Constant-time connectivity tests[1]

Philipp Klaus Krause

September 19, 2022

[1] This research was supported by the Deutsche Forschungsgemeinschaft (DFG), project "Graphstrukturtheorie im Übersetzerbau", 641649, KR 4970/1-1.

Property Testing

Property testing is about

- Fast (sublinear, often constant time) algorithms for approximate decision-making.
- Runtime depends on parameters, such as (average or maximum) degree of the input graph, maximum allowed error of the result, probability of the result being correct.
- Implemented constant time testers for connectivity, 2- and 3-edge connectivity, estimates of the nuber of connected components, distance to connectivity, 2-edge connectivity and eulerianity. All for sparse graphs. We also have a (not yet implemented) estimmate for the distance to 3-edge-connectivity.
- For 2-edge-connectivity and 3-edge-connectivity, our approaches have better runtime than previoulsy known approaches. This advantage carries over to tolerant testers based on them.

Models

- Bounded degree: $G = (V, E), \Delta \geq \Delta(G)$ is *ϵ-far* from having a property $\mathcal{P}$, if it cannot be transformed into a graph $G' \in \mathcal{P}, \Delta(G') \leq \Delta$ by at most $\epsilon\Delta|V|$ edge modifications. Complexities are given as functions of the maximum degree Δ.
- Sparse graph (aka. unbounded degree): $G = (V, E)$ is *ϵ-far* from having a property $\mathcal{P}$, if it cannot be transformed into a graph $G' \in \mathcal{P}$ by at most $\epsilon|E|$ edge modifications (i.e. edge insertions and edge deletions). Complexities are usually given as functions of the average degree d. When estimating distance to a property, often the distance is given in terms of edge modifications relative to the number of nodes (δ): $\delta|V| = \epsilon|E|$.

Connectivity Tests

Property	Model	Complexity
Connectivity (Goldreich 2002)	Δ	$O\left(\frac{\log^2\left(\frac{1}{\epsilon\Delta}\right)}{\epsilon}\right)$
Connectivity (Ron 2010)	d	$O\left(\frac{\log\left(\frac{1}{\epsilon d}\right)}{\epsilon^2 d}\right)$
Connectivity (Parnas 2002)	d	$O\left(\frac{\log\left(\frac{1}{\epsilon d}\right)}{\epsilon^2 d^2}\right)$
Connectivity	d	$O\left(-\log(1-p)\cdot\frac{\log\left(\frac{1}{\epsilon d}\right)}{\epsilon^2 d^2}\right)$
2-edge-connectivity (Goldreich 2002)	Δ	$O\left(\frac{\log^2\left(\frac{1}{\epsilon\Delta}\right)}{\epsilon}\right)$
2-edge-connectivity	d	$O\left(-\log(1-p)\cdot\frac{\log\left(\frac{1}{\epsilon d}\right)}{\epsilon^2 d^2}\right)$
3-edge-connectivity (Goldreich 2002)	Δ	$O\left(\frac{\log\left(\frac{1}{\epsilon\Delta}\right)}{\epsilon^2}\right)$
3-edge-connectivity	d	$O\left(-\log(1-p)\cdot\frac{\log\left(\frac{1}{\epsilon d}\right)}{\epsilon^3 d^3}\right)$
k-edge-connectivity (Goldreich 2002)	Δ	$O\left(\frac{k^3\log\left(\frac{1}{\epsilon\Delta}\right)}{\epsilon^{3-\frac{2}{k}}\Delta^{2-\frac{2}{k}}}\right)$
k-edge-connectivity (Parnas 2002)	d	$\tilde{O}\left(\frac{k^4}{\epsilon^4 d^4}\right)$
Eulerianity (Goldreich 2002)	Δ	$O\left(\frac{\log^2\left(\frac{1}{\epsilon\Delta}\right)}{\epsilon}\right)$
Eulerianity (Parnas 2002)	d	$O\left(\frac{\log\left(\frac{1}{\epsilon d}\right)}{\epsilon^2 d^2}\right)$
Eulerianity	d	$O\left(-\log(1-p)\cdot\frac{\log\left(\frac{1}{\epsilon d}\right)}{\epsilon^2 d^2}\right)$

