Karlsruhe Reports in Informatics 2011,17
Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

Density-Constrained Graph Clustering

Robert Görke, Andrea Schumm, and Dorothea Wagner

2011

KIT – University of the State of Baden-Wuerttemberg and National
Research Center of the Helmholtz Association

Density-Constrained Graph Clustering[⋆]

Robert Görke, Andrea Schumm, and Dorothea Wagner

Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Germany

Abstract. Clusterings of graphs are often constructed and evaluated with the aid of a quality measure. Numerous such measures exist, some of which adapt an established measure for graph cuts to clusterings. In this work we pursue the problem of finding clusterings which simultaneously feature guaranteed intra- and good intercluster quality. To this end we systematically assemble a range of cut-based bicriteria measures and, after showing $\mathcal{NP}$-hardness for some, focus on the classic heuristic of constrained greedy agglomeration. We identify key behavioral traits of a measure, (dis-)prove them for each one proposed and show how these translate to algorithmic efficiency. We close by giving ILP formulations.

1 Introduction

The guiding intuition in the field of graph clustering is "intracluster density vs. intercluster sparsity". Mathematical formalizations thereof abound, most of which, however, incorporate both aspects into a single criterion, which then serves as a quality measure for graph clusterings. Balance between the two aspects is a fine line and treating them separately allows to adjust their tradeoff as to fit given desiderata. While the recent literature on graph clustering (we recommend [9,7] for an overview) has mainly been focusing on large data sets and on single criteria such as Modularity [11], Kannan et al. [10] propose to minimize the cut between, subject to a guaranteed conductance within the clusters and show that this approach avoids the drawbacks of many simpler measures. This stepping stone in bicriterial graph clustering inspired Flake et al. [6], who give an algorithm with provable, but interdependent bounds on both intra- and a variant intercluster expansion. Brandes et al. [3] were the first to use a notion of intercluster conductance to experimentally evaluate clustering algorithms.

Together with sparsity, expansion and conductance are well-known and indisputable measures for quantifying the clarity of a cut, and each one suggests adaptions to measuring clusterings with respect to both aspects. However, only very few have so far been coined and used. We systematically assemble such measures and set our main focus on scrutinizing their behavior in the light of the question which combinations enable efficient greedy agglomeration, putting aside other algorithmic approaches [7]. This classic hierarchical technique used for clustering [7,2,5] starts with singletons and iteratively merges clusters, usually driven by some objective function, until a stopping criterion is met. We

[⋆] This work was partially supported by the DFG under grant WA 654/19-1

show that algorithmic efficiency and behavior strongly depends on three traits of a measure, roughly described as the range of feasible merges in terms of connectedness, the robustness of comparisons between merges and the monotonicity of a measure. For each established measure we prove or disprove these traits, leading both to qualitative insights and assertions on time and space complexity. We further motivate the use of a greedy heuristic by showing $\mathcal{NP}$-hardness for some of our problem statements and give a brief discussion how the constraints and objectives we use can be cast into integer linear programs. Systematic experiments evaluating how well these measures conform to human intuition and how well the proposed algorithms discover existing clusterings are beyond the scope of this work.

Notation and Preliminaries. Let $G = (V, E)$ be an undirected, unweighted, and simple graph[1]. We set $|V| =: n, |E| =: m$ and $\mathcal{C} = \{C_1, \ldots, C_k\}$ to be a partition of V. We call $\mathcal{C}$ a *clustering* of G and sets C_i *clusters*. Note that we restrict ourselves to disjoint clusters in this work. The cluster containing vertex v is $\mathcal{C}(v)$. A clustering is *trivial* if either $k = 1$ (*all-clustering*), or each cluster contains only one element (*singletons*). $\binom{\mathcal{C}}{2}$ denotes the set of all unordered pairs of clusters. We identify cluster C with the set of nodes it constitutes and with its vertex-induced subgraph of G. Then $E(\mathcal{C}) := \bigcup_{C \in \mathcal{C}} E(C)$ are called *intracluster* edges and $E \setminus E(\mathcal{C})$ *intercluster* edges. For two subsets A and B of V, $m_{A,B} := |\{\{u, v\} \in E \mid u \in A, v \in B\}|$ is the number of edges between A and B, $n_A := |A|$ is the number of vertices in A, $m_A := |E(A)|$ is its number of intracluster edges and $x_A := m_{A,V \setminus A}$ the number of intercluster edges incident to A. Further, the *volume* v_A of A is defined as $v_A := \sum_{v \in A} \deg(v)$. For $A \neq B \in \mathcal{C}$, we call $\{A, B\}$ a *merge* and abbreviate $AB := A \cup B$. Then, $\mathcal{C}_{\{A,B\}} := \mathcal{C} \setminus \{A, B\} \cup \{AB\}$ is the result of this merge. A *clustering measure* is a function that maps clusterings to real numbers, thereby assessing the quality of a clustering. We define high quality to correspond to high (low) values of intracluster (intercluster) measures.

2 Quality Measures for Clusterings

The bicriteria measures we construct and use in this work build upon conductance, expansion and density. The *conductance* of a cut (S, T) measures the bottleneck between S and T, defined as $\frac{m_{S,T}}{\min\{v_S, v_T\}}$; *expansion* substitutes volume by cardinality: $\frac{m_{S,T}}{\min\{n_S, n_T\}}$. The *density* (or *sparsity*) of a cut is $\frac{m_{S,T}}{n_S n_T}$, which equals the *uniform minimum-ratio cut*; the *density of a graph* is $\frac{m}{0.5n(n-1)}$, and the *conductance of a graph* is $\min_{S \subseteq V} \frac{m_{S,V \setminus S}}{\min\{v_S, v_{V \setminus S}\}}$ (expansion is analogous). *Intracluster measures* quantify how well the vertices within a cluster are interconnected. For *intercluster measures*, we distinguish two ways of measuring cuts: between pairs of clusters (*pairwise*), or cutting off a cluster (*isolating*). Hereby, isolated measures assess how well a cluster is separated from the remainder of the graph and pairwise measures how well the clusters are separated from one

[1] A *simple* graph in this work is both loopless and has no parallel edges.

another. To have these quantities express the quality of an entire clustering, we can either construct a worst-case measure (*minimum/maximum*) or an *average* measure. Density works analogously, however, it also lends itself to the natural idea of adding up all values before normalization (*global*). A simple alternative is globally counting intercluster edges. For convenience we list the measures thus constructed in Tables 1 and 2 and henceforth use their abbreviations.

The very evaluation of both the conductance and the expansion of a graph is $\mathcal{NP}$-hard ([1] and [8] respectively).While there are many ways to deal with this, it generally discourages the use of these functions as intracluster measures. [2] Density, by contrast, is well suited, yielding gid, mid and aid; we refer to a cluster C's contribution as $\mathsf{id}(C) = \frac{2m_C}{n_C(n_C-1)}$. Intercluster cuts, in turn, are efficiently computable. Thus, in accordance with the above classification, we define all twelve resulting intercluster measures (Tables 1 and 2), plus the two measures with a global nature: gxd and nxe. Adhering to our abbreviations, we denote individual clusters' contributions by $\mathsf{ixd}(C) := \frac{x_C}{n_C n_{V \setminus C}}$, $\mathsf{ixc}(C) := \frac{x_C}{\min\{v_C, v_{V \setminus C}\}}$ and $\mathsf{pxd}(\{C, D\}) := \frac{m_{C,D}}{n_C n_D}$, furthermore we call the number of intracluster edges $\mathsf{nie} := n - \mathsf{nxe}$. We generally define the intracluster density of a singleton to be 1, and, analogously, the intercluster quality of the all-clustering to be maximal. Any other choice is counterintuitive on trivial examples such as a clique or a clique plus an isolated vertex.

Table 1: Density and counting

intracluster density		
global	gid	$\frac{\sum_{C \in \mathcal{C}} m_C}{\sum_{C \in \mathcal{C}} \binom{n_C}{2}}$
minimum	mid	$\min\limits_{C \in \mathcal{C}} \frac{m_C}{\binom{n_C}{2}}$
average	aid	$\frac{1}{\lvert\mathcal{C}\rvert} \sum\limits_{C \in \mathcal{C}} \frac{m_C}{\binom{n_C}{2}}$

intercluster density		
global	gxd	$\frac{\sum_{A \neq B \in \mathcal{C}} m_{A,B}}{\sum_{A \neq B \in \mathcal{C}} n_A n_B}$
max. pw.	mpxd	$\max\limits_{A \neq B \in \mathcal{C}} \frac{m_{A,B}}{n_A n_B}$
max. is.	mixd	$\max\limits_{C \in \mathcal{C}} \frac{x_C}{n_C n_{V \setminus C}}$
av. pw.	apxd	$\binom{\lvert\mathcal{C}\rvert}{2}^{-1} \sum\limits_{\{A,B\} \in \binom{\mathcal{C}}{2}} \frac{m_{A,B}}{n_A n_B}$
av. is.	aixd	$\frac{1}{\lvert\mathcal{C}\rvert} \sum\limits_{C \in \mathcal{C}} \frac{x_C}{n_C n_{V \setminus C}}$

intercluster edges		
global	nxe	$\sum\limits_{\{A,B\} \in \binom{\mathcal{C}}{2}} m_{A,B}$

Qualitative Observations. While all proposed intra- and intercluster measures are based on the same intuition, there are fundamental differences in the

