GASP, a generalized framework for agglomerative clustering of signed graphs and its application to Instance Segmentation

Alberto Bailoni [*] [†] Constantin Pape [‡] [†] [*] Nathan Hütsch [*] Steffen Wolf [§]
Thorsten Beier [¶] Anna Kreshuk [†] Fred A. Hamprecht [*] [‖]

Abstract

We propose a theoretical framework that generalizes simple and fast algorithms for hierarchical agglomerative clustering to weighted graphs with both attractive and repulsive interactions between the nodes. This framework defines GASP, a Generalized Algorithm for Signed graph Partitioning[1]*, and allows us to explore many combinations of different linkage criteria and cannot-link constraints. We prove the equivalence of existing clustering methods to some of those combinations and introduce new algorithms for combinations that have not been studied before. We study both theoretical and empirical properties of these combinations and prove that some of these define an ultrametric on the graph. We conduct a systematic comparison of various instantiations of GASP on a large variety of both synthetic and existing signed clustering problems, in terms of accuracy but also efficiency and robustness to noise. Lastly, we show that some of the algorithms included in our framework, when combined with the predictions from a CNN model, result in a simple bottom-up instance segmentation pipeline. Going all the way from pixels to final segments with a simple procedure, we achieve state-of-the-art accuracy on the CREMI 2016 EM segmentation benchmark without requiring domain-specific superpixels.*

1. Introduction

In computer vision, the partitioning of weighted graphs has been successfully applied to tasks as diverse as image segmentation, object tracking and pose estimation. Most graph clustering methods work with positive edge weights only, which can be interpreted as similarities or distances between the nodes. These methods require users to specify the desired numbers of clusters (as in spectral clustering) or a termination criterion (e.g. in iterated normalized cuts) or even to add a seed for each object (e.g. seeded watershed or random walker).

Other graph clustering methods work with so-called *signed graphs*, which feature both positive and negative edge weights corresponding to attraction and repulsion between nodes. The advantage of signed graphs over unsigned graphs is that balancing attraction and repulsion allows us to obtain a clustering without defining additional parameters. A canonical formulation of the signed graph partitioning problem is the *multicut* or *correlation clustering* problem [15, 35]. This problem is NP-hard, though many approximate solvers have been proposed [8,47,62,77] together with greedy agglomerative clustering algorithms [36, 39, 51, 75]. Agglomerative clustering algorithms for signed graphs have clear advantages: they are parameter-free and efficient. Despite the fact that a variety of these algorithms exist, no overarching study has so far been conducted to compare their robustness and efficiency or to provide guidelines for matching an algorithm to the partitioning problem at hand.

Our first contribution is a simple theoretical framework that generalizes over agglomerative algorithms for signed graphs by linking them to hierarchical clustering (HC) on unsigned graphs (Section 3.2). This framework defines an underlying basic algorithm and allows us to explore its combinations with different linkage criteria and *cannot-link constraints* (see Fig. 1a, 1b, and Table 1). As second contribution, in Section 3.3, we formally prove that some of the combinations correspond to existing clustering algorithms, and introduce new algorithms for combinations which have not been explored before. By analyzing their theoretical properties, we also show that some of them define an ultrametric on the graph (see Table 1).

Third, we evaluate the algorithms on a large variety of both existing and synthetically generated signed graph clustering problems (Section 4). Fourth and finally, we also test

[*]Heidelberg Collaboratory for Image Processing (HCI/IWR), Heidelberg, Germany. {name.surname}@iwr.uni-heidelberg.de
[†]EMBL, Heidelberg, Germany. {name.surname}@embl.de
[‡]Institute for Computer Science, University Göttingen, Germany. constantin.pape@informatik.uni-goettingen.de
[§]MRC Laboratory of Molecular Biology, Cambridge, UK. swolf@mrc-lmb.cam.ac.uk
[¶]derthorstenbeier@gmail.com
[‖]Corresponding author
[1]Code available at: https://github.com/abailoni/GASP

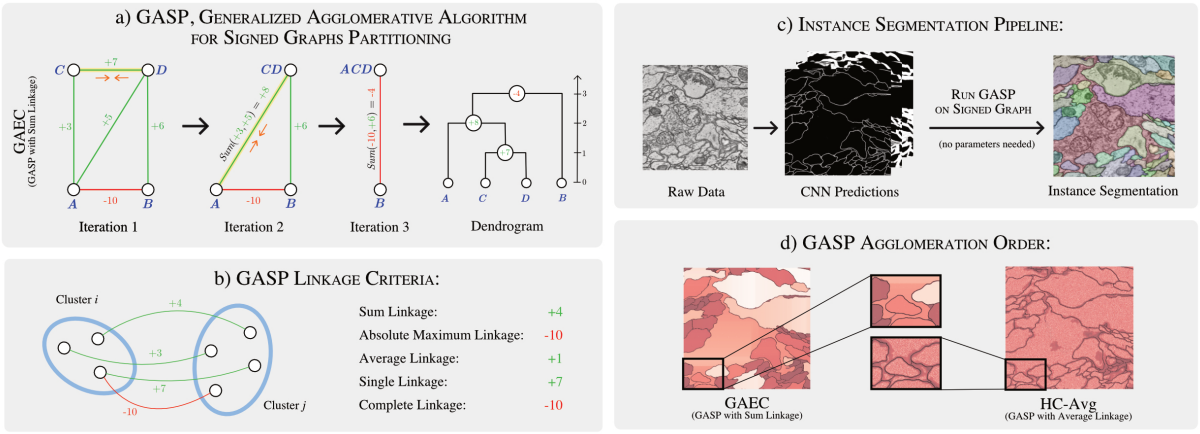


Figure 1. **(a)** Some iterations of GASP on a graph with attractive (green) and repulsive (red) interactions. At each iteration, the yellow edge with highest weight is contracted (example with sum linkage criterion is shown). **(b)** Linkage criteria used in this paper demonstrated on two small clusters (see definitions in Table 1 below). **(c)** Application of GASP to instance segmentation: we show raw data from the CREMI neuron-segmentation challenge and some predictions of our CNN model, where white pixels represent boundary evidence. **(d)** Seemingly similar linkage criteria can result in very different clustering dynamics, as shown in this example: color coded sequence of merges from early (white) via late (brown) to never (black).

