with probability $\geq 2/3$.

Background on property testing (contd.)

- Unlike testing graph properties in the *adjacency-matrix model*, only a few, very simple graph properties are known to be **testable** (i.e., query complexity is independent of n).
- For most of nontrivial graph properties, super-constant lower bounds exist.
  - k-colorability:
    - $\mathbf{\Omega(n)}$.
  - cycle-freeness: (We talk about it today)
    - $\mathbf{O(\frac{1}{\epsilon^3} + \frac{d}{\epsilon^2})}$ (two-sided error);
    - $\mathbf{\Omega(\sqrt{n})}$ (one-sided error).
  - having a vertex-cover of size ρn for a fixed $\rho > 0$:
    - $\mathbf{\Omega(n)}$.
  - having a dominating set of size ρn for a fixed $\rho > 0$:
    - **??** (known to be **non-testable**).
  - ...

Background on property testing (contd.)

- Unlike testing graph properties in the *adjacency-matrix model*, only a few, very simple graph properties are known to be **testable** (i.e., query complexity is independent of n).
- For most of nontrivial graph properties, super-constant lower bounds exist.
  - k-colorability:
    - $\mathbf{\Omega(n)}$.
  - cycle-freeness: (We talk about it today)
    - $\mathbf{O}(\frac{1}{\epsilon^3} + \frac{d}{\epsilon^2})$ (two-sided error);
    - $\mathbf{\Omega(\sqrt{n})}$ (one-sided error).
  - having a vertex-cover of size ρn for a fixed $\rho > 0$:
    - $\mathbf{\Omega(n)}$.
  - having a dominating set of size ρn for a fixed $\rho > 0$:
    - **??** (known to be **non-testable**).
  - ...

Outline

1 Background on property testing

2 Cycle-freeness

3 A two-sided-error property tester for cycle-freeness

Cycle-freeness

- A graph is cycle-free if it does not contain a cycle as a subgraph (or an induced subgraph).
- A connected graph with no cycles is a tree.
- A connected n-vertex graph with $n - 1$ edges is a tree.
- An n-vertex graph with no cycles is a forest.
  - A forest has $n - k$ edges (k: the number of components in the graph).

Deterministic: $O(n)$ time

- Using DFS, to determine if an n-vertex graph G has a cycle can be done in $O(n)$ time.

Outline

1 Background on property testing

2 Cycle-freeness

3 A two-sided-error property tester for cycle-freeness

A property tester for cycle-freeness in bounded-degree graphs

Algorithm: cycle-free-tester

Input: $G = (V, E)$ in an adjacency-list with bounded-degree d, $0 < \epsilon < 1$.

Uniformly and independently select $\ell = 2^{13}/\epsilon^2$ vertices from V;
for each selected vertex s **do**
  Perform a BFS starting from s until $\frac{8}{\epsilon d}$ vertices are reached or no more new vertices can be reached;
end for
if any of the above searches found a cycle **then**
  Output REJECT;
end if
Let $\hat{n}$ denote the number of vertices in the sample that belong to connected components of size $\geq \frac{8}{\epsilon d}$;
Let $\hat{m}$ denote *half* the sum of their degrees;
if $\frac{\hat{m}-\hat{n}}{\ell} \geq \frac{\epsilon d}{16}$ **then**
  Output REJECT;
else
  Output ACCEPT;
end if

Time complexity

- Steps 1–7 (mainly BFS) totally takes time $O\left(\ell \cdot \frac{1}{\epsilon d} \cdot d\right) = O\left(\frac{1}{\epsilon^2} \cdot \frac{1}{\epsilon}\right) = O\left(\frac{1}{\epsilon^3}\right)$.
- Steps 8–14 takes at most $\ell \cdot d = O\left(\frac{d}{\epsilon^2}\right)$ time.
  - Calculation of sum of degrees of the sampled ℓ vertices.

Rough ideas of the tester

- If G is cycle-free, then each of its components is a tree.
- If G is ϵ-far from being cycle-free, then it has many more edges within its components, where these edges (say **superfluous edges**) create cycles.

Rough ideas of the tester (contd.)