Table 2: Intracluster measures based on conductance and expansion

	intercluster conductance		intercluster expansion	
maximum pairwise	mpxc	$\max_{A \neq B \in \mathcal{C}} \frac{m_{A,B}}{\min\{v_A, v_B\}}$	mpxe	$\max_{A \neq B \in \mathcal{C}} \frac{m_{A,B}}{\min\{n_A, n_B\}}$
maximum isolated	mixc	$\max\limits_{C \in \mathcal{C}} \frac{m_{C,V \setminus C}}{\min\{v_C, v_{C \setminus S}\}}$	mixe	$\max\limits_{C \in \mathcal{C}} \frac{m_{C,V \setminus C}}{\min\{n_C, n_{C \setminus S}\}}$
average pairwise	apxc	$\frac{1}{\binom{\lvert\mathcal{C}\rvert}{2}} \sum\limits_{\{A,B\} \in \binom{\mathcal{C}}{2}} \frac{m_{A,B}}{\min\{v_A, v_B\}}$	apxe	$\frac{1}{\binom{\lvert\mathcal{C}\rvert}{2}} \sum\limits_{\{A,B\} \in \binom{\mathcal{C}}{2}} \frac{m_{A,B}}{\min\{n_A, n_B\}}$
average isolated	aixc	$\frac{1}{\lvert\mathcal{C}\rvert} \sum\limits_{C \in \mathcal{C}} \frac{m_{C,V \setminus C}}{\min\{v_C, v_{V \setminus C}\}}$	aixe	$\frac{1}{\lvert\mathcal{C}\rvert} \sum\limits_{C \in \mathcal{C}} \frac{m_{C,V \setminus C}}{\min\{n_C, n_{V \setminus C}\}}$

[2] Note that a bottom-up approach cannot use the approximation results used in [10].

way they assess particular clusterings. One important point is whether balanced clusterings, i.e., homogeneous cluster sizes, are rewarded or penalized. As an example aid has a tendency to favor unbalanced clusterings, as singletons degeneratively yield optimum values and it is easy to compensate the existence of a large cluster with poor intracluster density with an appropriate number of singletons. In contrast to that, mid rewards balanced clusterings, as clusters that are larger than the average are more likely to have low intracluster density and thus to be the qualitative bottleneck. Gid ranges somewhere between these extremes. Using the number of intercluster edges to measure intercluster quality clearly favors unbalanced clusterings, as cutting off small portions of the vertex set from the remainder of the graph usually cuts far fewer edges than partitioning the graphs in two blocks of roughly equal size. To some extent this effect can be compensated by combining nxe with an appropriate intracluster measure.

In the context of intercluster measures, another interesting aspect is how vertices that are only loosely connected to the remainder of the graph are handled. For example, singletons with degree one have a low intercluster density of $1/(|V|-1)$ but maximum intercluster conductance of one. Thus, algorithms minimizing intercluster conductance are prone to put "outsiders" in the clusters of their neighbors, while algorithms minimizing intercluster density will tend to consider these vertices as singletons. Both views can be motivated, depending on the desiderata: If a vertex is linked to just a single vertex of a larger group, it can be hardly considered as an integral part of this group and should thus be treated separately. On the other hand, this vertex has no links to other groups and thus, from its point of view, it clearly has a strong affiliation to the group of its neighbor.

2.1 Problem Statement

In the following we narrow down the myriad formalizations for combining intra- and intercluster quality and state the problem we focus on. Not only do these two aspects capture different properties of a clustering, they even tend to oppose each other: Fine clusterings easily achieve high intracluster density but poor intercluster quality, while the converse is true for coarse clusterings. In the light of a bottom-up strategy, intercluster density aspires a coarse clustering and starts out poorly, which suggests using it as the driving objective function. By contrast, intracluster density starts out with optimum quality, which, on the one hand, discourages using it as the driving force, but, on the other hand, suggests it as a suitable constraint. We thus formalize our problem statement as follows, an exemplary instance and its solution are given in Fig. 1.

*Problem (*Density-Constrained Clustering*).* Given a graph $G = (V, E)$, among all clusterings with an intracluster density of no less than α, find a clustering $\mathcal{C}$ with optimum intercluster quality.

2.2 Complexity

An exhaustive study of hardness results for all combinations of intra- and intercluster measures is beyond the scope of this work. We exemplarily show $\mathcal{NP}$-

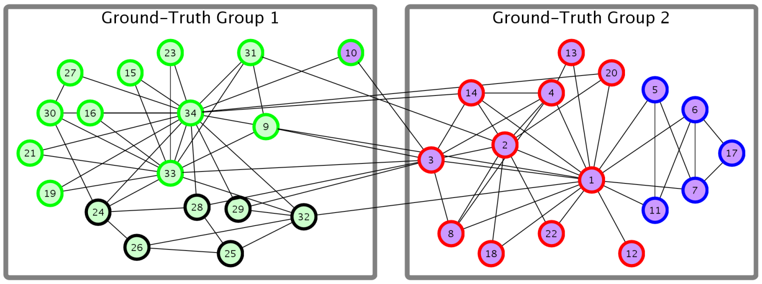


Fig. 1: Zachary's karate club [12] represents a social network and is traditionally used for a test of feasibility in the graph clustering literature. Groups represent the split of the network in reality, fill colors depict the optimal solution to our problem statement using nxe constrained by mid with $\alpha = 0.25$. Reviewing an optimal solution (see Sect. 4) helps judging this measure's usefulness independently of an algorithmic approach. For comparison, border colors indicate a Modularity-optimal clustering [11]. By contrast, aid yields the all-clustering with the exception of one singleton vertex (12), pointing out its undesirable tendency to allow degeneratively imbalanced clusterings.

hardness for Density-Constrained Clustering combining mid, aid or gid with nxe, but conjecture $\mathcal{NP}$-hardness for all remaining combinations. We use that, if we set $\alpha = 1$, the decision variant of our problem is equivalent to the following problem.

*Problem (*Min-Cut Clique Partition*).* Given a graph $G = (V, E)$, is there a partition $\mathcal{C}$ of V into cliques such that $\mathsf{nie}(\mathcal{C})$ is at least k?

This problem is similar to both the classic problem Partition into Cliques [8], which instead minimizes the *number* of cliques and the edge-maximizing variant of the $\mathcal{K}_r$-Packing Problem [4], which differs in that it only allows cliques with bounded size. To the best of our knowledge, Min-Cut Clique Partition has not yet been investigated. We reduce from Exact Cover by 3-Sets [8].

*Problem (*Exact Cover by 3-Sets, X3C*).* Given set $\mathcal{X}$ with $|\mathcal{X}| = 3q$ and collection $\mathcal{S}$ of 3-element subsets of $\mathcal{X}$. Does $\mathcal{S}$ contain an exact cover for $\mathcal{X}$, i.e., a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that every $x \in \mathcal{X}$ occurs in exactly one $S \in \mathcal{S}'$?

We transform an instance $I = (\mathcal{X}, \mathcal{S})$ of X3C with $|\mathcal{X}| =: n$ into a graph $G(I)$ as follows. For each $x \in \mathcal{X}$ we add a vertex v_x, and interconnect the resulting set $V_{\mathcal{X}}$ into an n-clique. Then, we map each set $S \in \mathcal{S}$ to an n-clique $K_n(S)$. For $x \in S$, we link v_x with each vertex in $K_n(S)$. A sketch of this reduction is given in Fig. 2, clearly, it is polynomial.

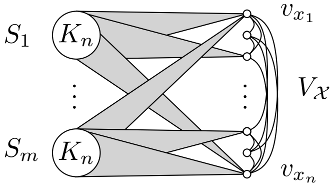


Fig. 2: Sketch of reduction

Lemma 1. *Let $I = (\mathcal{X}, \mathcal{S})$ be an instance of* X3C*. Then, I is solvable iff there exists a partition $\mathcal{C}$ of $G(I)$ into cliques such that $\mathsf{ixc}(\mathcal{C})$ is at least $|\mathcal{S}| \cdot \binom{n}{2} + n^2 + n$.*

Algorithm 1: Generic Greedy Agglomeration

```
Input  : graph $G = (V, E)$, function $\mathsf{objective}$, (constraint-) predicate $\mathsf{allowed}$
Output: clustering $\mathcal{C}$ of $G$
$C \leftarrow$ singletons
$\mathcal{A} \leftarrow \{\{A, B\} \in \binom{\mathcal{C}}{2} \mid \mathsf{allowed}(\mathcal{C}_{A,B})$ and $\mathsf{objective}(\mathcal{C}_{A,B}) \leq \mathsf{objective}(\mathcal{C})\}$
while $|\mathcal{C}| > 1$ and $\mathcal{A} \neq \emptyset$ do
    $M \leftarrow \arg\min_{M \in \mathcal{A}} \{\mathsf{objective}(\mathcal{C}_M)\}$
    $\mathcal{C} \leftarrow \mathcal{C}_M$
    $\mathcal{A} \leftarrow \{\{A, B\} \in \binom{\mathcal{C}}{2} \mid \mathsf{allowed}(\mathcal{C}_{A,B})$ and $\mathsf{objective}(\mathcal{C}_{A,B}) \leq \mathsf{objective}(\mathcal{C})\}$
return $\mathcal{C}$
```

Proof. $\Rightarrow$: Let $\mathcal{S}' \subseteq \mathcal{S}$ be an exact cover for $\mathcal{X}$. For each set $S \in \mathcal{S}$ we introduce a cluster containing $K_n(S)$. If $S = \{a, b, c\}$ is in $\mathcal{S}'$, this cluster also contains the vertices v_a, v_b and v_c. As $\mathcal{S}'$ is an exact cover, the set of these clusters is a partition $\mathcal{C}$ of the vertices in $G(I)$. $\mathcal{C}$ partitions the subgraph induced by $V_{\mathcal{X}}$ into triangles, inducing n intracluster edges in this subgraph. With this, it is easy to see that there are exactly $|\mathcal{S}| \cdot \binom{n}{2} + n^2 + n$ intracluster edges.

