A GENERALIZED RATIO CUT OBJECTIVE IN GRAPHS AND EFFICIENT ALGORITHM FOR SOLVING THE LOCALIZED GENERALIZED EXPANSION RATIO PROBLEM

A Thesis

by

SIJING YU

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Nate Veldt
Committee Members,	Jianer Chen
	Tiandong Wang
Head of Department,	Scott Schaefer

August 2023

Major Subject: Computer Science

ABSTRACT

In graph clustering, ratio cut objectives represent the ratio between the connectivity of the subgraph and some notation of the graph properties including size and density. These ratio cut objectives are widely studied and used for many tasks including graph partitioning. Specifically, the standard expansion ratio measures the ratio between subgraph's connections to the rest of the graph and the subgraph size. This thesis introduces a generalized version of the standard expansion ratio objective and studies a localized variant of the expansion ratio problem by presenting its connections to existing problems and numerical results on existing problems.

The generalized version of the expansion ratio concerned in this thesis replaces the subgraph size with a convex function of the set size, generalizing more than one existing objective functions. The localized variant of the expansion ratio problem removes the constraint of subgraph size while restricting the resulting subgraph to a given seed set of nodes. While the original expansion problem is NP-hard to solve, this thesis introduces a polynomial-time algorithm for the novel localized variant of the problem. By varying the convex function and tuning parameters, numerical experiments show solving this new problem with the generalized objective allows one implicitly control the size of the resulting subgraph.

ACKNOWLEDGMENTS

Foremost, I would like to express my deepest gratitude to my advisor Dr. Nate Veldt for his invaluable guidance and continuous patience throughout my master's study. Furthermore, his expertise, dedication, and commitment to academic excellence have constantly and greatly inspired me to push my boundaries and strive for intellectual growth.

I would like to thank Dr. Jianer Chen and Dr. Tiandong Wang, my committee members, for all their support throughout the defense. The completion of the whole process is impossible without their help.

I would also like to thank members of the graduate advising office, especially Karrie Bourquin. It is their support that helps me navigate challenging situations and provides important resources for my academic development.

Finally, I would like to express my heartfelt gratitude to my family for their unwavering love, support, and encouragement throughout my academic journey. This whole journey is impossible without their continued understanding and contributions.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Nate Veldt and Professor Jianer Chen of the Department of Computer Science and Professor Tiandong Wang of the Department of Statistics.

The theorems proved in Chapter 4 were conjectured and validated by Professor Nate Veldt. All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was partly supported by Professor Nate Veldt from Texas A&M University. There is no outside funding source to acknowledge for the research and completion of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

A graph is a mathematical structure of a pair $G = (V, E)$ where V is a set of elements named vertices or nodes, and E is a set of elements of paired vertices, capturing pairwise relationships between elements in V. A graph is weighted if there is a weight assigned to each edge $e \in E$, denoted by $G = (V, E, w)$ where $w : E \to \mathbb{R}_{\geq 0}$ assigns a non-negative weight to each of the edges.

Graph clustering is an important task in graph optimization, aiming to find subsets of nodes that are more well-connected to each other than the rest of the graph, characterized by more edges within the subset than the rest of the graph. As a fundamental task in graph-based data mining, graph clustering has many applications in various areas including data transformation, and biological networks with an example task of classifying gene expression data [1, 2].

One standard approach to graph clustering is to minimize a so-called *ratio cut* objective, which measures the ratio between edge cut for a set of nodes and some notion of the cluster's size. Common examples include graph expansion or graph conductance [3, 4]. However, in practice, NP-hardness is a common bottleneck in optimizing many objective functions including the expansion ratio problem, the normalized cut problem, and the conductance problem with details introduced in the next section [5, 6, 7]. Therefore, approximation techniques and important variants of the are proposed to approach the problem.

1.1 Spectral techniques

Spectral algorithms are widely used as the solution for these problems. The core idea of spectral methods is to find the eigenvectors of the matrix corresponding to the graph [8]. One commonly used objective in spectral clustering is the normalized cut [9].

There exists an approximation bound based on the second smallest eigenvalue, also defined as the Fiedler value, justifying the use of spectral methods [10, 6]. Many spectral methods are designed for specific tasks including graph clustering, and they are mostly based on the eigendecomposition of the graph's Laplacian matrices [11]. These methods can also be applied to address the converse problem of avoiding small cuts [12, 13, 14].

Other work includes methods based on spectral methods and convex optimization for the specific task of graph partitioning on random graphs [15]. While spectral algorithms work on the matrices, many of them suffer from obstacles when translating the result to a discrete output on the graph [6].

1.2 Approximation algorithms

There also exists a range of algorithms that are not spectral giving the approximate results. As optimizing conductance is a well-known NP-hard problem, approximation algorithms have been proposed based on linear programming [16, 17], semidefinite programming [18], and so-called cut-matching games [19, 20] as summarized in [2]. Specifically, Leighton and Rao introduced the first true approximation of $\mathcal{O}(\log n)$-approximations for the sparsest cut and graph conductance. A multicommodity flow-based linear programming relaxation is used [17] where the integrality gap is $\Omega(\log n)$. Therefore, novel strategies are needed for improving the approximation factors. A $\mathcal{O}(\sqrt{\log n})$-approximations for the sparsest cut, edge expansion, and graph conductance is proposed by Arora, Rao and Vazirani using semidefinite programming [18]. Many of the methods described above suffered from complexity implementation, thus more scalable methods are inspired and proposed [21] as mentioned by [2].

1.3 Local clustering

Considering the increasing sizes of graphs for clustering, problems of scalability encourage the development of local graph clustering algorithms that come with various theoretical guarantees [22]. Instead of outputting a global clustering from the complete graph, local clustering seeks to find the clusters around a given seed node or a given set of seed nodes [23]. Specifically, local graph algorithms are algorithms where the result is determined as the function of a local area of the graph instead of the complete graph. Therefore, the scalability issues are addressed by using time and memory resources only dependent on the size of the cluster returned instead of the entire graph [23].

Furthermore, some methods are strongly local meaning their runtime is only dependent on the size of the seed set instead of the entire graph. Many strongly-local methods have been developed for the task of graph clustering with satisfactory performances [20, 24].

In addition to theoretical guarantees for computational resources needed, local graph clustering algorithms have been shown to be useful for uncovering small-scale structures in large-scale graphs [25]. They are widely studied with applications in various areas [26].

