

Solving Minimum Cost Lifted Multicut Problems by Node Agglomeration

Amirhossein Kardoost and Margret Keuper

Data and Web Science Group
University of Mannheim, Germany
kardoost@informatik.uni-mannheim.de
keuper@uni-mannheim.de

Abstract. Despite its complexity, the minimum cost lifted multicut problem has found a wide range of applications in recent years, such as image and mesh decomposition or multiple object tracking. Its solutions are decompositions of a graph into an optimal number of segments which are optimized w.r.t. a cost function defined on a superset of the edge set. While the currently available solvers for this problem provide high quality solutions in terms of the task to be solved, they can have long computation times for more difficult problem instances. Here, we propose two variants of a heuristic solver (primal feasible heuristic), which greedily generate solutions within a bounded amount of time. Evaluations on image and mesh segmentation benchmarks show the high quality of these solutions.

1 Introduction

This paper tackles the fast, heuristic optimization of a graph decomposition problem, the minimum cost *lifted* multicut problem. The minimum cost lifted multicut problem has first been proposed in [1], where its promise for the decomposition of pixel grid graphs as well as 3D shape meshes has been shown. It has been successfully applied in fields like multiple object tracking [2] and motion segmentation [3] and became known to produce state-of-the-art results on the segmentation of electron microscopy stacks of neuronal structures [4, 5]. It is a generalization of the better-known minimum cost multicut problem [6, 7] (also referred to as correlation clustering problem). For example, formulations of the image segmentation problem [8] as minimum cost multicut problem have been considered in [9–23]. This formulation is attractive because the feasible solutions of the multicut problem relate one-to-one to the decompositions of a graph, and the multicut problem does not favor balanced solutions a priori [24]. Furthermore, multicut algorithms are easy to use: They take as input a graph and, for every edge, a real-valued cost of the incident vertices being in distinct components, and output a 01-labeling of the edges which induce a decomposition of the graph. Assigned real-valued cost of the incident vertices ideally reflects the logit of the probability of the edge being cut, i.e. $\log \frac{1-p_e}{p_e} + \log \frac{1-p^*}{p^*}$, for

a cut probability p_e at edge e , and a prior probability $p^* \in (0, 1)$ of cuts. Cut probability p_e is generated based on the estimated boundary [8] at edge e .

Generalizations of the minimum cost multicut problem in terms of higher-order edges have been considered in [3, 17, 20]. In contrast to explicit higher-order terms, the generalization to minimum cost *lifted* multicuts allows introduction of additive edge costs, which alter the cost of a specific solution without modifying the set of feasible solutions. In practice, this property turned out to be useful for example in image segmentation, where it is easy to decide for a pair of distant points whether they are in distinct components, yet potentially hard for a pair of neighboring points. The minimum cost lifted multicut problem enables to encode this long range information directly in edge costs.

While this formulation is extremely useful, it turned out to be truly harder than the (NP-hard) minimum cost multicut problem [25]. In practice, it can be solved using heuristic solvers such as KLj [1] or Fusion Moves [4]. For large problem instances such as the pixel-wise image segmentation instances from [1], such solvers can take several minutes to converge or might require greedy pre-clustering.

In this paper, we propose a modification of a very simple primal feasible heuristic, the greedy additive edge contraction (GAEC) algorithm from [1]. The proposed algorithm produces results that are close to the ones generated by KLj [1] with a computation time similar to the one from GAEC [1] and a guaranteed worst case complexity. We evaluate the proposed heuristic on instances of (1) image segmentation problems on the BSDS-500 dataset [8], (2) mesh segmentation problems on the Princeton Mesh Segmentation benchmark [26], and (3), the challenging 3D segmentation instance from the ISBI 2012 challenge on segmentation of neuronal structures from electron microscopy images used in [4, 5]. In all scenarios, the proposed algorithm can generate high quality results at a significantly reduced computation time.

2 Related Work

A canonical approach to solving instances of the minimum cost multicut problem is to solve the linear programming relaxation of the multicut polytope [6] as for example proposed in [27]. This procedure, with a subsequent thresholding to produce feasible solutions, has found wide application, e.g. in [16, 19, 20]. Yet, despite its NP-hardness, the minimum cost multicut problem has been solved to optimality for some practically relevant instances for example in [11, 28]. Yet, for larger problem instances, heuristic solvers such as Cut, Glue & Cut (CGC) [15], KL [29, 30] and its variant KLj [1], or the Fusion Moves approach [14] need to be employed. They have shown to generate good solutions in practice although no theoretical guarantee can be provided.

The latter two solvers, KLj [1] and Fusion Moves [4], can also be applied to the minimum cost *lifted* multicut problem which is of interest in this paper. Both have proven to provide good solutions in practice, generating results on par with the state-of-the-art in image segmentation in terms of the BSDS-500

benchmark [8], the mesh decomposition problem posed by the Princeton benchmark [26] and the ISBI2012 challenge on segmentation of neuronal structures in image stacks [31, 32]. For problem instances for example on the image segmentation task, these heuristics still need several minutes to converge, while Fusion Moves [4] has an edge on KLj in terms of computation time, at the price of a greedy data pre-clustering.

The proposed approach is built on the fully greedy approach GAEC which has been proposed as an initialization procedure for KLj in [1]. As GAEC, KLj and Fusion Moves, it can not provide any bounds for the proposed solution. However, in contrast to KLj and Fusion Moves, it can provide a guaranteed worst case complexity. In practice, it generates results with a computation time slightly higher than GAEC while the resulting objective values can be close to the ones from KLj.