- If many superfluous edges reside in "small" components,
  - $\Rightarrow$ many vertices are in these small components ($\because$ bounded degree).
  - $\Rightarrow$ Uniformly select a large enough number of vertices, with high probability we catch such a vertex and then by performing a (bounded) search we can find a cycle.

- If many superfluous edges reside in "big" components, we cannot exhaustively search in such components.
  - For this case, we count the sampled *vertices* belonging to big components and and the *edges* incident them.
  - The *discrepancy* between such edge count and the vertex count is believed to be large.
    - This is due to that the number of big components is relatively small.

Rough ideas of the tester (contd.)

- If many superfluous edges reside in "small" components,
  - $\Rightarrow$ many vertices are in these small components ($\because$ bounded degree).
  - $\Rightarrow$ Uniformly select a large enough number of vertices, with high probability we catch such a vertex and then by performing a (bounded) search we can find a cycle.

- If many superfluous edges reside in "big" components, we cannot exhaustively search in such components.
  - For this case, we count the sampled *vertices* belonging to big components and and the *edges* incident them.
  - The *discrepancy* between such edge count and the vertex count is believed to be large.
    - This is due to that the number of big components is relatively small.

Rough ideas of the tester (contd.)

- If many superfluous edges reside in "small" components,
  - ⇒ many vertices are in these small components ($\because$ bounded degree).
  - ⇒ Uniformly select a large enough number of vertices, with high probability we catch such a vertex and then by performing a (bounded) search we can find a cycle.

- If many superfluous edges reside in "big" components, we cannot exhaustively search in such components.
  - For this case, we count the sampled *vertices* belonging to big components and and the *edges* incident them.
  - The *discrepancy* between such edge count and the vertex count is believed to be large.
    - This is due to that the number of big components is relatively small.

Before we proceed with the proof of correctness of the tester, let us see the following useful observations first.

Discrepancy between the edge count and the vertex count

- A component is small (resp., big): it contains $< \frac{8}{\epsilon d}$ vertices (resp., $\geq \frac{8}{\epsilon d}$ vertices).
- Some further notations:
  - t: the number of big components in G.
  - n': the number of vertices in big components in G.
  - m': the number of edges in big components in G.
  - Big: the set of vertices in big components in G.

Observation*

For any graph G, we have $\left|\frac{\hat{m}-\hat{n}}{\ell} - \frac{m'-n'}{n}\right| \leq \frac{\epsilon d}{16}$ with probability $\geq \frac{2}{3}$.

- We ignore the proof here.

- A component is small (resp., big): it contains $< \frac{8}{\epsilon d}$ vertices (resp., $\geq \frac{8}{\epsilon d}$ vertices).
- Some further notations:
  - t: the number of big components in G.
  - n': the number of vertices in big components in G.
  - m': the number of edges in big components in G.
  - Big: the set of vertices in big components in G.

Observation*

For any graph G, we have $\left|\frac{\hat{m}-\hat{n}}{\ell} - \frac{m'-n'}{n}\right| \leq \frac{\epsilon d}{16}$ with probability $\geq \frac{2}{3}$.

- We ignore the proof here.

Proof of the “correctness” of the tester

Next, we shall prove that cycle-free-tester achieves the following conditions:

- if G is cycle-free, then cycle-free-tester outputs ACCEPT with probability $> \frac{2}{3}$;
- if G is ϵ-far from being cycle-free, then cycle-free-tester outputs REJECT with probability $> \frac{2}{3}$.

Recall the property tester...

Algorithm: cycle-free-tester

Input: $G = (V, E)$ in an adjacency-list with bounded-degree d, $0 < \epsilon < 1$.

```
Uniformly and independently select ℓ = 2^13/ε^2 vertices from V;
for each selected vertex s do
    Perform a BFS starting from s until 8/(εd) vertices are reached or
    no more new vertices can be reached;
end for
if any of the above searches found a cycle then
    Output REJECT;
end if
Let n̂ denote the number of vertices in the sample that belong to
connected components of size ≥ 8/(εd);
Let m̂ denote half the sum of their degrees;
if (m̂ − n̂)/ℓ ≥ εd/16 then
    Output REJECT;
else
    Output ACCEPT;
end if
```

