KARGER-KLEIN-TARJAN EXPECTED LINEAR TIME MST ALGORITHM

Robert Emerson

April 29, 2013

Randomized Algorithm

- Runs in O(n+m) with high probability
- Dependent on F-heavy edge lemma and cycle and cycle properties
- Combination of Boruvka’s Algorithm, random sampling and MST verification algorithms

Overview of KKT

- Run two Boruvka operations on the input graph
- Sample the remaining edges and discard each edge with probability $p=0.5$
  - Sampling lemma tells the expected number of F-light edges in resultant graph is n/p
- Recursively call the algorithm on resultant graph
- Use result of recursive call to identify and remove F-heavy edges according to the verification algorithm
- Recursively call algorithm again and combine result with edges contracted in the first Boruvka steps

MST Verification

- Reduces to the *tree-path-maxima* problem on branching trees to identify F-heavy edges
- Original algorithm used a path compression algorithm discovered by Dixon, Rauch, Tarjan(1992)
- V. King (1993,1997) proposed simpler algorithms using Boruvka trees and non-trivial edge encoding
- T. Hagerup (2009) uses ideas from King, combined with a new set operation, to simplify the algorithm

Results

- Code is still buggy, segfaults randomly at times.
- Graph size of 10,000,000 nodes
  - Initialization takes ~25 seconds
- KKT Algorithm: DNF
  - At this graph size runs out of memory in ~25 seconds
- Kruskal's Algorithm: ~35 seconds
  - Implementation from Boost C++ libraries
- At smaller graph sizes, Kruskal's Algorithm finishes before KKT

Links

- Code:
  - http://www.github.com/Robert-Emerson/6161_final
  - http://www.boost.org/doc/libs/1_53_0/libs/graph/doc/
- Papers:
  - KKT: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.3957
  - King: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.6311
  - Hagerup: http://link.springer.com/chapter/10.1007/978-3-642-11409-0_16