3 Optimization Problem

In this paper, we propose a heuristic solver for the minimum cost *lifted* multicut problem [1]. This problem is of interest, because its feasible solutions relate one-to-one to decompositions of a graph and because it provides a principled way to define a more general cost function decompositions than the minimum cost multicut problem. A decomposition of the graph $G = (V, E)$ is any partition Π of the vertices V such that every component $V' \in \Pi$ induces a connected subgraph of G . In the following, we define the minimum cost multicut problem [6, 7] and the minimum cost *lifted* multicut problem [1, 33].

3.1 Minimum Cost Multicut Problem

Given any graph $G = (V, E)$, a cost function $c : E \rightarrow \mathbb{R}$ and *edge* labels $y : E \rightarrow \{0, 1\}$, the optimization problem stated in (1) is an instance of the minimum cost multicut problem with respect to the graph G and costs c .

$$\begin{aligned} \min_{y \in \{0,1\}^E} \sum_{e \in E} c_e y_e \quad (1) \\ \text{s.t. } \forall C \in \text{cycles}(G) \quad \forall e \in C : y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'}. \end{aligned}$$

The inequality constraints stated over all cycles of G ensure that the edge labeling y induces a decomposition of G . It has been shown to be sufficient to ensure these constraints on the subset of all chordless cycles in [6]. The minimum cost multicut problem allows to assign a cost or reward (c_e) for every edge $e \in E$ to be cut.

3.2 Minimum Cost Lifted Multicut Problem

The minimum cost *lifted* multicut problem is defined w.r.t. a graph $G = (V, E)$ and a graph $G' = (V, E')$ with $E \subseteq E'$ and a cost function $c' : E' \rightarrow \mathbb{R}$ which allows to assign, for every edge in E' a cost or reward for being cut. Its feasible solutions relate one-to-one to decompositions of the graph G . Rigorously, for any undirected graph $G = (V, E)$, any $F = \binom{V}{2} \setminus E$ and any $c' : E \cup F \rightarrow \mathbb{R}$, the linear program written in Eq. (2) - (5) is an instance of the minimum cost lifted multicut problem w.r.t. G , F and c' .

$$\min_{y \in \{0,1\}^{E'}} \sum_{e \in E'} c'_e y_e \quad (2)$$

$$s.t. \quad \forall C \in \text{cycles}(G) \quad \forall e \in C : y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'} \quad (3)$$

$$\forall vw \in F \quad \forall P \in vw\text{-paths}(G) : y_{vw} \leq \sum_{e \in P} y_e \quad (4)$$

$$\forall vw \in F \quad \forall C \in vw\text{-cuts}(G) : 1 - y_{vw} \leq \sum_{e \in C} (1 - y_e) \quad (5)$$

The linear inequalities in Eq. (3) - (5) constrain y such that $\{e \in E | y_e = 1\}$ is a multicut of G . They ensure further that, for any edge $uv \in F$, $y_{uv} = 0$ iff there exists a path (uv -path) in the original graph G along which all edges are connected, i.e. labeled 0.

The set F of lifted edges is introduced to facilitate the modification of the cost function of the multicut problem without modifying the set of feasible solutions [1]. This is most intuitive in image segmentation. There, the connectivity should be defined on the image grid: two pixels in the same cluster should be connected in the image domain. However, for pairs of directly neighboring pixels, it is hard to decide whether they belong to the same segment or to different segments (due to blurred edges/textures). For pixels at a certain distance, it is easier to recognize boundaries separating one from the other. The lifted edge set F allows to introduce such information between distant pixels without modifying the connectivity [1].

4 Objectives

To generate fast solutions to minimum cost lifted multicut problems, we investigate the GAEC solver proposed in [1]. While other heuristic solvers such as [4] generate better solutions in terms of the resulting energy, GAEC is one of the fastest solvers currently available, although it operates directly on the nodes of

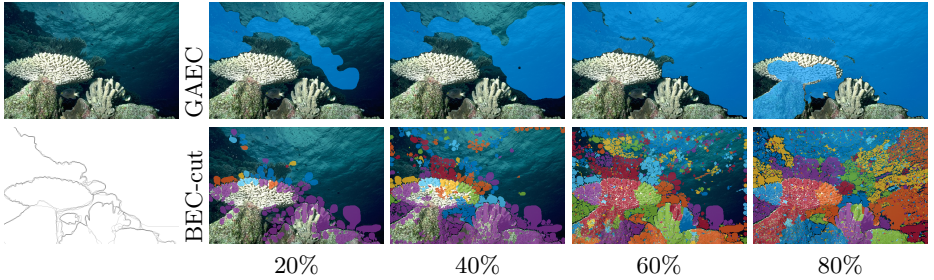


Fig. 1: For the BSDS-500 [8] image on the left (average ground-truth annotations depicted below), we show intermediate states during the execution of the GAEC solver [1] (top) and the proposed BEC-cut solver (bottom) after 20 %, 40%, 60% and 80% of the total merges have been executed. GAEC tends to generate large segments that merge points across object boundaries. These merges can not be “repaired”. In contract, BEC-cut generates and grows many segments simultaneously. It starts generating these segments in vicinity to the object boundaries.

the original graph without any local pre-clustering. GAEC provides deterministic solutions created in a greedy procedure of edge contractions and thus provides low computation times even for large and difficult problem instances [1]. While GAEC provides relatively low energy solutions on minimum cost multicut problems, it has major difficulties finding acceptable solutions to minimum cost *lifted* multicut problems. Where does this discrepancy come from? To analyze this effect, we visualize intermediate GAEC solutions in Fig. 1. We observe that large clusters are built pretty quickly. They tend to grow by adding small clusters or isolated nodes. In this scenario, long range repulsive edges are not in the scope of the greedy optimization until almost all nodes have been merged. Thus, the “difficult parts” of the problem, i.e. those parts possibly causing conflicts w.r.t. the objective, are only considered when most of the nodes have already been irreversibly merged.