$\Leftarrow$: Let $\mathcal{C}'$ be a partition of $G(I)$ into cliques such that the number of intracluster edges is at least $|\mathcal{S}| \cdot \binom{n}{2} + n^2 + n$. Without loss of generality we assume that each element in $\mathcal{X}$ belongs to at least one set in $\mathcal{S}$. We transform $\mathcal{C}'$ into a partition $\mathcal{C}$ of $G(I)$ into cliques without decreasing nie such that (i) each $K_n(S)$ is completely contained in a cluster and (ii) each v_x is contained in one of the clusters containing a $K_n(S)$ with $x \in S$. Assume that for a set $S = \{a, b, c\}$ $K_n(S)$ is fragmented, i.e., there is more than one cluster C in $\mathcal{C}'$ with $C \cap K_n(S) \neq \emptyset$. Then, each such C is contained in the larger clique $K_n(S) \cup \{v_a, v_b, v_c\}$. Thus, merging these clusters is valid and clearly increases nie, which yields (i). A vertex v_x that does not share a cluster with any $K_n(S)$ of an $S \ni x$ can contribute at most $|V_{\mathcal{X}} \setminus \{v_x\}| = n - 1$ intracluster edges. Thus, moving v_x to any cluster C containing a $K_n(S)$ with $x \in S$ increases nie, which yields (ii). In the transformed clustering $\mathcal{C}$, the cliques $K_n(S)$ contribute $|\mathcal{S}| \cdot \binom{n}{2}$, and the linkage between them and $V_{\mathcal{X}}$ contributes n^2 to nie. Thus, the subgraph G' induced by $V_{\mathcal{X}}$ must contain at least n intracluster edges, implying that the mean number of intracluster edges a vertex in G' is incident to is two. Since all sets contain only three elements, at most three of the vertices in $V_{\mathcal{X}}$ can share a cluster, hence, each vertex is incident to at most two intracluster edges in G'. It follows that $\mathcal{C}$ partitions G' into disjoint triangles. This means that either all vertices corresponding to the elements in a subset S share a cluster with $K_n(S)$, or none of them. Hence, $\mathcal{C}$ induces an exact cover of $\mathcal{X}$. □

Corollary 2. Min-Cut Clique Partitioning *is $\mathcal{NP}$-hard.*

3 Generic Greedy Agglomeration

The general structure of a greedy merge algorithm based on an objective function is given in Alg. 1. The idea is to choose from a constrained set of allowed merges the one that improves (w.l.o.g. minimizes) the objective function the most. The Modularity-based approach [5] fits into this concept if we set the objective function to be negative Modularity and use no constraint. Recalling our problem statement, our objective is to minimize intercluster density, subject to the restriction that no merge decreases intracluster density below a given threshold $\alpha \leq 1$. If $\mathsf{allowed}(\mathcal{C}_{A,B})$ only depends on A and B, in each step of Alg. 1 at most $2n-3$ elements are deleted from and at most $n-2$ new elements are inserted into $\mathcal{A}$. Together with the condition that elements can be compared in constant time and that $\mathcal{A}$ can be maintained in a heap, using benefits as keys, the time complexity of Alg. 1 is in $O(n^2 \log n)$. Before we detail this observation (Sect. 3.2), we first determine whether intercluster measures as objectives efficiently drive greedy agglomeration, in that they iteratively suggest eligible merges.

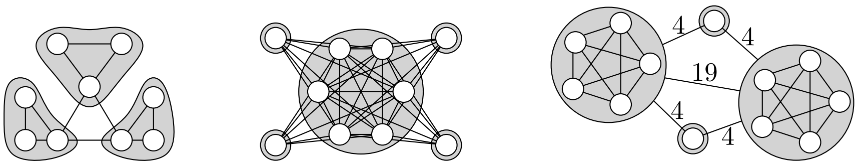
3.1 Merge Behavior

An objective function f is said to have *unbounded merge behavior* if for any clustering $\mathcal{C}$ with at least two clusters, there exist clusters $A, B \in \mathcal{C}$, such that merge $\{A, B\}$ does not increase f. We elucidate the merge behavior of each proposed intercluster measure, either by proving its unboundedness or by giving an example instance which poses a local minimum (summarized in Tab. 3 in App. C). We defer the proofs of all affirmative observations to App. B, and summarize them in the following proposition.

Proposition 3. *The intercluster measures* nxe*,* gxd*,* mixc*,* mixe*,* aixc*,* aixe*,* mixd *and* mpxd *exhibit unbounded merge behavior.*

Roughly speaking, the ingredient common to all proofs on maximum measures is the fact that, by investigating the adjacencies of the worst cluster B, we can always identify some worst contributor to B's value as an eligible partner for a merge. Likewise, aixc (aixe) allows us to find one or more clusters with a detrimental contribution, and then identify those adjacent clusters which are mainly liable for this as candidates for an improving merge. All the proofs are constructive in that they point out how to find a non-increasing merge.

Bounded Merge Behavior. From our set of fourteen objective functions, the remaining six do not have unbounded merge behavior, but can instead get stuck in local minima, such that no further merge is non-increasing. Thus, even without a constraint, the all-clustering cannot be reached. In Fig. 3 we give specific instances which are local minima of mpxc and mpxe (a), apxd, apxc and apxe (b), and of aixd (c). The common intuition for average measures is that a merge must not reduce the number of beneficial clusters (or pairs thereof) too dearly. Mpxc and mpxe are prone to local minima near balanced clusterings. Roughly speaking, this is due to the case distinction in the denominator of their base measures ruining arguments analogous to those in Proposition 17 for mpxd.



(a) mpxc, mpxe (b) apxd, apxc, apxe (c) aixd (edges are summarized)

Fig. 3: These instances illustrate bounded merge behavior. Given clustering $\mathcal{C}$ (grey), no further merge is non-increasing for the measures pointed out.

3.2 Impact of Clustering Measures on Running Times

We already gave conditions under which Alg. 1 can be implemented with a time complexity in $O(n^2 \log n)$. Here, we first review the constraints' impact, and then examine how the stated conditions can be relaxed without losing efficiency.

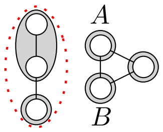
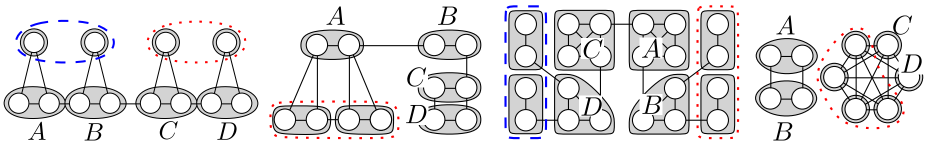


Fig. 4

Intra-Density. Using constraints potentially impedes quick agglomeration, as it does not suffice to determine the merge that improves the objective function the most. The good news is that if we use mid as a constraint, the feasibility of a merge only depends on the density of the merged cluster, which is clearly independent of the remainder of the clustering, and thus need only be checked once, incurring no penalty in running time. However, for gid and aid, this does not hold, as the status of a merge can change from allowed to disallowed and back again. In Fig. 4, starting from the gray clustering, merging the path to the left is not allowed if the constraint $\mathsf{gid}(\mathcal{C}) \geq 0.7$ is used. If singletons A and B are merged, this is allowed again, as the number of intracluster pairs increases. Figure 8 in App. C shows a similar example for aid.

As a heuristic approach, if it is possible to store all merges in a binary search tree, sorted by their benefit to the objective function, this tree can be traversed until we find a feasible merge. This might be more efficient than just searching through all possible merges, as the number of disallowed merges we encounter is limited to the number of more beneficial ones. However, in the worst case, it yields no improvement. We therefore focus our analysis on using mid as the constraint, supported by its good behavior in preliminary experiments, and leave a more efficient treatment of gid and aid open.

Locality. Independent of issues concerning the handling of constraints, in the following we resolve which properties an objective function f has to fulfill such that a set of feasible merges can efficiently be maintained in a heap. Intuitively, f should allow us to decide, without knowledge about the remainder of the clustering, which of two given merges is more beneficial to it. If the benefit exclusively depends on the participating clusters, as for nxe alone, this is immediate and decisions never change. For maximum isolated measures, by contrast, a merge



(a) mpxd, apxd (b) mpxc (c) mpxe (d) gxd

Fig. 5: Locality counterexamples: The base clustering consists of the gray clusters. In (a) and (c), if the blue, dashed merge is performed, merge $\{C, D\}$ is better, if the red, dotted merge is performed, $\{A, B\}$ is better. In (b) and (d), in the base clustering, $\{C, D\}$ is better than $\{A, B\}$, if the red, dotted merge is performed, the opposite holds.

can be non-improving at some point of the algorithm and then become improving again. The intuition is to require the existence of a relation $\leq_f$ that almost behaves like ordering the set of merges by their benefit for the objective function but allows for clever tie-breaking. To serve as a comparator in a priority queue, $\leq_f$ should closely resemble a total quasiorder on the set of all possible merges.[3] Informally, we call an objective function *local*, if it allows for such a relation. More formally, we get the following definition. Let us denote the set of all possible merges, i.e., the set of all unordered pairs of subsets of V, as $\mathcal{M}$.

