Practical Nearly-Linear-Time Approximation Algorithms for Hybrid and Overlapping Graph Clustering

Konstantinos Ameranis [1] **Lorenzo Orecchia** [1] **Kunal Talwar** [2] **Charalampos Tsourakakis** [3]

Abstract

In many graph-clustering applications, overwhelming empirical evidence suggests that communities and clusters are naturally overlapping, calling for novel *overlapping graph-partitioning algorithms* (OGP). In this work, we introduce a framework based on two novel clustering objectives, which naturally extend the well-studied notion of conductance to overlapping clusters and to clusters with hybrid vertex- and edge-boundary structure. Our main algorithmic contributions are nearly-linear-time algorithms $O(\log n)$-approximation algorithms for both these objectives. To this end, we show that the cut-matching framework of Khandekar et al. (2014) can be extended to overlapping partitions and give novel cut-improvement primitives that perform a small number of s-t maximum flow computations over the instance graph to detect sparse overlapping partitions near an input partition. Crucially, we implement our approximation algorithm to produce both overlapping and hybrid partitions for large graphs, easily scaling to tens of millions of edges, and test our implementation on real-world datasets against other competitive baselines.

1. Introduction

Detecting communities in real-world networks and clustering similarity graphs are major data mining tasks with a wide range of applications in graph mining, collaborative filtering, and bioinformatics. *Ratio-cut objectives* (also known as quotient-cut objectives) constitute a well-studied and commonly used family of graph partitioning problems (Hagen & Kahng, 1992; Abrahao et al., 2012). A ratio-cut objective measures the quality of a graph cut by the ratio of the weight of the edge cutset to the volume of the smaller side of the partition. Specifically, given a graph $G = (V, E, \mu \in \mathbb{R}^{|V|}_{\geq 0} w \in \mathbb{R}^{|E|}_{\geq 0})$ with non-negative edge weights w and a measure μ over vertices, the ratio-cut objective Ψ_G over partitions $(S, \bar{S})$ of V is defined as:

$$\Psi_G(S, \bar{S}) = \frac{w\big(E(S, \bar{S})\big)}{\min\{\mu(S), \mu(\bar{S})\}}. \tag{1}$$

The ratio-cut minimization problem asks us to minimize this objective over all partitions $(S, \bar{S})$, i.e., determine $\Psi(G) = \min_S \Psi_G(S, \bar{S})$. Ratio-cut objectives play a major role in graph clustering, as they include the widely used expansion ($\forall i \in V, \mu_i = 1$) and conductance ($\forall i \in V, \mu_i = \sum_{j \sim i} w_{ij}$), which is often taken to be the "gestalt" notion of graph clustering (Leskovec et al., 2009; Zahn, 1971).

In many real-world applications, it is desirable to allow entities to belong to more than one cluster. For instance, in biology a protein may belong to multiple protein complexes (Nepusz et al., 2012), in social networks an agent may be part of multiple communities (Ahn et al., 2010), and a political blog may reflect more than one party affiliation (Latouche et al., 2011). The seminal work of Leskovec et al. (2008) has shown, in numerous large-scale information and social networks, the existence of a core that spans most vertices and lacks community structure (aka "expander-like"), and the existence of numerous small communities with up to few hundreds of nodes that overlap with the core. As a result, standard graph clustering approaches (Abrahao et al., 2012), based on non-overlapping objectives such as ratio cuts, fail to recover the community structure in these ubiquitous datasets. We focus on the following fundamental open question:

Question 1. *Can we design a framework for* overlapping graph partitioning *(OGP) that allows for (i) a principled and intuitive mathematical formulation, together with (ii) solid worst-case approximation algorithms that (iii) scale gracefully to large networks?*

[1]Department of Computer Science, University of Chicago, Chicago, USA [2]Apple Inc [3]Department of Computer Science, Boston University, Boston, USA. Correspondence to: Konstantinos Ameranis <kameranis@uchicago.edu>, Lorenzo Orecchia <orecchia@uchicago.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Despite a recent a flurry of works on OGP (Ahn et al., 2010; Andersen et al., 2012; Arora et al., 2012; Bonchi et al., 2013; Khandekar et al., 2014; Mishra et al., 2007; Airoldi et al., 2008; Yang & Leskovec, 2013; Gopalan & Blei, 2013; Li et al., 2017; Palla et al., 2012; Tsourakakis, 2015; Whang et al., 2016), all prior works forgo at least one of these desired properties. By contrast, these properties are already satisfied by well-developed theory and software implementations for the non-overlapping ratio-cut objectives (Leighton & Rao, 1999; Arora et al., 2009; Leskovec et al., 2009; Shi & Malik, 2000; Orecchia et al., 2008), of which conductance is a special case. In this work, we provide problem formulations, algorithms and implementations that satisfy all parts of question 1.

Novel overlapping objectives We formulate natural generalizations of ratio-cut objectives for partitioning the graph into *two* overlapping partitions. As with other ratio-cut objectives, the balanced and k-way versions of overlapping problems can be reduced to the 2-way problem in standard ways (Kannan et al., 2004). The key idea behind our generalizations is to redefine the notion of the *boundary* of a cluster to contain *both* edges that leave the cluster and vertices that are shared with other clusters.

Definition 1. *An* overlapping partition[1] $[S,T]$ *of the vertex set* V *consists of two subsets* $S,T \subseteq V$ *such that* $S \cup T = V$. *The corresponding* edge-cutset $\delta_E[S,T]$ *and* vertex-cutset $\delta_V[S,T]$ *are:*

$$\delta_E[S,T] \stackrel{\text{def}}{=} E(S \setminus T, T \setminus S) \quad \textit{and} \quad \delta_V[S,T] \stackrel{\text{def}}{=} S \cap T,$$

As long as $S,T \neq S \cap T$, the edge-cutset $\delta_E[S,T]$ and the vertex-cutset $\delta_V[S,T]$ can be thought of as a generalized notion of boundary in V, as their removal disconnects the graph into at least two components associated to $S \setminus T$ and $T \setminus S$. We can now associate two ratio-cut-like measures to an overlapping partition:

$$q_E[S,T] \stackrel{\text{def}}{=} \frac{w(\delta_E[S,T])}{\min\{\mu(S),\mu(T)\}}, \tag{2}$$

$$q_V[S,T] \stackrel{\text{def}}{=} \frac{\mu(\delta_V[S,T])}{\min\{\mu(S),\mu(T)\}}. \tag{3}$$

In Section 3, for a parameter $\epsilon \in [0,1)$, we define the ϵ-*overlapping ratio-cut* (ϵ-ORC) problem to be the minimization of the edge ratio-cut $q_E[S,T]$ under the condition that the vertex ratio-cut $q_V[S,T] \leq \epsilon$, i.e., the overlap between S and T contains at most an ϵ-fraction of the measure of the smaller side of $[S,T]$. We also define a version of the problem with softer overlap constraints: for a parameter $\lambda \geq 0$, the λ-*hybrid ratio-cut* problem (λ-HCUT) is the minimization of $q_E[S,T] + \lambda \cdot q_V[S,T]$, i.e., the cost of cutting one unit of vertex boundary is λ times that of a unit of edge boundary.

Algorithm design Both λ-HCUT and ϵ-ORC are easily seen to be NP-hard. Existing metric relaxations and rounding algorithms (Leighton & Rao, 1999; Arora et al., 2004) for graph partitioning problems can be applied to obtain polylogarithmic approximations. However, solving such relaxations requires the computation of dense multicommodity flows on an edge- and vertex-capacitated version of the input graph, which needs quadratic time in the size of the input (Arora et al., 2010). To scale our computations to networks with tens of millions of edges on a single-machine, we rely on the *cut-matching* game of Khandekar, Rao and Vazirani (Khandekar et al., 2009), which computes approximate solutions to the formulation of Arora et al. (2004) by assuming oracle access to a *cut-improvement algorithm* (Andersen & Lang, 2008) for the desired ratio-cut problem. We provide two main **technical contributions**:

- the *first cut-improvement algorithm for* OGP *problems*, generalizing the graph version of Andersen & Lang (2008), while only requiring a polylogarithmic number of s-t maximum flow computations over a vertex-capacitated version of the input graph.
- the *first extension of the cut-matching game framework* to OGP problems, showing that the expander flow of paradigm of Arora et al. (2009) seamlessly ports over to the OGP setting.

Combining these contributions with recent advances in the fast solution of maximum flow problems (Chen et al., 2022), we obtain the first almost-linear-time $O(\log n)$ approximation to λ-HCUT and ϵ-ORC.

Empirical Evaluation We evaluate the performance of our proposed method `cm+improve` on graphs sampled from the Overlapping Stochastic Block Model (Abbe & Sandon, 2015) and on large real-world networks from the SNAP collection (Leskovec & Krevl, 2014). Our results show that `cm+improve` is competitive or outperforms baselines while scaling to graphs with over 10^7 edges.

2. Related work

Overlapping graph clustering. Overlapping community detection has been studied from a statistical viewpoint through the overlapping stochastic block model (Latouche et al., 2011; Abbe & Sandon, 2015). The problem remains largely open for general graphs that do not conform to such simple probabilistic models. Due to the importance of overlapping graph clustering, a variety of rigorous methods have been proposed based on different models of overlapping partitions. Arora et al. (2012) present an average-case analysis approach based on certain random graph models. Balcan

[1] We denote an overlapping partition by $[S,T]$ to clearly differentiate it from non-overlapping partitions $(S,\bar{S})$.

et al. (2012) consider a set-based latent structure, and extend the notion of (α, β)-communities originally proposed by Mishra et al. (2007). Other machine-learning methods also assume that nodes have latent features according to which they decide how to connect. Such methods can be seen as matrix factorization methods, and include notably mixed membership models (Airoldi et al., 2008), BIGCLAM (Yang & Leskovec, 2013), and several others (Andersen et al., 2012; Bonchi et al., 2013; Gopalan & Blei, 2013; Li et al., 2017; Palla et al., 2012; Tsourakakis, 2015; Whang et al., 2016).

Much less is known about algorithms with worst-case guarantees for overlapping clustering. Khandekar et al. (2014) consider the problem of minimizing the sum and the maximum of conductances for a set of communities under the constraint that each node belongs to at least one community. Their approach is based on the tree decomposition of Räcke (Räcke, 2008), with the authors themselves pointing out that their methods do not scale to large graphs.

A long line of work has focused on developing polynomial-time approximation algorithms for ratio-cut minimization, including spectral algorithm based on Cheeger's inequality (Alon & Milman, 1985), the multicommodity-flow-based Leighton-Rao $O(\log n)$ approximation algorithm (Leighton & Rao, 1999), and finally the current state-of-the-art approximation due to Arora, Rao and Vazirani (Arora et al., 2009), which combines spectral and flow techniques. In parallel, practitioners have developed many scalable graph-partitioning heuristics, including the Kernighan-Lin heuristic (Kernighan & Lin, 1970b) that is frequently used as a sub-routine for refining partitions (e.g., (Hendrickson & Leland, 1995)), the widely used METIS software (Karypis & Kumar, 1996; 1998; 1995), Graclus (Dhillon et al., 2007), and KaHIP that imposes balance constraints on the clusters (Sanders & Schulz, 2013).

3. Novel Overlapping Clustering Objectives

We model the input to our OGP formulation as consisting of a weighted undirected graph $G = (V, E, w, \mu)$ with arbitrary non-negative edge weights[2] $\{w_e \in \mathbb{Z}_{\geq 0}\}_{e \in E}$ and arbitrary non-negative vertex weights[2] $\{\mu_v \in \mathbb{Z}_{\geq 0}\}_{v \in V}$. Our main OGP problem is the *the ϵ-overlapping ratio-cut* (ϵ-ORC), which takes a parameter $\epsilon \in [0, 1]$ controlling the maximum size of the overlap $\delta_V[S, T]$:

$$\epsilon - \text{ORC}: \min_{S \cup T = V} q_E[S, T] = \min_{S \cup T = V} \frac{w(\delta_E[S, T])}{\min\{\mu(S), \mu(T)\}}$$
$$q_V[S, T] = \frac{\mu(\delta_V[S, T])}{\min\{\mu(S), \mu(T)\}} \leq \epsilon$$

In words, we are attempting to minimize the ratio between weight of the edges between $S \setminus T$ and $T \setminus S$, and the weight of the smaller of S and T, while constraining the weight of the overlap $S \cap T$ to be at most an ϵ-fraction of both S and T. The logic behind the choice of the ϵ-ORC objective is the realization that overlapping partitions fail to be detected by existing algorithms because they do not correspond to either sparse edge cuts or sparse vertex cuts.