We conclude that the GAEC algorithm would benefit from two aspects: (1) avoid creating very unbalanced clusters early on, such that repulsive edges are in the scope of intermediate regions early on, and (2) use information about repulsive edges directly as a criterion for merges.

5 Proposed Approach

We want to base our heuristic on the fast GAEC solver from [1] and follow a similar, greedy merging scheme such that running times remain affordable for large and difficult problem instances. In order to tackle aspects (1), avoid creating very unbalanced clusters early on, and (2), use information about repulsive edges in the merge criterion, we define an improved ranking scheme for the next best edge to be contracted. In the basic version of our proposed heuristic, we only

Algorithm 1: GAEC [1]

```

1  $\mathcal{E} := E, \mathcal{E}' := E'$ 
2  $\mathcal{V} := V$ 
3 foreach  $ab \in \mathcal{E}'$  do
4    $\chi_{ab} := c_{ab}$ 
5 while  $\mathcal{E} \neq \emptyset$  do
6    $ab := \operatorname{argmax}_{a'b' \in \mathcal{E}} \chi_{a'b'}$ 
7   if  $\chi_{ab} < 0$  then
8     break
9   contract  $ab$  in  $\mathcal{G}$  and  $\mathcal{G}'$ 
10  foreach  $ab \neq ab' \in \mathcal{E}'$  do
11     $\chi_{ab'} := \chi_{ab'} + \chi_{bb'}$ 

```

Algorithm 2: BEC

```

1  $\mathcal{E} := E, \mathcal{E}' := E'$ 
2  $\mathcal{V} := V$ 
3 foreach  $ab \in \mathcal{E}'$  do
4    $\chi_{ab} := c_{ab}$ 
5 while  $\mathcal{E} \neq \emptyset$  do
6    $ab := \operatorname{argmax}_{a'b' \in \mathcal{E}} \frac{\chi_{a'b'}}{|a'| + |b'|}$ 
7   if  $\chi_{ab} < 0$  then
8     break
9   contract  $ab$  in  $\mathcal{G}$  and  $\mathcal{G}'$ 
10  foreach  $ab \neq ab' \in \mathcal{E}'$  do
11     $\chi_{ab'} := \chi_{ab'} + \chi_{bb'}$ 

```

tackle aspect (1) by proposing a greedy “Balanced Edge Contraction” scheme. Instead of using the plain edge potentials as a merge criterion as in GAEC (see Alg. 1, line 6), we propose to normalize these potentials by the size of the components to be merged (Alg. 2, line 6). Note that this normalization only affects the order in which attractive edges are merged and not the actual edge potentials nor the cut objective.

In order to tackle aspect (2), we formulate a secondary merge criterion. Ultimately, we want to find a cut which minimizes the total energy of the cut (not maximize the energy of the joined components). We are motivated to encourage the solver to pursue this objective during the greedy optimization: We want to make join decisions that minimize the outgoing potentials of resulting components, i.e. minimize the intermediate cut.

To formalize this, let's look at the graph $G_d = (V, E_d)$ defined on G' (section 3.2), and assign to every vertex v in V a unary potential $d_v = \sum_{\{b|e_{vb} \in E'\}} c_{vb}$, i.e. the degree of v . The outgoing potentials of a component in G resulting from joining two vertices u and v can be computed as $d_u + d_v - 2c_{u,v}$. Then, we define for edges $uv \in E_d$, edge weights $c_{u,v}^d = d_u + d_v - 2c_{u,v}$. Minimizing these “dual” edge weights is used as a secondary join criterion in Alg. 3.

This modification of the order in which segments are agglomerated affects the intermediate stages of the solution during the optimization. Fig. 1 (bottom) shows such intermediate optimization stages after 20%, 40%, 60% and 80% of the total merges have been computed by the proposed algorithm. Unlike GAEC, it generates many small node agglomerations along both sides of the boundaries. Thus, repulsive terms on the boundary are in scope early on.

5.1 Algorithms

Alg. 2 and Alg. 3 are adaptations of greedy agglomeration, more specifically, greedy additive edge contraction (Alg. 1). Both take as input an instance of the

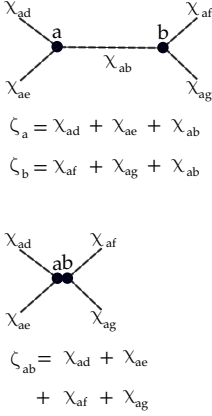


Fig. 2: Exemplary computation of ζ from χ before (top) and after (bottom) contraction of components a and b . χ encodes the sum of outgoing costs of a component.

Algorithm 3: BEC-cut

```

1   $\mathcal{E} := E, \mathcal{E}' := E'$ 
2   $\mathcal{V} := V$ 
3  foreach  $ab \in \mathcal{E}$  do
4     $\chi_{ab} := c_{ab}$ 
5  foreach  $a \in \mathcal{V}$  do
6     $\zeta_a := \sum_{\{b | e_{ab} \in E'\}} c_{ab}$ 
7  while  $\mathcal{E} \neq \emptyset$  do
8     $\mathcal{S} := \operatorname{argmax}_{a'b' \in \mathcal{E}} \frac{\chi_{a'b'}}{|a'| + |b'|}$ 
9     $ab := \operatorname{argmin}_{a'b' \in \mathcal{S}} \frac{\zeta_{a'} + \zeta_{b'} - 2\chi_{a'b'}}{|a'| + |b'|}$ 
10   if  $\chi_{ab} < 0$  then
11     break
12   contract  $ab =: \bar{a}$  in  $\mathcal{G}$  and  $\mathcal{G}'$ 
13    $\zeta_{\bar{a}} = \zeta_a + \zeta_b - 2\chi_{ab}$ 
14   foreach  $ab \neq ab' \in \mathcal{E}'$  do
15      $\chi_{ab'} := \chi_{ab'} + \chi_{bb'}$ 

```

lifted multicut problem (LMP) defined by $G = (V, E)$, F and $c : E \cup F \rightarrow \mathbb{R}$ (Eq. (2) - (5)) and construct as output a decomposition of the graph G . Both algorithms maintain a decomposition of G , represented by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose nodes $a \in \mathcal{V}$ are components of G and whose edges $ab \in \mathcal{E}$ connect any components a and b of G which are neighbors in G and $\chi : \mathcal{E} \rightarrow \mathbb{R}$ is an ordered sequence of costs for feasible joins. Objective values are computed w.r.t. the larger graph $G' = (V, E \cup F)$ and $c : E \cup F \rightarrow \mathbb{R}$.