1.4 Flow-based methods

Flow-based algorithms approach graph clustering by repeatedly solving minimum cut and maximum flow problems on the input graph. Many of these flow-based algorithms have guaranteed cut improvement for ratio cut objectives like conductance [27, 20, 28]. Furthermore, some of these algorithms are strongly local [24].

As many other algorithms can be cast to instances of network flow, flow-based algorithms achieve decent performances in many tasks including optimizing some ratio cut objectives.
Hochbaum shows a the solution to the relaxation of the expansion ratio problem can be found in polynomial time via computing a sequence of max-flow and min-cut problems [6].

1.5 Contributions

This thesis introduces a generalized version of the graph expansion objective. By replacing the term of subgraph size with a convex function of the subgraph size, the objective allows different extent of influences that the subgraph size can have on the result. Therefore, by changing the function, optimizing the objective allows implicitly controlling the size of the subgraph.

Considering the computational hardness of optimizing this new objective which follows from the original expansion ratio problem, this thesis introduces a localized variant of the corresponding optimization problem where the returning result is restricted to a given seed set of nodes.

Given the local variant of the problem, this thesis proposes polynomial-time flow-based algorithms based on properties induced by the convexity of the function in the objective and theoretical guarantees from techniques used in hypergraph cut problems. Dependent on the flow algorithm, theoretical guarantees are derived to prove that this localized problem is solvable in practice.

1.6 Experiments and code

In addition to theoretical results, experimental results are presented in Chapter 4. These numerical results mainly demonstrate (i) the effects of the introduced objective function in controlling the size of the resulting graph, and (ii) the connections between our introduced problem and previous optimization problems.

2. PRELIMINARIES AND RELATED WORK

This chapter presents the mathematical definitions and the technical background for optimization problems considered in this thesis. Starting with mathematical notations on graphs, this chapter then gives an overview of several important objective functions.

In addition, this chapter introduces previous work on optimizing the objectives and their applications, with the MQI algorithm for "quotient-style" objectives explained in detail. An important contribution of the thesis is to solve the proposed problem using gadget expansion techniques of hypergraph gadgets. Therefore, mathematical details of existing graph reduction techniques and theoretical properties for the hypergraph cut problem are also covered.

2.1 Graph basics

Let $G = (V, E, w)$ be a weighted directed graph with edge weight $w : E \to \mathbb{R}^+$, $|V| = n$ and $|E| = m$. For each directed edge $(i, j) \in E$, we assume weights $w_{ij} > 0$. An undirected edge (i, j) is interpreted as having two directed edges (i, j) and (j, i). Let $d_v = |N_v|$ denote the degree of a node v where N_v is the set of nodes that share an edge with v. Given a set of nodes $S \subseteq V$ and its complement set $\bar{S} = V \backslash S$, we have following definitions of *cut* and *volume*:

$$\begin{aligned}\textbf{cut}(S) &= \textbf{cut}(S, \bar{S}) = \sum_{u \in S, v \in \bar{S}} w_{uv} \\ \textbf{vol}(S) &= \sum_{v \in S} d_v\end{aligned}$$

Then the classic graph cut problem is to find the subset $S \subset V$ that minimizes the cut value, namely $\textbf{cut}(S)$. In addition, if we identify two special *terminal* nodes s and t in V, then the **minimum** s-t **cut** problem is to find the set of edges with minimum sum of edge weights such that there are no paths from s to t if the set is removed, formally denoted as

$$\min_{S \subset V} \textbf{cut}(S) \quad \text{subject to } s \in S, t \in \bar{S} \tag{2.1}$$

Note that the minimum s-t cut problem focuses on a directed graph and only aims to cut flows from s to t. Therefore, instead of any edges with two endpoints in S and $\bar{S}$, only edges crossing from S to $\bar{S}$ incur penalties.

2.2 Hyperedge and splitting functions

Let $\mathcal{H} = (V, \mathcal{E})$ denote a hypergraph where each hyperedge $e \in \mathcal{E}$ is a subset of nodes in V. For each hyperedge $e \in \mathcal{E}$, we associate a *splitting function* $\mathbf{w}_e : A \subseteq e \to \mathbb{R}_{\geq 0}$ for splitting e among two clusters, mapping subsets $A \subseteq e$ to a nonnegative penalty. For the purpose of this thesis, only one single hyperedge e is considered instead of an entire hypergraph, and we have following properties regarding a hyperedge splitting function.

Definition 1. *For any $A, B \subseteq e$, the function $\boldsymbol{w}_e$ is*
(1) *submodular if $\boldsymbol{w}_e(A) + \boldsymbol{w}_e(B) \geq \boldsymbol{w}_e(A \cap B) + \boldsymbol{w}_e(A \cup B)$;*
(2) *cardinality-based if $\boldsymbol{w}_e(A) = \boldsymbol{w}_e(B)$* when $|A| = |B|$.

Then we have the following observation, following the definition of the concavity of functions.

Observation 1. *The splitting function $\boldsymbol{w}_e : A \subseteq e \to \mathbb{R}_{\geq 0}$ is submodular if it can be expressed as a concave function $g : [0, |e|] \to \mathbb{R}$ such that $\boldsymbol{w}_e(A) = g(|A|)$.*

This follows directly from the concavity of the function, and it is also easy to observe that such $\mathbf{w}_e$ is also cardinality-based by $\mathbf{w}_e(A) = \mathbf{w}_e(B) = g(|A|) = g(|B|)$ when $|A| = |B|$.

2.2.1 Hypergraph cut

Analogous to the cut of simple graphs, based on the splitting function for each hyperedge, the hypergraph cut for a subset $S \subset V$ is defined as

$$\mathbf{cut}_{\mathcal{H}}(S) = \sum_{e \in \mathcal{E}} \mathbf{w}_e(e \cap S).$$

Since only one hyperedge e is involved in the focus of this thesis, the corresponding minimum s-t cut problem for this hypergraph of one hyperedge is denoted as

$$\min_{S\subset V} \mathbf{cut}_{\mathcal{H}}(S) = \min_{S\subset V} \sum_{e\in\mathcal{E}} \mathbf{w}_e(e\cap S) = \min_{S\subset V} \mathbf{w}_e(S) \quad \text{subject to } s\in S, t\in \bar{S} \tag{2.2}$$

where s and t are assigned source and sink nodes, similar to the minimum s-t problem on simple graphs. It is proved that this problem can be solved for any submodular cardinality-based splitting function via reduction to a graph s-t cut problem [29].

2.3 Ratio cut objective functions

As an important category of objective functions used in graph clustering, ratio cut objective functions have been widely studied. This thesis will mainly focus on the expansion ratio (with its optimization problem) and several other closely related objective functions.