GASP	Sum Linkage $\sum_{e\in E_{ij}} w_e$	Absolute Maximum Linkage w_e with $e = \arg\max_{t\in E_{ij}} \lvert w_t\rvert$	Average Linkage $\sum_{e\in E_{ij}} w_e \Big/ \lvert E_{ij}\rvert$	Single Linkage $\max_{e\in E_{ij}} w_e$	Complete Linkage $\min_{e\in E_{ij}} w_e$
Unsigned graphs	-	**HC-Single**	**HC-Avg**	**HC-Single**	**HC-Complete**
Signed graphs	GAEC [39]	**Mutex Watershed** [75]	**HC-Avg**	**HC-Single**	**HC-Complete**
Signed graphs + cannot-link constraints	HCC-Sum	**Mutex Watershed** [75]	HCC-Avg	HCC-Single	**HC-Complete**

Table 1. Conceptual contribution: Properties of clustering algorithms included in the proposed GASP framework, given a linkage criterion, a type of graph (signed or unsigned) and the optional use of cannot-link constraints. New constrained hierarchical clustering algorithms (HCC) proposed in this paper are highlighted in yellow. For algorithms typeset in bold font we prove that they define an ultrametric on the graph (Eq. 3). For algorithms in the green box we show that they are weight-shift invariant (Prop. 3.2). Notation: E_{ij} denotes the set of edges connecting two clusters $S_i, S_j \subseteq V$.

the algorithms on *instance segmentation* – a computer vision task consisting of assigning each pixel of an image to an object instance – by partitioning graphs whose edge weights are estimated by a CNN (see Fig. 1c and Section 4.2). Our experiments show that the choice of linkage criterion markedly influences how clusters are grown by the agglomerative algorithms (Fig. 1d), making some linkage methods more suited for certain types of clustering problems. We benchmark the clustering algorithms by focusing on their efficiency, robustness and tendency to over- or under-cluster. On instance segmentation, we show that the agglomerative algorithms outperform recently proposed spectral clustering methods, and that average-linkage based agglomerative algorithms achieve state of the art results on the CREMI 2016 challenge for neuron segmentation of 3D electron microscopy image volumes of brain tissue.

2. Related work

Proposal-free instance segmentation methods adopt a bottom-up approach by directly grouping pixels into instances. In the last years, there has been a growing interest in such methods that do not involve object detection because, in certain types of data, object instances cannot be approximated by bounding boxes [5, 41]. Some use metric learning to predict high-dimensional associative pixel embeddings that map pixels of the same instance close to each other [21, 23, 49, 60] and then retrieve final instances by applying a clustering algorithm [44]. Other recent methods let the model predict the relative coordinates of the instance center [13, 59] or, given a pixel (x, y), they train a model to generate the mask of the instance located at (x, y) [70].

Edge detection also experienced recent progress thanks

to deep learning, both on natural images [29, 43, 55, 76] and biological data [18, 50, 57, 69]. In neuron segmentation for connectomics, a field of neuroscience we also address in our experiments, boundaries are converted to final instances with subsequent postprocessing and superpixel-merging: some use a combinatorial framework [10], others use loopy graphs [37, 45] or trees [26, 52, 54, 57, 72] to represent the region merging hierarchy. Flood-filling networks [32] and MaskExtend [57] used a CNN to iteratively grow one region/neuron at the time. A structured learning approach was also proposed in [28, 71].

Agglomerative graph clustering has often been applied to instance segmentation [3, 53, 66, 68] because of its efficiency as compared to other divisive approaches like graph cuts. Novel termination criteria and merging strategies have often been proposed: the agglomeration in [56] deploys fixed sets of merge constraints; the popular graph-based method [24] stops the agglomeration when the merge costs exceed a measure of quality for the current clusters. The optimization approach in [40] performs greedy merge decisions that minimize a certain energy, while other pipelines use classical linkage criteria, e.g. average linkage [50, 55], median [28] or a linkage learned by a random forest classifier [42, 61].

Clustering of signed graphs has the goal of partitioning a graph with both attractive and repulsive cues. Finding an optimally balanced partitioning has a long history in combinatorial optimization [16, 30, 31]. NP-hardness of the *correlation clustering* problem was shown in [7], while the connection with graph multicuts was made by [22]. Modern integer linear programming solvers can tackle problems of considerable size [2], but accurate approximations [8, 62, 77], greedy agglomerative algorithms [36, 39, 51, 74] and persistence criteria [47, 48] have been proposed for even larger graphs. Another line of research is given by spectral clustering methods that, on the other hand, require the user to specify the number of clusters in advance. Recently, some of these methods have been generalized to graphs with signed weights [14, 19, 46], whereas others let the user specify must-link and cannot-link constraints between clusters [20, 65, 73].

This work reformulates the clustering algorithms of [39, 51, 75] in a generalized framework and adopts ideas from the proposal-free instance segmentation methods [50, 55, 75] to predict edge weights of a graph.

3. Generalized framework for agglomerative clustering of signed graphs

3.1. Notation

Graph formalism – We consider an undirected simple edge-weighted graph $\mathcal{G}(V, E, w^+, w^-)$ with both attractive and repulsive edge attributes. The weight function $w^+ : E \to \mathbb{R}^+$ associates to every edge a positive scalar attribute $w_e^+ \in \mathbb{R}^+$ representing a merge affinity or a similarity measure. On the other hand, $w^- : E \to \mathbb{R}^+$ associates to each edge a split tendency $w_e^- \in \mathbb{R}^+$. Graphs of the type $\mathcal{G}(V, E, w^+, w^-)$ are often defined as *signed graphs* $\mathcal{G}(V, E, w)$, featuring positive and negative edge weights $w_e \in \mathbb{R}$. Following the theoretical considerations in [48], we define signed weights as $w_e = w_e^+ - w_e^-$.

Multicut objective – We call the set $\Pi = \{S_1, \dots, S_K\}$ a *clustering* or *partitioning* if $V = \cup_{S\in\Pi} S$, $S \cap S' = \emptyset$ for different clusters S, S' and every cluster $S \in \Pi$ induces a connected subgraph of $\mathcal{G}$. For any clustering Π of $\mathcal{G}$, we denote as $E_\Pi^0 = \{e_{uv} \in E \,|\, \exists S \in \Pi : u, v \in S\}$ the set of edges linking nodes in the same cluster. Its complementary set $E_\Pi^1 = E \setminus E_\Pi^0$ of edges linking nodes belonging to distinct clusters, is known as the *multicut* of $\mathcal{G}$ associated to clustering Π. The instance of the NP-hard *minimum cost multicut problem* w.r.t. $\mathcal{G}(V, E, w_e)$ is the task of finding a clustering that optimally balances the attraction and repulsion in the graph and is given by the following binary integer program:

$$\min_\Pi \sum_{e\in E} w_e x_e^\Pi, \qquad \text{where} \quad x_e^\Pi = \begin{cases} 1 & \text{if } e \in E_\Pi^1 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Linkage criteria and hierarchical trees – Let the interaction $\mathcal{W}(S \cup S') \in \mathbb{R}$ between two clusters S, S' be defined as a function, named *linkage criterion*, depending on the weights of *all* edges connecting clusters S and S'. The linkage criteria tested in this article are listed and defined in Table 1. A *dendrogram* T is a rooted binary tree[2] representing the merging order of an agglomerative algorithm, such that the leaves of the tree are in one-to-one correspondence with V and each node of the tree represents a merge between two clusters. Let $T_\mathrm{R}, T_\mathrm{L} \subset T$ denote the subtrees rooted at the two children of the root node in T. For any two leaves $u, v \in V$, let $T[u \vee v]$ be the subtree rooted at the least common ancestor $(u \vee v) \in T$ of nodes u and v (furthest from the root), and let $\texttt{leaves}(T[u \vee v]) \subseteq V$ be the set of leaves of this subtree. Given an agglomerative algorithm with merging tree T, let $h_T : V \times V \to \mathbb{N}$ denote the *dendrogram-height* of each $(u \vee v) \in T$, which is defined as the iteration number at which nodes $u, v \in V$ were merged by the algorithm (see example in Fig. 1a). We also define $\mathcal{W}_T(u, v)$ as the signed interaction $\mathcal{W}(S \cup S')$ between the two clusters S, S' that were merged at iteration $h_T(u, v)$:

$$\mathcal{W}_T(u, v) \equiv \mathcal{W}\big(\texttt{leaves}(T_\mathrm{R}[u\vee v]) \cup \texttt{leaves}(T_\mathrm{L}[u\vee v])\big) \tag{2}$$

[2] In general, one could look at trees that are not binary. However, the algorithms discussed in this paper always generate binary hierarchical trees, so nothing would be gained by this generalization.