Consider the emblematic Zachary's Karate Club social network (Zachary, 1977) in Figure 1 in which two karate clubs S and T overlap on a subset $S \cap T$. This intersection contains a small number of nodes that are well-connected to both communities. At the same time, there also exists a small number of edges directly between $S \setminus T$ and $T \setminus S$, possibly because of second-order interactions between the nodes. In this setting, neither an edge-based graph partitioning algorithm nor a vertex-based one succeeds in detecting the overlapping partition S and T. The former suffers a large penalty if it separates either S or T from $S \cap T$, as a large number of edges is cut. The latter cannot identify $S \cap T$ because it is not a vertex separator, i.e., S and T are not disconnected by the removal of $S \cap T$. Our objective ϵ-ORC enables us to interpolate between the edge- and vertex-based cuts to optimize over *hybrid* cuts, as shown in Subfigure (b).

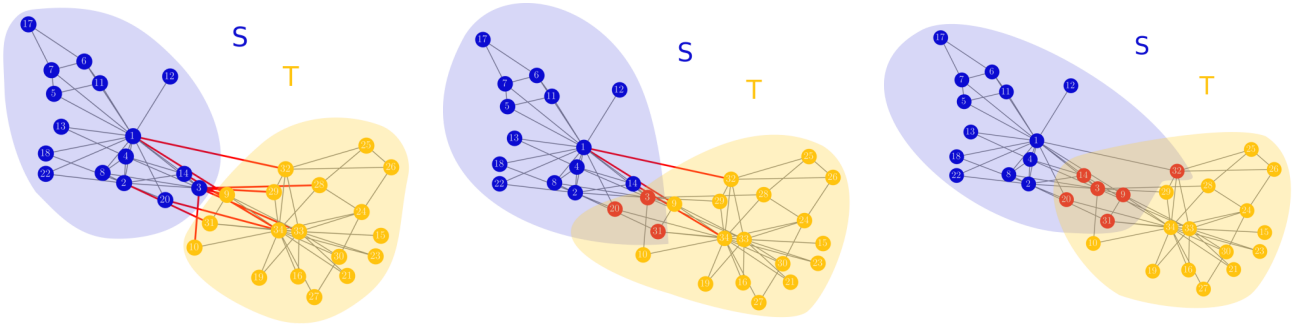
3.1. Hybrid Graph Partitioning

The λ-HCUT problem provides an unconstrained, computationally easier, version of the the ϵ-ORC problem, where the overlap fraction $q_V[S, T]$ is controlled via a penalty term $\lambda \cdot q_V[S, T]$ directly in the objective. For a parameter $\lambda \geq 0$, we define the λ-HCUT objective $q_{G,\lambda}$ as:

$$q_{G,\lambda}[S, T] \stackrel{\text{def}}{=} q_E[S, T] + \lambda \cdot q_V[S, T]$$
$$= \frac{w(\delta_E[S, T]) + \lambda \cdot \mu(\delta_V[S, T])}{\min\{\mu(S), \mu(T)\}}$$

The λ − HCUT problem consists of the minimization of $q_{G,\lambda}[S, T]$ over all overlapping partition $[S, T]$ of V, i.e., determining $q(G, \lambda) = \min_{[S,T]} q_{G,\lambda}[S, T]$. To minimize the objective $q_{G,\lambda}$, we are looking to separate the graph into two disconnected components $S \setminus T$ and $T \setminus S$ by removing a small set of edges $\delta_E[S, T]$ and a small set of vertices $\delta_V[S, T]$. The parameter λ regulates the relative cost of cutting one unit of edge-weight compared to one unit of vertex-weight. By varying λ, the λ-HCUT problem interpolates between edge-based ($\lambda \geq \max_{i \in V}(\sum_{i \sim j} w_{ij}/\mu_i)$)

[2] We assume integral weights for the rest of the paper. Problems with rational weights can be reduced to the integral case by an appropriate scaling. Our complexity guarantees will depend (logarithmically) on the magnitude of the largest weight.



(a) $\epsilon = 0$. Edge-based partition. (b) $\epsilon = .11$. Overlapping partition. (c) $\epsilon = .22$. Vertex-based partition.

Figure 1: Visualizations of overlapping solutions to ϵ-ORC for different values of $\epsilon \in [0, 1]$ on the Karate graph (Zachary, 1977), with μ set to the degree measure. Partition sides are yellow and blue. Overlap is red. Cut edges are also red. Solutions were computed using our algorithm `cm+improve`.

and vertex-based ($\lambda \leq \min_{i \in V}(\Sigma_{i \sim j} w_{ij}/\mu_i)$) partitions via hybrid partitions cutting both edges and vertices.

Interpreting the parameter λ as a Lagrangian multiplier yields a simple relation between λ-HCUT and ϵ-ORC: optimal solutions to λ-HCUT yield optimal solutions to the ϵ-ORC problem.

Lemma 1. *For any $\lambda \geq 0$, let $[S, T]$ be an α-approximate optimal solution for the λ-HCUT problem. Define $\epsilon = \mu(\delta_V[S,T])/\min\{\mu(S),\mu(T)\}$. Then, $[S, T]$ is an optimal solution to the ϵ-ORC problem.*

This lemma can be easily generalized to α-approximate optimal solution, as long as we allow for a bi-criterion approximation for the ϵ-ORC problem, where the output overlapping partition is only required to have $\delta_V[S, T] \leq \alpha\epsilon$. While it may be tempting to use this approach to reduce the optimization of ϵ-ORC to that of λ-HCUT, by performing a search over the Lagrangian multiplier λ, this is not possible in general, as approximately optimal solutions for ϵ-ORC for some values of ϵ may not be approximately optimal for any λ. Fortunately, our algorithmic approach, described in the next section, still allows us to solve both problems, essentially by carrying out an analogue of the the proposed reduction for localized, convex versions of the two problems. For this reason, we first describe an algorithm for the λ-HCUT problem in the next section.

4. Efficient Approximation Algorithms

The λ-HCUT problem is NP-hard, as it generalizes edge-based and vertex-based graph partitioning problems, capturing the minimum-conductance problem as a special case. Spectral methods yield provable non-trivial approximation guarantees only for the minimum-conductance problem, as Cheeger's Inequality does not extend to generic vertex measures other than the degree measure. For this reason, we consider approximation algorithms based on metric relaxations of graph partitioning problems (Leighton & Rao, 1999; Arora et al., 2004). Such relaxations yield polynomial-time poly-logarithmic approximations for both edge- and vertex-based ratio-cut problems (Feige et al., 2005), including Ψ_G. Indeed, these methods can be adapted to yield the same approximation for the λ-HCUT problem. However, the convex programs arising from these relaxations have a cubic number of constraints and generally require the solution of dense multi-commodity flow problems (Arora et al., 2010) over G, drastically limiting the scalability of this approach.

Due to the practical importance of graph partitioning, a number of works have focused on designing scalable algorithms that match the poly-logarithmic approximation ratios afforded by the metric relaxations while only using s-t maximum flow computations, rather than the more time-consuming multi-commodity flows. A particularly simple framework for this reduction is the *cut-matching* game of khandekar2009graph,thesis, which computes approximate solutions to the (Arora et al., 2004) formulation by assuming oracle access to a *cut-improvement algorithm*, which is implemented via the solution of a small number of s-t maximum-flow computations over the instance graph. We follow this approach and design a novel cut-improvement algorithm, `HybridImprove`, for the λ-HCUT problem, and a closely related cut-improvement procedure , `OverlapImprove`, for the ϵ-ORC problem.

For an instance of the λ-HCUT problem G, let the parameter W_G denote the maximum weight $\max\{\max_{e \in E}\{w_e\}, \max_{i \in V}\{\lambda\mu_i\}\}$. Let $T_f(m, n, C)$ be the time complexity of solving a s-t maximum flow problem over an edge- and vertex-capacitated graph with m arcs, n vertices and integral arc capacities bounded by C. The following is our main result, yielding a logarithmic approximation to the λ-HCUT problem.

Theorem 2. *For an input graph $= (V, E, w, \mu)$, the cut-matching framework applied to* `HybridImprove`

yields a $O(\log|V|)$-approximation algorithm for the λ-HCUT problem. The running time is $[O(|E|) + T_f(O(|E|), O(|V|), O(W_G))] \cdot \text{poly}(\log|V| \cdot \log W_G)$.

By the same method, we also obtain a similar result for the ϵ-ORC problem, with a bi-criterion approximation, which is standard for constrained graph-partitioning problems, e.g., balanced graph partitioning (Leighton & Rao, 1999).

Theorem 3. *For an input graph $= (V, E, w, \mu)$, let $R = \max_{i,j} \mu_i/\mu_j$. The cut-matching framework applied to* `OverlapImprove` *outputs an overlapping partition $[S, T]$ such that $q_V[S, T] \le (R + 1) \cdot \epsilon$ and for all overlapping partitions $[A, B]$ with $q_V[A, B] \le \epsilon$*

$$q_E[S, T] \le O(\log|V|) \cdot q_E[A, B].$$

The running time is $[O(|E|) + T_f\big(O(|E|), O(|V|), O(W_G)\big)] \cdot \text{poly}(\log|V| \cdot \log W_G)$.

If we choose the algorithm of Goldberg & Rao (1998) as our s-t maxflow solver, the total running time for both algorithms becomes $O(|V||E|^{1/2} \cdot \text{poly}(\log|V| \cdot \log W_G)$. To obtain the advertised almost-linear running times, there are two approaches based on approximate maximum-flow computations:

- Following Khandekar et al. (2009) , we can approximately compute the maximum flow by running blocking flows of length up to $O\left(|V|/q(G,\lambda)\right)$ to obtain a running time of $O\left(|E|/q(G,\lambda) \cdot \text{poly}(\log|V| \cdot \log W_G)\right)$.
- By the recent work of Chen et al. (2022), the edge- and vertex-capacitated flow can be computed in $O(|E|^{1+o(1)})$ time.

In our implementation, we use the HIPR implementation (Cherkassky et al., 1994) of the push-relabel method, which has proved to be very efficient in practice.

4.1. Cut Improvement for λ-HCUT

A cut-improvement algorithm (Kernighan & Lin, 1970a; Fiduccia & Mattheyses, 1982; Andersen & Lang, 2008) for a ratio-cut problem takes as input a candidate partition $(S_0, \bar{S}_0)$ and outputs a nearby partition $(S, \bar{S})$ with an improved objective $\Psi_G(S, \bar{S}) \le \Psi_G(S_0, \bar{S}_0)$. A practical approach to solving HGP is to use a generic partitioning algorithm to find a non-overlapping cut (S, T), where $T = \bar{S}$, together with a *cut improvement* procedure that includes vertices in the overlap $S \cap T$ as to minimize $q_{G,\lambda}$. A natural cut-improvement heuristic, which we refer to as `GreedyImprove`, is to repeatedly loop over all vertices u on the boundary of S and T and greedily include $u \in S \cap T$ if the inclusion decreases the value of the λ-HCUT objective. This heuristic will serve as a competitor to our algorithm in the empirical evaluation of Section 5. Unfortunately, `GreedyImprove` does not yield any global approximation guarantees.

Our first significant algorithmic contribution is the design and analysis of a novel cut-improvement method for the λ-HCUT problem. Our algorithm `HybridImprove` generalizes previous flow-based improvement algorithms (Andersen & Lang, 2008; Lang & Rao, 2004) to the hybrid cut setting. However, our approximation guarantees for `HybridImprove` are much sharper, as previous results cannot be directly deployed in the cut-matching game. Our guarantees are more easily stated if we first extend the definition of the non-overlapping ratio-cut objective Ψ_G to overlapping partitions $[A, B]$ in the following natural way:

$$\Psi_G([A, B]) = \max_{S \subseteq A, \bar{S} \subseteq B} \frac{w\big(E(S, \bar{S})\big)}{\min\{\mu(A), \mu(B)\}} \tag{4}$$

Here the numerator in the ratio-cut $\Psi_G([A, B])$ for an overlapping partition $[A, B]$ is the worst (maximum) edge-cutset weight over all ways of splitting the overlap $A \cap B$ between $S \subseteq A$ and $\bar{S} \subseteq B$.

Given an input non-overlapping partition $(S_0, \bar{S}_0)$, `HybridImprove` outputs an improved overlapping partition $[S, T]$, together with a certificate that lower-bounds the ratio-cut of partitions near $[S, T]$. This dual certificate takes the form of a bipartite μ-regular graph H between $(S_0, \bar{S}_0)$. Indeed, the dual problem solved by `HybridImprove` is exactly that of routing the largest possible multiple of a bipartite μ-regular graph across the input partition $(S_0, \bar{S}_0)$ into an edge- and vertex-capacitated version of G. The execution of `HybridImprove` only requires a small number of s-t maxflow computations over a directed version on G.

