Supplementary: Faster Multi-Object Segmentation using Parallel Quadratic Pseudo-Boolean Optimization

Niels Jeppesen, Patrick M. Jensen, Anders N. Christensen, Anders B. Dahl, and Vedrana A. Dahl
Department of Applied Mathematics and Computer Science
Technical University of Denmark, Kgs. Lyngby, Denmark
{niejep,patmjen,anym,abda,vand}@dtu.dk

1. Proof of equivalent labeling

Theorem: Assume we have two (not necessarily equal) minimum cuts, C_1 and C_2, of the same extended graph. Assume that for each solution we run the algorithm described in [1, 3] to extract minimum cuts C'_1 and C'_2 that label the maximum number of nodes. Then, C'_1 and C'_2 will label the same nodes.

Proof: We use the notation from Kolmogorov and Rother [[3], Choosing a minimum cut] and prove by contradiction. Assume there is a node u that is labeled by C'_1 but not by C'_2. The algorithm of [1, 3] will label all nodes p, where p and the corresponding flipped node $\overline{p}$ are not in the same strongly connected component of the residual graph.

Therefore, since u was not labeled by C'_2, it must be in the same strongly connected component as the corresponding flipped node, $\overline{u}$. By the Ford-Fulkerson theorem (as stated in [3]), this means there is no minimum cut which separates u and $\overline{u}$.

However, since u was labeled by the minimum cut C'_1, there is at least one minimum cut which separates u and $\overline{u}$. Thus, we have a contradiction and C'_2 must label all nodes that C'_1 does. Interchanging C'_1 and C'_2 and repeating the proof shows that C'_1 and C'_2 must label the same nodes. □

2. Performance of parallel EIBFS

The results presented in Table 3 in our paper show that our *parallel* EIBFS performs worse than the other tested algorithms, including serial EIBFS. The discussion of the performance of EIBFS would shift the focus of our work, so we omitted it from our paper. Still, we would like to share our findings on the matter, as they may be of interest.

While testing the algorithms, we investigated the cause of the poor performance of parallel EIBFS by profiling the code. We determined that the reason for the poor performance is that EIBFS must maintain a number of invariants in its internal data structures. When edges are modified in a solved graph, these invariants must be restored before solving can (re)start. Our tests showed that this incurred significant overhead which made the parallel version slower than the serial. Thus, EIBFS does not seem like an optimal choice for either QPBO or bottom-up merging, which both rely on modifying edges after solving. This also aligns with the results in [[2], Table 2] showing that EIBFS is not as dominant in the dynamic setting.

References

[1] Bengt Aspvall, Michael F Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979. 1

[2] Andrew V Goldberg, Sagi Hed, Haim Kaplan, Pushmeet Kohli, Robert E Tarjan, and Renato F Werneck. Faster and More Dynamic Maximum Flow by Incremental Breadth-First Search. In *Proceedings of the European Symposium on Algorithms (ESA)*, pages 619–630, 2015. 1

[3] Vladimir Kolmogorov and Carsten Rother. Minimizing non-submodular functions with graph cuts-a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(7):1274–1279, 2007. 1