Definition 4. *An objective function f on clusterings is* local*, if there exists a relation $\leq_f$ on $\mathcal{M} \times \mathcal{M}$ such that for any clustering $\mathcal{C}$ and for all M_1, M_2, M_3 in $\mathcal{M}$ with $M_1 \cup M_2 \cup M_3 \subseteq \mathcal{C}$, the following holds:*

$$\begin{array}{lll} M_1 \leq_f M_2 \text{ or } M_2 \leq_f M_1 & & \approx \textit{totality} \\ M_1 \leq_f M_2 \wedge M_2 \leq_f M_3 & \Longrightarrow M_1 \leq_f M_3 & \approx \textit{transitivity} \\ M_1 \leq_f M_2 & \Longrightarrow f(\mathcal{C}_{M_1}) \leq f(\mathcal{C}_{M_2}) & \textit{consistency with } f \end{array}$$

Based on the above considerations, we will now state sufficient conditions for both non-locality (Lemma 5) and locality (Lemmas 6,8), thereby resolving locality for all our objective functions. We exemplarily state the short proof of Lemma 8, but defer all other proofs to App. C.

Lemma 5. *Let f be a clustering measure. If there exists a graph with two clusterings $\mathcal{C}$ and $\mathcal{D}$ both containing two merges M_1 and M_2 such that $f(\mathcal{C}_{M_1}) < f(\mathcal{C}_{M_2})$ and $f(\mathcal{D}_{M_1}) > f(\mathcal{D}_{M_2})$, then f is not local.*

For mpxd, apxd, mpxc, mpxe and gxd, Figure 5 shows examples where the preconditions of Lemma 5 hold, implying that these measures are not local. The rough idea behind the examples in Figs. 5a-5c is that a low pairwise intercluster quality between two clusters can be improved by merging one of the partners with a third cluster. Figure 5d exploits that merging large clusters becomes more attractive if the global intercluster density is low. An instance proving the nonlocality of apxe and apxc is given in Fig. 9 in App. C.

[3] We do not need a proper total quasiorder as we never have to compare pairs of merges that cannot coexist in a clustering, e.g., because the clusters considered intersect.

Lemma 6. *Let f be an objective function such that $f(\mathcal{C})$ can be expressed as $f(\mathcal{C}) = \max_{C \in \mathcal{C}} f'(C)$, with $f'(C)$ solely depending on C. Then, f is local.*

Proof sketch. Let $\mathcal{C}$ be a clustering and $\mathcal{D} = \{D_1, \ldots, D_k\} \subseteq \mathcal{C}$. Then we define $L(\mathcal{D}) := (f'(D_{i_1}) \geq \ldots \geq f'(D_{i_k}))$ to be the non-increasing sequence of function values of all $D_i \in \mathcal{D}$. Let $\leq_\ell$ be the lexicographical order on sequences of real numbers. The proof relies on the observation that for arbitrary merges $M_1 = \{A, B\}$ and $M_2 = \{C, D\} \subseteq \mathcal{C}$, $L(\mathcal{C}_{M_1}) \leq_\ell L(\mathcal{C}_{M_2})$ implies $f(\mathcal{C}_{M_1}) \leq f(\mathcal{C}_{M_2})$. We show that $L(\mathcal{C}_{M_1}) \leq_\ell L(\mathcal{C}_{M_2})$ is equivalent to $L(\{f'(A \cup B), f'(C), f'(D)\}) \leq_\ell L(\{f'(C \cup D), f'(A), f'(B)\})$ and that the latter relation satisfies locality.

Corollary 7. Mixd*,* mixc *and* mixe *are local.*

Lemma 8. *Let f be an objective function such that $f(\mathcal{C})$ can be expressed as $f(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} f'(C)$, with $f'(C)$ solely depending on C. Then, f is local.*

Proof. Let $\mathcal{C}$ be an arbitrary clustering containing four clusters A, B, C and D. Then, $f(\mathcal{C}_{A,B}) \leq f(\mathcal{C}_{C,D})$ implies that

$$\underbrace{f'(A \cup B) - f'(C) - f'(D)}_{:=k_{A,B}} \leq \underbrace{f'(C \cup D) - f'(C) - f'(D)}_{:=k_{C,D}}$$

As for each cluster, f' is independent of the remainder of $\mathcal{C}$, this inequation shows that $\leq_f := \{(\{A, B\}, \{C, D\}) \mid k_{A,B} \leq k_{C,D}\}$ is a quasiorder on $\mathcal{M} \times \mathcal{M}$ such that $\{A, B\} \leq_f \{C, D\}$ implies $f(\mathcal{C}_{A,B}) \leq f(\mathcal{C}_{C,D})$. Thus, f is local. □

Corollary 9. Aixd*,* aixc *and* aixe *are local.*

For nxe, it is easy to see that choosing $\leq_f$ such that $\{A, B\} \leq_f \{C, D\}$ is equivalent to $m_{A,B} \geq m_{C,D}$ satisfies the definition of locality. We have now proven or disproven the locality of all intercluster measures (for a summary see Tab. 3 in App. C). Note that all proofs of locality are constructive in that they induce comparators which can be used to efficiently maintain the set of all possible merges considered by Alg. 1 in a priority queue. For the maximum functions, triples of real numbers can be used as keys, compared as described in the proof of Lemma 6. Using average functions, it suffices to store the values $k_{A,B}$ defined in the proof of Lemma 8. All keys as well as the density of a new cluster can be computed and compared in constant time if, for any two clusters A and B, the values v_A, x_A, n_A and $m_{A,B}$ are maintained. Summarizing, we obtain the following corollary.

Corollary 10. *Algorithm 1 combining* mid *with* mixd*,* mixc*,* mixe*,* aixd*,* aixc*,* aixe *or* nxe *can be implemented with a running time in $O(n^2 \log n)$.*

Disconnected Merges. Whenever no single edge links two clusters, intuitively, merging them should not be beneficial to the clustering, or at least, such a merge should not be the best option. In the light of our bicriterial approach, an objective function which does encourage such a *disconnected merge* is naturally opposed by the separate mechanism of a constraint on the intracluster density. Superficially, this resolves the issue for non-degenerate instances; however, a more accurate assertion is algorithmically relevant: If we can rule out disconnected merges, it suffices to maintain only the set of *connected* merges in the heap (see Alg. 1), of which there are at most m (instead of $\Omega(n^2)$). This implies linear space and—given locality—$O(md \log n)$ time complexity, where d denotes the height of the dendrogram.[4] For sparse graphs ($m \in O(n)$), this bound can be an improvement, since d usually approaches $\log n$ (but never exceeds n). This analysis has initially been observed for Modularity [5], which enforces connected merges. It uses that, for each level of the dendrogram, only $O(m)$ heap entries are updated. In the following we resolve the question whether our objective functions enforce connected merges.

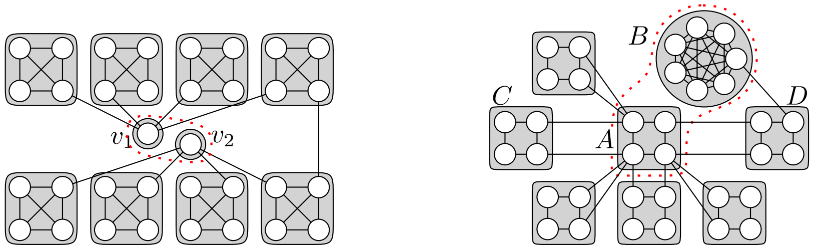
We say an objective function f *enforces connected merges* if for any pair of clusters $C \neq D \in \mathcal{C}$ with $m_{C,D} = 0$ and $f(\mathcal{C}) - f(\mathcal{C}_{\{C,D\}}) > 0$ (i.e., an improving, disconnected merge), there exist clusters $A \neq B \in \mathcal{C}$ such that $m_{A,B} > 0$ and $f(\mathcal{C}_{\{C,D\}}) > f(\mathcal{C}_{\{A,B\}})$ (i.e., a better, connected merge). In fact, only nxe and gxd enforce connected merges in general (see Tab. 3 in App. C). Both measures never even benefit at all from disconnected merges: The former does not change, and the latter even deteriorates. The circumstances under which all other measures potentially encourage disconnected merges are intuitively illustrated in Fig. 6. If most clusters are reasonable, merging two clusters with a particularly ill contribution to the measure can be the best option (Fig. 6a). For all pairwise measures, this is immediate, and it is also not hard to see for all average measures, as, roughly speaking, the number of bad contributors decreases. While the above arguments fail for mixc, mixe and mixd, a disconnected merge of a bad cluster with a very good one can be the best option for them (Fig. 6b and Fig. 10 in App. C). Note that it is always possible to artificially restrict the set of allowed merges to connected ones, yielding a modified greedy algorithm. Evaluating the practical impact of such a restriction shall be subject to an experimental study; in our preliminary experiments, we observed none. We summarize our positive observations in the following corollary, which can be extended to also apply to any local objective function if we restrict ourselves to connected merges.

