Minimum Degree Heuristic

Eli Fox-Epstein

June 30, 2016

This abstract corresponds to the submission for the 2016 PACE Challenge. We are entering Track A and provides a serial, heuristic implementation. Source code is available at: `https://github.com/elitheeli/2016-pace-challenge`.

This entry is provided as a baseline, implementing simple strategy: the Minimum Degree Heuristic [1]. In the time given, this implementation runs the heuristic repeatedly, maintaining the best decomposition.

Implementation Details. This implementation is written in C++11. Tree decompositions are represented by two vectors: one is a vector of bags (each of which is a vector of vertex IDs) and one stores parent-child relationships.

The graph is represented using an adjacency matrix (compactly represented as a `std::vector<bool>`), as well as a adjacency list for each vertex (using a `std::vector`) to support efficient neighborhood iteration and a vector tracking vertex degrees. At any given time, the adjacency lists contain a *superset* of the neighbors of a vertex: on iteration through a neighborhood, each potential neighbor is verified with the matrix and discarded if it is not a true neighbor.

The algorithm maintains a priority queue on the vertices, keyed by degree plus a small amount of random noise. The noise ensures a random choice among the vertices with minimum degree. Three vectors are maintained: one to store the order in which vertices are selected, one mapping a vertex to whether or not it has been selected, and one tracking the degree of a vertex when enqueued. Initially, all vertices are enqueued. When processing a vertex v, if v has already been seen, it is simply discarded. If its degree is less than its current degree, it is re-enqueued with the correct degree. Finally, if its degree is correct, it is added to the ordering, marked as visited, and its neighbors are connected into a clique and then disconnected from v. Some neighbors may be re-enqueued if their degrees decreased by losing their edge to v.

References

[1] H. M. Markowitz. The elimination form of the inverse and its application to linear programming. *Management Science*, 3(3):255–269, 1957.