2.3.1 Expansion ratio

For a given graph G, the expansion ratio is a measure of the connectivity of the entire graph, given by finding the set minimizing the expansion ratio as defined below. It is also called the edge expansion and is equivalent to the Cheeger constant when .

$$\min_{S\subseteq V, |S|\leq \frac{|V|}{2}} \frac{\mathbf{cut}(S)}{|S|} \tag{2.3}$$

2.3.2 Uniform sparsest cut

For a given graph G, the optimization of the uniform sparsest cut objective is defined as

$$\min_{S\subseteq V} \frac{\mathbf{cut}(S)}{|S||\bar{S}|} \tag{2.4}$$

Since $\frac{|V|}{2} \leq |\bar{S}| \leq |V|$, optimizing the sparest cut is the same as solving the expansion ratio problem on a graph up to a factor of 2.

2.3.3 Minimum bisection

A bisection $(S, \bar{S})$ for a given graph G is a partition of nodes into two sets S and $\bar{S}$ such that their sizes differ by at most one. Then the minimum bisection problem is defined as

$$\min_{S \subseteq V,\ \ |(|S| - |\bar{S}|)| \leq 1} \mathbf{cut}(S) \tag{2.5}$$

This problem can be proved to share a set of optimal solutions to the new problem this thesis introduces with details in the next section.

2.3.4 Conductance

Defined as the ratio between the number of connections between the subgraph to the rest of the graph and the minimum volume of S and $\bar{S}$, the conductance is an important objective in graph optimization:

$$\phi(S) = \frac{\mathbf{cut}(S)}{\min(\mathbf{vol}(S), \mathbf{vol}(\bar{S}))} \tag{2.6}$$

Naturally, the corresponding optimization problem is to minimize the measure. A set of optimal conductance can be interpreted as a basic bottleneck when exploring the community structure of the graph [2]. As the conductance values must fall between 0 and 1, another way to interpret them is to treat them as probabilities [2].

2.3.5 Normalized cut

This objective is related to the conductance in the way that it differs by at most a factor of two from conductance [30]

$$\frac{\mathbf{cut}(S)}{\mathbf{vol}(S)} + \frac{\mathbf{cut}(S)}{\mathbf{vol}(\bar{S})} = \frac{\mathbf{cut}(S)(\mathbf{vol}(S) + \mathbf{vol}(\bar{S}))}{\mathbf{vol}(S)\mathbf{vol}(\bar{S})} = \mathbf{vol}(V)\frac{\mathbf{cut}(S)}{\mathbf{vol}(S)\mathbf{vol}(\bar{S})} \tag{2.7}$$

Both the conductance and the normalized cut are NP-hard to minimize [31] on the entire graph but the localized variants can be minimized in polynomial-time by repeatedly solving maximum flow problem [27, 12, 20, 24, 28].

2.3.6 Size-normalized cut

This quantity can be used to bound the expansion ratio of the graph, and can be viewed as an unweighted normalization version of the normalized cut problem where the weight of all nodes is set to one, resulting in the name of "size normalized". In addition, we can see that the uniform sparsest cut problem has the same set of optimal solutions as the size-normalized problem defined as

$$\min_{S \subseteq V} \frac{\textbf{cut}(S)}{|S|} + \frac{\textbf{cut}(S)}{|\bar{S}|} \tag{2.8}$$

by observing that

$$\frac{\textbf{cut}(S)}{|S|} + \frac{\textbf{cut}(S)}{|\bar{S}|} = \frac{\textbf{cut}(S)(|S| + |\bar{S}|)}{|S||\bar{S}|} = |V|\frac{\textbf{cut}(S)}{|S||\bar{S}|} \quad \text{where } |V| = n \text{ is a constant.}$$

2.4 Hyperedge reduction techniques

A cardinality-based splitting function $\mathbf{w}_e$ on an k-node hyperedge involved in this thesis can be characterized by $k-1$ penalty scores w_i for $i \in \{1, 2, ..., k-1\}$ where w_i is the penalty of having i nodes on the source-side of the cut [29]. Since this thesis only considers a single hyperedge, with a source node s and a sink node t being introduced, the original hypergraph s-t problem discussed by Veldt [29] is presented in (2.2).

In order to reduce the problem to a graph problem where flow-based algorithms can be applied, an asymmetric cardinality-based gadget (ACB-gadget) is used and constructed as below, following Definition B.2. from [32]

- For each hyperedge e, introduce a new auxiliary node v_e,
- For each $v \in e$, add directed edges (v, v_e) with weight $a \cdot (k - b)$ and (v_e, v) with weight $a \cdot b$ respectively.

Then this ACB-gadget models the following splitting function

$$\mathbf{w}_{a,b}(S) = a \cdot \min\{i \cdot (k - b), (k - i) \cdot b\},$$

where k is the number of nodes in hyperedge and i is the number of nodes on the source-side of the cut. The theoretical guarantee of applying graph reduction techniques in this thesis is based on Theorem 4.8 in [29], where we only illustrate the case concerned in this thesis.

Theorem 1. *For a cardinality-based splitting function $\mathbf{w}_e$, the problem presented in (2.2) is graph reducible if $\mathbf{w}_e$ is submodular.*

Proof. A problem is *graph reducible* if can be modeled by some splitting function, Theorem 4.8 in [29] states that the instance of asymmetric cardinality-based HYPER-ST-CUT is graph reducible if every splitting function is submodular. For cardinality-based $\mathbf{w}_e$, the problem (2.2) is an instance of cardinality-based HYPER-ST-CUT when the entire hypergraph only contains a single hyperedge e. Therefore, for $\mathbf{w}_e(S) = g(|S|)$ where $g : [0, |e|] \to \mathbb{R}$ is concave, we have $\mathbf{w}_e$ being both submodular and cardinality-based as the only splitting function. As a result, problem (2.2) is graph reducible and can be modeled by the ACB-gadgets presented above. $\square$

2.5 Related work

Minimizing the connections of the subgraph to the rest of the graph while some notion of the graph size is expected to be maximized, optimization problems with the objectives above are desired for tasks like graph clustering and partitioning in nature. This subsection will discuss some of the existing work on approximating or relaxing objective functions discussed above, as well as applications like graph partitioning.