Corollary 11. *Algorithm 1 combining* mid *with* nxe *can be implemented with a running time in $O(md \log n)$ and linear space complexity.*

4 Integer Linear Programs

In this section we sketch out how the problem statements we deal with can be cast into integer linear programs. Since an in-depth treatment of this topic is beyond the scope of this work, we divert details to App. A, which, however, is still far from exhaustive. The foundation of all our formulations is a set $\mathcal{X}$ of $O(n^2)$ binary variables X_{uv}, which indicate whether vertices u and v share a

[4] A dendrogram is a binary forest with singletons as leaves, and inner vertices representing the merge operations of an agglomerative process.



(a) apxc, apxe, apxd, aixc, aixe, aixd, mpxc, mpxe, mpxd

(b) mixc, mixe

Fig. 6: Given the gray clusterings, disconnected merges (red, dotted) yield the highest improvement for the objective functions pointed out. Thus, all objective functions, except nxe and gxd, potentially favor disconnected merges (for mixd, see Fig. 10, App. C).

cluster ($X_{uv} = 1$) or not ($X_{uv} = 0$). Since for a clustering, $\mathcal{X}$ constitutes an equivalence relation on V, $O(n^3)$ constraints suffice to ensure transitivity within $\mathcal{X}$. Based on $\mathcal{X}$, we can model many convenient values, e.g., whether an edge e is internal to $\mathcal{C}(v)$, the number of intercluster edges of $\mathcal{C}(v)$, $|\mathcal{C}(v)|$ or $|\mathcal{C}|$.

A general obstacle for modeling our measures linearly are the division operations necessary for almost all variants. We can substitute such operations in a brute-force manner: We multiply binary *indicator variables* $J_{v,P}$, which equal 1 if and only if $\mathcal{C}(v)$ meets properties P, with precomputed intermediate results for P, and take the sum over all possible configurations of P. We exemplify the usage of $J_{v,P}$ with measure aixd. To calculate $\mathsf{ixd}(\mathcal{C}(v))$, we need P to contain $n_{\mathcal{C}(v)}$ and $x_{\mathcal{C}(v)}$. Thus, we add constraints forcing $J_{v,\tilde{x},\tilde{n}}$ to equal 1 if and only if $\tilde{n} = n_{\mathcal{C}(v)} \wedge \tilde{x} = x_{\mathcal{C}(v)}$ (see App. A.1). We can then use these indicators as follows: Let $A_v := \sum_{\tilde{x}=0}^{m} \sum_{\tilde{n}=1}^{n} J_{v,\tilde{x},\tilde{n}} \cdot \frac{\tilde{x}}{\tilde{n}(n-\tilde{n})\cdot\tilde{n}}$, which yields $\mathsf{ixd}(\mathcal{C}(v)) = \sum_{w\in\mathcal{C}(v)} A_w$ and thus $\mathsf{aixd}' := \sum_{v\in V} A_v = |\mathcal{C}|\cdot\mathsf{aixd}(\mathcal{C})$. We can circumvent a division by $|\mathcal{C}|$ by more subtle means, avoiding an asymptotic increase in complexity: For all possible values $\tilde{k}$ which $|\mathcal{C}|$ can assume, we add constraints $\mathsf{aixd}(\mathcal{C}) \geq \frac{1}{\tilde{k}} \cdot \mathsf{aixd}' - M_{\tilde{k}}$, with $M_{\tilde{k}}$ being large if $\tilde{k} \neq |\mathcal{C}|$. When minimizing $\mathsf{aixd}(\mathcal{C})$, only the one inequality using the correct $\tilde{k}$ actually constrains, thus this yields the correct result.

Summing up, we end up with $O(n^2m)$ variables, where the number of indicator variables dominates the term. Beyond the $O(n^3)$ constraints used for keeping the base variables $\mathcal{X}$ consistent, and the lesser number of constraints for helper variables, every single indicator variable requires a constant number (roughly 4 for aixd) of constraints, which yields $O(n^2m)$ constraints.

Generally, we can use this method to construct ILPs for all our measures, however, some measures require prohibitively large sets P of properties. For instance, conductance requires additional properties representing the case distinction in the denominator and the volume which roughly incurs a factor $2m$ in the number of variables (and constraints); all pairwise measures require indicators for *pairs* of vertices, adding another factor of n^2 or even nm. Used as

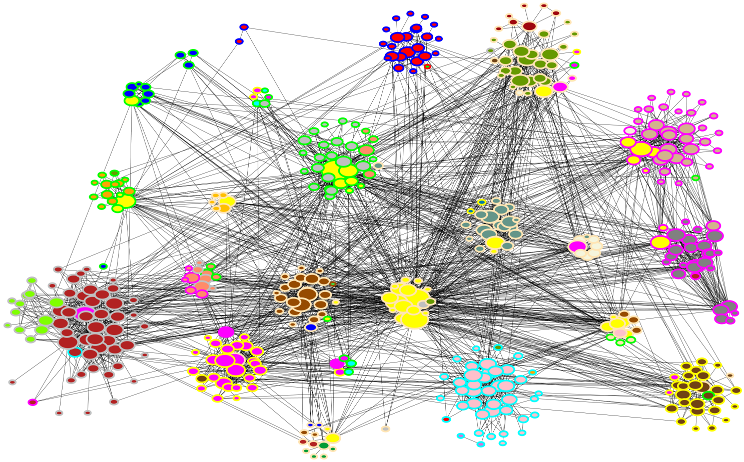


Fig. 7: This graph is a three-month snapshot of the email traffic at KIT's CS department, groups represent chairs, which serve as a ground truth (vertices are scaled by degree, $n = 472, m = 2845$). We ran Alg.1 using mid with $\alpha = 0.25$ and aixc to arrive at the color-clustering. Border colors indicate a Modularity-based clustering [5].

constraints (not as objective functions), the intracluster measures gid and mid are simpler to model, while aid again seems to require a construction as above.

5 Concluding Remarks

Established measures for graph cuts lend themselves well for precisely expressing desiderata on graph clusterings. Despite the scarce attention this approach has received from the graph clustering literature so far, existing studies did indicate its appropriateness. With a focus on finding graph clusterings that feature guaranteed intra- and high intercluster quality, we revived this ansatz and systematically formalized bicriteria quality measures based on expansion, conductance and sparsity. The classification of these measures with respect to their behavior in the context of greedy agglomeration yields conditions that render this widespread heuristic efficient, namely the locality and the connectedness of a measure, which we observed to coincide with common intuition about what is a good cut-based clustering measure. On top of that, we showed that a guaranteed density inside each cluster is especially suited to constrain agglomeration and that most definitions of intercluster quality do not suffer from local minima. We complemented our findings by exemplarily showing $\mathcal{NP}$-hardness for some variants of our problem statement and sketching ILPs for all. An experimental evaluation of density-constrained graph clustering and the adaption to local greedy optimization and to weighted graphs shall be subject to future work. We illustrate the outcome of greedy agglomeration combining guaranteed intracluster density with high average isolated intercluster conductance in Fig. 7.

References

1. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, and A. Marchetti-Spaccamela. *Complexity and Approximation - Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 2nd edition, 2002.
2. P. Berkhin. A Survey of Clustering Data Mining Techniques. In J. Kogan, C. Nicholas, and M. Teboulle, editors, *Grouping Multidimensional Data: Recent Advances in Clustering*, pages 25–71. Springer, 2006.
3. U. Brandes, M. Gaertler, and D. Wagner. Engineering Graph Clustering: Models and Experimental Evaluation. *ACM Journal of Experimental Algorithmics*, 12(1.1):1–26, 2007.
4. F. Chataigner, G. Manic, Y. Wakabayashi, and R. Yuster. Approximation algorithms and hardness results for the clique packing problem. *Electronic Notes in Discrete Mathematics*, 29:397–401, 2007.
5. A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(066111), 2004.
6. G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. Graph Clustering and Minimum Cut Trees. *Internet Mathematics*, 1(4):385–408, 2004.
7. S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75–174, 2009.
8. M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of $\mathcal{NP}$-Completeness*. W. H. Freeman and Company, 1979.
9. A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.
10. R. Kannan, S. Vempala, and A. Vetta. On Clusterings - Good, Bad and Spectral. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS'00)*, pages 367–378, 2000.
11. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(026113), 2004.
12. W. W. Zachary. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33:452–473, 1977.