3.2. The GASP algorithm

Our main contribution is a generalized agglomerative algorithm for signed graph partitioning (GASP) that generalizes hierarchical clustering (HC) to signed graphs. The framework, defined in the following, encompasses several known and new agglomerative algorithms on display in Table 1, which are differentiated by the linkage criterion employed, similarly to HC.

In Algorithm 1, we provide simplified pseudo-code for the proposed GASP algorithm. GASP implements a bottom-up approach that starts by assigning each node to its own cluster and then iteratively merges pairs of adjacent clusters. The algorithm proceeds in three phases.

In phase one, GASP selects the pair of clusters with the highest absolute interaction $|\mathcal{W}(S \cup S')|$, so that the most attractive and the most repulsive pairs are analyzed first. If the interaction is repulsive and the algorithm option *addCannotLinkConstraints* is `True`, then the two clusters are constrained so that their members can never merge in subsequent steps of phase one. If the interaction is attractive, then the clusters are merged, provided that they were not previously constrained. After each merge, the interaction between the merged cluster and its neighbors is updated according to one of the linkage criteria $\mathcal{W}(S \cup S')$ listed in Table 1. Phase one terminates when all the remaining clusters are either constrained or share repulsive interactions. Note that, on unsigned graphs, in phase one all nodes are merged into a single cluster and GASP is then equivalent to a standard hierarchical clustering algorithm.

Phase two: Now that the clusters have grown in size, the algorithm removes the constraints previously introduced in phase one and merges all the clusters that still share an attractive interaction, merging the most attractive one first[3]. The final clustering Π^* returned by GASP is found at the end of phase two and it is then composed of clusters sharing only mutual repulsive interactions.

Finally, in phase three, the algorithm keeps merging all clusters until only a single one is left and then returns the hierarchical tree T^* representing the full sequence of merging steps. The algorithm was implemented using a standard HC implementation with computational complexity $\mathcal{O}(N^2 \log N)$ (details left in Appendix A6.1).

3.3. GASP: New and existing algorithms

In this paper we focus on five linkage methods (see columns of Table 1). Many more linkage criteria have been applied to unsigned graphs [24, 28, 61], involving median-based[4] or size-regularized methods, but we decided to focus this paper on those five criteria because they represent the most popular choices.

Algorithm 1 GASP

Input: Graph $\mathcal{G}(V, E, w^+, w^-)$; linkage criterion $\mathcal{W}$; boolean addCannotLinkConstraints
Output: Final clustering Π^*, rooted binary hierarchical tree T^*

Initial clustering: $\Pi = \{\{v_1\}, \dots, \{v_{|V|}\}\}$
Initialize hierarchical tree T^* with leaf nodes $V = \{v_1, \dots, v_{|V|}\}$
Initialize cluster interactions with $w_e = w_e^+ - w_e^-, \forall e \in E$
// Phase 1: Merge positive interactions (possibly using constraints)
Push incident nodes of every edge $e \in E$ to priority queue (PQ) with priority $|w_e|$
repeat
  Pop $S, S' \in \Pi$ with highest interaction $|\mathcal{W}(S \cup S')|$ from PQ
  if $[\mathcal{W}(S \cup S') > 0]$ **and** $[S, S'$ **not** constrained$]$ **then**
    Merge clusters S, S' and update hierarchical tree T^*
    Update interactions & constraints with neighboring clusters
  else if addCannotLinkConstr **and** $[\mathcal{W}(S \cup S') \leq 0]$ **then**
    Add CannotLink Constraint between S and S'
until $[PQ$ is empty$]$
// Phase 2: Remove constraints & merge all positive interactions
Push signed interactions $\mathcal{W}(S \cup S')$ to PQ, $\forall S, S' \in \Pi$
repeat
  Pop $S, S' \in \Pi$ with highest interaction $\mathcal{W}(S \cup S')$ from PQ
  if $[\mathcal{W}(S \cup S') > 0]$ **then**
    Merge clusters S, S' and update hierarchical tree T^*
    Update interactions with neighboring clusters
until $[\mathcal{W}(S \cup S') \leq 0]$
Save the final clustering $\Pi^* \leftarrow \Pi$
// Phase 3: Merge negative interactions until one single cluster is left
repeat
  Pop $S, S' \in \Pi$ with highest interaction $\mathcal{W}(S \cup S')$ from PQ
  Merge clusters S, S' and update hierarchical tree T^*
  Update interactions with neighboring clusters
until [Only one cluster is left in Π]
return Π^*, T^*

Sum Linkage – On signed graphs, the sum of two attractive (or repulsive) interactions is still attractive (repulsive). On the other hand, on unsigned graphs, a strong attractive interaction could be obtained by summing many weak interactions, which depending on the application could be undesirable. This explains why, to our knowledge, an agglomerative algorithm with sum linkage has never been used on unsigned graphs. On signed graphs, such an algorithm was pioneered in [39, 51] and was named Greedy Agglomerative Edge Contraction (GAEC)[5]. GAEC always makes the *greedy choice* that most decreases the multicut objective defined in Eq. 1 each time two clusters with positive interaction are merged[6]. The authors of [51] propose an algorithm named *GreedyFixation*, which is equivalent to phase one of GASP using cannot-link-constraints and a sum linkage.

[3] Note that in the version of GASP without *cannotLinkConstraints*, nothing happens in phase two because all remaining interactions are repulsive.

[4] Median linkage is also implemented in our library (see implementation details Appendix A6.1).

[5] An algorithm equivalent to GAEC was recently independently re-proposed in [12].

[6] In general, GASP cannot be seen as a local search algorithm of the multicut problem (for details see Appendix A6.2).

However, running both phase one and two of GASP with sum linkage (algorithm named HCC-Sum in this paper) performed better than GreedyFixation in our experiments.