Balanced Edge Contraction (BEC) Starting from the decomposition into single nodes, in every iteration, a pair of neighboring components is joined for which the join decreases the objective value. While for the basic GAEC algorithm, an edge is picked such that the objective value decreases maximally, we propose to weight the prospective gain by the size of the components. This weighting encourages components of similar size to merge earlier than components of different size if both merges are advantageous. Intuitively, this should lead to several balanced clusters (as opposed to one large cluster and many single nodes) at intermediate optimization states. If no join strictly decreases the objective value, the algorithm terminates.

BEC-cut Starting from the decomposition into single nodes, at every iteration, a pair of neighboring components is joined for which the join decreases the objective value. If a unique pair of neighboring components exists whose

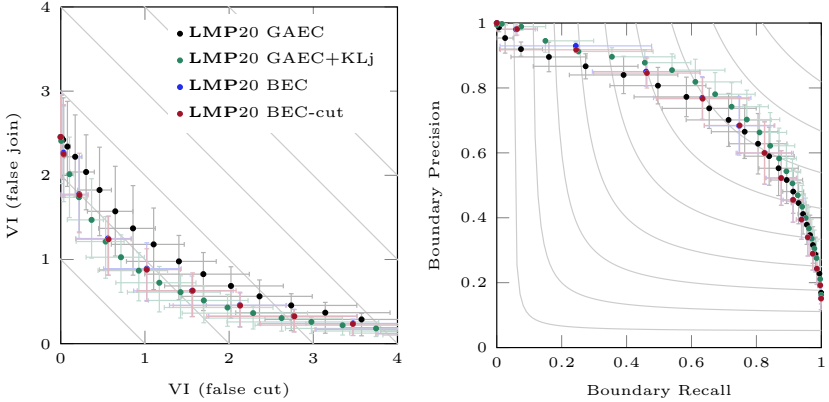


Fig. 3: Depicted above is an evaluation of the heuristics GAEC, (Alg. 1), KLj [1] and the proposed BEC (Alg. 2) and BEC-cut (Alg. 3) on the large and difficult lifted multicut problem instances from [1] with lifting radius 20 (LMP20). These instances address the image decomposition problem posed by the BSDS-500 benchmark [8]. On the **left**, the variation of information (VI), split additively into a distance due to false cuts and a distance due to false joins is depicted (lower is better), on the **right**, the accuracy of boundary detection, split into recall and precision is shown (higher is better). Error bars depict the 0.25 and 0.75-quantile. Result of the proposed heuristics figure in between those of the solver GAEC and KLj in both metrics. In the region metric VI, results of the proposed solvers BEC and BEC-cut are very close to results of the KLj.

join strictly decreases the *size-weighted* objective value maximally, this join is executed. From the set of all possible joins that decrease the weighted objective value maximally, the join is executed which minimizes the sum of outgoing costs, i.e. the prospective value ζ of the resulting component. Compare Fig.2 for an illustration of the computation of ζ . If no join strictly decreases the objective value, the algorithm terminates.

It is important to notice that the balancing criteria in our heuristic methods never affects the original objective function, which is defined in section 3.2.

Implementation. Our implementation is built upon GAEC [1]. As GAEC, it uses ordered adjacency lists for the graph \mathcal{G} and for a graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ whose edges $ab \in \mathcal{E}'$ connect any components a and b of \mathcal{G} for which there is an edge $vw \in E \cup F$ with $v \in a$ and $w \in b$. It uses a *disjoint set data structure* for the partition of V and a *priority queue* for an ordered sequence of costs $\chi : \mathcal{E} \rightarrow \mathbb{R}$ of feasible joins, with a secondary sorting criterion $\bar{\chi} : \mathcal{E} \rightarrow \mathbb{R}$, $\bar{\chi}_{ab} = \frac{\zeta_a + \zeta_b - 2\chi_{ab}}{|a| + |b|}$ defined on ζ for Alg. 3 (comp. line 9). As GAEC, its worst-case time complexity $O(|V|^2 \log |V|)$ is due to a sequence of at most $|V|$ contractions, in each of which at most $\deg \mathcal{G}' \leq |V|$ edges are removed, each in time $O(\log \deg \mathcal{G}') \in O(\log |V|)$.

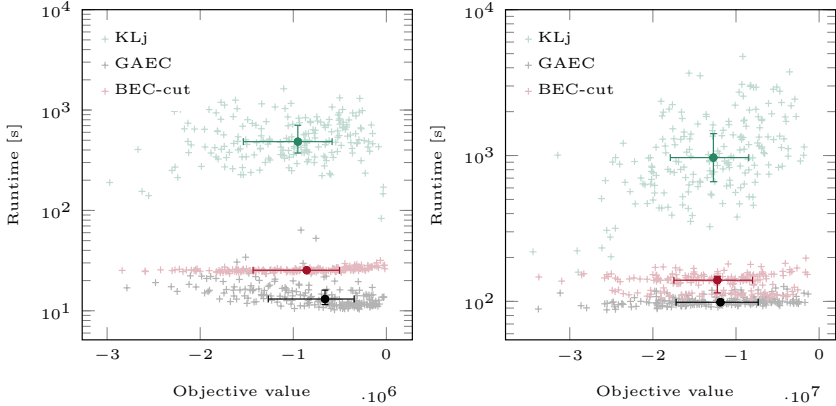


Fig. 4: Depicted above is a comparison of Alg. 3 (BEC-cut) with GAEC and KLj [1]. Every point corresponds to one instance of the lifted multicut problem [1] defined w.r.t. one test image in the BSDS-500 benchmark [8] with lifting radius 10 (**left**) and 20 (**right**). The computation times of the proposed algorithm BEC-cut are close to the ones from GAEC while the resulting energy is improved.