A Remarks on ILPs

A.1 Building Blocks for Bicriterial Clustering via ILP

***Node-Node-Equivalence* Variables X_{uv}:** "is $C(u) = \mathcal{C}(v)$?" To render our base variables X_{uv} consistent, we use constraints ensuring their transitivity (Eq. 12); reflexivity and symmetry are immediate, since X_{uv} and X_{vu} are the same variable and since we can simply set $X_{vv} \equiv 1$.

$$\mathcal{X} = \{X_{uv} \mid \{u, v\} \in \binom{V}{2}\} \text{ with } X_{uv} = \begin{cases} 1 & \text{if } C(u) = \mathcal{C}(v) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

***Internal-Edge-Indicator* Variables E_{ev}:** "is e inside $\mathcal{C}(v)$?" Using constraints for a binary And (Eq. 14), we align E_{ev} with $\mathcal{X}$ to behave as follows:

$$\mathcal{E} = \{E_{ev} \mid e \in E, v \in V\} \text{ with } E_{\{i,j\}v} = X_{iv} \text{ AND } X_{jv} \tag{2}$$

***External-Edge-Indicator* Variables L_{ev}:** "is e an intercluster edge of $\mathcal{C}(v)$?" Determining e's end-vertices $\{i, j\}$, we align L_{ev} with $\mathcal{X}$ by constraints for binary XOR using X_{iv} and X_{jv} (Eq. 15) which ensures the following behavior:

$$\mathcal{L} = \{L_{ev} \mid e = \{i, j\} \in E, v \in V\} \text{ with } L_{\{i,j\}v} = X_{iv} \text{ \textsc{Xor} } X_{jv} \tag{3}$$

***Cluster-Size* Variables $Z_{v\ell}$:** "is $|\mathcal{C}(v)| = \ell$?" For each $v \in V$ we can write $|\mathcal{C}(v)| := \sum_{u \in V} X_{uv}$, and then use constraints for value matching (see Eq. 16) to align $Z_{v\ell}$ with $\mathcal{X}$ such that the following holds:

$$\mathcal{Z}(V) = \{Z_{v\ell} \mid v \in V, \ell \in \{1, \ldots, n\}\} \text{ with } Z_{v\ell} = \begin{cases} 1 & |\mathcal{C}(v)| = \ell \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

***Über-Node* Variables $\ddot{U}_v$:** "is v über-vertex of $\mathcal{C}(v)$?" We call vertex $v = \min_{u \in C(w)}\{u\}$ the *über-vertex* of $C(w)$, i.e., the smallest index in a cluster.

$$\ddot{\mathcal{U}}(V) = \{\ddot{U}_v \mid v \in V\} \text{ with } \ddot{U}_v = \begin{cases} 1 & v = \min_{u \in \mathcal{C}(v)}\{u\} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$\forall v \in V : \ddot{U}_v \leq 1 - \frac{1}{n}(\sum_{u<v} X_{uv}) \text{ and } \ddot{U}_v \geq 1 - \sum_{u<v} X_{uv} \tag{6}$$

Note that from $\mathcal{E}$, $\mathcal{L}$ and $\ddot{\mathcal{U}}$, we can obtain the values $m_{\mathcal{C}(v)}$, $x_{\mathcal{C}(v)}$, $v_{\mathcal{C}(v)}$ and $|\mathcal{C}|$, respectively, and even $m_{\mathcal{C}(v),\mathcal{C}(u)}$. Building upon the variables above, we can express all measures discussed in this work, however, with no claim about minimal complexity. The following example models the constraint $\mathsf{mid} \geq \alpha$:

$$\forall v \in V : \sum_{e \in E} E_{ev} - \alpha \cdot (\sum_{\ell \in \{1,\ldots,n\}} \binom{\ell}{2} \cdot Z_{v\ell}) \geq 0 \tag{7}$$

$\mathcal{C}(v)$*-Indicator* Variables: "$J_{v,P}$ = does $\mathcal{C}(v)$ meet properties P?" We use these variables as brute-force building blocks for substituting divisions, which we require but cannot formulate linearly. In the following we show how to define appropriate variables $J_{v,P}$ for aixd. For aixd we require P to contain the variables necessary to precompute $\mathsf{ixd}(\mathcal{C}(v))$, which are x_v and n_v. Thus we build a variable for each candidate combination of v, $x_{\mathcal{C}(v)}$ and $n_{\mathcal{C}(v)}$:

$$\mathcal{J}^{\mathsf{aixd}}(V) = \{J_{v,\tilde{x},\tilde{n}} \mid v \in V, \tilde{x} \in \{0, \ldots, m\}, \tilde{n} \in \{1, \ldots, n\}\} \tag{8}$$

$$\text{with } J_{v,\tilde{x},\tilde{n}} = \begin{cases} 1 & \text{if } x_{\mathcal{C}(v)} = \tilde{x} \text{ and } n_{\mathcal{C}(v)} = \tilde{n} \;\; \text{(i.e., } \mathcal{C}(v) \text{ meets } P\text{)} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

But these two properties can easily be enforced as follows, using binary constraints for value matching, as shown in Eq. 16 for general settings, and variables

L_{ev} and X_{uv} as defined above:

$$\text{0 for wrong } \tilde{x}\text{: } J_{v,\tilde{x},\tilde{n}} \leq 1 - \frac{\tilde{x} - \sum_{e \in E} L_{ev}}{m+1}, \quad J_{v,\tilde{x},\tilde{n}} \leq 1 - \frac{\sum_{e \in E} L_{ev} - \tilde{x}}{m+1},$$

$$\text{0 for wrong } \tilde{n}\text{: } J_{v,\tilde{x},\tilde{n}} \leq 1 - \frac{\tilde{n} - \sum_{u \in V} X_{uv}}{n}, \quad J_{v,\tilde{x},\tilde{n}} \leq 1 - \frac{\sum_{u \in V} X_{uv} - \tilde{n}}{n},$$

$$\text{force the one fitting variable to 1: } \forall v \in V : \sum_{\tilde{x}=0}^{m} \sum_{\tilde{n}=1}^{n} J_{v,\tilde{x},\tilde{n}} = 1 \tag{10}$$

With the set of all variables $J_{v,\tilde{x},\tilde{n}}$ at our disposal, we can now compute the value $\mathsf{ixd}(\mathcal{C}(v)) = \frac{x_{\mathcal{C}(v)}}{n_{\mathcal{C}(v)} n_{V \setminus \mathcal{C}(v)}}$ as follows (details on how to proceed with aixd are given in Sect. 4):

$$\mathsf{ixd}(\mathcal{C}(v)) = \sum_{\tilde{x}=0}^{m} \sum_{\tilde{n}=1}^{n} \left(J_{v,\tilde{x},\tilde{n}} \cdot \frac{\tilde{x}}{\tilde{n}(n-\tilde{n})} \right) \tag{11}$$

A.2 Sets of Constraints for Basic Tasks

Transitivity **Constraints:** "$A_{ab} \wedge A_{bc} \Rightarrow A_{ac}$"

$$\forall \{a,b,c\} \in \binom{\text{Set}}{3} : \begin{cases} A_{ab} + A_{bc} - A_{ac} \leq 1 \\ A_{ab} + A_{ac} - A_{bc} \leq 1 \\ A_{ac} + A_{bc} - A_{ab} \leq 1 \end{cases} \tag{12}$$

Reflexivity **and** ***Symmetry*** **Constraints:** "$A_{aa} = \text{TRUE}$" and $A_{ab} \Rightarrow A_{ba}$"

$$\forall a \in \text{ Set } : A_{aa} = \text{ TRUE} \qquad \text{and} \qquad \forall \{a,b\} \in \binom{\text{Set}}{2} : A_{ab} = A_{ba} \tag{13}$$

Binary AND ***Constraints:*** "$T = \bigwedge_{i=1}^{t} A_i$"

$$T \leq \frac{1}{t} \sum_{i=1}^{t} A_i \quad \text{and} \quad T \geq \sum_{i=1}^{t} A_i - (t-1) \tag{14}$$

Binary XOR ***Constraints:*** "$T = A_a$ XOR A_b"

$$T \leq 2 - A_a - A_b \qquad T \leq A_a + A_b \qquad T \geq A_a - A_b \qquad T \geq -A_a + A_b \tag{15}$$

Binary Value-Match **Constraints:** "$T_{t\ell} = (t == \ell)$?" We set t to be the target value, then ℓ searches t's range and tries to match it. An example showing how these constraints can be used for our purposes is given around Eq. 10.

$$T_{t\ell} \leq 1 - \frac{1}{|\text{range}|}(\ell - t), \quad T_{t\ell} \leq 1 - \frac{1}{\text{range}}(t - \ell), \quad \sum_{\ell \in \text{range}} T_{t\ell} = 1 \tag{16}$$

B Proof of Lemma 3 (Unlimited Merge Behavior)

Proof (of Lemma 3). Since this lemma summarizes a number of results, we split the proof into Lemma 14 for mixc, Lemma 15 for aixc, Lemma 17 for mixd and mpxd, Corollary 16 for mixe and aixe and Corollary 12 for gxd and nxe. □

A Vectorial Point of View. For gxd we introduce a convenient point of view we shall repeatedly use in the following. Gxd is defined as $\frac{\sum_{A\neq B\in\mathcal{C}} m_{A,B}}{\sum_{A\neq B\in\mathcal{C}} n_A n_B}$; if we identify a fraction with the two-dimensional vector given by using the numerator and the denominator as the coordinates, we can interpret gxd as the sum of the vectors given by the individual fractions $\frac{m_{A,B}}{n_A n_B}$. The crucial observation is, that there either exists at least one steeper vector $\frac{m_{C,D}}{n_C n_D} > \mathsf{gxd}(\mathcal{C})$, or all are equally steep: $\forall A \neq B : \frac{m_{A,B}}{n_A n_B} = \mathsf{gxd}(\mathcal{C})$. For a formal proof, see Lemma 13. In terms of gxd, merge $\{C, D\}$ is equivalent to vectorially subtracting from $\frac{\sum_{A\neq B\in\mathcal{C}} m_{A,B}}{\sum_{A\neq B\in\mathcal{C}} n_A n_B}$ contributor $\frac{m_{C,D}}{n_C n_D}$. If we pick $\{C, D\}$ to be the steepest contributor, we get $\mathsf{gxd}(\mathcal{C}_{\{C,D\}}) \leq \mathsf{gxd}(\mathcal{C})$, a non-increase. Obviously, nxe cannot ever increase and thus has unbounded merge behavior. Together with Lemma 13 we thus have:

Corollary 12. Nxe *and* gxd *have unbounded merge behavior.*

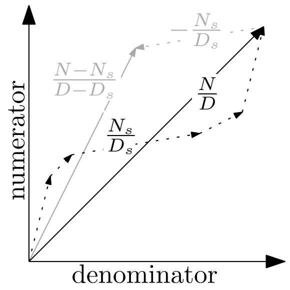
Lemma 13. *Given* $N = \sum_{i=1}^k N_i$ *and* $D = \sum_{i=1}^k D_i$ *with* $\beta = \frac{N}{D}$ *and suppose* $N_i, D_i \geq 0$, $D > 0$ *then exactly one of the following cases holds:*

1. $\exists \ell, s \in [1,\ldots,k] : \dfrac{N_\ell}{D_\ell} > \beta > \dfrac{N_s}{D_s}$ *(strictly larger/smaller contributors)*
2. $\forall i \in [1,\ldots,k] : \dfrac{N_i}{D_i} = \beta$ *(complete equality)*

Proof. We rewrite β as a convex combination of its contributors $\beta_i := \frac{N_i}{D_i}$:

$$\beta = \frac{N}{D} = \frac{\sum_{i=1}^k N_i}{D} = \frac{\sum_{i=1}^k \frac{N_i}{D_i} \cdot D_i}{D}$$
$$= \frac{\sum_{i=1}^k \beta_i \cdot D_i}{D} = \sum_{i=1}^k \beta_i \cdot \frac{D_i}{D}$$

Since $\sum_{i=1}^k D_i = D$, this is indeed a convex combination. Thus, if not all β_i equal β, we have $\beta \in (\min_{s=1}^k \beta_s, \max_{\ell=1}^k \beta_\ell)$, which immediately yields the claim. □



The vectorial point of view of Lemma 13: In a convex combination, if not all contributors are equal, at least one is smaller. The gray vectors illustrate the effect of subtracting a less steep contributor.

Lemma 14. Mixc *has unbounded merge behavior.*

Proof. We focus on the worst cluster $B = \arg\max_{C \in \mathcal{C}} \mathsf{ixc}(C)$ (which need not be unique), and distinguish three cases by v_B. W.l.o.g. we assume $|\mathcal{C}| \geq 3$.

$v_B \geq m$: We express $\beta := \mathsf{ixc}(B)$ as the vectorial sum of other clusters' contributions: $\beta = \frac{x_B}{2m - v_B} = \frac{\sum_{C \neq B} m_{C,B}}{\sum_{C \neq B} v_C}$. There must either exist $A \in \mathcal{C}$ with $\frac{m_{A,B}}{v_A} > \beta$, or we have $\forall C \neq B : \frac{m_{C,B}}{v_C} = \beta$. Since $\mathsf{ixc}(C) \geq \frac{m_{C,B}}{v_C}$, the former case yields $\mathsf{ixc}(A) > \beta$, contradicting our choice of B; for the same reason, the latter case either also yields such a cluster A, or we have $\forall C \neq B : \mathsf{ixc}(C) = \beta$. But then merging any $C, D \neq B$ is non-increasing: $\mathsf{ixc}(CD) = \beta$.

$v_B \leq m$ and $\forall C \neq B : v_B + v_C \geq m$: For an arbitrary merge $\{C, B\}$, by the above arguments, there must either exist A with $\mathsf{ixc}(A) > \mathsf{ixc}(CB)$, or we have $\forall D \neq C, B : \mathsf{ixc}(D) = \mathsf{ixc}(CB)$. In either case merge $\{C, B\}$ was non-increasing.

$\exists A \neq B : v_A + v_B \leq m$: Merge AB yields $\mathsf{ixc}(AB) = \frac{x_A + x_B}{v_A + v_B} \leq \beta$, since $\frac{x_A}{v_A} \leq \frac{x_B}{v_B} = \beta$, and thus is a non-increase. □

Lemma 15. Aixc *has unbounded merge behavior.*

Proof. For clusters $C, D \in \mathcal{C}$ with $v_C + v_D \leq m$, we will use that merging them yields $\mathsf{ixc}(CD) = \frac{x_C + x_D - 2m_{C,D}}{v_C + v_D} \leq \max\{\mathsf{ixc}(C), \mathsf{ixc}(D)\}$. We will show that for each clustering $\mathcal{C}$ with $\mathsf{aixc}(\mathcal{C}) = \beta$, there exists a merge that does not increase the objective function. We divide the clusters of an arbitrary clustering $\mathcal{D}$ into "bad" and "good" clusters, $\mathcal{B}_\mathcal{D} = \{B \in \mathcal{D} \mid \mathsf{ixc}(B) > \beta\}$ and $\mathcal{G}_\mathcal{D} = \{H \in \mathcal{D} \mid \mathsf{ixc}(H) \leq \beta\}$, respectively. Note that for each clustering $\mathcal{D}$, it holds that:

$$|\mathcal{D}| \cdot \mathsf{aixc}(\mathcal{D}) = |\mathcal{D}| \cdot \beta + \underbrace{\textstyle\sum_{B \in \mathcal{B}_\mathcal{D}} (\mathsf{ixc}(B) - \beta)}_{:= b_\mathcal{D}} - \underbrace{\textstyle\sum_{H \in \mathcal{G}_\mathcal{D}} (\beta - \mathsf{ixc}(H))}_{:= g_\mathcal{D}}$$

From this, it directly follows that $\mathsf{aixc}(\mathcal{D}) \leq \beta$ iff $b_\mathcal{D} \leq g_\mathcal{D}$. If we find $B_1 \neq B_2 \in \mathcal{B}_\mathcal{C}$ with $v_{B_1} + v_{B_2} \leq m$, merge $\{B_1, B_2\}$ leaves $\mathcal{G}_\mathcal{C}$ untouched and reduces $b_\mathcal{C}$ by at least $\min\{\mathsf{ixc}(B_1) - \beta, \mathsf{ixc}(B_2) - \beta\} > 0$, and thus improves. Therefore, assume $\forall B_1 \neq B_2 \in \mathcal{B}_\mathcal{C} : v_{B_1} + v_{B_2} > m$. We distinguish four cases by $|\mathcal{B}_\mathcal{C}|$:

$|\mathcal{B}_\mathcal{C}| \geq 4 :$ Since $\sum_{C \in \mathcal{C}} v_C = 2m$, there must exist B_1, B_2 with $v_{B_1} + v_{B_2} \leq m$.

$|\mathcal{B}_\mathcal{C}| = 3 :$ We sort $\mathcal{B}_\mathcal{C}$ by $\mathsf{ixc}(B_1) \geq \mathsf{ixc}(B_2) \geq \mathsf{ixc}(B_3)$. Merge B_1B_2 yields

$$\mathsf{ixc}(B_1B_2) = \frac{x_{B_1B_2}}{\sum_{C \neq B_1, B_2} v_C} = \frac{\sum_{C \neq B_1, B_2} m_{B_1B_2, C}}{\sum_{C \neq B_1, B_2} v_C} \leq \frac{\sum_{C \neq B_1, B_2} x_C}{\sum_{C \neq B_1, B_2} v_C} < \mathsf{ixc}(B_3),$$

which shows that merge $\{B_1, B_2\}$ reduces $b_\mathcal{C}$, leaving $\mathcal{G}_\mathcal{C}$ untouched.

$|\mathcal{B}_\mathcal{C}| = 2 :$ Analogous to case $|\mathcal{B}_\mathcal{C}| = 3$, we can see that $\mathsf{ixc}(B_1B_2) < \beta$.

$|\mathcal{B}_\mathcal{C}| = 1 :$ If, for any $H \in \mathcal{G}_\mathcal{C}$ we have $\mathsf{ixc}(BH) \leq \beta$, merge $\{B, H\}$ is improving; thus, assume $\forall H \in \mathcal{G}_\mathcal{C} : \mathsf{ixc}(BH) > \beta$ $(*)$. This implies (analogous to case $|\mathcal{B}_\mathcal{C}| = 3$) that $v_B + v_H \leq m$ for all $H \in \mathcal{G}_\mathcal{C}$. For merge $\{B, H\}$, we denote by $\Delta_b(H) := x_B/v_B - x_{BH}/v_{BH}$ the change of $b_\mathcal{C}$, and the change of $g_\mathcal{C}$ by $\Delta_g(H) := \beta - x_H/v_H$; the crucial observation is that any merge $\{B, H\}$ for which $\varDelta_b(H) > \varDelta_g(H)$ holds, improves; but such a merge must exist:

$$(|\mathcal{G}_\mathcal{C}|+1)\beta = \sum_{H\in\mathcal{G}_\mathcal{C}} \Big(\frac{x_H}{v_H}\Big) + \frac{x_B}{v_B} = \sum_{H\in\mathcal{G}_\mathcal{C}} \Big(\frac{x_H}{v_H} + \frac{m_{B,H}}{v_B}\Big) \leq \sum_{H\in\mathcal{G}_\mathcal{C}} \Big(\frac{x_H}{v_H} + \frac{2m_{B,H}}{v_B}\Big)$$

$$= \sum_{H\in\mathcal{G}_\mathcal{C}} \Big(\frac{x_H}{v_H} + \frac{x_B}{v_B} - \frac{x_B - 2m_{B,H}}{v_B}\Big) \leq \sum_{H\in\mathcal{G}_\mathcal{C}} \Big(\frac{x_H}{v_H} + \frac{x_B}{v_B} - \underbrace{\frac{x_B - 2m_{B,H} + x_H}{v_B + v_H}}_{=\mathsf{ixc}(BH)}\Big)$$