Measures including conductance, expansion ratio, sparsest and normalized cut are frequently used in graph clustering and partitioning, being approximation reducible within a constant factor [17]. Leighton and Rao introduced the first true approximation of $\mathcal{O}(\log n)$-approximations for the sparsest cut and graph conductance. A multicommodity flow-based linear programming relaxation is used [17] where the integrality gap is $\Omega(\log n)$. Therefore, novel strategies are needed for improving the approximation factors. A $\mathcal{O}(\sqrt{\log n})$-approximations for the sparsest cut, edge expansion, and graph conductance is proposed Arora, Rao and Vazirani using semidefinite programming [18]. Algorithms based on so-called cut-matching games have also been proposed to optimize conductance [20]. Many of the methods described above suffered from complexity implementation, thus more scalable methods are inspired and proposed [21].

2.5.1 MaxFlow Quotient-Cut Improvement (MQI) problem and algorithm

Related to the problem studied in this thesis and specified by Lang and Rao, the MQI problem seeks to improve the quality of graph clustering quantified by quotient-style graph objectives using flow-based algorithms [12, 2]. Given an input graph $G = (V, E)$ and a seed set $R \subset V$ with the constraint $\mathbf{vol}(R) \leq \frac{\mathbf{vol}(G)}{2}$, the algorithm returns a "better" cluster represented by minimized conductance. To illustrate, the basic MQI problem is

$$\min_{S \subseteq R} \frac{\mathbf{cut}(S)}{\mathbf{vol}(S)} \tag{2.9}$$

With the constraint $\mathbf{vol}(R) \leq \frac{\mathbf{vol}(G)}{2}$, the above problem can be represented as

$$\min_{S \subseteq R} \ \phi(S) \tag{2.10}$$

where $\phi(S)$ is the conductance score defined in equation **2.6**. The constraint $S \subseteq R$ with $\mathbf{vol}(R) \leq \frac{\mathbf{vol}(G)}{2}$ is important for the conductance problem to be solvable in polynomial time [2]. In fact, it is NP-hard to even find a set S with $\mathbf{vol}(S) \leq \frac{\mathbf{vol}(G)}{2}$ where the set of the minimum conductance is contained. Lang and Rao propose the following algorithm targeting the MQI problem in 2.6 that the objective decreases monotonically at each iteration.

The convergence of the algorithm is given in the original paper as a corollary of Theorem 3.4 where δ_i denotes the value objective function evaluated at S_i.

Theorem 2. *[12, 2] Given an undirected, connected graph with nonnegative weights G and a subgraph for the seed set R where $\mathbf{vol}(R) \leq \mathbf{vol}(\bar{R})$. The sequence δ_k decreases monotonically at each iteration of MQI.*

Since this "improvement" of the partition is guaranteed for each iteration from the above theorem, the maximally balanced partition given by Metis [33], a fast implementation of multi-

Algorithm 1 MQI (Lang and Rao, 2004)

Require: : G, R
  $k := 1$
  $S_1 := R$
  $\delta_1 := \phi(S_1)$.
  while not returning **do**
    Solve $S_{k+1} :=$ argmin$_{S \subseteq R}$**cut**$(S) - \delta_k$**vol**(S)
    If $\phi(S_{k+1}) < \delta_k$ then
      $\delta_{k+1} := \phi(S_{k+1})$
    else
      δ_k is optimal, **return** S_k.
    $k := k + 1$
  end while

resolution Fiduccia-Mattheyses [34] is a desired starting point for balanced cuts as the input for the MQI algorithm. In addition, this algorithm is introduced due to it being an important inspiration of the algorithms presented in this thesis.

3. THE GENERALIZED EXPANSION RATIO AND SPARSEST CUT

Inspired by the original expansion ratio presented in section 2.3, we introduce a new generalized objective and the corresponding optimization problem.

Definition 2. *Given $G = (V, E)$, the* ***generalized expansion ratio*** *problem is defined as*

$$\min_{|S| \leq \frac{|V|}{2}} \frac{\boldsymbol{cut}(S)}{g(|S|)} \tag{3.1}$$

where $g : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a monotonically non-decreasing convex function, and the ratio to be minimized is naturally the generalized expansion ratio.

By the definition above, we have the following observation naturally.

Observation 2. *There exists an instance of the generalized graph expansion ratio problem equivalent to the original graph expansion ratio in* (2.1).

This equivalence is easy to be observed by having $g(|S|) = |S|$. For the purpose of simplicity and variability, this thesis mainly considers instances of $g(|S|) = |S|^\alpha$ where $\alpha \in [1, \infty)$, satisfying g being a monotonically non-decreasing convex function by having a positive second derivative.

In addition, the hardness of the generalized expansion ratio problem follows from having the original expansion ratio problem which is NP-hard as an instance [35, 36]. A related hardness result can also be observed as follow.

Observation 3. *Without constraints $|S| \leq \frac{|V|}{2}$ and g being convex, the hardness of the generalized expansion ratio problem follows by having the (uniform) sparsest cut problem (2.4) as a special case. This can be seen from having $g(|S|) = |S|(|V| - |S|) = |S||\bar{S}|$ where $|S|(|V| - |S|)$ is a concave function with respect to $|S|$ with fixed $|V|$.*

Analogous to the generalized expansion ratio, we also introduce a new generalized version of the sparsest cut objective and establish its connection to other problems presented in section 2.3.

Definition 3. *Let $G = (V, E)$, for any subset $S \subseteq V$, the* ***generalized sparsest cut*** *problem is defined as*

$$\min_{S \subseteq V} \frac{\textbf{\textit{cut}}(S)}{g(|S|)} + \frac{\textbf{\textit{cut}}(S)}{g(|\bar{S}|)} \tag{3.2}$$

where $g : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a monotonically non-decreasing convex function.

Similarly, we can have the following observation having $g(|S|) = |S|$.

Observation 4. *There exists an instance of the generalized sparsest cut problem equivalent to the (uniform) sparsest cut problem, thus the size-normalized cut problem presented in* (2.4) *and* (2.8).

3.1 Connections to existing optimization problems

While the optimization problems above generalize the existing expansion and sparsest cut problems in the nature of their definitions, they are also related to other optimization problems. Recall the minimum bisection problem in (2.5) as $\min\limits_{S \subseteq V,\ |(|S| - |\bar{S}|)| \leq 1} \textbf{cut}(S)$, we can derive the following connection between the minimum bisection problem and the generalized sparsest cut problem for $g(|S|) = |S|^\alpha$.