6 Experiments

We evaluate the proposed heuristics on types of problem instances and w.r.t. three different tasks: lifted pixel-grid graphs [1] defined on the image segmentation problems of the BSDS-500 [8] benchmark, lifted graphs on 3D shape meshes [1] defined on the Princeton Mesh Segmentation benchmark [26] and the lifted superpixel adjacency graph [5] defined on the ISBI 2012 challenge data, a volumetric electron microscopy recording of neuronal structures [31, 32].

6.1 Image Decomposition

On the image segmentation task posed by the BSDS-500 benchmark [8], we apply the proposed solvers BEC 2 and BEC-cut 3 to the minimum cost lifted multicut instances proposed in [1]. These problem instances represent pixel grid graphs. Lifted “long range” edges are inserted to connect all pixels within a pre-defined pixel radius to provide more “global” information to the optimization problem. Lifting radii of 20 pixels showed best performance in [1].

Fig. 3 shows the results of the proposed solvers in comparison to GAEC and KLj initialized with GAEC in terms of the variation of information (VI) [34] (Fig. 3 left, lower is better) and the boundary precision and recall (Fig. 3 right, higher is better). For the evaluation on these metrics results at different levels of granularity need to be generated. This can be done by modifying the prior probability of a cut in the lifted multicut problem instances. Every point in the plots of Fig. 3 shows, for one cut prior and algorithm, the average over all test images in the BSDS-500 benchmark.

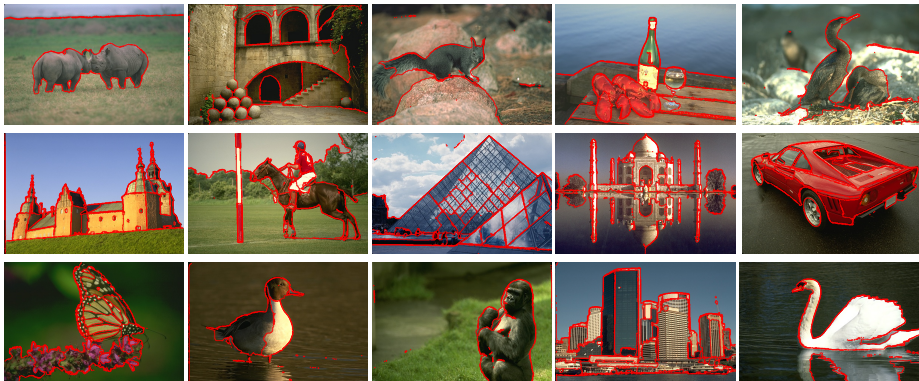


Fig. 5: Qualitative segmentation results of the proposed BEC-cut solver on instances from the BSDS-500 benchmark. The lifted multicut problems are computed with lifted edges between all points at distance less than 10.

While BEC and BEC-cut show almost equivalent performance, the improvement over GAEC [1] especially on the region metric VI is evident.

Fig.4 shows scatter plots of the computation times and resulting objective values for the algorithms GAEC and KLj [1] and the proposed BEC-cut. With respect to the objective value, BEC-cut yields a clear improvement over GAEC while it is significantly faster than KLj.

Some of the qualitative results are shown in Fig. 5 for the proposed BEC-cut solver and lifting radius 10. Although its greedy merging procedure, it can produce nicely closed contours on these problem instances.

6.2 Mesh Segmentation

In this section, we evaluate the proposed methods on the Princeton Mesh Segmentation benchmark [26] which provides 380 meshes consisting 19 object categories with human-generated ground-truths [26].

On this dataset, the instance of the minimum cost lifted multicut problem for each mesh is generated based on Keuper et al. [1]. More specifically, the local edge costs for $e \in E$ are computed using curvatures and dihedral angle features which are computed based on the mesh data. The lifted edges are then inserted up to a fixed node distance of 70 (defined on the graph) and for cut prior 0.55 as proposed in [1] using their code¹. Consequently, the created problem instances are solved by the proposed solvers. Some of the qualitative results for success and failure cases for the mesh segmentations are depicted in Fig. 8 and 7, respectively.

Tab.1 shows a comparison of the resulting objective functions and the consumed optimization time of the different approaches. On this dataset, the optimization based on KLj when it is initialized with either GAEC or BEC converges

¹ <https://www.mpi-inf.mpg.de/fileadmin/inf/d2/levinkov/iccv-2015/code.tar.gz>

Table 1: Average computation time and objective value of the different solvers on the Princeton Mesh Segmentation Benchmark.