Distance Estimates

Property	Model	Complexity
Connectivity (Chazelle 2005)	d	$O\left(\frac{d\log\left(\frac{d}{\delta}\right)}{\delta^2}\right)$
Connectivity (Marko 2005)	d	$O\left(\frac{1}{\epsilon^4 d^4}\right) = O\left(\frac{1}{\delta^4}\right)$
Connectivity (Berenbrink 2014)	d	$O\left(\frac{1}{\delta^4}\right)$, exp.: $O\left(\frac{\log\left(\frac{1}{\delta}\right)}{\delta^2}\right)$
Connectivity	d	$O\left(\frac{\log\left(\frac{1}{\delta}\right)}{(1-p)\delta^2}\right)$
2-edge-connectivity	d	$O\left(\frac{\log\left(\frac{1}{\delta}\right)}{(1-p)\delta^2}\right)$
3-edge-connectivity	d	$O\left(\frac{\log\left(\frac{1}{\delta}\right)}{(1-p)\delta^3}\right)$
k-edge-connectivity (Marko 2005)	d	$O\left(\frac{k^6\log\left(\frac{k}{\epsilon d}\right)}{\epsilon^6 d^6}\right) = O\left(\frac{k^6\log\left(\frac{k}{\delta}\right)}{\delta^6}\right)$
Eulerianity (Marko 2005)	d	$O\left(\frac{1}{\epsilon^4 d^4}\right) = O\left(\frac{1}{\delta^4}\right)$
Eulerianity	d	$O\left(\frac{\log\left(\frac{1}{\delta}\right)}{(1-p)\delta^2}\right)$

Distance to 2-edge-connectivity

Lemma (Eswaran and Tarjan 1976):
Let $G = (V, E)$ be a graph. Let c_1 be the number of connected components of G, that are 2-edge-connected. Let c_2 be the number of 2-edge-connected components of G, that are connected to the rest of G by a single bridge. If $|V| > 2$ and $c_1 + c_2 > 1$, the minimal number of edge modifications necessary to make G 2-edge-connected is $\lceil \frac{c_2}{2} \rceil + c_1$.

Distance to 2-edge-connectivity

```
int × int zshg_component_2(s, 𝔱)
{
  1 Do first depth−first search from s for up to 𝔱 + 1 nodes.
    Let n1 be the number of nodes found.

  2 Do second depth−first search from s for up to 𝔱 + 1 nodes,
    never traversing an edge of the search tree of 1 in the same direction.
    Let n2 be the number of nodes found.

  if (n1 == n2 == 𝔱 + 1) // Not in a 2−edge−connected component of size at most 𝔱
    return (0, 0);

  3 Do third depth−first search from s for up to 𝔱 nodes,
    only considering nodes found in 2,
    never traversing an edge of the search tree of 2 in the same direction.
    Let n3 be the number of nodes found.

  if (n2 == n3 == n1) // In 2−edge−connected component of size n3 that is 2−edge−connected
    return (n3, 0);
  else if (n2 == n3) // In 2−edge−conn. component of size n3 conn. to rest by single bridge
    return (0, n3);
  else
    return (0, 0);
}
```

Distance to 2-edge-connectivity

```
float zshg2_component(δ, p, n)
{
  r := 7 / (2(1−p)δ^2);
  a_1 := a_2 := 0;

  ℓ := r(3 log_e(2/δ) + 6);

  for(i := 0; i < r; i := i + 1)
  {
    s := rand_index(n);
    x := rand_range(2/δ);
    (b_1, b_2) := zshg_component_2(s, x, &ℓ);
    a_1 := a_1 + b_1;
    a_2 := a_2 + b_2;
  }

  return (a_1 n / r, a_2 n / r);
}
```

Distance to 2-edge-connectivity

- From the returned values of zshg2_component we can calculate the estimated distance.
- Correctness can be proven using basic stochastics.
- Number of queries made is bounded by ℓ.

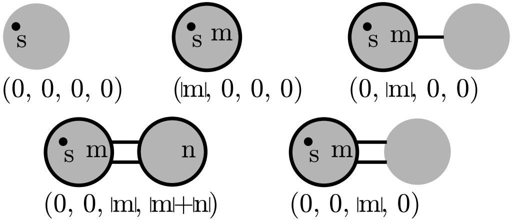
Distance to 3-edge-connectivity

Lemma (Naor et alii 1997 for connected graphs, generalized by Marko 2005):
Let $G = (V, E)$ be a graph and $k \in \mathbb{N}$. The distance of G to k-edge-connectivity is $\frac{1}{|E|} \left\lceil \frac{\Phi_k(V)}{2} \right\rceil$.

Distance to 3-edge-connectivity

Let C_0 be the number of of 3-class-leaves that are connected components, let C_1 be the number of of 3-class-leaves that are connected to the rest of the graph by a single bridge, let C_2 be the number of 3-class-leaves that are connected to the rest of the graph by a exactly two edges, let C_3 be the number of connected components that are 2-edge-connected and contain exactly 2 3-edge-leaves.

$$\Phi_3(V) = 3C_0 + 2C_1 + 1C_2 + 1C_3 - 3.$$



Remarks

- While the results on correctness also hold for multigraphs, the results on query complexity don't.
- All algorithms can be parallelized easily so are the implementations. The parallel versions offer an advantage when multiple pending queries can be answered more efficiently, e.g. in the case of large (too big to fit into RAM) graphs stored on an SSD (current SSDs typically achieve maximum throughput for random reads at about 16 simultaneous pending reads) or in the case of the queries being processed by a remote server on a network.
- Free source code at http://zshg.sourceforge.net/