AbsMax Linkage – This linkage method is also specific to signed graphs, since on unsigned graphs it would be equivalent to single linkage. Here, we prove that the Mutex Watershed Algorithm [75] can be seen as an agglomerative algorithm with AbsMax linkage (proofs of the following three propositions are given in Appendix A6.3):

Proposition 3.1. *The GASP Algorithm 1 with AbsMax linkage, with or without cannot link constraints, returns the same final clustering* Π^*_{AbsMax} *also returned by the Mutex Watershed Algorithm (MWS) [75], which has empirical complexity* $\mathcal{O}(N \log N)$.

Average, Single, and Complete Linkage – These three linkage criteria have been thoroughly studied on unsigned graphs, but never - until very recently - on signed graphs. In concurrent independent related work [12], the authors prove that applying these three linkage methods to a signed graph is equivalent to applying them to the unsigned graph obtained by shifting all edge weights by a constant. Here, we prove which of the algorithms studied here are "intrinsically signed" and do not have this invariance-property:

Proposition 3.2. *We call an agglomerative algorithm "weight-shift invariant" if the dendrogram* T *returned by the algorithm is invariant w.r.t. a shift of all edge weights* w_e *by a constant* $\alpha \in \mathbb{R}$. *Among the variations of GASP, only hierarchical clustering with Average (HC-Avg), Single (HC-Single), and Complete linkage (HC-Complete) are weight-shift-invariant (see green box in Table 1).*

Although average and single linkage methods have been largely studied on unsigned graphs, to our knowledge, they have never been combined with cannot-link constraints on signed graphs[7], so we name these algorithms *HCC-Avg* and *HCC-Single*.

Algorithms defining an ultrametric – The connection between agglomerative algorithms and ultrametrics[8] is well known. Usually, ultrametrics are associated to strictly positive *similarity* or *dissimilarity* measures on a graph. In our framework, a trivial ultrametric is always given by the height h_T of the dendrogram. However, for some of the GASP variations, we now define an ultrametric based on the edge weights and the signed interactions between clusters, generalizing what has been done for HC on unsigned graphs [33, 58]. To define this measure and prove its ultrametric property, we first map the signed interaction $\mathcal{W}_T$ defined in Eq. 2 to positive "pseudo-distances" $d_T : V \times V \to \mathbb{R}^+$:

$$d_T(u,v) \equiv \begin{cases} 0 & \text{if } u = v \\ M - \mathcal{W}_T(u,v) & \text{if } u \neq v \end{cases} \quad \forall u, v \in V \tag{3}$$

$$\text{where} \quad M \equiv \epsilon + \max_{u',v' \in V,\, u' \neq v'} \mathcal{W}_T(u',v') \tag{4}$$

and where $\epsilon > 0$. We then prove the following proposition:

Proposition 3.3. *Among the algorithms included in the GASP framework (see Table 1), only Mutex Watershed and hierarchical clustering with Average (HC-Avg), Single (HC-Single) and Complete linkage (HC-Complete) define an ultrametric* (V, d_{T^*}), *where* d_{T^*} *is defined in Eq. 3 and* T^* *is the tree returned by the GASP Algorithm 1.*

In summary, in this section we have extended the family of HC algorithms [33, 58] with "weight-based ultrametrics" to signed graphs. Next, we move to their empirical evaluation.

4. Experiments

4.1. Signed graph clustering problems

We evaluate the agglomerative clustering algorithms included in our framework on a large collection of both synthetic and real-world graphs with very different structures. The size of the graphs ranges from a few hundred to hundreds of millions of edges. In this way, we will highlight the strengths and limitations of the different linkage criteria introduced in the last section.

Synthetic SSBM graphs – We first consider synthetic graphs generated by a signed stochastic block model (SSBM). We use an Erdős-Rényi random graph model $\mathcal{G}(N,p)$ with N vertices and edge probability p. Following the approach in [19], we partitioned the graph into k *ground-truth* clusters, such that edges connecting vertices belonging to the same cluster (different clusters, respectively) have Gaussian distributed edge weights centered at $\mu = 1$ ($\mu = -1$, respectively) and with standard deviation $\sigma = 0.1$. To model noise, the sign of the edge weights is flipped independently with probability η.

Existing signed graphs – We use clustering instances from the OpenGM benchmark [34] as well as biomedical segmentation instances [62]. The dataset *Image Segmentation* contains planar region-adjacency-graphs (RAG) that are constructed from superpixel adjacencies of photographs. The *Knott-3D* datasets contains 3D-RAGs arising from volume images acquired by electron microscopy (EM). The set *Modularity Clustering* contains complete graphs constructed from clustering problems on small social networks.

[7] Note that Complete linkage methods return the same clustering whether constraints are enforced or not (proof in Lemma A6.3, in Appendix).

[8] A metric space (X,d) is an *ultrametric* if, for every $x,y,z \in X$, $d(x,y) \leq \max\{d(x,z), d(y,z)\}$.

Clustering problem	Graph Type	$\#I$	$\lvert V\rvert$	$\lvert E\rvert$	Multicut objective values (average across instances, lower is better) GAEC [39]	HCC-Sum	MWS [75]	HC-Avg	HCC-Avg
Modularity Clustering [11]	*complete*	6	34-115	561-6555	-0.457	-0.453	-0.073	**-0.467**	**-0.467**
Image Segmentation [1]	*RAG*	100	156-3764	439-10970	**-2,955**	-2,953	-2,901	-2,903	-2,896
Knott-3D (150-300-450) [2]	*3D-RAG*	24	572-17k	3381-107k	**-36,667**	-36,652	-35,200	-35,957	-35,631
CREMI-3D-RAG (OurCNN)	*3D-RAG*	3	134k-157k	928k-1065k	**-1,112,287**	**-1,112,286**	-1,109,731	-1,112,177	-1,112,100
Fruit-Fly Level 1-4 [62]	*3D-RAG*	4	5m-11m	28m-72m	**-151,022**	-151,017	-150,879	-150,909	-150,876
CREMI-gridGraph (OurCNN)	*gridGraph*	15	39m	140m	-73,317,601	-73,328,867	-73,330,568	**-73,502,947**	-73,474,856
Fruit-Fly Level Global [62]	*3D-RAG*	1	90m	650m	**-151,688**	-151,596	-146,315	-150,466	-150,171

Table 2. List of compared signed graph clustering problems: for each, we specify the number of instances $\#I$, number of nodes $|V|$, and number of edges $|E|$ per instance. We compare algorithms in the GASP framework by their value of the multicut objective defined in Eq. 1 (lower is better).

The *Fruit-Fly* 3D-RAG instances were generated from volume image scans of fruit fly brain matter. Instances *Level 1-4* are progressively simplified versions of the global problem obtained via block-wise domain decomposition [62].

Grid-graphs from CNN predictions – We also evaluate the clustering methods on the task of neuron segmentation in EM image volumes using training data from the CREMI 2016 EM Segmentation Challenge [27]. We train a 3D U-Net [17, 67] using the same architecture as [28] and predict long-and-short range affinities as described in [50]. The predicted affinities $a_e \in [0, 1]$, which represent how likely it is for a pair of pixels to belong to the same neuron segment, are then mapped to signed edge weights $w_e = a_e - 0.5$, resulting in a 3D grid-graph having a node for each pixel/voxel of the image[9]. We divided the three CREMI training samples, consisting of ∼196 million voxels each, into five sub-blocks for a total of 15 clustering problems (named *CREMI-gridGraph* in Table 2). See Appendix A6.5 for extended details about training, data augmentation, and how we remove tiny clusters left after running GASP on the *CREMI-gridGraph* clustering problems.