	GAEC [1]	KLj-GAEC [1]	BEC	BEC-cut	KLj-BEC-cut	KLj-BEC
Avg. Comp. time [s]	576	755	589	574	797	755
Avg. Objective Value (lower is better)	-17840450	-18988930	-18484140	-18057480	-18991920	-18989332

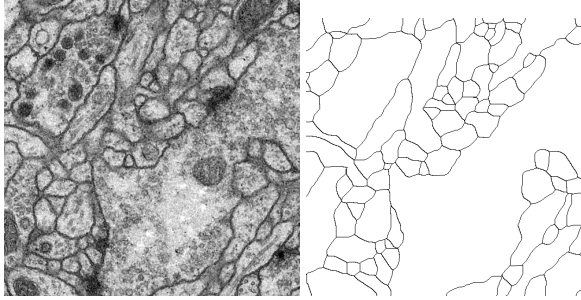


Fig. 6: First frame of the stack in the test data of ISBI 2012 with the corresponding segmentation boundaries are shown. The results for this dataset are acquired with BEC-cut solver.

surprisingly fast. This might be an indication for a meaningful initial segmentation in these cases. Among the greedy heuristics BEC yields the lowest objective values. Initializing KLj with the proposed BEC yields a better objective value than KLj initialized with GAEC [1].

Tab. 2 provides a comparison between different solvers in terms of Rand’s Index (RI) (higher is better) and Variation of Information (VI) (lower is better). In addition to report the plain results of our proposed solvers, we evaluate the quality of solutions from KLj initialized with BEC, i.e. KLj-BEC. The iterative, local move making heuristic KLj [1] is known to benefit from good initializations.

The evaluation shows that both proposed heuristics outperform GAEC. Specifically, BEC yields an improvement of 0.05 on the RI and a decrease of 0.08 in the VI. On this dataset, BEC performs better than BEC-cut. In fact, it yields a RI value close to the one from KLj-GAEC with the same lifting parameters. KLj, initialized with BEC yields slightly better results than KLj initialized with GAEC in terms of the VI. The column “KLj-GAEC_{opt}” in Tab. 2 corresponds to the results from Keuper et al. [1], which are obtained by KLj-GAEC when the lifting radius and cut prior were optimized per instance. KLj, initialized by BEC almost meets these results.

Table 2: Resulting RI and VI score for the 3D mesh segmentation instances of the Princeton Mesh Segmentation Benchmark [26]. We evaluate the proposed solvers BEC and BEC-cut and compare to GAEC [1] and KLj [1] initialized by GAEC. We also evaluate KLj initialized with our results from BEC. “KLj-GAEC_{opt}” represents the results from Keuper et al. [1] when the lifting radius and cut prior were optimized per instance.

	GAEC [1]		BEC		BEC-cut		KLj-GAEC [1]		KLj-BEC		KLj-GAEC _{opt} [1]	
	RI	VI	RI	VI	RI	VI	RI	VI	RI	VI	RI	VI
Human	0.77	1.90	0.84	1.77	0.85	1.74	0.86	1.63	0.86	1.60	0.87	1.79
Cup	0.83	0.53	0.88	0.44	0.86	0.45	0.88	0.40	0.89	0.37	0.90	0.39
Glasses	0.61	1.38	0.82	0.85	0.78	1.04	0.84	0.78	0.84	0.77	0.90	0.68
Airplane	0.89	0.93	0.91	0.92	0.91	0.86	0.91	0.84	0.91	0.85	0.92	0.83
Ant	0.93	0.70	0.97	0.48	0.96	0.52	0.97	0.43	0.97	0.44	0.98	0.42
Chair	0.85	0.85	0.92	0.67	0.89	0.84	0.92	0.58	0.92	0.58	0.93	0.55
Octopus	0.96	0.41	0.97	0.41	0.97	0.44	0.97	0.37	0.97	0.37	0.98	0.33
Table	0.87	0.48	0.91	0.40	0.83	0.63	0.93	0.29	0.93	0.29	0.94	0.29
Teddy	0.92	0.76	0.93	0.76	0.92	0.78	0.95	0.55	0.95	0.55	0.96	0.50
Hand	0.67	1.75	0.80	1.47	0.76	1.72	0.84	1.23	0.84	1.23	0.85	1.32
Plier	0.72	1.38	0.84	1.12	0.81	1.19	0.88	0.93	0.91	0.85	0.93	0.84
Fish	0.73	1.20	0.75	1.26	0.76	1.30	0.80	1.12	0.80	1.11	0.80	1.09
Bird	0.90	1.01	0.92	0.98	0.91	1.01	0.92	0.91	0.92	0.91	0.93	0.99
Armadillo	0.88	1.88	0.91	1.75	0.90	1.87	0.91	1.63	0.91	1.62	0.92	1.48
Bust	0.57	2.12	0.66	2.32	0.63	2.46	0.68	2.29	0.68	2.29	0.69	2.25
Mech	0.80	0.66	0.81	0.72	0.80	0.66	0.84	0.60	0.84	0.59	0.84	0.59
Bearing	0.79	0.76	0.82	0.75	0.83	0.70	0.84	0.69	0.84	0.69	0.84	0.69
Vase	0.77	1.02	0.80	1.06	0.77	1.13	0.84	0.87	0.84	0.87	0.84	0.87
FourLeg	0.81	1.94	0.82	1.98	0.81	2.04	0.83	1.87	0.83	1.89	0.84	1.72
Average	0.81	1.14	0.86	1.06	0.84	1.13	0.88	0.95	0.88	0.94	0.89	0.93

6.3 ISBI 2012 Challenge

In the ISBI 2012 challenge, a stack of electron microscopy images of neuronal structures [31, 32] is provided for segmentation. The images are recorded by a serial section transmission electron microscope from the Drosophila larva ventral nerve cord.

We applied the proposed BEC and BEC-cut solvers to the instance of the minimum cost lifted multicut problem defined in [5]. The instance is based on a superpixel adjacency graph, i.e. every node corresponds to a superpixel and is connected to its direct neighbors. An additional *lifted* edge set encodes long range information. All edge weights are learned from training data using two random forest classifiers, one of them is trained to predict whether two adjacent superpixels should be assigned to the same cluster, the other one learns this prediction for non-adjacent superpixels. The trained classifiers are used to assign weights to the local and lifted edges, respectively. Tab. 3 shows results we achieve on this data with our heuristic solvers BEC and BEC-cut. At a significantly reduced computation time, we can get results close to the state-of-the-art in terms of objective value as well as segmentation quality. Fig. 6 represents the first frame in the stack of the test data from ISBI 2012 with the corresponding segmentation boundaries which are resulted from the proposed BEC-cut solver.

