MaxFlow Revisited: An Empirical Comparison of Maxflow Algorithms for Dense Vision Problems

Tanmay Verma
tanmay08054@iiitd.ac.in
IIIT-Delhi
Delhi, India

Dhruv Batra
dbatra@ttic.edu
TTI-Chicago
Chicago, USA

Motivation. Over the past two decades, algorithms for finding the maximum amount of flow possible in a network (or max-flow) have become the workhorses of modern computer vision and machine learning – from optimal (or provably-approximate) inference in sophisticated discrete models [2, 12] to enabling real-time image processing [16]. Perhaps the most prominent role of max-flow is due to the work of Hammer [10] and Kolmogorov and Zabih [12], who showed that a fairly large class of energy functions – sum of submodular functions on pairs of boolean variables – can be efficiently and *optimally* minimized via a reduction to max-flow. Max-flow also plays a crucial role in approximate minimization of energy functions with multi-label variables, triplet or higher order terms, global terms, and terms encoding label costs.

Given the wide applicability, it is important to ask *which* max-flow algorithm should be used. There are numerous algorithms for max-flow with different asymptotic complexities and practical run-time behaviour. Broadly speaking, there are three main families of max-flow algorithms:

1. Augmenting-Path (AP) variants: algorithms [1, 5, 6, 7, 9] that maintain a valid flow during the algorithm, *i.e.* always satisfying the capacity and flow-conservation constraints.
2. Push-Relabel (PRL) variants: algorithms [4, 8] that maintain a *preflow*, *i.e.* satisfy the capacity constraints but may violate the conservation constraints to have flow excess at nodes (but never a deficit).
3. Pseudoflow (HPF) variants: algorithms [3, 11] that maintain a *pseudoflow*, *i.e.* satisfy the capacity constrains but may violate the conservation constrains to allow flow excess and deficit at nodes.[1]

Boykov and Kolmogorov [1] compared AP and PRL algorithms on a number of computer vision problems, and found that their own algorithm (BK) was the fastest algorithm in practice, even though they could only prove a very loose asymptotic complexity bound of $O(n^2mC)$, where n is the number of nodes, m is the number of edges and C is the max-flow value.

Goal. The central thesis of this work is that since this comparison a decade ago, the models used in computer vision and the *kinds* of inference problems we solve have changed significantly. Specifically, while [1] only considered 4-connected grid MRFs, the models today involve high-order terms, long-range connections, hierarchical MRFs and even global terms. The effect of all these modifications is to make the underlying max-flow graph significantly denser, thus causing the complexity of the algorithm of [1] to become a concern. It is time to revisit this comparison.

Contribution. The goal of this paper is to compare the runtimes of different max-flow algorithms, to investigate if the conclusions of Boykov and Kolmogorov [1] are still valid for current-day *dense* problems, and find out which algorithm is most suited for modern vision problems. One key contribution of our study is that it includes recently proposed algorithms – Pseudoflow [3, 11] and Incremental Breadth-First-Search (IBFS) [9] – which were not developed at the time [1] was written, and thus were absent from their comparison.

Problems. We tested a number of max-flow algorithms on the following:

1. **Synthetic Instances.** We created synthetic max-flow graphs with a basic grid structure and randomly added long-range edges depending on a density parameter.
2. **ALE Graphs.** We used the max-flow graphs created during alpha-expansion by the Automatic Labeling Environment (ALE) of Ladicky *et al*. [13] on PASCAL VOC 2010 segmentation images.
3. **Deconvolution.** We used the QPBO max-flow graph on the binary image deconvolution CRF instance from Rother *et al*. [15]. This is an extremely dense graph and the problem is not submodular.
4. **Super Resolution.** We used the QPBO max-flow graph on the super-resolution CRF instances of [15].
5. **Texture Restoration.** We used the QPBO max-flow graph on the Brodatz texture D103 model from [15].
6. **DTF Graphs.** Decision Tree Field (DTF) [14] is a recently introduced model that combines random forests and conditional random fields. We used the 100 instances provided by Nowozin *et al*. [14] and saved the QPBO-graphs to file.
7. **3D Segmentation.** Finally, we also evaluated all algorithms on the standard benchmark for such studies, the binary 3D (medical) segmentation instances from the University of Western Ontario `http://vision.csd.uwo.ca/maxflow-data/`.

We note that all of the previous studies were restricted to 3D segmentation, and problems 2-6 have never been used to evaluate max-flow algorithm, yet they are in some sense more representative of modern problems.

Findings: Our paper has the following findings:

1. **Choice of Algorithm *Matters*.** In all applications, the fastest algorithms is *orders of magnitude* faster than the slowest algorithm.
2. **BK Scales Poorly with Density.** Our results show that the motivating hypothesis of this study is correct. In a number of cases (Synthetic, Deconv, 3Dseg) BK starts out fairly competitive at low densities but very quickly becomes the slowest algorithm.
3. **New Kids in Town: IBFS and HPF.** In a number of applications we considered, both IBFS and HPF significantly outperform BK, IBFS more consistently so than HPF.
4. **Clever Data-structures Matter.** We found the data-structures used by BK to be particularly efficient. In a number of applications (see *e.g.* SuperRes, Texture-Restoration), BK maxflow time is longer than IBFS but the maxflow+initialization time is shorter.

We hope that the results of our study guide practitioners in picking the correct implementation for their problems.

[1] Interestingly, the key difference between Push-Relabel and Pseudoflow algorithms is not the concept of pseudoflow rather the admissibility of certain push schemes.

[1] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.

[2] Yuri Boykov, Olga Veksler, and Ramin Zabih. Efficient approximate energy minimization via graph cuts. *PAMI*, 20(12):1222–1239, 2001.

[3] Bala G. Chandran and Dorit S. Hochbaum. A computational study of the pseudoflow and push-relabel algorithms for the maximum flow problem. *Oper. Res.*, 57:358–376, March 2009. ISSN 0030-364X. doi: 10.1287/opre.1080.0572.

[4] Boris V. Cherkassky and Andrew V. Goldberg. On implementing the push-relabel method for the maximum flow problem. *Algorithmica*, 19(4):390–410, 1997.

[5] E. A. Dinic. Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation. *Soviet Math Doklady*, 11:1277–1280, 1970.

[6] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19:248–264, April 1972. ISSN 0004-5411.

[7] L. R. Ford and D. R. Fulkerson. Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[8] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35:921–940, October 1988. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/48014.61051. URL `http://doi.acm.org/10.1145/48014.61051`.

[9] Andrew V. Goldberg, Sagi Hed, Haim Kaplan, Robert E. Tarjan, and Renato F. Werneck. Maximum flows by incremental breadth-first search. In *Proceedings of the 19th European conference on Algorithms*, ESA'11, pages 457–468, 2011. ISBN 978-3-642-23718-8.

[10] P.L. Hammer. Some network flow problems solved with pseudo-boolean programming. *Operations Research*, 13:388–399, 1965.

[11] Dorit S. Hochbaum. The pseudoflow algorithm: A new algorithm for the Maximum-Flow problem. *Operations Research*, 56(4):992–1009, July 2008.

[12] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[13] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical CRFs for object class image segmentation. *ICCV*, 2009.

[14] Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, and Pushmeet Kholi. Decision tree fields. In *ICCV*, 2011.

[15] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *CVPR*, June 2007. doi: 10.1109/CVPR.2007.383203.

[16] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "Grabcut": interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 2004.