3D-RAG from CNN-predictions – Lastly, we use the predictions of our CNN model to generate three graph instances (one for each CREMI training sample, named *CREMI-3D-RAG* in Table 2), which have very similar structure to the *Knott-3D* and *Fruit-Fly* instances. We obtain these problems by using a pipeline that is very common in neuron segmentation: a watershed algorithm generates superpixels and from those a 3D region-adjacency graph is built, where edge weights are given by the CNN predictions averaged over the boundaries of adjacent superpixels (details in Appendix A6.5).

4.2. Comparison of results and discussion

Multicut objective values – In Table 2, we report the values of the multicut objective obtained for clustering with different GASP algorithms[10]. Although many heuristics were proposed to better optimize this objective [8, 9, 38], these methods are out of the scope of this paper, since they do not scale to the largest graph instances considered here. By looking at results in Table 2, we observe that GAEC almost always achieves the lowest objective values, expect in the *CREMI-gridGraph* instances. Despite this, on graphs where a ground truth clustering is known, GAEC does not achieve the lowest ARAND errors (see Tables 3a) and 3b)).

Size of growing clusters: Sum vs Avg linkage – In all the studied clustering problems, we empirically observe that sum-linkage algorithms like GAEC grow clusters one after the other, as shown in Fig. 1d and Fig. 3 by the agglomeration order of GAEC[11]. This is intuitively explained by the following: initially, many of the most attractive edge weights have very similar values; when the two nodes u, v with the highest attraction are merged, there is a high chance that they will have a common neighboring node t belonging to the same cluster; thus, the interaction between the merged nodes uv and t is likely assigned to the highest priority, because it is given by the sum of two highly attractive edge weights. This will then start a "chain reaction" where only a single cluster is agglomerated at the time. In the following, we will show how this unique *flooding strategy* of the sum-linkage methods can be both an advantage or a disadvantage, depending on the type of clustering problem.

Comparison to spectral clustering – The spectral clustering methods for signed graphs SPONGE$_{sym}$ and SPONGE proposed by [19] achieved state of the art performances on SSBM synthetic graphs. Their competitive performances are also confirmed by our experiments in Fig. 2. However, these methods do not scale up to the large graph instances considered here and they also require the user to specify the true number of clusters in advance, which is not known for other graph instances tested in this paper. In Appendix, Table A6, we report the scores achieved by these methods on a much smaller sub-instance of the *CREMI-gridGraph* problem: even when the true number of clusters is specified in advance for the spectral methods, they perform much worse than other GASP algorithms, with an

[9] To map affinities to signed weights, we also tested the *logarithmic mapping* proposed in [2, 25], but it performed worse in our experiments.

[10] Objective values achieved by Single and Complete linkage methods are much worse compared to other algorithms and are reported in Table A5, in Appendix.

[11] This *flooding agglomeration-strategy* of GAEC was also observed in [36].

	ARAND Error	VOI split	VOI merge	Runtime (s)
HC-Avg	**0.0487**	0.387	0.258	2344
HCC-Avg	0.0492	0.389	0.259	2892
MWS [75]	0.0554	0.440	0.249	688
GAEC [39]	0.0856	0.356	0.338	4717
HCC-Sum	0.0872	0.365	0.337	4970
HC-Complete	0.9211	4.536	**0.211**	1020
HC-Single	0.9264	**0.060**	4.887	**312**
HCC-Single	0.9264	**0.060**	4.887	6440

a) *CREMI-gridGraph (OurCNN)*

	ARAND Error	VOI split	VOI merge	Runtime (s)
HC-Avg	**0.0896**	0.603	0.323	86
HCC-Avg	0.0898	0.600	0.325	87
GAEC [39]	0.0905	0.606	0.323	89
HCC-Sum	0.0910	0.608	0.323	**85**
MWS [75]	0.1145	0.825	0.295	86
HCC-Single	0.5282	**0.437**	1.367	88
HC-Single	0.5282	**0.437**	1.367	**85**
HC-Complete	0.5654	2.253	**0.249**	86

b) *CREMI-3D-RAG (OurCNN)*

	Needs superpixels?	CREMI Score	ARAND Error	VOI split	VOI merge
OurCNN: 3D-RAG + LiftedMulticut	✗	**0.221**	**0.108**	**0.339**	0.115
GASP: OurCNN + gridGraph + HCC-Avg	□	0.224	0.113	0.361	0.085
GASP: OurCNN + gridGraph + HC-Avg	□	0.224	0.114	0.364	0.083
PNI CNN [50]	✗	0.228	0.116	0.345	0.106
LSI-Masks [6]	□	0.246	0.125	0.383	0.107
GASP: OurCNN + 3D-RAG + HCC-Avg	✗	0.257	0.132	0.438	**0.063**
GASP: OurCNN + 3D-RAG + HC-Avg	✗	0.262	0.135	0.448	**0.063**
MALA CNN + MC [28]	✗	0.276	0.132	0.490	0.089
CRU-Net [78]	✗	0.566	0.229	1.081	0.389

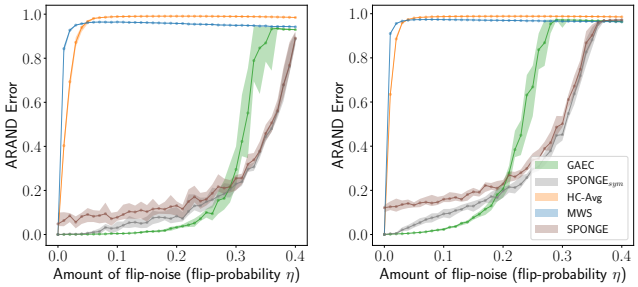
c) CREMI Challenge leader-board

Table 3. **Tables (a-b)**: Scores and run times of algorithms in the GASP framework on the *CREMI-gridGraph* and *CREMI-3D-RAG* clustering problems: average linkage methods achieved the best accuracy. Measures shown are: Adapted-Rand error (ARAND, lower is better); Variation of Information (VOI) [4] (VOI-merge for under-clustering error and VOI-split for over-clustering error, lower values are better). **Table (c)**: Current leading entries in the CREMI challenge leaderboard (November 2021). CREMI-score is given by the geometric mean of (VOI-split + VOI-merge) and ARAND error (lower is better).

accuracy penalty of almost 50%. For these reasons, we exclude them from our other comparison experiments.