Theorem 4. *Let $G = (V, E, w \in \mathbb{R}^E_{\ge 0}, \mu \in \mathbb{R}^V_{\ge 0})$ be an undirected weighted graph and $(S_0, \bar{S}_0)$ be a non-overlapping partition of G. Assume without loss of generality that $\mu(S_0) \le \mu(\bar{S}_0)$ and define $\kappa \stackrel{\text{def}}{=} \mu(S_0)/\mu(\bar{S}_0) \in (0, 1]$. On input $(G, (S_0, \bar{S}_0), \lambda \ge 0)$, the* `HybridImprove` *algorithm outputs:*

- *a weighted graph $H = (V, E_H, u \in \mathbb{R}^{E_H}_{\ge 0}, \mu)$, with the same vertex weights as G, such that*
  - *H is bipartite across $(S_0, \bar{S}_0)$,*
  - *the weighted degree of a vertex i in H is μ_i if $i \in S_0$ and $\kappa \cdot \mu_i$ if $i \in \bar{S}_0$.*
- *an overlapping partition $[S, T]$ such that for all overlapping partitions $[A, B]$,*

$$\frac{q_{G,\lambda}([A, B])}{q_{G,\lambda}([S, T])} \ge \Psi_H([A, B]) \tag{5}$$

The running time of `HybridImprove` *is* $(T_f(O(|E|), O(|V|), O(W_G)) + |E|) \cdot O(\log(W_G \cdot |V|)$.

A crucial property of HybridImprove is the approximation result of Equation 5 for the output overlapping partition $[S, T]$. In particular, it guarantees that *for all* overlapping partitions $[A, B]$, including ones potentially far from $(S_0, \bar{S}_0)$, the objective of the output partition $[S, T]$ is within a factor $\Psi_H([A, B])$ of $q_{G,\lambda}([A, B])$, i.e., the more edges of H cross $[A, B]$, the better approximation to $q_{G,\lambda}([A, B])$ we have. Applying this reasoning to the optimal cut for the λ-HCUT objective yields the following corollary:

Corollary 5. *Let $[A^\star, B^\star]$ be the overlapping partition minimizing $q_{G,\lambda}$. On input $(S_0, \bar{S}_0)$,* HybridImprove *outputs an overlapping partition $[S, T]$ which is a $1/\Psi_H([A^\star,B^\star])$ approximation to the optimum $q_{G,\lambda}([A^\star, B^\star])$.*

Hence, we can obtain a good approximation ratio for the λ-HCUT problem, if we use HybridImprove to query a cut $(S_0, \bar{S}_0)$ that forces H to have many edges cross the optimal overlapping partition $[A^\star, B^\star]$. The details of the HybridImprove algorithm and a full proof of Theorem 4 can be found in the Appendix.

4.2. Cut Improvement for ϵ-ORC

Recall that it is not generally possible to reduce the ϵ-ORC problem to a sequence of calls to an oracle for the HCUT problem with different λ values. Fortunately, the same reduction strategy works instead when applied at the cut improvement level: we can obtain a cut improvement algorithm OverlapImprove for the ϵ-ORC algorithm simply via performing binary search on λ in the HybridImprove algorithm. The full details of OverlapImprove and the proof of the following analogue of Theorem 4 are described in the Appendix. An exact analogue of Corollary 5 also holds.

Theorem 6. *Under the same assumptions of Theorem 4, let R be the largest ratio between vertex weights, i.e., $R = \max_{i,j} \mu_i/\mu_j$. The algorithm* OverlapImprove *on input $(G, (S_0, \bar{S}_0), \epsilon \in (0, 1)$ outputs:*

- *a weighted graph $H = (V, E_H, u \in \mathbb{R}_{\geq 0}^{E_H}, \mu)$, with the same properties as in Theorem 4,*
- *an overlapping partition $[S, T]$, $q_V[S, T] \leq (R + 1)\epsilon$, such that for all overlapping partitions $[A, B]$ with $q_V[A, B] \leq \epsilon$:*

$$\frac{q_E([A, B])}{q_E([S, T])} \geq \Psi_H([A, B]). \tag{6}$$

4.3. Reduction to the Cut-Matching Game

The cut-matching game is an interactive game between a cut player $\mathcal{C}$ and a matching player $\mathcal{M}$ over a vertex set V with vertex measure μ. Starting with an empty graph over V, at each iteration t, $\mathcal{C}$ plays a partition $(S_t, \bar{S}_t)$ of V with $\mu(S_t) \leq \mu(\bar{S}_t)$. The matching player $\mathcal{M}$ responds by placing a bipartite μ-regular graph H_t across $(S_t, \bar{S}_t)$, i.e., a bipartite graph such that every vertex $v \in S_t$ has degree μ_v and every vertex $u \in \bar{S}_t$ has degree $\mu(S_t)/\mu(\bar{S}_t) \cdot \mu_u$. Let $\bar{H}_T = \frac{1}{T} \cdot \sum_{t=1}^{T} H_t$ be the average of the graphs added by $\mathcal{M}$ up to time T. As T goes to infinity, the goal of the cut player is to maximize the minimum ratio-cut quotient $\Psi(\bar{H}_T, \mu)$, while the matching player aims to minimize the same quantity. In words, the cut player aims to select sparse cuts to force the matching player to make $\bar{H}_t$ more expander-like. Conversely, the matching player will try to add edges to $\bar{H}_t$ while preserving some sparse cuts. The following theorem (Orecchia et al., 2008; Orecchia, 2011) gives an efficient strategy for the cut player, which is based on Matrix Multiplicative Weight Updates (Tsuda et al., 2005).

Theorem 7. *(Orecchia et al., 2008; Orecchia, 2011) There exists a strategy for the cut player $\mathcal{C}$ such that, for any play of the matching player $\mathcal{M}$, we have $\Psi(\bar{H}_T, \mu) \geq \Omega(1/\log |V|)$ for $T = O(\log^2 |V|)$. At time t, the cut $(S_t, \bar{S}_t)$ played by this strategy can be computed as a function of $\bar{H}_t$ in time $O(|E(\bar{H}_t)| \cdot \text{polylog}(\mu(V)))$.*

With this strategy in hand, we are ready to sketch the proof of our main results on the approximation of λ-HCUT (Theorem 2) and ϵ-ORC (Theorem 3). Pseudocode for resulting generic algorithm is given as Algorithm 1. A learning interpretation of these results is that the cut player strategy is using Matrix Multiplicative Weight Updates to boost the local approximation guarantees of HybridImprove and OverlapImprove to a global guarantee by carefully choosing which cut-improvement problems are solved. The complete proof can be found in the Appendix.

Proof Sketch for Theorem 2 and Theorem 3. The cut-improvement algorithm (HybridImprove or OverlapImprove) takes the role of the matching player in the cut-matching game, i.e., at every iteration t, the matching player's response to $(S_t, \bar{S}_t)$ is the μ-regular bipartite certificate H_t output by HybridImprove on input $(S_t, \bar{S}_t)$. By choosing input cuts $(S_t, \bar{S}_t)$ using the strategy of Theorem 7, we can guarantee that after $T = O(\log^2 |V|)$, we have that for any overlapping partition $[A, B]$

$$\max_{1 \leq t \leq T} \Psi_{H_t}([A, B]) \geq \frac{1}{T} \sum_{t=1}^{T} \Psi_{H_t}([A, B]) \geq \Psi_{\bar{H}_t}([A, B]) = \Omega\left(\frac{1}{\log |V|}\right),$$

where the second inequality is a consequence of the definition in Equation 5. Applying this statement to the optimal overlapping partition $[A^\star, B^\star]$, it must be the case for some $t^\star$ that $\Psi_{H_t}([A^\star, B^\star]) \geq \Omega(1/\log |V|)$. Hence, by Corollary 5, the overlapping partition output by the

cut-improvement algorithm at iteration $t^\star$ is a $O(\log |V|)$-approximation to the optimum. □

Algorithm 1 Generic cut-matching game algorithm

Input: graph instance $G = (V, E, w, \mu)$
Output: overlapping cut$[S, T]$
$H_0 \leftarrow G$ {Certificate initialization}
for $t \leftarrow 1, \cdots, T = O(\log^2(n))$ **do**
  $(S_t, \bar{S}_t) \leftarrow \mathcal{C}(H_{t-1})$ {Cut player's move}
  $M_t, [A_t, B_t] \leftarrow$ `Improve`$(G, S_t, \bar{S}_t)$
  $H_t = H_{t-1} + M_t$ {Certificate update}
end for
return best partition in $\{(A_t, B_t)\}_{t \in [T]}$

5. Empirical Evaluation

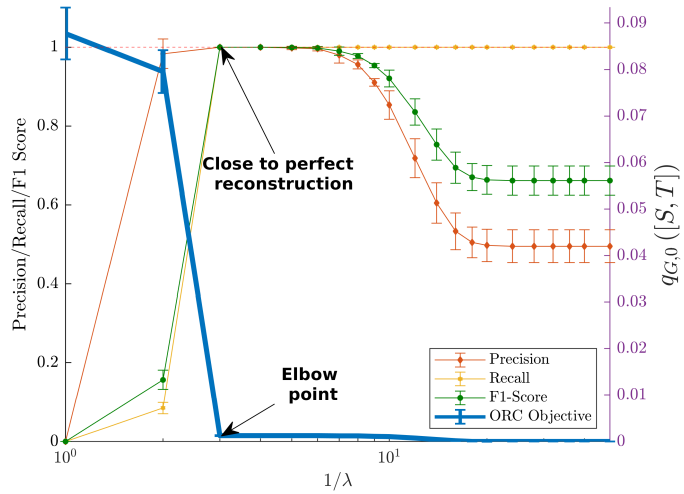
The main challenge in comparing `cm+improve` with existing algorithms is the lack of closely related methods for OGP problems, as statistical methods are highly tuned to the structure and parameters of the model. `BIGCLAM` (Yang & Leskovec, 2014), a popular method for detecting overlapping communities performs very poorly in our testbed, likely because it optimizes a very different notion of objective, more akin to detecting smaller obvious clusters, rather than partitioning the whole graph. This challenge is compounded by the lack of other well-defined objective functions for the task of overlapping clustering, which is actually our motivation in defining ϵ-ORC and λ-HCUT. In order to make a meaningful comparison to other algorithms, we post-process the partitions generated by our competitors via one of two HCUT-based cut improvement procedures: the `GreedyImprove` heuristic described in Section 4.1 or our `HybridImprove` algorithm. Thanks to this post-processing, we can now expand our set of competitors to include popular algorithms for non-overlapping clustering, such as spectral clustering methods (Von Luxburg, 2007) and `METIS`, and use our OGP objectives without arbitrarily skewing the playing field.

Our Implementation of `cm+improve`: The cut-matching strategy of Theorem 7 is implemented in MATLAB, while the cut-improvement algorithm `HybridImprove` is single-threaded C++, using Goldberg's s-t-maxflow solver `HIPR`, which implements the push-relabel algorithm (Goldberg; Cherkassky & Goldberg, 1997). Our implementation performs some numerical approximations to minimize the number of calls to `HIPR` and, as a result, departs slightly from the theoretical description of the `HybridImprove` algorithm. These optimizations are described in the Appendix. Throughout our evaluation, we set μ to be the degree measure of the graph, as other methods are already tuned to minimize conductance. For experiments requiring the output to be a balanced overlapping partition, we implemented a simple heuristic modification of `HybridImprove`, by only routing a fraction of the maximum flow in `HybridImprove`, as suggested by Khandekar et al. (2009). All assets are currently accessible (sup, 2021) and will become publicly available under the BSD license after publication. All experiments were conducted on an institutional cluster on machines with 24 Cores (2x 24 core Intel Xeon Silver 4116 CPU @ 2.10GHz), 48 threads and 128GB RAM.

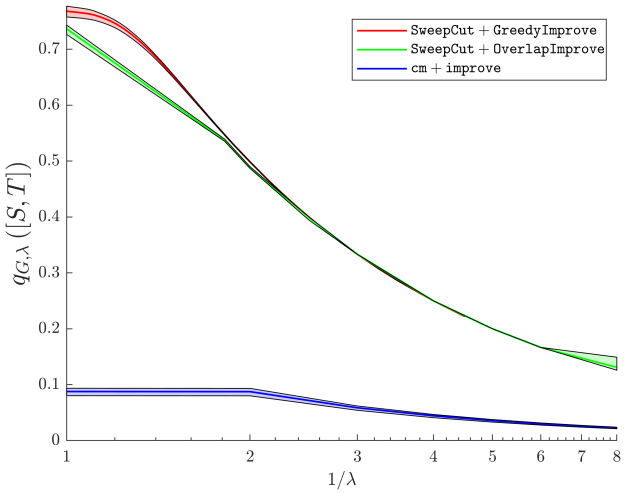
Competitors: The `SweepCut` algorithm is the classic spectral approach to graph partitioning: it performs a sweep of the second eigenvector of the normalized Laplacian (Chung, 1997) and outputs the threshold cut with minimum conductance. This is then fed into our overlapping post-processor. `METIS` (Karypis & Kumar, 1995) is a software suite for solving edge-based graph-partitioning and producing fill reducing orderings for sparse matrices. Its high-quality results, speed and over 25 years of support make it one of the most widely used packages for these tasks. The multi-scale graph-coarsening approach championed by `METIS` yields an extremely fast algorithm, whose accuracy varies with the choice of randomness used in the coarsening step. For a fair comparison, we run `METIS` on many random seeds, for a total time comparable to that of our `cm+improve` on the same instance. The more sophisticated overlap-specific algorithm `BIGCLAM` (Yang & Leskovec, 2013) solves the ERM problem, where each edge between two nodes comes from a shared community and nodes with no shared communities have a very small chance of connecting. In essentially all cases, we found that `BIGCLAM` fails to partition the whole graph, often leaving $> 50\%$ nodes in no community. `BIGCLAM` also tends to output small clusters, even when better, more balanced overlapping partitions clearly exist. This probability reflects a radical difference between `BIGCLAM`'s objective function and standard isoperimetry-based definitions of graph partitioning. We postpone the quantitative results on the comparison of `BIGCLAM` to `cm+improve` to the Appendix.