Theorem 3. *Given a graph $G = (V, E)$ where $|V|$ is even so $|S| = |\bar{S}|$ for some $S \subset V$, there exists some $p \in \mathbb{R}_{\geq 1}$ such that $\forall \lambda \geq p$, $S^* = \operatorname*{argmin}\limits_{S \subseteq V, |S| = |\bar{S}|} \textbf{cut}(S)$ is the optimal solution to* (3.2) *when $g(|S|) = |S|^p$.*

Proof. Let $|V| = n$, we have $|S^*| = |\bar{S^*}| = \frac{n}{2}$. For any $S \subseteq V$, let $\mu = \frac{\textbf{cut}(S^*)}{\textbf{cut}(S)}$, $|S| = x|V| = xn$, then $x \in [0, 1]$, and we have

$$\frac{\textbf{cut}(S^*)}{g(|S^*|)} + \frac{\textbf{cut}(S^*)}{g(|\bar{S^*}|)} = \frac{2\,\textbf{cut}(S^*)}{(\frac{n}{2})^\alpha} = \frac{2^{\alpha+1}\,\textbf{cut}(S^*)}{n^\alpha}$$

First, we can derive that $h(x) = \frac{1}{x^\alpha} + \frac{1}{(1-x)^\alpha}$ is convex for $\alpha \geq 1$, and $x \in (0, 1)$ with a minimum at $x = 0.5$ by observing $h''(x) > 0$, and $h'(x) = 0$ if only if $x = 0.5$.

Since $h(0.5) = 2^{\alpha+1}$, we have $h(x) > 2^{\alpha+1}$ for $x \in (0, 0.5) \cup (0.5, 1)$, $\alpha \geq 1$. Then we can compute that

$$\lim_{\alpha \to \infty} \frac{1}{2^{\alpha+1}} \left(\frac{1}{x^\alpha} + \frac{1}{(1-x)^\alpha}\right) = \infty,\ x \in (0, 0.5) \cup (0.5, 1) \tag{3.3}$$

By the formal definition of limit, we have that for any μ, there exists a $\delta > 0$ such that for all $\alpha \geq 1, p \neq \alpha$,

$$0 < |\alpha - p| < \delta \implies \frac{1}{2^{\alpha+1}}(\frac{1}{x^\alpha} + \frac{1}{(1-x)^\alpha}) > \mu$$

for $x \in (0, 0.5) \cup (0.5, 1)$. Therefore, for any μ defined above, there must exist a p such that $\mu\, 2^{p+1} \leq \frac{1}{x^p} + \frac{1}{(1-x)^p}$. This implies that for subset $S \subseteq V$, $x \in (0, 0.5) \cup (0.5, 1)$, there exists a $p \geq 1$ such that

$$\mathbf{cut}(S^*)2^{p+1} \leq (\frac{1}{x^p} + \frac{1}{(1-x)^p})\mathbf{cut}(S) \tag{3.4}$$

$$\implies \frac{2\,\mathbf{cut}(S^*)2^p}{n^p} \leq \frac{\mathbf{cut}(S)}{(nx)^p} + \frac{\mathbf{cut}(S)}{(n(1-x))^p} \tag{3.5}$$

$$\implies \frac{2\,\mathbf{cut}(S^*)}{|S^*|^p} \leq \frac{\mathbf{cut}(S)}{|S|^p} + \frac{\mathbf{cut}(S)}{|\bar{S}|^p} \tag{3.6}$$

By the limit obtained in (3.5), we have (3.6) holds for any $\lambda \geq p$. Therefore, there exists a $p \geq 1$ such that for all $S \subseteq V$,

$$\frac{2\,\mathbf{cut}(S^*)}{|S^*|^p} \leq \frac{\mathbf{cut}(S)}{|S|^p} + \frac{\mathbf{cut}(\mathrm{S})}{|\bar{S}|^p} \text{ , and}$$

$$\forall\lambda \geq p, \quad \frac{2\,\mathbf{cut}(S^*)}{|S^*|^\lambda} \leq \frac{\mathbf{cut}(S)}{|S|^\lambda} + \frac{\mathbf{cut}(\mathrm{S})}{|\bar{S}|^\lambda}.$$

The theorem establishes that for large enough α in the generalized sparsest cut problem with $g(|S|) = |S|^\alpha$shares the optimal solution to the minimum bisection problem. Therefore, in practice, the minimum bisection problem may be solved by approaching the generalized sparsest cut problem for $g(|S|) = |S|^\alpha$ with properly chosen α's. This is also shown in the experimental results in Chapter 5.

□

4. THE LOCAL GENERALIZED EXPANSION RATIO PROBLEM AND ALGORITHMS

Based on the generalized expansion ratio problem above, we then introduce a localized version of the generalized expansion ratio problem with $g : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ being convex, i.e. $g(x) - 2g(x + 1) + g(x + 2) \leq 0$ holds for any $x \in \mathbb{Z}_{\geq 0}$, and propose two polynomial-time algorithms.

Definition 4. *Given a graph $G = (V, E, w)$ and a seed set of nodes $R \subset V$, the* ***local generalized expansion ratio*** *problem is defined by*

$$\min_{S \subseteq R} \frac{\boldsymbol{cut}(S)}{g(|S|)} \tag{4.1}$$

where $g : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a monotonically non-decreasing convex function.

While the original generalized expansion ratio problem suffers from computational hardness, we propose two polynomial-time algorithms for solving this localized variant based on repeatedly applying flow-based algorithms.

4.1 Algorithm I

This algorithm mainly uses the properties of g being a convex function. The cut value of a set in the auxiliary graph constructed below gives the value of a variant of the objective function. Therefore, by solving the minimum s-t cut problem on the resulting graph, the subset returned minimizes the objective function. First, we will introduce the graph construction and theoretical guarantees on the entire graph, and then move to the localized version with modifications in implementation.