GASP on synthetic SSBM graphs – GASP algorithms using cannotLinkConstraints are not expected to perform well on these graphs, because of the type of employed sign noise, so we focus our comparison only on the GAEC, HC-Avg and MWS algorithms (using Sum, Average, and AbsMax linkage methods, respectively). Empirically, we observe that GAEC is the agglomerative algorithm performing best on SSBM graphs, on par with spectral method SPONGE$_{sym}$ (see Fig. 2). Given the simple properties of SSBM graphs, we can now give a detailed explanation of these empirical results. In SSBM graphs, the number of edges E_{ij} connecting two clusters S_i, S_j is proportional to the product $|S_i| \cdot |S_j|$ of cluster sizes. With Sum or Avg linkage methods, due to the law of large numbers, the flipping noise is "averaged out" as soon as the set E_{ij} becomes larger and clusters grow in size. On the other hand, when clusters are small, it can happen that, for few clusters, several of their edges in E_{ij} are flipped and the algorithm makes a mistake by merging two clusters belonging to different ground truth communities. From this observation, it follows that the *flooding strategy* of the sum-linkage algorithm GAEC is a very good strategy on these types of graphs, because clusters are immediately grown in size (see dendrograms in Fig. 3). Average linkage method HC-Avg instead performs much worse on these graphs because it grows small equally-sized clusters and makes several wrong merge-decisions at the beginning. Lastly, the MWS algorithm is not expected to perform well on these graphs because of the high sensitivity of the AbsMax linkage to flipping noise. In Proposition A6.2 (see Appendix), we prove that, at every iteration, the MWS algorithm makes a mistake with at least probability η, independently on the sizes of the two clusters that are popped from priority queue. In summary, for the SSBM, we can obtain a deep understanding of the dynamics induced by various linkage criteria, and find that GAEC gives highest accuracy by a large margin.

Scores on CREMI instance-segmentation – SSBM



a) $k = 20, p = 0.1$ b) $k = 50, p = 0.2$

Figure 2. ARAND errors (median values over 20 experiments, lower is better) on synthetic graphs generated with SSBM. We consider k ground truth communities of random size. Graphs have $N = 10000$ nodes; edges are randomly added with probability p.

graphs are *non-planar*, and every edge has the same probability to be present in the graph. On the other hand, the *gridGraph* and *3D-RAG* graphs of Table 2 are sparse and have a very regular structure: regardless of whether a node represents a pixel or a superpixel, it will only have edge connections with its neighbors in the image (up to a certain hop distance). Tables 3a)-3b) show that average linkage methods (HC-Avg, HCC-Avg) strongly outperform other methods on *CREMI-gridGraph* instances and also achieve the best scores on *CREMI-3D-rag* graphs. Sum-based linkage methods (GAEC, HCC-Sum) have a two times higher ARAND error on grid-graphs and often return under-clustered segments (see failure cases in Fig. 4). This suggests that the *flooding strategy* observed previously in the sum-linkage methods does not work on grid-graphs, because in this setup edge weights are predicted by a CNN and noise is strongly spatially-correlated [12]. To fully test this hypothesis, we conduct a set of experiments where the CNN predictions are perturbed by adding structured noise and simulating addi-

[12] This effect is not as strong on *3D-RAG* graphs, because edge weights are computed by averaging CNN predictions (and noise) over the boundaries of adjacent supervoxels.

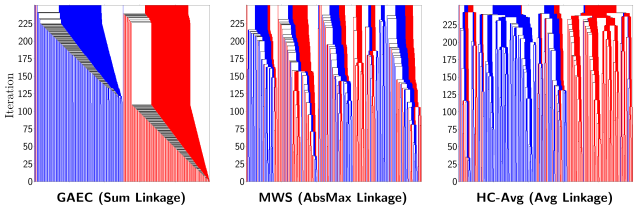


Figure 3. Clustering dynamics and accuracy of GASP variations on stochastic block models. The dendrograms result from three versions of GASP on a synthetic graph generated with SSBM (250 nodes, edge probability $p = 0.05$, flipping probability $\eta = 0.1$). Red and blue colors show which of the two equal-sized ground-truth communities each node belongs to. At the top, dendrograms are truncated at the level of the final clustering Π^* returned by GASP.

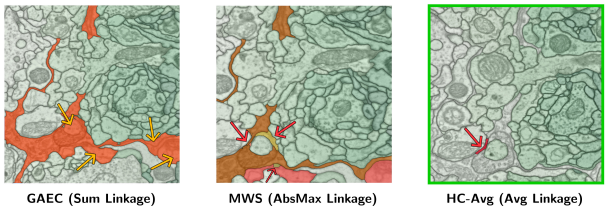


Figure 4. Failure cases of three versions of GASP applied to neuron segmentation. Only *wrongly* segmented regions are highlighted in different warm colors. Red arrows point to wrongly split regions; yellow arrows point to false merge errors. HC-Avg returned the best segmentation. Data is 3D, hence the same color could be assigned to parts of segments that appear disconnected in 2D.

tional artifacts like "holes" in the boundary evidence[13]. The plot in Fig. 5 confirms that HC-Avg and HCC-Avg are very robust algorihtms on this data, followed by Sum-linkage algorithms and the Mutex Watershed algorithm (MWS). It is not a surprise that the AbsMax linkage used by MWS is not robust to this type of structured noise. However, the scores and runtimes in Table 3a) prove how MWS can achieve high accuracy with 70% lower runtime compared to HC-Avg.

Complete and Single Linkage – We use these two linkage methods as baselines to highlight the difficulty of the studied graph clustering problems listed in Table 2. Scores in Tables 3a)-3b) show their poor performance: Single linkage hierarchical clustering (HC-Single), which here is equivalent to thresholding the edge weights at $w_e = 0$ and computing connected components in the graph, often returned few big under-segmented clusters. HC-Complete returned instead a lot of over-segmented clusters.

Results on CREMI challenge – Table 3c) shows that the HCC-Avg and HC-Avg clustering algorithms achieve state-of-the-art accuracy on the CREMI challenge, when combined with predictions of our CNN. Most of the other entries (apart from *LSI-Masks* [6]) employ super-pixels based post-processing pipelines and cluster 3D-region-adjacency graphs. As we show in Table 3b), using superpixels considerably reduces the size of the clustering problem and, consequently, the post-processing time. However, our method operating directly on pixels (*gridGraph + HCC-Avg*) achieves better performances than superpixel-based methods (*3D-RAG + HCC-Avg*) and does not require the parameter tuning necessary to obtain good super-pixels, which is usually highly dataset dependent. To scale up our method operating on pixels, we divided each test-volume into four sub-blocks, and then combined the resulting clusterings by running the algorithms again on the combined graph. The method *3D-RAG + LiftedMulticut* based on the lifted multicut approximation of [10] achieves the best scores overall, but it takes into account different information through the lifted edge weights that also depend on additional raw-data and shape information from highly engineered super-pixels.

[13] See Appendix A6.6 for details about how we perturbed the *CREMI-gridGraph* problems by using Perlin noise [63, 64], which is one of the most common gradient noises used in procedural pattern generation.