Table 3: Comparison of the objective value and run time of our proposed solvers, BEC and BEC-cut with Fusion Move algorithm with the Randomized proposal generator (FM-R) [4] are provided. The results are generated based on the defined minimum cost lifted multicut problem for ISBI 2012 Challenge [31, 32]. Leader board: http://brainiac2.mit.edu/isbi_challenge/leaders-board-new

Algorithm	Comp. time [s]	Objective Value (lower is better)	V^{Rand}	V^{Info}
FM-R [4]	10.43	-1.948524	0.9822	0.9884
BEC	2.53	-1.944767	-	-
BEC-cut	2.59	-1.944964	0.9811	0.9883



Fig. 7: Some of the failure cases for mesh decomposition are shown, where the instance of the lifted multicut problem is solved with BEC-cut. For success cases refer to the Fig. 8.

7 Conclusion

In this paper, we propose two heuristic solvers BEC and BEC-cut. Both are modifications of the greedy agglomerative heuristic GAEC and provide a bounded worst case complexity. We evaluate our solvers on three application scenarios: image decomposition, 3D mesh segmentation and segmentation of neuronal structures from volumetric electron microscopic data. In all scenarios, the proposed solvers yield results close to the ones from more elaborate solvers KLj [1] or Fusion Moves [4], while the computation time is significantly lower.

Acknowledgements We acknowledge funding by the DFG project KE 2264/1-1. We also acknowledge the NVIDIA Corporation for the donation of a Titan Xp GPU.

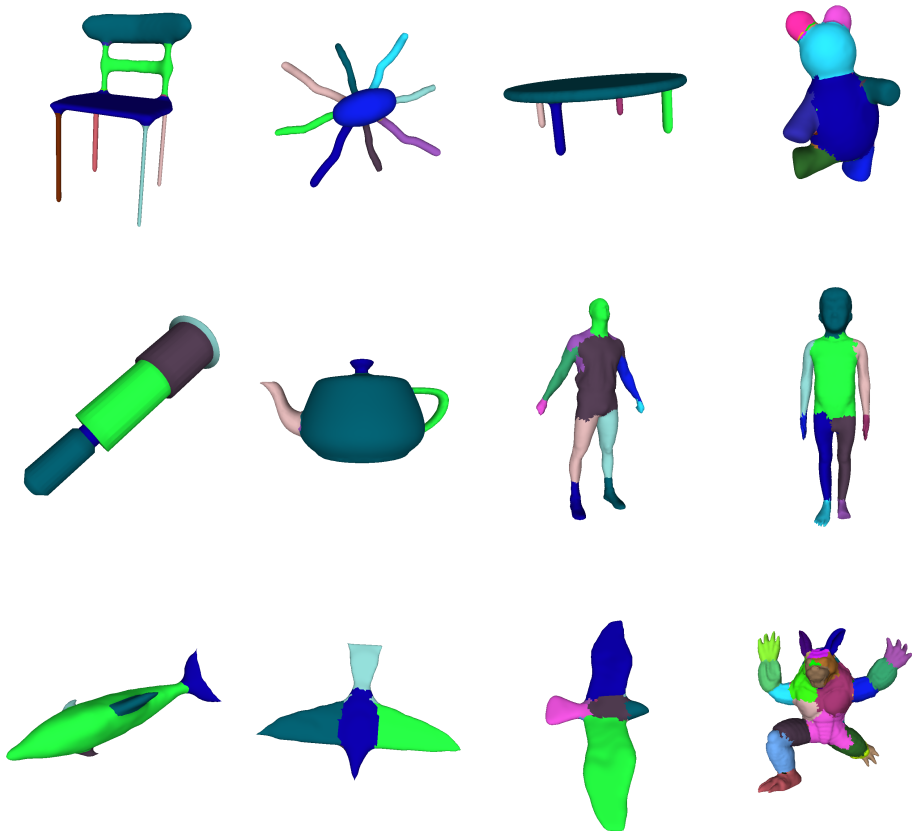


Fig. 8: Depicted above is a sample of success cases for mesh decompositions generated by solving an instance of the lifted multicut problem with BEC-cut.

References

1. Keuper, M., Levinkov, E., Bonneel, N., Lavoué, G., Brox, T., Andres, B.: Efficient decomposition of image and mesh graphs by lifted multicuts. 2015 IEEE International Conference on Computer Vision (ICCV) (2015) 1751–1759 [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#)
2. Tang, S., Andriluka, M., Andres, B., Schiele, B.: Multi people tracking with lifted multicut and person re-identification. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2017) [1](#)
3. Keuper, M.: Higher-order minimum cost lifted multicuts for motion segmentation. In: The IEEE International Conference on Computer Vision (ICCV). (2017) [1](#), [2](#)
4. Beier, T., Andres, B., Köthe, U., Hamprecht, F.A.: An efficient fusion move algorithm for the minimum cost lifted multicut problem. In: ECCV. (2016) [1](#), [2](#), [3](#), [4](#), [13](#)