The case where G is cycle-free

- cycle-free-tester never outputs REJECT in Step 6.
- $m' - n' = -t \leq 0$.
- Recall that with probability $\geq 2/3$, the inequality $\left| \frac{\hat{m}-\hat{n}}{\ell} - \frac{m'-n'}{n} \right| \leq \frac{\epsilon d}{16}$ holds.
- The inequality $(\hat{m} - \hat{n})/\ell < \epsilon d/16$ holds with probability $\geq 2/3$, thus the algorithm accepts G in Step 13 with probability $\geq 2/3$.

The case when G is ϵ-far from being cycle-free

- For a graph G with t connected components, n vertices and m edges, we define $m - (n - t) \geq 0$ to be the number of superfluous edges in G.
- G is ϵ-far from being cycle-free $\Rightarrow$ the number of superfluous edges $\geq \frac{1}{2}\epsilon dn$.
- Let us consider two cases:

Case 1: There are $\geq \epsilon dn/4$ superfluous edges inside **small** components

- Consider a small component having s superfluous edges.
  - This component must contain $\geq \frac{2s}{d}$ vertices.
- The total number of vertices in small components that contain superfluous edges $\geq \epsilon n/2$.
- Hence, no cycle is detected in Step 2 with probability $< (1 - \epsilon/2)^{\ell} < 1/3$.

Case 2: There are $\geq \epsilon dn/4$ superfluous edges inside **big** components

- Recall that
  - t: the number of big components;
  - n': the number of vertices in big components;
  - m': the number of edges in big components.
- $m' - (n' - t) \geq \frac{\epsilon dn}{4}$
- Note that $t \leq \frac{n}{8/\epsilon d} = \frac{\epsilon dn}{8}$.
  - We have $\frac{m'-n'}{n} \geq \frac{\epsilon d}{8}$.
- Recall that with probability $\geq 2/3$, the inequality $\left|\frac{\hat{m}-\hat{n}}{\ell} - \frac{m'-n'}{n}\right| \leq \frac{\epsilon d}{16}$.
  - $\frac{\hat{m}-\hat{n}}{\ell} > \frac{\epsilon d}{16}$ with probability $\geq 2/3$.
  - Thus the algorithm returns REJECT in Step 11 with probability $\geq 2/3$.

Thank you,

and

Happy Teachers' Day!

Proof of Observation*

- For $i = 1, \ldots, \ell$, let χ_i be a 0–1 random variable that equals 1 iff the ith selected vertex (say, v_{k_i}) belongs to Big.
  - $\hat{n} = \sum_{i=1}^{\ell} \chi_i$.
  - $\mathbf{E}[\chi_i] = \sum_{v_{k_i} \in \text{Big}} 1 \cdot \mathbf{Pr}[v_{k_i} \text{ is selected}] = n' \cdot \frac{1}{n} = \frac{n'}{n}$.
  - $\therefore \mathbf{E}[\hat{n}] = \frac{n'\ell}{n}$.

Chernoff bounds

Some useful Chernoff bounds:

- Let $X_1, X_2, \ldots, X_n$ be a series of *mutually independent Bernoulli random variables* with $S = \sum_{i=1}^{n} X_i$ and $\mu = \mathbf{E}[S]$. Assume that, for all i, $\mathbf{Pr}[X_i = 1] = p$ for some $p > 0$, then

$$\begin{aligned} \mathbf{Pr}[S \geq (1+\delta)\mu] &\leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}, \text{ for any } \delta > 0; \\ \mathbf{Pr}[|S-\mu| \geq \delta\mu] &\leq 2e^{-\mu\delta^2/3} \text{ for } \delta \in (0,1). \end{aligned}$$

Proof of Observation* (contd.)