5. Conclusion

We have presented a unifying framework for agglomerative clustering of graphs with both positive and negative edge weights. This framework allowed us to explore new combinations of constraints and linkage criteria and to perform a consistent evaluation of all algorithms in it. We have then analyzed several theoretical and empirical properties of these algorithms. On instance segmentation, algorithms based on an average linkage criterion outperformed all the others: they proved to be simple and robust approaches to process short- and long-range predictions of a CNN. On biological images, these simple average agglomeration algorithms achieve state-of-the-art results without requiring the user to spend much time tuning complex task-dependent pipelines based on super-pixels.

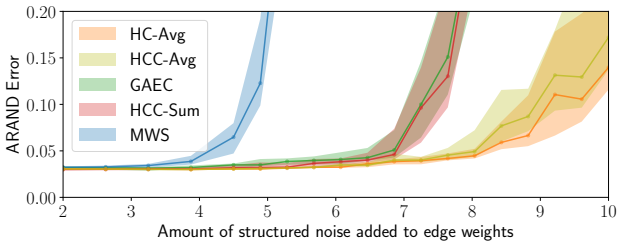


Figure 5. ARAND errors (median values over 20 experiments, lower is better) on *CREMI-gridGraph* clustering problems perturbed with structured noise. Average-linkage algorithms proved to be the most robust.

References

[1] Bjoern Andres, Jörg H Kappes, Thorsten Beier, Ullrich Köthe, and Fred A Hamprecht. Probabilistic image segmentation with closedness constraints. In *2011 International Conference on Computer Vision*, pages 2611–2618. IEEE, 2011. 6

[2] Bjoern Andres, Thorben Kroeger, Kevin L Briggman, Winfried Denk, Natalya Korogod, Graham Knott, Ullrich Koethe, and Fred A Hamprecht. Globally optimal closed-surface segmentation for connectomics. In *European Conference on Computer Vision*, pages 778–791. Springer, 2012. 3, 6

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011. 3

[4] Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireşan, Alessandro Giusti, Luca M Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, 9:142, 2015. 7

[5] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5221–5229, 2017. 2

[6] Alberto Bailoni, Constantin Pape, Steffen Wolf, Anna Kreshuk, and Fred A Hamprecht. Proposal-free volumetric instance segmentation from latent single-instance masks. *arXiv preprint arXiv:2009.04998*, 2020. 7, 8

[7] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56(1-3):89–113, 2004. 3

[8] Thorsten Beier, Björn Andres, Ullrich Köthe, and Fred A Hamprecht. An efficient fusion move algorithm for the minimum cost lifted multicut problem. In *European Conference on Computer Vision*, pages 715–730. Springer, 2016. 1, 3, 6

[9] Thorsten Beier, Thorben Kroeger, Jorg H Kappes, Ullrich Kothe, and Fred A Hamprecht. Cut, glue & cut: A fast, approximate solver for multicut partitioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 2014. 6

[10] Thorsten Beier, Constantin Pape, Nasim Rahaman, Timo Prange, Stuart Berg, Davi D Bock, Albert Cardona, Graham W Knott, Stephen M Plaza, Louis K Scheffer, et al. Multicut brings automated neurite segmentation closer to human performance. *Nature Methods*, 14(2):101, 2017. 3, 8

[11] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE transactions on knowledge and data engineering*, 20(2):172–188, 2007. 6

[12] Morteza Haghir Chehreghani. Hierarchical correlation clustering and tree preserving embedding. *arXiv preprint arXiv:2002.07756*, 2020. 4, 5

[13] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab. *arXiv preprint arXiv:1910.04751*, 2019. 2

[14] Kai-Yang Chiang, Joyce Jiyoung Whang, and Inderjit S Dhillon. Scalable clustering of signed networks using balance normalized cut. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 615–624. ACM, 2012. 3

[15] Sunil Chopra and Mendu R Rao. On the multiway cut polyhedron. *Networks*, 21(1):51–89, 1991. 1

[16] Sunil Chopra and Mendu R Rao. The partition problem. *Mathematical Programming*, 59(1-3):87–115, 1993. 3

[17] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 6

[18] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012. 3

[19] Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. SPONGE: A generalized eigenproblem for clustering signed networks. In *AISTATS*, 2019. 3, 5, 6

[20] Mihai Cucuringu, Ioannis Koutis, Sanjay Chawla, Gary Miller, and Richard Peng. Simple and scalable constrained clustering: a generalized spectral method. In *Artificial Intelligence and Statistics*, pages 445–454, 2016. 3

[21] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 2

[22] Erik D Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2-3):172–187, 2006. 3

[23] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P Murphy. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017. 2

[24] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004. 3, 4

[25] Jenny Rose Finkel and Christopher D Manning. Enforcing transitivity in coreference resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 45–48. Association for Computational Linguistics, 2008. 6

[26] Jan Funke, Fred A Hamprecht, and Chong Zhang. Learning to segment: training hierarchical segmentation under a topological loss. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 268–275. Springer, 2015. 3

[27] Jan Funke, Stephan Saalfeld, Davi Bock, Srini Turaga, and Eric Perlman. Cremi challenge. https://cremi.org., 2016. Accessed: 2019-11-15. 6

[28] Jan Funke, Fabian David Tschopp, William Grisaitis, Arlo Sheridan, Chandan Singh, Stephan Saalfeld, and Srinivas C Turaga. Large scale image segmentation with structured loss

based deep learning for connectome reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 3, 4, 6, 7

[29] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. SSAP: Single-shot instance segmentation with affinity pyramid. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[30] Martin Grötschel and Yoshiko Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45(1-3):59–96, 1989. 3

[31] Martin Grötschel and Yoshiko Wakabayashi. Facets of the clique partitioning polytope. *Mathematical Programming*, 47(1-3):367–387, 1990. 3

[32] Michał Januszewski, Jörgen Kornfeld, Peter H Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. High-precision automated reconstruction of neurons with flood-filling networks. *Nature methods*, 15(8):605, 2018. 3

[33] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967. 5

[34] Joerg Kappes, Bjoern Andres, Fred Hamprecht, Christoph Schnorr, Sebastian Nowozin, Dhruv Batra, Sungwoong Kim, Bernhard Kausler, Jan Lellmann, Nikos Komodakis, et al. A comparative study of modern inference techniques for discrete energy minimization problems. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1328–1335, 2013. 5

[35] Jörg Hendrik Kappes, Markus Speth, Björn Andres, Gerhard Reinelt, and Christoph Schnörr. Globally optimal image partitioning by multicuts. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 31–44. Springer, 2011. 1

[36] Amirhossein Kardoost and Margret Keuper. Solving minimum cost lifted multicut problems by node agglomeration. In *ACCV 2018, 14th Asian Conference on Computer Vision*, Perth, Australia, 2018. 1, 3, 6

[37] Verena Kaynig, Amelio Vazquez-Reina, Seymour Knowles-Barley, Mike Roberts, Thouis R Jones, Narayanan Kasthuri, Eric Miller, Jeff Lichtman, and Hanspeter Pfister. Large-scale automatic reconstruction of neuronal processes from electron microscopy images. *Medical image analysis*, 22(1):77–88, 2015. 3