5.1. Overlapping Stochastic Block Model

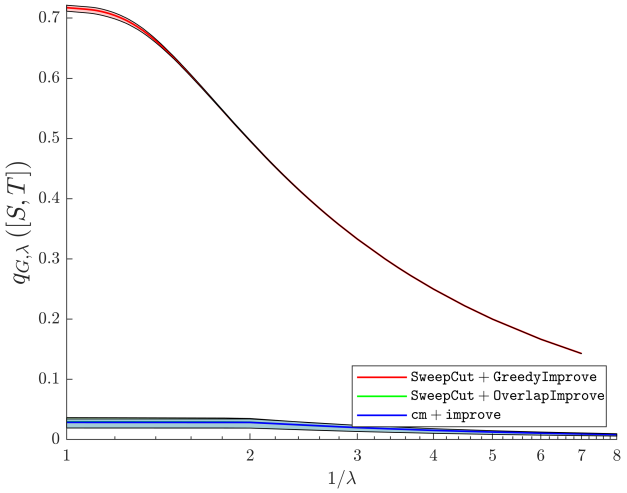
The first goal of our evaluation is to assess whether our OGP objectives reflect meaningful overlapping clustering structure on datasets where the ground-truth overlapping clusters are known. To address this question, we study the statistical performance of `cm+improve` in recovering overlapping partitions on graphs generated by the Overlapping Stochastic Block Model (OSBM) of Abbe and Sandon (Abbe & Sandon, 2015), the most generic statistical model of an overlapping clustering. This is an instance of the general stochastic block model in which the ground-truth partition is a tripartition (L, C, R) corresponding to an overlapping partition (S, T) where $L = S \setminus T, R = T \setminus S, C = S \cap T$.



(a)



(b)



(c)

Figure 2: (a) Statistical performance of `cm+improve` in the recovery of the ground-truth overlap C on 5 samples on balanced OSBM with parameters $(|C|, p, \epsilon) = (100, 4, 0.05)$. (b) Comparison on balanced OSBM with parameters $(|C|, p, q, \epsilon) = (10, 4, 4, \epsilon = 0.05)$. (c) Comparison on balanced OSBM with parameters $(|C|, p, q, \epsilon) = (10, 4, 4, \epsilon = 0.05)$.

Each pair of vertices $\{i, j\}$ is added to the edge set independently with probability $p \cdot \log n/n$ if both vertices belong to either S or T. If neither S nor T contains both i and j, they are connected with the smaller probability $\epsilon \cdot \log n/n$. The $\log n/n$ scaling is standard and ensures connectedness of the resulting graph. Besides the values of p and ϵ, our experiments varied the balance of the communities in the generated graphs and the size of the overlap. We also attempted to vary the probability assigned to pairs in the overlap $S \cap T$, but could not detect significant differences in the behavior of the algorithm, A full description of all settings is found in the Appendix.

Results The results were remarkably consistent across the size of the graph. We display here highlights of the results for the smallest graphs with $n = 10^4$. Figure 2(a) shows how the recovery performance of `cm+improve` changes as $1/\lambda$ increases and the size of the overlap grows. On the same y-axis, we also show how the contribution of the edge cutset to $q_{G,\lambda}([S, T)]$, i.e., the corresponding ϵ-ORC value. The overlap starts empty for small $1/\lambda$ on the left. As $1/\lambda$ increases, `cm+improve` starts including in the overlap $S \cap T$ vertices from the true C, boosting precision. Once the overlap is large enough, the recall follows so that we obtain essentially perfect recovery at $\lambda = 3$. At the same time, the edge cutset has significantly shrunk, as we have switched from cutting edges incident to C to cutting vertices in C.

If we continue increasing the overlap after this point, `cm+improve` will start adding vertices incident to edges in $E(S \setminus T, T \setminus S)$ to the overlap, until we reach a vertex cut. In this last phase, the edge cutsets decreases slowly, as the vertices added to the overlap do not belong to the C, but are endpoints of the more rare edges in $E(L, R)$. Indeed, the sharp elbow in the ϵ-ORC objective coincides with perfect recovery of the overlap, demonstrating that our method does not require prior knowledge of the overlap size. By contrast, all other algorithms in our test bed generally achieve poor recovery. We focus here on the comparison with `SweepCut` as the spectral approach comes with strong guarantees in stochastic block models. `METIS` performs entirely analogously. The example of Figure 2(b) is typical of its behavior when the overlap becomes large enough ($\approxeq \sqrt{n}$)). It shows that `cm+improve` outperforms both post-processings of `SweepCut` by an order of magnitude on the λ-HCUT objective. We do not show statistical information here because the precision and recall of `SweepCut` are both 0 for all values of λ, i.e., even after postprocessing `SweepCut` fails to find any vertex in the true overlap C. This phenomenon can be explained as follows: because of the sparsity of the ground-truth L and R and the relative density higher density of C, `SweepCut` finds outputs smaller cuts entirely contained within L or R. As the overlap is large, these cuts cannot be rounded to C by `GreedyImprove` or `HybridImprove`.

Figure 2(c) shows the same setup for a smaller ground-truth overlap $|C| = \Theta(\log n)$. In this case, the output of `SweepCut` is not too far from the optimal overlapping partition. Indeed, the `HybridImprove` post-processing matches the performance of `cm+improve` and achieves the same statistical performance. On the other hand, the heuristic `GreedyImprove` post-processing still fails to recover any vertices of C.

Our results support the overall superiority of global algorithms targeting overlapping measures of graph partitioning, such as `cm+improve`, over algorithms based on local improvement of edge-based cuts. Such standard approaches appear to fail in detecting overlapping clusters, even in the simple case of OSBM and even when given access to a very powerful overlapping cut-improvement in `HybridImprove`. We believe that this makes a powerful case for the adoption of OGP objective functions and algorithms in practice.

5.2. Information and Social Networks

In the next set of experiments, we evaluate the performance and efficiency of different methods on the λ-cut objective on a number of social and information networks from the SNAP database (Yang & Leskovec, 2015; Leskovec & Krevl, 2014). We now focus on finding *balanced* overlapping partitions for two reasons: i) our best competitor `METIS` is biased towards outputting balanced cuts, and ii) such partitions plausibly contain more interesting structural information and could be used to recursively decompose network. We find that `cm+improve` performs comparably to `METIS+HybridImprove`, showing that overlapping cuts in real networks tend to be somewhat correlated with sparse edge-based cuts. Unfortunately, the running time of `cm+improve` quickly becomes infeasible for networks with over 10^7 edges. We are confident that an optimized implementation taking greater advantage of parallelism and randomness will allow `cm+improve` to scale to even larger graphs. Due to space limitations, full quantitative results appear in the Appendix.

Recursive Bisection: In Section E.4 of the Appendix, we also highlight how recursive application of `cm+improve` to the DBLP co-authorship graph yields multi-way partitions that recover different areas of Computer Science and detect overlap between different interest communities.

References

Supplementary material, 2021. URL https://drive.google.com/drive/folders/1RK-Q_8_S6LFxmWC9oRhlZAicNWb-gcax?usp=sharing.

Abbe, E. and Sandon, C. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 670–688, 2015. doi: 10.1109/FOCS.2015.47.

Abrahao, B., Soundarajan, S., Hopcroft, J., and Kleinberg, R. On the separability of structural classes of communities. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 624–632, 2012.

Ahn, Y.-Y., Bagrow, J. P., and Lehmann, S. Link communities reveal multiscale complexity in networks. *nature*, 466(7307):761–764, 2010.

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. *Network Flows: Theory, Algorithms, and Applications*. Prentice hall, 1993.

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. Mixed membership stochastic blockmodels. *Journal of machine learning research*, 9(Sep):1981–2014, 2008.

Alon, N. and Milman, V. D. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38(1):73–88, 1985.

Andersen, R. and Lang, K. An algorithm for improving graph partitions. In *SODA '08 Proc. 19th ACM-SIAM Symp. Discret. algorithms*, pp. 651–660, 2008.

Andersen, R., Gleich, D. F., and Mirrokni, V. Overlapping clusters for distributed computation. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 273–282. ACM, 2012.

Arora, S., Rao, S., and Vazirani, U. Expander flows, geometric embeddings and graph partitioning. In *STOC '04 Proc. thirty-sixth Annu. ACM Symp. Theory Comput.*, pp. 222–231, New York, NY, USA, 2004. ACM. ISBN 1-58113-852-0. doi: http://doi.acm.org/10.1145/1007352.1007355.

Arora, S., Rao, S., and Vazirani, U. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009.

Arora, S., Hazan, E., and Kale, S. $O(\sqrt{\log(n)})$ approximation to sparsest cut in $\tilde{O}(n^2)$. *SIAM J. Comput.*, 39:1748–1771, 01 2010.

Arora, S., Ge, R., Sachdeva, S., and Schoenebeck, G. Finding overlapping communities in social networks: toward a rigorous approach. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pp. 37–54. ACM, 2012.

Balcan, M.-F., Borgs, C., Braverman, M., Chayes, J., and Teng, S.-H. I like her more than you: Self-determined communities. Technical report, 01 2012.

Bonchi, F., Gionis, A., and Ukkonen, A. Overlapping correlation clustering. *Knowledge and information systems*, 35(1):1–32, 2013.

Chen, L., Kyng, R., Liu, Y. P., Peng, R., Gutenberg, M. P., and Sachdeva, S. Maximum Flow and Minimum-Cost Flow in Almost-Linear Time. Technical Report arXiv:2203.00671, arXiv, April 2022. URL http://arxiv.org/abs/2203.00671. arXiv:2203.00671 [cs] type: article.

Cherkassky, B. and Goldberg, A. On implementing the push—relabel method for the maximum flow problem. *Algorithmica*, 19:390–410, 1997.

Cherkassky, B. V., Goldberg, A. V., and Radzik, T. Shortest paths algorithms: theory and experimental evaluation. In *SODA '94*, pp. 516–525, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics. ISBN 0-89871-329-3. URL http://dl.acm.org/citation.cfm?id=314464.314638.

Chung, F. R. K. *Spectral Graph Theory*. American Mathematical Society, 1997.

Dhillon, I. S., Guan, Y., and Kulis, B. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007.

Feige, U., Hajiaghayi, M., and Lee, J. R. Improved approximation algorithms for minimum-weight vertex separators. In *Proc. thirty-seventh Annu. ACM Symp. Theory Comput. - STOC '05*, 2005. ISBN 1581139608. doi: 10.1145/1060590.1060674.

Fiduccia, C. M. and Mattheyses, R. M. A linear-time heuristic for improving network partitions. In *DAC '82*, pp. 175–181, 1982.

Goldberg, A. Hipr version 3.7. /http://www.avglab.com/andrew/soft.html. Last retrieved December 2019. License: Attribution.

Goldberg, A. V. and Rao, S. Beyond the flow decomposition barrier. *J. ACM*, 45(5):783–797, September 1998. ISSN 0004-5411. doi: 10.1145/290179.290181. URL https://doi.org/10.1145/290179.290181.

Gopalan, P. K. and Blei, D. M. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):14534–14539, 2013.

Hagen, L. and Kahng, A. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992. doi: 10.1109/43.159993.

Hendrickson, B. and Leland, R. W. A multi-level algorithm for partitioning graphs. *SC*, 95(28):1–14, 1995.

Kannan, R., Vempala, S., and Vetta, A. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51 (3):497–515, 2004.

Karypis, G. and Kumar, V. Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0. 1995.

Karypis, G. and Kumar, V. Parallel multilevel graph partitioning. In *IPPS*, pp. 314–319, 1996.

Karypis, G. and Kumar, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, December 1998. ISSN 1064-8275. doi: 10.1137/S1064827595287997. URL http://dx.doi.org/10.1137/S1064827595287997.

Kernighan, B. W. and Lin, S. An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. J.*, 49(2): 291–307, February 1970a.

Kernighan, B. W. and Lin, S. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(2):291–307, 1970b.