4.1.1 Main idea

For given $G = (V, E)$ and a seed set $R \subset V$ where $|R| = r$, inspired by Kawase and Miyauchi on the densest subgraph problem [37], we construct the auxiliary graph (U, A, w_ζ) as follows:

- Introduce source node s and sink node t

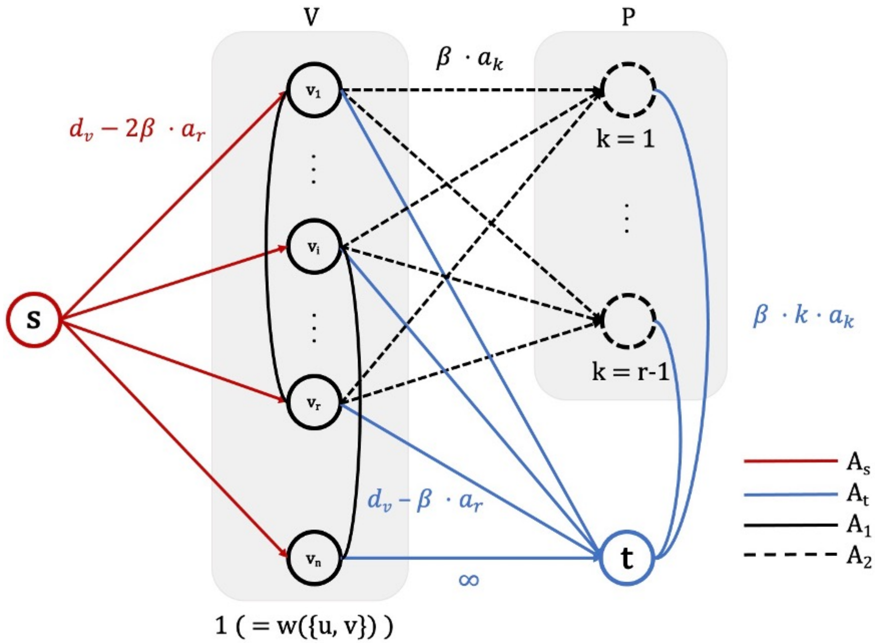


Figure 4.1: Auxiliary Graph

- Let $U = V \cup P \cup \{s, t\}$ where $P = \{p_1, p_2, ..., p_{r-1}\}$

- Let $A = A_s \cup A_t \cup A_1 \cup A_2$ where

  $A_s = \{(s, v) | \ v \in V\}$,

  $A_t = \{(p, t) | \ p \in P\}$

  $A_1 = \{(u, v) | \ \{u, v\} \in E\}$,

  and $A_2 = \{(v, p) | \ v \in V, p \in P\}$.

- Define the edge weights $w_\zeta : A \to \mathbb{R}_{\geq 0}$ as

$$w_\zeta(e) = \begin{cases} d(v) - 2\beta \cdot a_r & (e = (s, v) \in A_s) \\ 1 \ (= w(\{u, v\})) & (e = (u, v) \in A_1) \\ \beta \cdot a_k & (e = (v, p_k) \in A_2 \text{ for } k \in [1, n-1]) \\ \beta \cdot k \cdot a_k & (e = (p_k, t) \in A_t \text{ for } k \in [1, n-1]) \\ -\beta \cdot a_r + d(v) & (e = (v, t) \in A_t) \end{cases}$$

where a_k = $\begin{cases} g(k+1)+g(k-1)-2g(k) & k=1,...,r-2,r-1 \\ g(n-r-1)-g(n-r) & k=r \end{cases}$

Thus $a_k \geq 0$ for $k = 1,...,r$ because g is non-decreasing and convex We have the following lemma.

Lemma 4. *Let (X,Y) be any minimum s-t cut in the network (U,A,w_ζ) such that $X \cup Y = U$ and $X \cap Y = \emptyset$, and let $S = X \cap V$. Then, the cost of (X,Y) is equal to* $\mathbf{cut}(S) + \mathbf{vol}(V) - \beta \cdot g(|S|) - 2\beta|V| \cdot a_r + \beta g(0)$.

Proof. Similar to [37], we first show that for any positive integer $s \leq n-r$, it holds that

$$\sum_{i=1}^{r} min\{i,s\} \cdot a_i = g(0) - g(s).$$

By the definition of a_k, we get

$$\begin{aligned}\sum_{j=i}^{r} a_j &= g(r-1) - g(r) + \sum_{j=i}^{r-1}(g(j+1)+g(j-1)-2g(j)) \\ &= g(r-1) - g(r) - \sum_{j=i}^{r-1}((g(j)-g(j+1)) + (g(j)-g(j-1))) \\ &= g(i-1) - g(i)\end{aligned}$$

Thus, we have

$$\sum_{i=1}^{r} min\{i,s\} \cdot a_i = \sum_{i=1}^{s}\sum_{j=i}^{r} a_j = \sum_{i=1}^{s}(g(i-1)-g(i)) = g(0) - g(s).$$

We can observe that when $|S| > k$, $p_k \in X$ and $p_k \in Y$ if $|S| < k$. Consequently, the cost of the minimum cut (X, Y) is

$$
\begin{aligned}
&\sum_{v \in V \setminus S} (d(v) - 2\beta \cdot a_r) + \sum_{\{u,v\} \in E: u \in S, v \in V \setminus S} w(\{u,v\}) + \beta \cdot \sum_{i=1}^{r-1} min\ \{i, |S|\} \cdot a_i + \sum_{v \in S} (-\beta \cdot a_r + d(v)) \\
= \ & \mathbf{vol}(V \setminus S) - |V \setminus S| \cdot 2\beta \cdot a_r + \mathbf{cut}(S) + \beta \cdot \sum_{i=1}^{r-1} min\ \{i, |S|\} \cdot a_i - \beta \cdot |S| \cdot a_r + \mathbf{vol}(S) \\
= \ & \mathbf{vol}(V \setminus S) + \mathbf{vol}(S) + \mathbf{cut}(S) - |V \setminus S| \cdot 2\beta \cdot a_r + \\
& (\beta \cdot \sum_{i=1}^{r-1} min\ \{i, |S|\} \cdot a_i + \beta \cdot |S| \cdot a_r) - 2\beta \cdot |S| \cdot a_r \\
= \ & \mathbf{vol}(V) + \mathbf{cut}(S) - 2\beta |V| \cdot a_r + \beta \cdot \sum_{i=1}^{r} min\ \{i, |S|\} \cdot a_i \\
= \ & \mathbf{cut}(S) + \mathbf{vol}(V) - \beta \cdot g(|S|) - 2\beta |V| \cdot a_r + \beta g(0)
\end{aligned}
$$

□

We observe that

$$
\mathbf{vol}(V) + \mathbf{cut}(S) - 2\beta |V| \cdot a_r - \beta g(s) + \beta g(0) \leq \mathbf{vol}(V) - 2\beta |V| \cdot a_r + \beta g(0)
$$

if and only if $\mathbf{cut}(S) - \beta g(s) \leq 0$ which implies $\frac{\textbf{cut(S)}}{g(|S|)} \leq \beta$. Therefore, since $\mathbf{vol}(V) - 2\beta |V| \cdot a_r + \beta g(0)$ is a constant, for a given β, by solving the minimum s-t cut problem on the auxiliary graph constructed above, we can conclude if there exists a set $S \subset V$ satisfying $\frac{\textbf{cut(S)}}{g(|S|)} \leq \beta$. By repeatedly solving the minimum s-t cut problem on the graph with decreasing β, the objective can be minimized. However, the main obstacle in practice is that the algorithm will always return the entire graph since the cut value is zero. This is also part of the motivation for the localized version of the problem where we propose practical algorithms with applications.