[38] Brian W Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970. 6

[39] Margret Keuper, Evgeny Levinkov, Nicolas Bonneel, Guillaume Lavoué, Thomas Brox, and Bjorn Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1751–1759, 2015. 1, 2, 3, 4, 6, 7

[40] B Ravi Kiran and Jean Serra. Global–local optimizations by hierarchical cuts and climbing energies. *Pattern Recognition*, 47(1):12–24, 2014. 3

[41] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. Instancecut: from edges to instances with multicut. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5008–5017, 2017. 2

[42] Seymour Knowles-Barley, Verena Kaynig, Thouis Ray Jones, Alyssa Wilson, Joshua Morgan, Dongil Lee, Daniel Berger, Narayanan Kasthuri, Jeff W Lichtman, and Hanspeter Pfister. RhoanaNet pipeline: Dense automatic neural annotation. *arXiv preprint arXiv:1611.06973*, 2016. 3

[43] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint arXiv:1511.07386*, 2015. 3

[44] Shu Kong and Charless C Fowlkes. Recurrent pixel embedding for instance grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9018–9028, 2018. 2

[45] Nikola Krasowski, Thorsten Beier, Graham W Knott, Ullrich Koethe, Fred A Hamprecht, and Anna Kreshuk. Improving 3D EM data segmentation by joint optimization over boundary evidence and biological priors. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 536–539. IEEE, 2015. 3

[46] Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. SIAM, 2010. 3

[47] Jan-Hendrik Lange, Bjoern Andres, and Paul Swoboda. Combinatorial persistency criteria for multicut and max-cut. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6093–6102, 2019. 1, 3

[48] Jan-Hendrik Lange, Andreas Karrenbauer, and Bjoern Andres. Partial optimality and fast lower bounds for weighted correlation clustering. In *International Conference on Machine Learning*, pages 2898–2907, 2018. 3

[49] Kisuk Lee, Ran Lu, Kyle Luther, and H Sebastian Seung. Learning dense voxel embeddings for 3D neuron reconstruction. *arXiv preprint arXiv:1909.09872*, 2019. 2

[50] Kisuk Lee, Jonathan Zung, Peter Li, Viren Jain, and H Sebastian Seung. Superhuman accuracy on the SNEMI3D connectomics challenge. *arXiv preprint arXiv:1706.00120*, 2017. 3, 6, 7

[51] Evgeny Levinkov, Alexander Kirillov, and Bjoern Andres. A comparative study of local search algorithms for correlation clustering. In *German Conference on Pattern Recognition*, pages 103–114. Springer, 2017. 1, 3, 4

[52] Ting Liu, Cory Jones, Mojtaba Seyedhosseini, and Tolga Tasdizen. A modular hierarchical approach to 3D electron microscopy image segmentation. *Journal of neuroscience methods*, 226:88–102, 2014. 3

[53] Ting Liu, Mojtaba Seyedhosseini, and Tolga Tasdizen. Image segmentation using hierarchical merge tree. *IEEE transactions on image processing*, 25(10):4596–4607, 2016. 3

[54] Ting Liu, Miaomiao Zhang, Mehran Javanmardi, Nisha Ramesh, and Tolga Tasdizen. SSHMT: Semi-supervised hierarchical merge tree for electron microscopy image segmentation. In *European Conference on Computer Vision*, pages 144–159. Springer, 2016. 3

[55] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. Affinity derivation and graph

merge for instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 686–703, 2018. 3

[56] Filip Malmberg, Robin Strand, and Ingela Nyström. Generalized hard constraints for graph segmentation. In *Scandinavian Conference on Image Analysis*, pages 36–47. Springer, 2011. 3

[57] Yaron Meirovitch, Alexander Matveev, Hayk Saribekyan, David Budden, David Rolnick, Gergely Odor, Seymour Knowles-Barley, Thouis Raymond Jones, Hanspeter Pfister, Jeff William Lichtman, et al. A multi-pass approach to large-scale connectomics. *arXiv preprint arXiv:1612.02120*, 2016. 3

[58] Glenn W Milligan. Ultrametric hierarchical clustering algorithms. *Psychometrika*, 44(3):343–346, 1979. 5

[59] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8837–8845, 2019. 2

[60] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017. 2

[61] Juan Nunez-Iglesias, Ryan Kennedy, Toufiq Parag, Jianbo Shi, and Dmitri B Chklovskii. Machine learning of hierarchical clustering to segment 2D and 3D images. *PloS one*, 8(8):e71715, 2013. 3, 4

[62] Constantin Pape, Thorsten Beier, Peter Li, Viren Jain, Davi D Bock, and Anna Kreshuk. Solving large multicut problems for connectomics via domain decomposition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–10, 2017. 1, 3, 5, 6

[63] Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985. 8

[64] Ken Perlin. Noise hardware. *Real-Time Shading SIGGRAPH Course Notes*, 2001. 8

[65] Syama Sundar Rangapuram and Matthias Hein. Constrained 1-spectral clustering. In *AISTATS*, volume 30, page 90, 2012. 3

[66] Zhile Ren and Gregory Shakhnarovich. Image segmentation by cascaded region agglomeration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2018, 2013. 3

[67] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 6

[68] Philippe Salembier and Luis Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE transactions on Image Processing*, 9(4):561–576, 2000. 3

[69] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 265–273. Springer, 2018. 3

[70] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. AdaptIS: Adaptive instance selection network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7355–7363, 2019. 2

[71] Srinivas C Turaga, Kevin L Briggman, Moritz Helmstaedter, Winfried Denk, and H Sebastian Seung. Maximin affinity learning of image segmentation. pages 1865–1873, 2009. 3

[72] Mustafa Gokhan Uzunbas, Chao Chen, and Dimitris Metaxas. An efficient conditional random field approach for automatic and interactive neuron segmentation. *Medical image analysis*, 27:31–44, 2016. 3

[73] Xiang Wang, Buyue Qian, and Ian Davidson. On constrained spectral clustering and its applications. *Data Mining and Knowledge Discovery*, 28(1):1–30, 2014. 3

[74] Steffen Wolf, Alberto Bailoni, Constantin Pape, Nasim Rahaman, Anna Kreshuk, Ullrich Köthe, and Fred A Hamprecht. The mutex watershed and its objective: Efficient, parameter-free graph partitioning. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3724–3738, 2020. 3

[75] Steffen Wolf, Constantin Pape, Alberto Bailoni, Nasim Rahaman, Anna Kreshuk, Ullrich Kothe, and FredA Hamprecht. The mutex watershed: Efficient, parameter-free image partitioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 546–562, 2018. 1, 2, 3, 5, 6, 7

[76] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proc. ICCV'15*, pages 1395–1403, 2015. 3

[77] Julian Yarkony, Alexander Ihler, and Charless C Fowlkes. Fast planar correlation clustering for image segmentation. In *European Conference on Computer Vision*, pages 568–581. Springer, 2012. 1, 3

[78] Tao Zeng, Bian Wu, and Shuiwang Ji. DeepEM3D: approaching human-level performance on 3D anisotropic EM image segmentation. *Bioinformatics*, 33(16):2555–2562, 2017. 7