Progress Log

Albert Kutsyy

19th April 2021

General

- My laptop is 3x faster than the server (tried running a graph) for single-thread performance. This is in line with CPU benchmarks.
- Largest graph I can run: EU-email (265214 nodes, 420045 edges), 30mins for my implementation of Brandes (and a whole lot more for JGraphT [at least 1 hour for 8%])

Brandes++

1. So first off the major issue of using an optimized c library for one thing and python for another
2. They also don't say how to get correctly formatted inputs which is annoying
3. The bit about removing clarifying sentences (which say that this isn't as magical as they say it is) in the second draft
4. arraygraph: `https://ocw.mit.edu/courses/electrical-engineering-and-computer-scienc 6-895-theory-of-parallel-systems-sma-5509-fall-2003/projects/kasheff. pdf`

Geisberger

location: `http://algo2.iti.kit.edu/schultes/hwy/betweenness.pdf` (geisberger) additional: `https://www.yumpu.com/en/document/view/16871703/better-approximation-of-betwee` thesis)

- The big study says they use geisberger. Their source code just calls networkKit ApproxBetweenness2 (their code was last updated June 21, 2017) (`https://github.com/ecrc/BeBeCA/blob/master/Source_Code/src/NetworKitApps.cpp`) This was then renamed to EstimateBetweenness (`https://github.com/networkit/networkit/blob/master/networkit/cpp/centrality/EstimateBetweenness.cpp`).

  This is documented at `https://networkit.github.io/dev-docs/python_api/centrality.html`, which says that it uses the algorithm described in Geisberger. HOWEVER, the source code actually implements the version of Brandes (2008) which is described by Geisberger.

- So it says "decrement" in section 3.3. It doesn't specify by what, and it isn't 1. I found the thesis (above) that Geisberger's paper is based on, which describes to decrement v by 1/sigma(v) (it also describes a different position to decrement than Geisberger does. It also actually describes the algorithm! However, it shouldn't be 1/sigma(v), it should be 1/sigma(w) where w is the thing popped off of the stack.

- I don't think a published implementation actually exists

- The sampling actually requires the following steps repeated numSamples times: sample parents, run canonical betweenness.

Brandes 2008

1. Can't actually find the proper one anywhere except `https://www.worldscientific.com/doi/abs/10.1142/S0218127407018403` where it's $30...

2. Pre-print here: `https://kops.uni-konstanz.de/bitstream/handle/123456789/5772/estimations.pdf?sequence=1&isAllowed=y`

3. Good description in Geisberger though

Bader

1. Good for getting top 1