Applying the Chernoff bound below:

$$\begin{aligned}
\mathbf{Pr}\left[\left|\frac{\hat{n}}{\ell} - \frac{n'}{n}\right| \geq \frac{\epsilon}{32}\right] &\leq \mathbf{Pr}\left[\left|\hat{n} - \frac{n'}{n}\cdot\ell\right| \geq \frac{\epsilon}{32}\cdot\ell\right] \\
&\leq \mathbf{Pr}\left[\left|\hat{n} - \frac{n'}{n}\cdot\ell\right| \geq \left(\frac{\epsilon}{32}\cdot\frac{n}{n'}\right)\cdot\frac{n'}{n}\cdot\ell\right] \\
&\leq 2e^{-\frac{1}{3}\cdot\frac{n'}{n}\cdot\ell\cdot\left(\frac{\epsilon n}{32n'}\right)^2} \quad \text{(note that } \ell = 2^{13}/\epsilon^2\text{)} \\
&\leq 2\cdot e^{-8/3} \\
&< \frac{1}{6}.
\end{aligned}$$

- Thus with probability $\geq 5/6$, we have $\left|\frac{\hat{n}}{\ell} - \frac{n'}{n}\right| < \frac{\epsilon}{32}$.

Proof of Observation* (contd.)

- Similarly, for $i = 1, \ldots, \ell$ let ϕ_i be a random variable that equals the **degree** of the ith selected vertex **if it belongs to a big component**, and 0 otherwise.
- Then $\hat{m} = \frac{1}{2} \sum_{i=1}^{\ell} \phi_i$, and $\mathbf{E}[\hat{m}] = \frac{m'\ell}{n}$.
  - $\mathbf{E}[\phi_i] = \sum_{v_{k_i} \in \text{Big}} d(v_{k_i}) \cdot \mathbf{Pr}[v_{k_i} \text{ is selected}] = 2m' \cdot \frac{1}{n}$.
  - Note that $0 \le \phi_i \le d$ for each i.

A Hoeffding's bound

A Hoeffding's bound:

- Let $X_1, X_2, \ldots, X_n$ be a series of mutually independent bounded Bernoulli random variables (i.e., $a_i \leq X_i \leq b_i$, for some positive real a_i and b_i), then for $\alpha > 0$

$$\mathbf{Pr}[|S - \mu| \geq \alpha] \leq 2e^{-2\alpha^2/\sum_{i=1}^{n}(b_i - a_i)^2}.$$

Proof of Observation* (contd.)

Applying the previous Hoeffding's bound below:

$$\begin{aligned}
\mathbf{Pr}\left[\left|\frac{\hat{m}}{\ell} - \frac{m'}{n}\right| \geq \frac{\epsilon d}{32}\right] &= \mathbf{Pr}\left[\left|\hat{m} - \frac{m'\ell}{n}\right| \geq \frac{\epsilon d \ell}{32}\right] \\
&\leq 2e^{-\frac{2\cdot\left(\frac{\epsilon d \ell}{32}\right)^2}{\ell\cdot d^2}} \\
&= 2e^{-\frac{2\cdot\frac{\epsilon^2\cdot d^2\cdot\ell^2}{2^{10}}}{\ell\cdot d^2}} \\
&= 2e^{-\frac{\epsilon^2\cdot\frac{2^{13}}{\epsilon^2}}{2^9}} \\
&= 2e^{-16} \\
&< \frac{1}{6}.
\end{aligned}$$

Thus with probability $\geq 5/6$, we have $\left|\frac{\hat{m}}{\ell} - \frac{m'}{n}\right| < \frac{\epsilon d}{32}$.

Proof of Observation* (contd.)

Here we have:

- With probability $\geq 5/6$, we have $\left|\frac{\hat{n}}{\ell} - \frac{n'}{n}\right| < \frac{\epsilon}{32}$ (say (i)), and
- With probability $\geq 5/6$, we have $\left|\frac{\hat{m}}{\ell} - \frac{m'}{n}\right| < \frac{\epsilon d}{32}$ (say (ii)).

$\mathbf{Pr}[\text{(i) or (ii) is not satisfied}] < \frac{1}{3}$.

Thus with probability $\geq 2/3$, the inequality $\left|\frac{\hat{m}-\hat{n}}{\ell} - \frac{m'-n'}{n}\right| \leq \frac{\epsilon d}{16}$ holds.

Thank you,

and

Happy Teachers' Day!