To see the second inequality, observe the following: By $(*)$ $\mathsf{ixc}(BH) > \beta$ and since $H \in \mathcal{G}_\mathcal{C}$, $\mathsf{ixc}(H) \leq \beta$; thus the vectorial addition of $\frac{x_H}{v_H}$ must have decreased $\frac{x_B - 2m_{B,H}}{v_B}$, and thus increases the right hand side. Since we have only $|\mathcal{G}_\mathcal{C}|$ summands, at least one summand exceeds β:

$$\exists H_0 \in \mathcal{G}_\mathcal{C} : \frac{x_{H_0}}{v_{H_0}} + \frac{x_B}{v_B} - \frac{x_B - 2m_{B,H_0} + x_{H_0}}{v_B + v_{H_0}} > \beta \;\Rightarrow\; \varDelta_b(H_0) > \varDelta_g(H_0) \qquad \square$$

A careful inspection of the proofs of Lemma 14 and Lemma 15 reveals, that if we substitute any occurrence of volume v_C by the size n_C of a cluster, the conclusions remain correct, but translate the results from conductance to expansion:

Corollary 16. Mixe *and* aixe *have unbounded merge behavior.*

Lemma 17. Mixd *and* mpxd *have unbounded merge behavior.*

Proof. We first consider mixd. For a worst cluster $B = \arg\max_{C\in\mathcal{C}} \mathsf{ixd}(C)$ (which need not be unique) with $\mathsf{ixd}(B) =: \beta$, we show how to find a non-increasing merge. Merge CB yields $\mathsf{ixd}(CB) = \frac{x_C + x_B - 2m_{C,B}}{n_C(n-n_C)+n_B(n-n_B)-2n_C n_B}$, which is the vectorial addition of $\mathsf{ixd}(C)$, $\mathsf{ixd}(B)$ and the *correction* $\kappa_C = \frac{2m_{C,B}}{2n_C n_B}$. We can decompose $\mathsf{ixd}(B) = \frac{\sum_{C\neq B} 2m_{C,B}}{\sum_{C\neq B} 2n_C n_B}$, into such corrections κ_C, whence we see that there must exist $A \in \mathcal{C}$ with $\kappa_A \geq \mathsf{ixd}(B)$. Thus, for merge $\{A, B\}$, $\mathsf{ixd}(AB)$ is the vectorial sum of $\mathsf{ixd}(B)$ plus $\mathsf{ixd}(A)$ (at most as steep) minus κ_A (at least as steep), yielding $\mathsf{ixd}(AB) \leq \beta$, a non-increase.

No we turn to mpxd. Let $\{AB\} = \arg\max_{M\in\mathcal{M}} \mathsf{pxd}(M)$, merge $\{A, B\}$ undoes $\beta = \mathsf{pxd}(\{A, B\})$ and cannot cause new, worse values $\mathsf{pxd}(\{AB, C\})$:

$$\forall C \neq AB : \mathsf{pxd}(\{AB, C\}) = \frac{m_{AB,C}}{n_{AB} n_C} = \frac{m_{A,C} + m_{B,C}}{n_A n_C + n_B n_C}$$
$$\leq \max\{\mathsf{pxd}(\{A, C\}), \mathsf{pxd}(\{B, C\})\} \leq \beta \qquad \square$$

C Additional Proofs and Results

C.1 Proof of Lemma 5 (Sufficient Condition for Non-Locality)

Proof (of Lemma 5). Assume f is local, let $\leq_f$ be a relation on $\mathcal{M}\times\mathcal{M}$ satisfying the definition of locality. From $f(\mathcal{C}_{M_1}) < f(\mathcal{C}_{M_2})$ it follows that $M_2 \leq_f M_1$ does not hold and $f(\mathcal{D}_{M_1}) > f(\mathcal{D}_{M_2})$ implies that $M_1 \leq_f M_2$ does not hold. This means that M_1 and M_2 are not comparable, contradicting the choice of $\leq_f$. $\square$

Table 3: Summary of the properties of all intercluster measures

measure	abbr.	merge beh.	enf. conn. merges	local
number of intercluster edges	nxe	unbounded	Y	Y
global intercluster density	gxd	unbounded	Y	N
max. is. intercluster density	mixd	unbounded	N	Y
avg. is. intercluster density	aixd	bounded	N	Y
max. pw. intercluster density	mpxd	unbounded	N	N
avg. pw. intercluster density	apxd	bounded	N	N
max. is. intercluster conductance	mixc	unbounded	N	Y
avg. is. intercluster conductance	aixc	unbounded	N	Y
max. pw. intercluster conductance	mpxc	bounded	N	N
avg. pw. intercluster conductance	apxc	bounded	N	N
max. is. intercluster expansion	mixe	unbounded	N	Y
avg. is. intercluster expansion	aixe	unbounded	N	Y
max. pw. intercluster expansion	mpxe	bounded	N	N
avg. pw. intercluster expansion	apxe	bounded	N	N

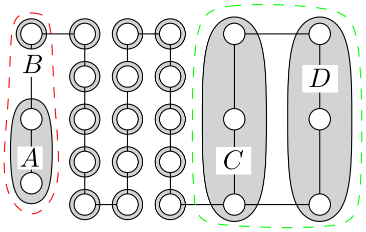


Fig. 8: Suppose we use objective function nxe, constraint aid and an appropriate value of α. Starting with singletons, let us start with a merge that yields cluster A. Merge $\{A, B\}$ will decrease aid, any further decrease in the number of clusters will worsen this effect. Suppose we reach clustering $\mathcal{C}$ (gray), we even have $\mathsf{aid}(\mathcal{C}_{\{A,B\}}) < \mathsf{aid}(\mathcal{C}_{\{C,D\},\{A,B\}})$, as two "bad" clusters are summarized into a single one, which mildly lifts aid. Thus, if we set α to the average of these two values, merge $\{A, B\}$ is infeasible in $\mathcal{C}$ but as soon as merge $\{C, D\}$ is performed, regains feasibility. Obviously, it again loses feasibility as soon as we continue merging other singletons.

C.2 Proof of Lemma 6 (Locality of Maximum Isolated Measures)

Proof (of Lemma 6). We prove the claim constructively by defining a relation $\leq_f$ and showing that it fulfills the required properties for the definition of locality.

Let $\mathcal{C}$ be a clustering and $\mathcal{D} = \{D_1, \ldots, D_k\} \subseteq \mathcal{C}$. Then we define $L(\mathcal{D}) := (f'(D_{i_1}) \geq \ldots \geq f'(D_{i_k}))$ to be the non-increasing sequence of function values of all $D_i \in \mathcal{D}$. Let $\leq_\ell$ be the lexicographical order on sequences of real numbers. The first observation we make is that for arbitrary merges M_1 and $M_2 \subseteq \mathcal{C}$, $L(\mathcal{C}_{M_1}) \leq_\ell L(\mathcal{C}_{M_2})$ implies $f(\mathcal{C}_{M_1}) \leq f(\mathcal{C}_{M_2})$. Since the lexicographical order on sequences of reals is a partial order, $\leq_\ell$ is transitive and reflexive. If we can find an equivalent relation $\leq_f$ on $M \times M$, i.e., a relation such that $M_1 \leq_f M_2$ iff $L(\mathcal{C}_{M_1}) \leq_\ell L(\mathcal{C}_{M_2})$, independent of $\mathcal{C} \setminus (M_1 \cup M_2)$, this yields the locality of f.

The crucial idea is to just consider that part of $L(\mathcal{C})$ which changes if either of the two merges $M_1 = \{A, B\}$ and $M_2 = \{C, D\}$ is performed. The affected entries of $L(\mathcal{C})$ are $\{f'(A), f'(B), f'(C), f'(D)\}$. Performing M_1 replaces these by $\{f'(A \cup B), f'(C), f'(D)\}$, and performing M_2 replaces them by $\{f'(A), f'(B), f'(C \cup D)\}$. Since $L(\mathcal{C} \setminus \{f'(A), f'(B), f'(C), f'(D)\})$ remains unchanged in either case, it suffices to compare the two replacements $L(\{f'(A \cup B), f'(C), f'(D)\})$ and $L(\{f'(A), f'(B), f'(C \cup D)\})$ instead of $L(\mathcal{C}_{M_1})$ and $L(\mathcal{C}_{M_2})$. Thus, if we define $\leq_f$ such that $\{A, B\} \leq_f \{C, D\}$ iff $L(\{f'(A \cup B), f'(C), f'(D)\}) \leq_\ell L(\{f'(C \cup D), f'(A), f'(B)\})$, $\leq_f$ is a relation that satisfies the definition of locality. □

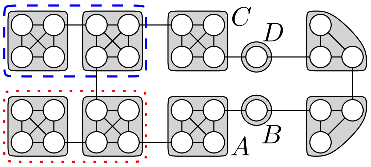


Fig. 9: Counterexample proving the nonlocality of apxe and apxc: The base clustering consists of the gray clusters. If the blue, dashed merge is performed, merge $\{C, D\}$ is better, if the red, dotted merge is performed, $\{A, B\}$ is better. See Fig. 5 for the remaining counterexamples on locality.

Fig. 10: Given the gray clusterings, the disconnected merge $\{A, B\}$ (red, dotted) yields the highest improvement in mixd. For examples on other measures, see Fig. 6.