5. Beier, T., Pape, C., Rahaman, N., Prange, T., Berg, S., Bock, D.D., Cardona, A., Knott, G.W., Plaza, S.M., Scheffer, L.K., Koethe, U., Kreshuk, A., Hamprecht, F.A.: Multicut brings automated neurite segmentation closer to human performance. *Nature Methods* **14** (2017) [1](#), [2](#), [9](#), [12](#)
6. Chopra, S., Rao, M.: The partition problem. *Mathematical Programming* **59** (1993) 87–115 [1](#), [2](#), [3](#)
7. Deza, M.M., Laurent, M.: *Geometry of Cuts and Metrics*. Springer (1997) [1](#), [3](#)
8. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *IEEE TPAMI* **33**(5) (2011) [1](#), [2](#), [3](#), [5](#), [8](#), [9](#)
9. Alush, A., Goldberger, J.: Ensemble segmentation using efficient integer linear programming. *TPAMI* **34** (2012) 1966–1977 [1](#)
10. Andres, B., Kappes, J.H., Beier, T., Köthe, U., Hamprecht, F.A.: Probabilistic image segmentation with closedness constraints. In: *ICCV*. (2011) [1](#)
11. Andres, B., Kröger, T., Briggman, K.L., Denk, W., Korogod, N., Knott, G., Köthe, U., Hamprecht, F.A.: Globally optimal closed-surface segmentation for connectomics. In: *ECCV*. (2012) [1](#), [2](#)
12. Andres, B., Yarkony, J., Manjunath, B.S., Kirchhoff, S., Turetken, E., Fowlkes, C., Pfister, H.: Segmenting planar superpixel adjacency graphs w.r.t. non-planar superpixel affinity graphs. In: *EMMCVPR*. (2013) [1](#)
13. Bagon, S., Galun, M.: Large scale correlation clustering optimization. *CoRR abs/1112.2903* (2011) [1](#)
14. Beier, T., Hamprecht, F.A., Kappes, J.H.: Fusion moves for correlation clustering. In: *CVPR*. (2015) [1](#), [2](#)
15. Beier, T., Kroeger, T., Kappes, J., Köthe, U., Hamprecht, F.: Cut, glue, & cut: A fast, approximate solver for multicut partitioning. In: *CVPR*. (2014) [1](#), [2](#)
16. Kappes, J.H., Speth, M., Andres, B., Reinelt, G., Schnörr, C.: Globally optimal image partitioning by multicuts. In: *EMMCVPR*. (2011) [1](#), [2](#)
17. Kappes, J.H., Speth, M., Reinelt, G., Schnörr, C.: Higher-order segmentation via multicuts. *CoRR abs/1305.6387* (2013) [1](#), [2](#)
18. Kappes, J.H., Swoboda, P., Savchynskyy, B., Hazan, T., Schnörr, C.: Probabilistic correlation clustering and image partitioning using perturbed multicuts. In: *SSVM*. (2015) [1](#)
19. Kim, S., Nowozin, S., Kohli, P., Y., C.D.: Higher-order correlation clustering for image segmentation. In: *NIPS*. (2011) [1](#), [2](#)
20. Kim, S., Yoo, C.D., Nowozin, S.: Image segmentation using higher-order correlation clustering. *IEEE TPAMI* **36** (2014) 1761–1774 [1](#), [2](#)
21. Nowozin, S., Jegelka, S.: Solution stability in linear programming relaxations: Graph partitioning and unsupervised learning. In: *ICML*. (2009) [1](#)
22. Yarkony, J., Ihler, A., Fowlkes, C.: Fast planar correlation clustering for image segmentation. In: *ECCV*. (2012) [1](#)
23. Yarkony, J., Zhang, C., Fowlkes, C.: Hierarchical planar correlation clustering for cell segmentation. In: *EMMCVPR*. (2015) [1](#)
24. Shi, J., Malik, J.: Normalized cuts and image segmentation. *TPAMI* **22** (2000) 888–905 [1](#)
25. Hornáková, A., Lange, J.H., Andres, B.: Analysis and optimization of graph decompositions by lifted multicuts. In: *ICML*. (2017) [2](#)
26. Chen, X., Golovinskiy, A., Funkhouser, T.: A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)* **28** (2009) [2](#), [3](#), [9](#), [10](#), [12](#)
27. Demaine, E.D., Emanuel, D., Fiat, A., Immorlica, N.: Correlation clustering in general weighted graphs. *Theoretical Computer Science* **361** (2006) 172–187 [2](#)

28. Kappes, J.H., Andres, B., Hamprecht, F.A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B.X., Kröger, T., Lellmann, J., Komodakis, N., Savchynskyy, B., Rother, C.: A comparative study of modern inference techniques for structured discrete energy minimization problems. *IJCV* (2015) [2](#)
29. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal* **49** (1970) 291–307 [2](#)
30. Kappes, J., Andres, B., Hamprecht, F., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B., Kröger, T., Lellmann, J., Komodakis, N., Savchynskyy, B., Rother, C.: A comparative study of modern inference techniques for structured discrete energy minimization problems. *International Journal of Computer Vision* **115** (2015) 155–184 [2](#)
31. Cardona, A., Saalfeld, S., Preibisch, S., Schmid, B., Cheng, A., Pulkas, J., Tomanek, P., Hartenstein, V.: An integrated micro- and macroarchitectural analysis of the drosophila brain by computer-assisted serial section electron microscopy. *PLOS Biology* **8** (2010) 1–17 [3](#), [9](#), [12](#), [13](#)
32. Carreras, I., Turaga, S., Berger, D., San, D., Giusti, A., Gambardella, L., Schmidhuber, J., Laptev, D., Dwivedi, S., Buhmann, J., Liu, T., Seyedhosseini, M., Tasdizen, T., Kamensky, L., Burget, R., Uher, V., Tan, X., Sun, C., Pham, T., Bas, E., Uzunbas, M., Cardona, A., Schindelin, J., Seung, H.: Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy* **9** (2015) 1–13 [3](#), [9](#), [12](#), [13](#)
33. Andres, B.: Lifting of multicuts. *CoRR* **abs/1503.03791** (2015) [3](#)
34. Meilă, M.: Comparing clusterings—an information based distance. *J. Multivar. Anal.* **98** (2007) 873–895 [9](#)