Khandekar, R., Rao, S., and Vazirani, U. Graph partitioning using single commodity flows. *Journal of the ACM (JACM)*, 56(4):19, 2009.

Khandekar, R., Kortsarz, G., and Mirrokni, V. On the advantage of overlapping clusters for minimizing conductance. *Algorithmica*, 69(4):844–863, 2014.

Lang, K. and Rao, S. A flow-based method for improving the expansion or conductance of graph cuts. In Bienstock, D. and Nemhauser, G. (eds.), *Integer Programming and Combinatorial Optimization*, pp. 325–337, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-25960-2.

Latouche, P., Birmelé, E., and Ambroise, C. Overlapping stochastic block models with application to the French political blogosphere. *Annals of Applied Statistics*, 5(1):309–336, 2011. ISSN 19326157. doi: 10.1214/10-AOAS382.

Leighton, T. and Rao, S. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999.

Leskovec, J. and Krevl, A. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, 2014.

Leskovec, J., Lang, K., Dasgupta, A., and Mahoney, M. W. Statistical properties of community structure in large social and information networks. In *Proceeding of the 17th international conference on World Wide Web*, pp. 695–704. ACM, 2008.

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

Li, P., Dau, H., Puleo, G., and Milenkovic, O. Motif clustering and overlapping clustering for social network analysis. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9. IEEE, 2017.

Mishra, N., Schreiber, R., Stanton, I., and Tarjan, R. E. Clustering social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pp. 56–67. Springer, 2007.

Nepusz, T., Yu, H., and Paccanaro, A. Detecting overlapping protein complexes in protein-protein interaction networks. *Nature methods*, 9(5):471, 2012.

Orecchia, L. *Fast Approximation Algorithms for Graph Partitioning using Spectral and Semidefinite-Programming Techniques*. PhD thesis, EECS Department, University of California, Berkeley, May 2011. URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-56.html.

Orecchia, L., Schulman, L. J., Vazirani, U. V., and Vishnoi, N. K. On partitioning graphs via single commodity flows. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pp. 461–470, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605580470. doi: 10.1145/1374376.1374442. URL https://doi.org/10.1145/1374376.1374442.

Palla, K., Knowles, D., and Ghahramani, Z. An infinite latent attribute model for network data. *arXiv preprint arXiv:1206.6416*, 2012.

Räcke, H. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 255–264. ACM, 2008.

Sanders, P. and Schulz, C. Think Locally, Act Globally: Highly Balanced Graph Partitioning. In *Proceedings of the 12th International Symposium on Experimental Algorithms (SEA'13)*, volume 7933 of *LNCS*, pp. 164–175. Springer, 2013.

Shi, J. and Malik, J. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

Tsourakakis, C. Provably fast inference of latent features from networks: with applications to learning social circles and multilabel classification. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015*, pp. 1111–1121, 2015.

Tsuda, K., Rätsch, G., and Warmuth, M. K. Matrix exponentiated gradient updates for on-line learning and bregman projection. *Journal of Machine Learning Research*, 6(34):995–1018, 2005. URL http://jmlr.org/papers/v6/tsuda05a.html.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Whang, J. J., Gleich, D. F., and Dhillon, I. S. Overlapping community detection using neighborhood-inflated seed expansion. *IEEE Transactions on Knowledge and Data Engineering*, 28(5):1272–1284, 2016.

Yang, J. and Leskovec, J. Community-affiliation graph model for overlapping network community detection. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 1170–1175, 12 2012. ISBN 978-1-4673-4649-8. doi: 10.1109/ICDM.2012.139.

Yang, J. and Leskovec, J. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 587–596, 2013.

Yang, J. and Leskovec, J. Overlapping communities explain core–periphery organization of networks. *Proceedings of the IEEE*, 102(12):1892–1902, 2014.

Yang, J. and Leskovec, J. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.

Zachary, W. W. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

Zahn, C. T. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on computers*, 100(1):68–86, 1971.

A. Cut Improvement Algorithms

A.1. The `HybridImprove` Algorithm

Specification The `HybridImprove` algorithm takes the following **inputs**:

1. an undirected graph $G = (V, E, w, \mu)$ with non-negative integral edge weights $\{w_e\}_{e \in E}$ and non-negative integral vertex weights $\{\mu_i\}_{i \in V}$.
2. a non-overlapping partition $(S_0, \bar{S}_0)$ of V.
3. a value $\lambda \geq 0$ for which we seek to minimize $q_{G,\lambda}$.

The `HybridImprove` algorithm returns the following **outputs**:

1. an overlapping partition $[S, T]$,
2. a weighted graph $H = (V, E_H, w_H \in \mathbb{R}_{>0}^{E_H})$.

Assume without loss of generality that $\mu(S_0) \leq \mu(\bar{S}_0)$ and define $\kappa \stackrel{\text{def}}{=} \mu(S_0)/\mu(\bar{S}_0) \in (0, 1)$.

The flow network G_α The algorithm starts by building an auxiliary flow network G_α, parametrized by $\alpha \geq 0$ from G. To support vertex capacities, for each vertex $v \in V$, G_α contains two vertices labeled v_{IN} and v_{OUT}, together with a directed edge $(v_{\text{IN}}, v_{\text{OUT}})$ with capacity $\alpha \cdot \lambda \cdot \mu_i$. Every edge $\{u, v\} \in E$ yields two directed arcs $(u_{\text{OUT}}, v_{\text{IN}})$ and $(v_{\text{OUT}}, u_{\text{IN}})$ of capacity $\alpha \cdot w_{uv}$ in G_α. Finally, G_α contains two auxiliary nodes, a source s and a sink t. They are connected to the rest of graph based on the input partition $(S_0, \bar{S}_0)$ as follows:

- for all $v \in S_0$, there is an arc (s, v_{IN}) with capacity μ_i;
- for all $v \in \bar{S}_0$, there is an arc (v_{OUT}, t) with capacity $\kappa \cdot \mu_i$.

The capacity on the $\bar{S}_0$-side are scaled down by κ to ensure that the total capacity of the trivial source cut equals the total capacity of the trivial sink cut. As we aim to set α to be large enough such that both these cuts are saturated, these capacities can also be interpreted as demands we want to concurrently route from S_0 to $\bar{S}_0$. The construction of G_α is illustrated in Figure 3.

Searching over the α parameter The `HybridImprove` algorithm aims to find the minimum value $\alpha = \alpha^*$ such that a single-commodity flow can be routed from s to t while fully saturating the source cut and the sink cut, i.e., while routing $\alpha \cdot \mu(S_0)$ units of flow. To do so, it performs a binary search over α by testing, for each α whether the required flow can be routed by solving the corresponding s-t maximum-flow. Assuming that the graph G is connected and that the edge- and vertex-weights, together with the parameter λ are integral, α can range between $1/\mu(V)$ and $w(E) + \lambda\mu(V)$, so that $O(\log(|V| \cdot W_G)$ rounds suffice to compute α^*.

Returning the output Once the algorithm has identified the optimal value α^*, it extracts a non-trivial s-t mincut (A, B) in G_{α^*}. Such a mincut is guaranteed to exist by the variational definition of α^*. The output overlapping partition $[S, T]$ of G is formed as following:

1. A vertex $i \in V$ is placed in S if $v_{\text{IN}} \in A$.
2. A vertex $i \in V$ is placed in T if $v_{\text{OUT}} \in B$.

In particularly, vertices for which both conditions hold are placed in the overlap $S \cap T$.

Finally, the algorithm computes a flow-path decomposition of the flow routed into G_α using dynamic trees (Ahuja et al., 1993; Khandekar et al., 2009), to obtain a list of flow paths routed from S_0 to $\bar{S}_0$. The demands routed by these paths are defined to be the graph H returned by `HybridImprove`.

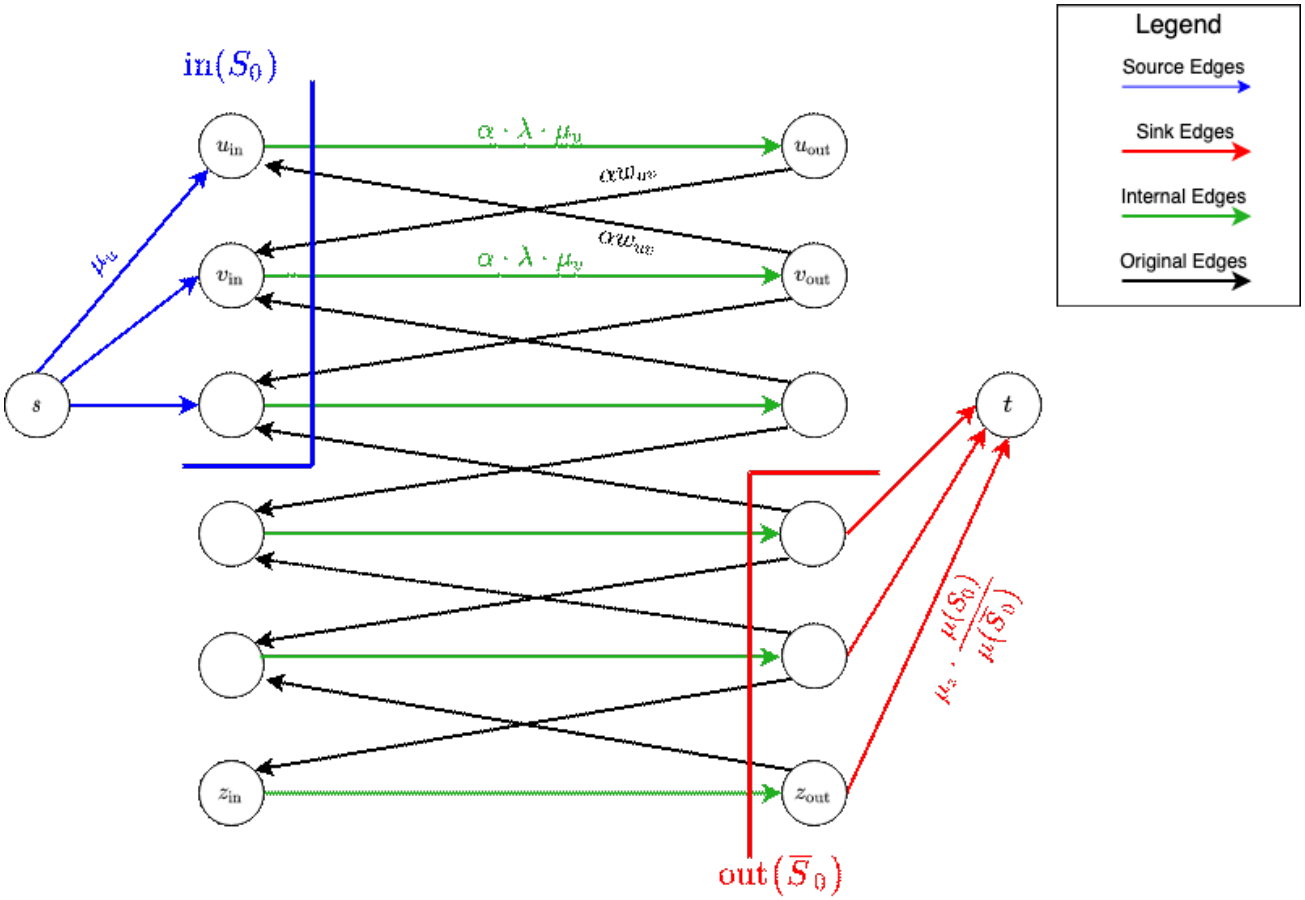


Figure 3: The flow network G_α for a path graph on 6 vertices, for a bisection $(S, \bar{S}_0)$ into connected components.

A.2. The `OverlapImprove` Algorithm

Specification The `OverlapImprove` algorithm takes the following **inputs**:

1. an undirected graph $G = (V, E, w, \mu)$ with non-negative integral edge weights $\{w_e\}_{e \in E}$ and non-negative integral vertex weights $\{\mu_i\}_{i \in V}$.
2. a non-overlapping partition $(S_0, \bar{S}_0)$ of V.
3. a value $\epsilon \in [0, 1]$ for which we seek to minimize $q_{G,\lambda}$.

The `OverlapImprove` algorithm returns the following **outputs**:

1. an overlapping partition $[S, T]$,
2. a weighted graph $H = (V, E_H, w_H \in \mathbb{R}_{>0}^{E_H})$.

Assume without loss of generality that $\mu(S_0) \leq \mu(\bar{S}_0)$ and define $\kappa \stackrel{\text{def}}{=} \mu(S_0)/\mu(\bar{S}_0) \in (0, 1)$.