4.1.2 Localized algorithm

Based on the graph construction above, in order to guarantee the resulting subset as a subset of seed set R, we make sure to exclude nodes in $\bar{R}$. To exclude $\bar{R}$, in implementation, the easiest way is to put infinite weight for every (v, t) where $v \in \bar{R}$. However, this method may incur computational problems in practice due to infinite weight. Therefore, we merge all nodes in $\bar{R}$ with t.

4.1.2.1 Localization

Given (U, A, w_ζ) constructed above, we merge all nodes in $\bar{R}$ with t by replacing $\bar{R} \cup t$ with a "supernode" t' where for every $u \in R$ the weight of (u, t) becomes $\sum_{v \in \bar{R}} w_{r,u}$. Therefore, (U, A, w_ζ) becomes (U', A', w'_ζ) where $U' = R \cup P \cup \{s, t\}$ and $A' = \{(u, v) | (u, v) \in A$ and u, v cannot both belong to $\bar{R}\}$. The correctness of this idea for solving (3.1) follows from (i) nodes from $\bar{R}$ cannot be returned in the result after being merged (ii) the cost of any cut of the set returned still follows Lemma 3 since the total weights from any $v \in R$ to $\bar{R}$ remains the same. Therefore, solving the minimum s-t cut problem with decreasing β still works to minimize the objective with the local constraint. The detailed algorithm for the localized problem is as follows.

Algorithm 2 Minimizing generalized expansion ratio – Algorithm I

Require: : G, R
  $G \leftarrow$ constructed as described above as (U', A', w'_ζ)
  $\beta := \infty$
  $\beta_{new} \leftarrow \frac{\mathbf{cut}(R)}{g(|R|)}$
  $S = R$
  while $\beta_{new} < \beta$ **do**
    $S_{best} \leftarrow S$
    $\beta \leftarrow \beta_{new}$
    $S \leftarrow$ argmin $\mathbf{cut}(S) - \beta g(|S|)$
    $\beta_{new} \leftarrow \frac{\mathbf{cut}(S)}{g(|S|)}$
  end while
  return S_{best}

A bound on the number of iterations for Algorithm 2 is derived below by slightly adapting techniques used by Anderson and Lang [27, 24].

Theorem 5. *Algorithm 2 will need to solve the min-cut objective at most* ***cut****(R) times.*

Proof. Let $f_\beta(S) = \textbf{cut}(S) - \beta g(|S|)$, then consider two consecutive iterations in which Algorithm 2 successfully outputs sets with improved generalized expansion ratio. Let S_i denote the set returned after the $(i-1)$st iteration, so $S_i = \text{argmin } f_{\beta_{i-1}}(S)$ for some $\beta_{i-1} = \frac{\textbf{cut}(S_{i-1})}{g(|S_{i-1}|)}$ and set $\frac{\textbf{cut}(S)}{g(|S_{i+1}|)} = \frac{\textbf{cut}(S_i)}{g(|S_i|)} < \beta_{i-1}$. Similarly we have $S_{i+1} = \text{argmin } f_{\beta_i}(S)$ and $\frac{\textbf{cut}(S)}{g(|S|_{i+1})} = \frac{\textbf{cut}(S_{i+1})}{g|S_{i+1}|} < \beta_i$. Then we have

$$
\begin{aligned}
f_{\beta_{i-1}}(S_i) &= \textbf{cut}(S_i) - \beta_{i-1} g(|S_i|) + g(|V|) \\
&= \beta_i g(|S_i|) - \beta_{i-1} g(|S_i|) + g(|V|) \\
&= g(|S_i|)(\beta_i - \beta_{i-1}) + g(|V|)
\end{aligned}
$$

and similarly

$$
f_{\beta_{i-1}}(S_{i+1}) = g(|S_{i+1}|)(\beta_{i+1} - \beta_{i-1}) + g(|V|)
$$

Since S_i minimizes $f_{\beta_{i-1}}$ we have $f_{\beta_{i-1}}(S_i) \leq f_{\beta_{i-1}}(S_{i+1})$, and this implies

$$
g(|S_i|)(\beta_i - \beta_{i-1}) \leq g(|S_{i+1}|)(\beta_{i+1} - \beta_{i-1}).
$$

Since $(\beta_{i+1} - \beta_{i-1}) < (\beta_i - \beta_{i-1}) < 0$, we have $g(|S_{i+1}|) < g(|S_i|)$. Therefore, both $\frac{\textbf{cut}(S)}{g(|S|)}$ and $g(|S|)$ are strictly decreasing between non-terminating iterations of the algorithm. Consequently, $\textbf{cut}(R)$ must strictly decrease at each iteration correspondingly. Considering the graph is unweighted, there are at most $\textbf{cut}(R)$ iterations in the algorithm. □

4.1.3 Runtime

A runtime bound dependent on the number of nodes and edges can also be given based on different max-flow algorithms. Let $T_{mf}(N, M)$ denote the time to solve a max-flow problem with N nodes and M edges. Then, given $G = (V, E)$ with $|V| = n$, $|M| = m$ and a seed set of nodes R such that $|R| = r$, the final auxiliary graph is (U', A', w'_ζ) where $U' = R \cup P \cup \{s, t\}$ and

$A' = \{(u,v)|(u,v) \in A$ and u, v cannot both belong to $\bar{R}\}$. Recall $P = \{p_1, p_2, ..., p_{r-1}\}$ and $A = A_s \cup A_t \cup A_1 \cup A_2$ with $A_s = \{(s,v)|\ v \in V\}$, $A_t = \{(p,t)|\ p \in P\}$, $A_1 = \{(u,v)|\ \{u,v\} \in E\}$, and $A_2 = \{(v,p)|\ v \in V, p \in P\}$. Therefore, there are $|U'| = r + r - 1 + 2 = 2r + 1$ nodes and at most $|A| \le r + m + r(r-1) + (2r-1) = r^2 + 2r + m - 1$ edges. Therefore, the time required to solve the max-flow problem in each iteration is bounded by $T_{mf}(2r+1, r^2+2r+m-1)$. A recent $\tilde{O}(M + N^{1.5})$-time algorithm for the maximum s-t flow problem by van den Brand et al. would generate a runtime guarantee of $\tilde{O}(r^2 + 2r + m - 1 + (2r+1)^{1.5})$ [38].