Description The `OverlapImprove` algorithm is obtained by performing a binary search on the input λ of the `HybridImprove` algorithm, starting at $\max_i \Sigma_{j \sim i} w_{ij}/\mu_i$. If the output overlapping partition $[S, T]$ has $q_V[S, T] \geq \epsilon$, then λ is reduced. Otherwise, it is increased. The process eventually stops in polylogarithmic iterations for λ^* and α^* such that two s-t mincuts exists in G_{α^*}, corresponding to overlapping partitions $[S_1, T_1]$ with $q_V[S_1, T_1] \leq \epsilon$ and $[S_2, T_2]$ with $q_V[S_2, T_2] \geq \epsilon$. The submodularity of the cut function implies that we must necessarily have $\delta_V[S_1, T_1] \subseteq \delta_V[S_2, T_2]$. Hence, adding a single vertex from the overlap $\delta_V[S_2, T_2]$ to the overlap $\delta_V[S_1, T_1]$ yields a new overlapping partition that also corresponds to a s-t mincut in G_{α^*}. By the bound R on the ratio of weights, we have that the resulting overlapping partition $[S, T]$ has $q_V[S, T] \leq (R + 1)\epsilon$.

B. Cut Improvement Analysis: Proof of Theorem 3

B.1. Valid s-t cuts and corresponding overlapping partitions

We start by proving some simple lemmata about the `HybridImprove` construction, which is essentially a reduction from the overlapping improvement problem to a family of $s - t$ minimum cut problems on bipartite flow networks G_α. To do this end, we define a subset of s-t cuts in G_α that can be put in bijection with overlapping partitions of G.

Definition 2. *An s-t cut (A', B') of G_α is valid if, for all vertices v, $v_{\text{IN}} \in B'$ implies $v_{\text{OUT}} \in B'$. Equivalently, for all v, $v_{\text{OUT}} \in A'$ implies $v_{\text{IN}} \in A'$.*

The relevance of this definition is shown by the following lemma.

Lemma 8. *An s-t mincut (S', T') in G_α is valid.*

Proof. Suppose there exists $v \in V$ such that $v_{\text{IN}} \in B'$ and $v_{\text{OUT}} \in A'$. Including v'_{OUT} in B' decreases the capacity of the s-t cut as the only arc going into v_{OUT} is the arc $(v_{\text{IN}}, v_{\text{OUT}})$, which has strictly positive capacity. □

The bijection between valid s-t cuts in G_α and corresponding overlapping partitions of G is constructed as follows: a valid s-t cut (A', B') maps to the overlapping partition $[A, B]$ such that $v \in A$ if $v_{\text{IN}} \in A'$ and $v \in B$ if $v_{\text{OUT}} \in B'$. Notice that both of these conditions will hold for a vertex in the overlap $A \cap B$. By the validity of (A', B'), we deduce that $A \cup B = V$, so that $[A, B]$ is indeed an overlapping partition of s-t. Similarly, for an overlapping partition $[A, B]$, with $\mu(A) \leq \mu(B)$ we can construct a valid s-t cut (A', B') as follows: if $v \in A \setminus B$, let $v_{\text{IN}}, v_{\text{OUT}} \in A'$; if $v \in B \setminus A$, let $v_{\text{IN}}, v_{\text{OUT}} \in B'$; if $v \in A \cap B$, let $v_{\text{IN}} \in A'$ and $v_{\text{OUT}} \in B'$.

The following lemma describes the relation between the capacity of a valid s-t cut (S', T') in G_α and the edge- and vertex-cutsets of the corresponding overlapping partition $[S, T]$. For a subset C' of vertices in G_α, we denote by C'_{IN} its IN vertices and by C'_{OUT} its OUT vertices.

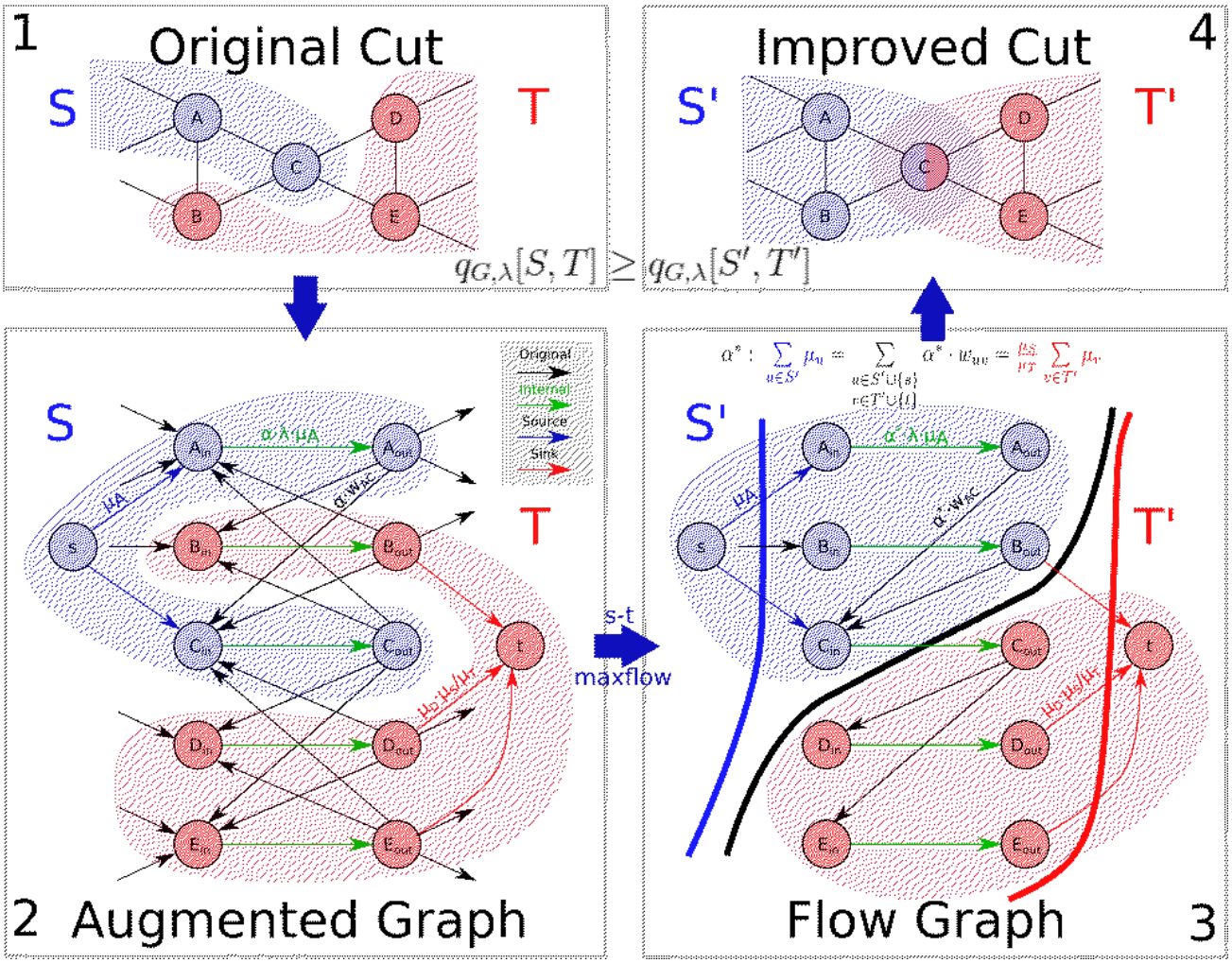


Figure 4: A valid s-t mincut (S', T') of G_α, together with the corresponding overlapping partition.

Lemma 9. *Let (S', T') be a valid s-t cut in G_α and let $[S, T]$ be the corresponding overlapping partition. Then:*

$$\mathrm{cap}_{st}(S', T') = \alpha \cdot w(\delta_E[S, T]) + \alpha \cdot \lambda \cdot \mu(\delta_V[S, T]) + \mu(\bar{S} \cap S_0) + \kappa\mu(\bar{T} \cap \bar{S}_0).$$

Proof. The capacity of (S', T') can be written in terms of the capacities between different subsets of G_α. By the construction of G_α (see also Figure 4), we obtain the following:

$$\mathrm{cap}_{st}(S', T') = \mathrm{cap}_{st}(S'_{\text{IN}}, T'_{\text{OUT}}) + \mathrm{cap}_{st}(S'_{\text{OUT}}, T'_{\text{IN}}) + \mathrm{cap}_{st}(\{s\}, T'_{\text{IN}}) + \mathrm{cap}_{st}(S'_{\text{OUT}}, \{t\}).$$

Notice now that $\mathrm{cap}_{st}(S'_{\text{IN}}, T'_{\text{OUT}}) = \alpha \cdot \lambda \cdot \mu(\delta_V[S, T])$, as the only arcs going from the IN-side to the OUT-side are the internal edges of vertices included in the overlap. Similarly for the second term, $\mathrm{cap}_{st}(S'_{\text{OUT}}, T'_{\text{IN}}) = w(\delta_E[S, T])$, as the only arcs going the opposite way correspond to original edges in $\delta_E[S, T]$. Finally, the last two terms arise from vertices in S_0 that were moved directly to the opposite side $\bar{S}$ and of vertices in $\bar{S}_0$ that were switched over to $\bar{T}$. An example of such a vertex is vertex y in Figure 4. By the choice of capacities from s and to t, we have $\mathrm{cap}_{st}(\{s\}, T'_{\text{IN}}) = \mu(\bar{S} \cap S_0)$ and $\mathrm{cap}_{st}(S'_{\text{OUT}}, \{t\}) = \kappa\mu(\bar{T} \cap \bar{S}_0)$, completing the proof. □

B.2. Splits of overlapping partitions

Next, we discuss the notion of a non-overlapping split of an overlapping partition $[A, B]$, formalizing the notion behind the definition of $\Psi_{G,\mu}$.

Definition 3. *Let $[A, B]$ be an overlapping partition of vertex set V. A non-overlapping split $(C, \bar{C})$ of $[A, B]$ is a non-overlapping partition of V such that $C \subseteq A$ and $\bar{C} \subseteq B$. The non-overlapping split of $[A, B]$ according to a non-overlapping partition $(S, \bar{S})$ is the non-overlapping split that assigns vertices in $A \cap B$ to C or $\bar{C}$ based on their location in $(S, \bar{S})$, i.e.:*

$$C = \bar{B} \cup (A \cap S) \subseteq A$$
$$\bar{C} = \bar{A} \cup (B \cap \bar{S}) \subseteq B$$

We will need the following fact about the relation between flows across $[A, B]$ and splits of $[A, B]$ in G_α.

Lemma 10. *Let $[A, B]$ be an overlapping partition of V and $(C, \bar{C})$ be a split of $[A, B]$. Let (A', B') and $(C', \bar{C}')$ denote the corresponding s-t cuts in G_α. For an s-t maximum flow in G_α, the following holds:*

$$\mathrm{netflow}_{st}(A', B') \geq \mathrm{netflow}_{st}(C', \bar{C}').$$

The same holds with equality if (A', B') is a non-trivial s-t mincut of G_α and $(C, \bar{C})$ is the split according to $(S_0, \bar{S}_0)$.

Proof. Consider any $v \in A \cap B$ in the overlap of $[A, B]$. In the flow network G_α, we have $v_{\text{IN}} \in A'$ and $v_{\text{OUT}} \in B'$. Because all the flow out of v_{IN} and into v_{OUT} runs along the internal arc $(v_{\text{IN}}, v_{\text{OUT}})$, shifting v_{IN} or v_{OUT} to the other side of the s-t can only decrease the netflow. For the second part of the lemma, further assume the same $v \in S_0$. Then equality holds as long as there is no flow into v_{IN} via arcs coming from B'. Similarly, for $v \in \bar{S}_0$, equality holds if there is no flow from v_{OUT} to vertices in A'. This is the case if (A', B') is an s-t mincut of G_α. □

B.3. Flow and cuts in G_{α^*} and their relation with the demand graph H

Next, we use the demand graph H, which has been routed from S_0 to $\bar{S}_0$ in G_{α^*}, to construct lower bounds on the hybrid quotient-cut $q_{G,\lambda}([A, B])$ of an overlapping partition $[A, B]$:

Lemma 11. *For any overlapping partition $[A, B]$ of G:*

$$q_{G,\lambda}([A, B]) \geq \frac{\Psi_{H,\mu}([A, B])}{\alpha^*}$$

Equality is achieved for any overlapping partition $[S, T]$ corresponding to a non-trivial s-t mincut in G_{α^*}, yielding the stronger bound:

$$q_{G,\lambda}([S, T]) = \frac{\Psi_{H,\mu}([A, B])}{\alpha^*} \leq \frac{1}{\alpha^*}$$

Proof. Consider an overlapping partition $[A, B]$ of V and its corresponding s-t mincut (A', B') in G_{α^*}. Take also any non-overlapping split $(C, \bar{C})$ of $[A, B]$ and its corresponding s-t cut $(C', \bar{C}')$. By the s-t maxflow-mincut theorem and Lemma 10, for an s-t maxflow on G_{α^*}, we have:

$$\mathrm{cap}_{st}(A', B') \geq \mathrm{netflow}_{st}(A', B') \geq \mathrm{netflow}_{st}(C', \bar{C}').$$

Now, we can easily relate $\mathrm{netflow}_{st}(C', \bar{C}')$ to the cut $(C, \bar{C})$ in H :

$$\mathrm{netflow}_{st}(C', \bar{C}') = w_H(E(C, \bar{C})) + \mu(\bar{C} \cap S_0) + \kappa\mu(C \cap \bar{S}_0). \tag{7}$$

Compare this lower bound on $\mathrm{cap}_{st}(A', B')$ with the result of Lemma 9:

$$\mathrm{cap}_{st}(A', B') = \alpha^* \cdot w(\delta_E[A, B]) + \alpha^* \cdot \lambda \cdot \mu(\delta_V[A, B]) + \mu(\bar{A} \cap S_0) + \kappa\mu(\bar{B} \cap \bar{S}_0). \tag{8}$$

By the definition of $(C, \bar{C})$, we have that $\bar{A} \subseteq \bar{C}$ and $\bar{B} \subseteq C$. Hence, the last two terms in Equation 7 dominate the last two terms of Equation 8. Combining Equation 7 and Equation 8, we then get:

$$\alpha^* \cdot w(\delta_E[A, B]) + \alpha^* \cdot \lambda \cdot \mu(\delta_V[A, B]) \geq w_H(E(C', \bar{C}'))$$

Dividing both sides by $\min\{\mu(A), \mu(B)\}$ completes the proof of the first part of the lemma as:

$$q_{G,\lambda}([A, B]) \geq \frac{\Psi_{H,\mu}([A, B])}{\alpha^*}$$

For the second part, the s-t maximum-flow minimum-cut theorem and Lemma 10 ensure that for a non-trivial s-t minimum cut (S', T') and its split $(C, \bar{C})$ according to S_0:

$$\mathrm{cap}_{st}(S', T') = \mathrm{netflow}_{st}(S', T') = \mathrm{netflow}_{st}(C', \bar{C}').$$

Moreover, we have that $\bar{C} \cap S_0 = \bar{A} \cap S_0$ and $C \cap \bar{S}_0 = \bar{B} \cap \bar{S}_0$, so that the last two terms in Equations 7 and 8 cancel exactly, yielding:

$$\alpha^* \cdot w(\delta_E[S, T]) + \alpha^* \cdot \lambda \cdot \mu(\delta_V[S, T]) = w_H(E(C, \bar{C})).$$

By construction of α^*, we also know that the capacity of any non-trivial s-t minimum cut in $G_{\alpha*}$ equals that of the trivial s-t cut S, which is $\mu(S_0)$. Hence:

$$\begin{aligned} w_H(C, \bar{C}) &= \mu(S_0) - \mu(\bar{A} \cap S_0) - \kappa\mu(\bar{B} \cap \bar{S}_0) \\ &\leq \min\{\mu(A \cap S_0), \kappa\mu(B \cap \bar{S}_0)\} \\ &\leq \min\{\mu(A), \mu(B)\}. \end{aligned}$$

Equivalently, we have that $\Psi_{H,\mu}([S, T]) \leq 1$. Together with the first part of the lemma, this yields:

$$q_{G,\lambda}([S, T]) = \frac{q_H(S, T)}{\alpha^*} \leq \frac{1}{\alpha^*}.$$

□

B.4. Proof of Theorem 3

We are now ready to complete the proof of the main theorem:

Proof. By construction, the demand graph H is bipartite between S_0 and $\bar{S}_0$. Moreover, H is induced by an s-t maximum flow on G_{α^*}, so that each vertex $i \in S_0$ routes μ_i units of flow to $\bar{S}_0$ and each vertex $j \in \bar{S}_0$ routes $\kappa\mu_j$ units of flow to S_0. Hence, the degree in H of $i \in S_0$ is μ_i and the degree in H of $j \in \bar{S}_0$ is $\kappa\mu_j$, as required.

For any overlapping partition (A, B), combining the two part of Lemma 11 ensures that

$$q_{G,\lambda}([A, B]) \geq \frac{\Psi_{H,\mu}([A, B])}{\alpha^*} \geq \Psi_{H,\mu}([A, B]) \cdot q_{G,\lambda}([S, T]),$$

which is the required approximation guarantee. The proof for `OverlapImprove` is entirely analogous.

To bound the running time of `HybridImprove`, we notice that, for a connected [3], α^* must lie in the interval $\left[\frac{1}{|E|W_G}, |V|W_G\right]$, or it is not possible to have the trivial s-t cut have the same capacity as a non-trivial s-t minimum cut. Hence, performing binary search requires $O(\log|V| + \log W_G)$ s-t maxflow computations on graphs G_α in which capacities can be rescaled to be integral and at most $|E|W_G^2$. This yields the first term in the promised running time. The second term $|E|\log(W_G|V|)$ accounts for the computation of graph H, which is achieved by a flow-path decomposition of the s-t maximum flow in G_{α^*} via dynamic trees (Goldberg & Rao, 1998).

□

The following simple corollary show that `HybridImprove` is indeed a cut-improvement algorithm, in that it always improves the initial input partition.

Corollary 12.

$$q_\lambda(S,T) \le q(S_0, \bar{S}_0),$$

Proof. Because H is bipartite across $(S_0, \bar{S}_0)$ and has degrees proportional to μ, we have $q_H(S_0, \bar{S}_0) = 1$. The result then follows from the quotient cut guarantee of the main theorem. □

C. Other Proofs

Lemma 13 (Lemma 1 in main body)**.** *For any $\lambda \ge 0$, let (L, C, R) be an optimal solution for the λ-HCUT problem. Let $S \stackrel{\text{def}}{=} L \cup C$ and $T \stackrel{\text{def}}{=} R \cup C$ and define $\epsilon = {\mu(\delta_V(S,T))}/{\min\{\mu(S),\mu(T)\}}$. Then, (S,T) is an optimal solution to the ϵ-ORC problem.*

Proof. Suppose (S,T) is not optimal. Then, there exists a different overlapping clustering (S',T') of smaller objective value with $\mu(S' \cap T') \le \min\{\mu(S), \mu(T)\} \le \epsilon$. Let $L' \stackrel{\text{def}}{=} S' \setminus T'$, $R' \stackrel{\text{def}}{=} T' \setminus S'$ and $C = S' \cap T'$. Hence, we have:

$$\begin{aligned} q_\lambda(L', C', R') &= \frac{w(\delta_E(S',T')) + \lambda \cdot \mu(\delta_V(S',T'))}{\min\{\mu(S'), \mu(T')\}} \\ &\le \text{ORC}(S',T') + \lambda\epsilon < \text{ORC}(S,T) + \lambda\epsilon = q_\lambda(L,C,R). \end{aligned}$$

This contradicts the optimality of (L,C,R). □

Proof of Theorem 2. It remains to prove the running time result. This is a simple consequence of Theorem 5 in the main body. The total number of calls to `HybridImprove` is $T = O(\log^2|V|)$, which yields the polylog bound in the theorem. The total cost of computing the cut strategy is at most $O(\log^2|V|) \cdot \log(|V|W_G) \cdot |E(\bar{H}_T)|$, as $|E(\bar{H}_t)|$ is monotonically increasing. However, by construction $|E(\bar{H}_T)| \le \sum_{t=1}^{T} |E(H_t)| \le O(|E(G)| \cdot \log(W_G|V|))$ by the proof of Theorem 3 in the main body. □

D. Implementation Details

Our implementation is available online (sup, 2021). It departs in a small number of places from the theoretical description of the `HybridImprove` algorithm. We highlight them here:

- Following other implementations of cut-improvement algorithms by a subset of co-authors for standard ratio-cut objectives, rather than performing binary search over α, we initialize α to be the ${1}/{q_{G,\lambda}(S_0,\bar{S}_0)}$. After each maxflow operation, we extract the overlapping partition $[A, B]$c from the s-t mincut in G_α and update α to be ${1}/{q_{G,\lambda}(A,B)}$.
- We limit the precision of the value α, a rational number, by approximating it up to a 1.001-factor using Farey sequences. When running the maximum flow operation, we scale all capacities in G_α by the denominator in our representation of α to ensure all capacities become integral.

[3] The partitioning problem is trivial if the graph G is disconnected

- For the results in this paper, we use a simpler DFS-based algorithm to compute the flow-path decomposition of the flow routed on G_α. We are currently in the process of implementing dynamic trees to further speed up this operation.

E. Empirical Evaluation

E.1. Comparison with BIGCLAM

Communities	Average community size	Nodes absent
2	3591.5	75931
5	1881.4	78334
20	1030.15	69333
100	594.3	46549

Table 1: BIGCLAM node coverage statistics in cExtractedDblp graph with $n = 83114$ vertices

In order to evaluate our algorithm's performance we considered BIGCLAM (Yang & Leskovec, 2013) and its predecessor AGMFIT (Yang & Leskovec, 2012). The latter was disqualified because it required a quadratic number of iterations, which is computationally infeasible for larger graphs. The successor BIGCLAM replaces the discrete step in the EM algorithm with a continuous one, requiring far fewer iterations. While the running time is no longer a problem, the partitions output by BIGCLAM do not cover the entire graph. A majority of the nodes belongs to no community, even when the algorithm is allowed to output a large number of communities. This could ameliorated if only a few nodes were absent from the communities. However, when a majority of the nodes are unclassified, there is no easy way to convert BIGCLAM's output into a partitioning scheme without radically changing the algorithm. The problem definition of BIGCLAM is very elegant, but unfortunately the current implementation cannot be used as a comparison baseline. It is also conceivable that BIGCLAM may optimize an objective that is inherently different from ratio-cut objectives. Indeed, BIGCLAM has no restriction against including high degree nodes in the overlap, which is sub par in our problem definition, as the cost for a node to be included is proportional to its degree. BIGCLAM may serve as a more useful benchmark when considering the problem of detecting small overlapping communities in the periphery of a large information network . Indicatively, in Table 1, we present the average community size and the total uncovered nodes for a number of settings of the parameter regulating the number of communities output.

E.2. OSBM Experiments

Table 2a describes the coefficients applied to the scaling $\log n/n$ to define the probabilities of including edge between different parts of the ground-truth tripartition in the OSBM model. Table 2b displays the different parameters choices for the graph generation in our OSBM experiments.

	L	R	C
L	p	ϵ	p
R	ϵ	p	p
C	p	p	q

(a) Edge probability coefficients between two vertices in the OSBM model.

Size	L-R-C	p-q
10,000	0.45 - 0.45 - 0.10	4-2
30,000	0.45-0.45-0.01	4-4
100,000	0.6-0.3-0.1	4-6
	0.745-0.245-0.01	4-8

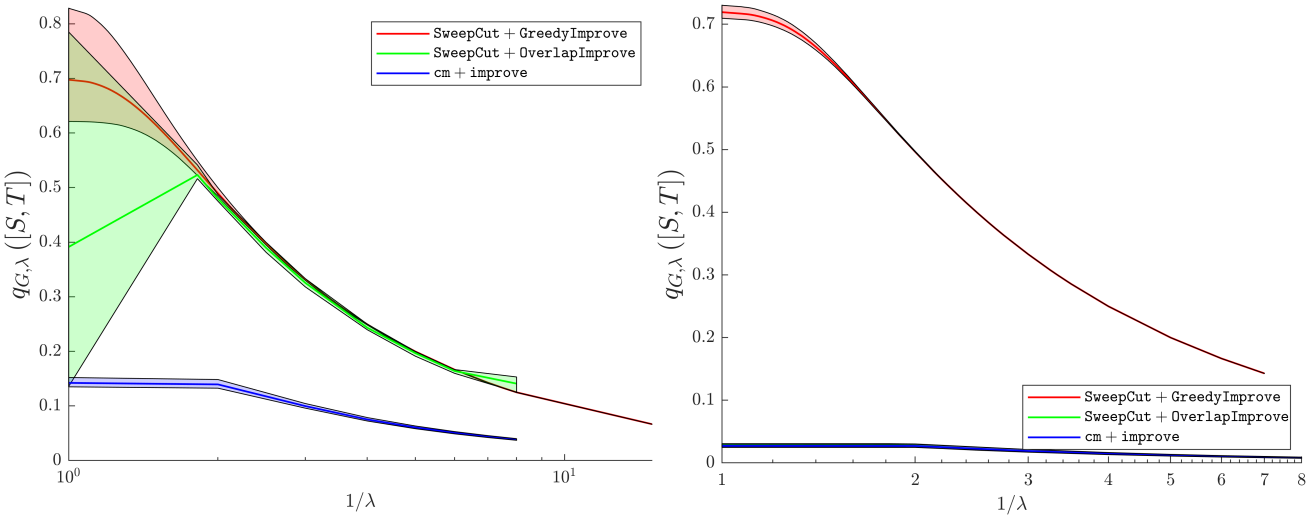
(b) Values used for generating OSBM graphs. $\epsilon = 0.05$.

The results of the experiments on OSBM were remarkably consistent across the size of the graph and the choice of p and q. Here and in the paper, we display results for the smallest graphs with $n = 10^4$. Further study is required to find interesting settings of p and q where threshold phenomena may arise. Figure 5 shows the comparison between cm + improve and SweepCut for the case when the partition is unbalanced, as given by the third and fourth entry of second column in Table 2b. The main body of the paper includes the same results for the balanced choice (first and second entry of second column in

Network	n	m	time
DBLP	83,114	409,541	2-4min
Amazon	334,863	925,872	15-18min
Youtube	1,134,890	2,987,624	55-75min
LiveJournal	3,997,962	17,340,594	5-8h

Table 3: Social networks overlook and range of running times on 5 executions.

Table 2b). As in the balanced case, for large overlap size $|C|$, both versions of the `SweepCut` algorithm fail to detect the ground-truth overlap and have much larger λ-HCUT values. However, in this case, `SweepCut+HybridImprove` actually often achieves the same value of `cm` + `improve` when only looking for an edge partition, but is unable to maintain that performance when looking for overlaps. Once again, this suggests the need for algorithms that explicitly target overlapping clustering notions, rather than locally improving edge-based notions.



(a) Comparison on unbalanced OSBM with parameters $(|C|, p, q, \epsilon) = (100, 4, 4, \epsilon = 0.05)$.

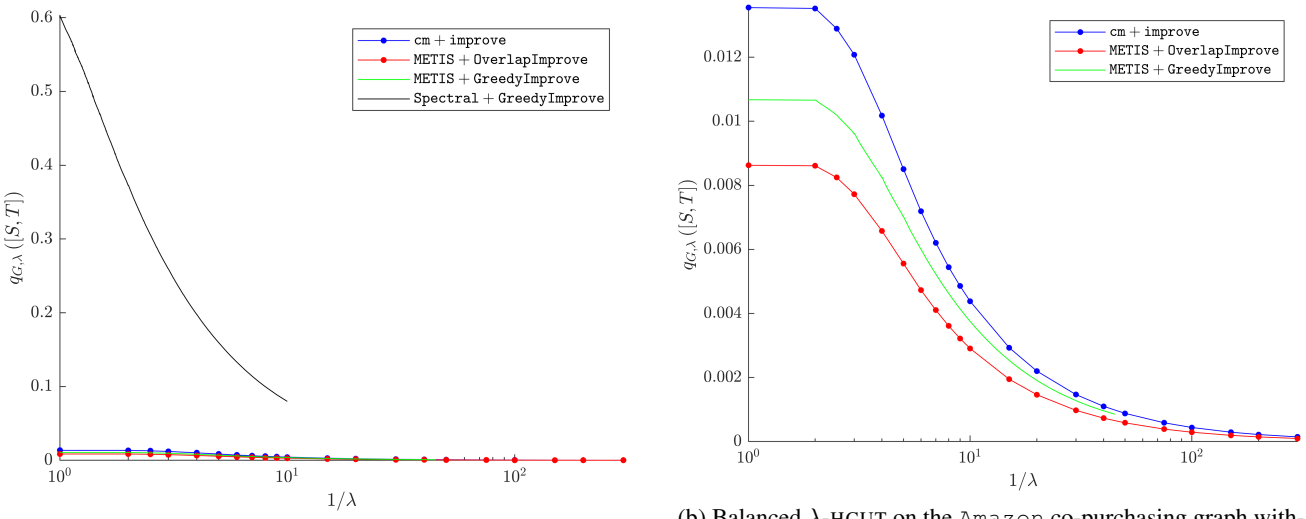
(b) Comparison on unbalanced OSBM with parameters $(|C|, p, q, \epsilon) = (10, 4, 4, \epsilon = 0.05)$

Figure 5: Performance of `SweepCut` against `cm+improve` on graphs from the OSBM model with $n = 10^4$. Each graph displays the performance over 5 samples from the model. The shaded area shows the minimum and maximum values over the samples, while the bold curve represents the average.

E.3. Large Social and Information Networks

We now describe the quantitative results of the comparison between `cm+improve` and `METIS` for the task of finding balanced overlapping partitions on the large networks in the SNAP database (Leskovec & Krevl, 2014). Table 3 displays our selection of graphs, together with their diverse sizes, and the running time required by `cm+improve`. Below, we focus on the performance as measured by the λ-HCUT objective $q_{G,\lambda}$.

We start with the Amazon graph in Figure 6. The left subfigure here includes data for `SweepCut+GreedyImprove`, showing a performance that is two orders of magnitude worse than `cm+improve` or `METIS`. The right subfigure excludes `SweepCut+GreedyImprove` allowing us to have a closer comparison with `METIS`. It shows that `METIS`, with either post-processing, slightly outperforms `cm+improve`, especially for large λ, i.e., edge-based cuts. This is not surprising, as `METIS` is highly optimized to find sparse balanced edge cuts. The fact that this advantage persists when searching for overlapping clusters may be an indication of the lack of meaningful overlapping structure over balanced cuts for this network.



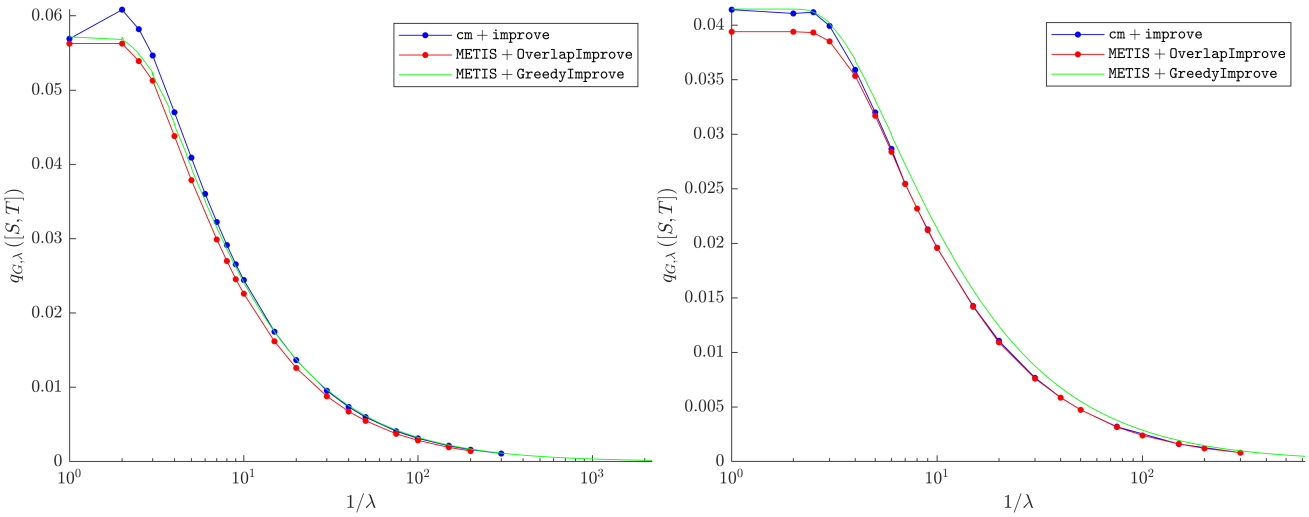
(a) Balanced λ-HCUT on `Amazon` copurchasing graph.

(b) Balanced λ-HCUT on the `Amazon` co-purchasing graph without `Spectral + GreedyImprove`.

Figure 6

It is also possible that `cm+improve` may require a larger number of iterations to achieve its optimal performance or that our balanced heuristic needs to be refined.

The results for the Youtube and DBLP graphs are shown in Figure 7. For these graphs, `cm+improve` essentially matches the performance of `METIS` which is a testament to the power of our algorithm framework, even in the non-overlapping settings. The fact



(a) Balanced λ-HCUT on Youtube social graph.

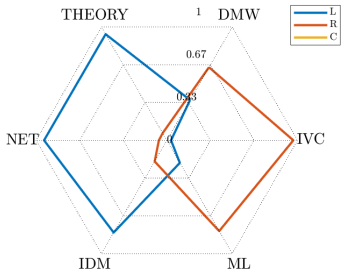
(b) Balanced λ-HCUT on DBLP co-authorship graph.

Figure 7

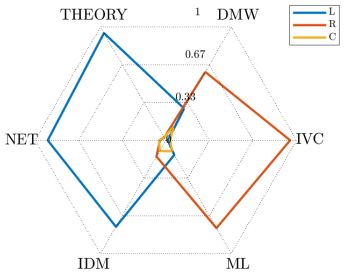
Future work should further address this comparison by relaxing the balanced constraint or using overlapping balanced clustering recursively to produce overlapping decompositions of the network. Such decompositions may detect meaningful overlapping structure at smaller sizes and in localized areas of the graph.

Visualization of Overlapping Clusters in DBLP co-authorship network The DBLP dataset for co-authorship in the academic field of Computer Science gives us the opportunity to visualize the overlapping communities discovered by `cm+improve` and compare them with the known clustering based on the venue of the each paper. Specifically, we built the

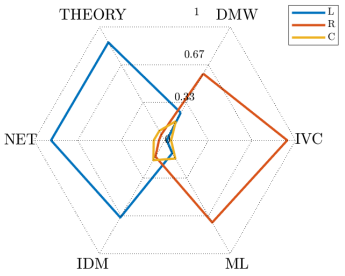
co-authorship network creating for every paper p a weighted clique on the d-coauthors of paper p. The weight of this clique is equal to $\frac{1}{d}$ to ensure that each paper carries the same amount of information, i.e., the resulting random walk on the graphs consists of choosing a paper uniformly at random and sampling a co-author in this paper uniformly. The natural setting for the measure weight μ is then the degree measure of the graph. The results of running the balanced version of `cm+improve` as λ decreases are displayed below. We verified our results against `METIS+HybridImprove`, which outputs essentially the same tripartitions.



(a) $\lambda = 1$. Edge-based partition.



(b) $\lambda = 1/10$. Small overlap.



(c) $\lambda = 1/300$. Large overlap.

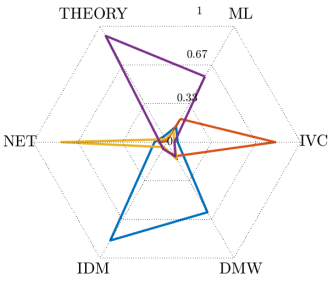
Figure 8: Spider plots showing the composition of the tripartition (L, C, R) output by `cm+improve`. For each set $X \in \{L, C, R\}$ of this partition and each subarea Y of Computer Science represent, we show the total edge volume of X coming from papers in area Y over the total edge volume of papers in Y.

The overlapping partitions found are very-well correlated with the edge-based partition capturing a clustering of subareas corresponding to a left cluster $\{\text{THEORY}, \text{NET}, \text{IDM}\}$ and a right cluster $\{\text{DMW}, \text{IVC}, \text{ML}\}$. As we saw above, this is expected as the real-world networks in our testbed do not appear to exhibit an overlapping balanced clustering that is different from the non-overlapping ones. As the first subfigure shows, a few papers from DMW and ML contribute to the left cluster. Indeed, as the overlap grows nodes with large degree in DMW and ML are the first to be included.

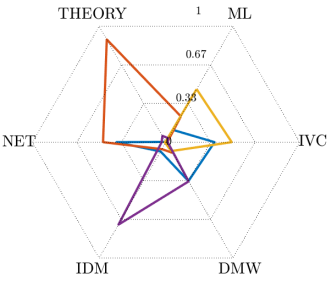
E.4. Recursive Overlapping Bisections

Our algorithm `cm+improve` can be used as a black box to **recursively bisection** the graph and identify multiple overlapping communities. When our algorithm is run without a balance constraint the successive cuts identify the "whiskers" around the core of the graph (Leskovec et al., 2009). For example, running recursive bisectioning on the Amazon dataset requires 52 cuts before a cut of significant volume is returned. All cuts up to that point have hundreds of nodes, meaning less than 1% of the graph.

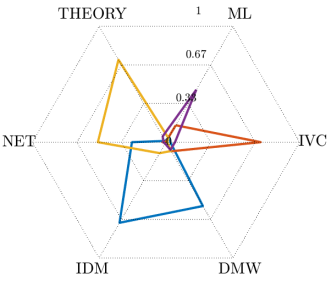
Balanced multicuts are of greater interest as they partition the graph in more interpretable parts. For example, in the DBLP co-authorship graph we can further refine the found communities and relate them more closely to specific areas.



(a) $\lambda = 1/2$. Edge-based partition.



(b) $\lambda = 1/10$. Small overlap.



(c) $\lambda = 1/300$. Large overlap.

Figure 9: Partitioning the DBLP co-authorship graph in four communities. The edge based partitions sharply correlate to specific computing areas. As the overlap increases communities become more rounded, losing their clear definition.