Based on the flow-based algorithm, we then propose another algorithm with faster runtime using techniques of gadget reduction.

4.2 Algorithm II

We then introduce the second flow-based algorithm with improved runtime, sharing the same idea of repeatedly solving the minimum s-t cut problem on an auxiliary graph with decreasing β from **cut**$(S) - \beta g(|S|) + c$ with constant c.

4.2.1 Main idea

The auxiliary graph is constructed by combining two parts accounting for two terms respectively, namely **cut**(S) and $-\beta g(|S|)$.

Lemma 6. *Given $G_1 = (V, E_1, w_1)$, $G_2 = (V, E_2, w_2)$, let $G_3 = (V, E_3, w_3)$ where $E_3 = E_1 \cup E_2$ and $\forall (i,j) \in E_1 \cup E_2, w_3(i,j)) = w_1(i,j) + w_2(i,j)$, if $S \subseteq V$ induces cut costs of c_1 and c_2 on G_1 and G_2 respectively, S induces a cut cost of $c_1 + c_2$ on G_3.*

Proof. By the definition of cut, for any subset $S \subset V$, we have $c_1 = \underset{G_1}{\mathbf{cut}}(S) = \sum_{i \in S, j \in \bar{S}} w_1(i,j)$

and $c_2 = \underset{G_2}{\mathbf{cut}}(S) = \sum_{i \in S, j \in \bar{S}} w_2(i,j)$. Therefore,

$$\begin{aligned}
\underset{G_3}{\mathbf{cut}}(S) &= \sum_{i \in S, j \in \bar{S}} w_3(i,j) \\
&= \sum_{i \in S, j \in \bar{S}} (w_1(i,j) + w_2(i,j)) \\
&= \sum_{i \in S, j \in \bar{S}} w_1(i,j) + \sum_{i \in S, j \in \bar{S}} w_2(i,j) = \underset{G_1}{\mathbf{cut}}(S) + \underset{G_2}{\mathbf{cut}}(S) = c_1 + c_2.
\end{aligned}$$

□

Therefore, based on the above lemma, $c_1 + c_2$ can be minimized by solving min-cut problem on G_3 where G_3 is constructed as above for given G_1 and G_2 with $c_1 = \underset{G_1}{\mathbf{cut}}(S)$ and $c_2 = \underset{G_2}{\mathbf{cut}}(S)$. Having the input graph G as G_1 gives $c_1 = \underset{G_1}{\mathbf{cut}}(S) = \mathbf{cut}(S)$. Then if we could construct a desired G_2 satisfying $c_2 = \underset{G_2}{\mathbf{cut}}(S) = c - \beta g(|S|)$, then combining two graphs as described above into G_3 gives a desired graph reduction for the objective $\mathbf{cut}(S) - \beta g(|S|) + c$. First, we establish the correctness of this idea of graph reduction for the term $c - \beta g(|S|)$.

Lemma 7. *Given a graph $G = (V, E)$, there exists a graph $G' = (V', E')$ where for any $S \subset V$, the cost of cut induced by S on G' is equivalent to $c - \beta g(|S|)$, namely $\underset{G'}{\mathbf{cut}}(S) = c - \beta g(|S|)$.*

Proof. Since g is convex, for any constant c and $\beta \geq 0$, $h(|S|) = c - \beta g(|S|)$ is concave where $h : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$. By Observation 1, $\mathbf{w}_e(S) = h(|S|)$ is a submodular function where $\mathbf{w}_e : S \subseteq e \to \mathbb{R}_{\geq 0}$ is a splitting function. Therefore, by Theorem 1, the problem presented in (2.2) with $\mathbf{w}_e(S) = h(|S|)$ is graph reducible, that is,

$$\min_{S \subset V} \mathbf{w}_e(S) = \min_{S \subset V} h(|S|) = \min_{S \subset V} c - \beta g(|S|) \quad \text{subject to } s \in S, t \in \bar{S}. \tag{4.2}$$

can be modeled by a gadget splitting function. Let the resulting graph from reduction be $G' = (V', E')$, then we have $\underset{G'}{\mathbf{cut}}(S) = c - \beta g(|S|)$ for any $S \subset V$. □

With the guarantee that it can be reduced to a graph cut problem, the specific techniques are adopted from [39]'s graph reduction techniques. Modeling $\mathbf{w}_e$ can be achieved by combining a set of *cardinality-based(CB) gadgets* [29] where a CB-gadget is a subgraph parametrized by positive weights (a, b). As introduced in section 2.4, after introducing an auxiliary node v_e, for each $v \in V$, introduce directed edges (v, v_e) and (v_e, v) with weights $a \cdot (k - b)$ and $a \cdot b$ respectively, then the function $\mathbf{w}_e(S) = a \cdot \min\{|S| \cdot (i - b), (k - |S|) \cdot b\}$ can be modeled. This can be understood by considering having i nodes on the s-side, then placing v_e on the s-side would incur a cut penalty of $ab(k - i)$, while placing v_e on the t-side gives a penalty of $ai(k - b)$. Therefore, to solve the min-cut problem in (4.2), the smaller of the two cut penalties will be chosen. Then the graph of a combination of these CB-gadgets with corresponding edge weights (a, b)'s is able to model problem (4.2).

4.2.1.1 Sparse reduction strategies

While it was shown that any cardinality-based submodular function $\mathbf{w}_e$ can be modeled by combining $|e| - 1$ CB-gadgets, the runtime can be largely improved using techniques introduced in [39]. For the purpose of this thesis, only related theoretical guarantees and reduction details will be presented from the original work. Since this thesis focuses on the exact problem without approximation error, the approximation ratio ε is set to 0 in the original problem. Given a concave function g, there are two major steps:

- Solve a *sparsest approximate reduction problem (SpAR)* [29], which gives the minimum number of linear pieces required to model the function g where each linear piece corresponds to a CB-gadget. The solution also outputs values needed to compute the weights of edges introduced by the auxiliary nodes.

- With the information of the class piecewise linear functions, its correspondence to the combined gadget functions can be established in Lemma 1 in [29]. As a result, the gadgets can be built, combined, and added to the graph.

By Theorem 6 in [39], for the graph considered in this thesis with a single hyperedge, the above procedures will construct a graph with $\mathcal{O}(n)$ nodes and $\mathcal{O}(n^2)$ edges. In fact, the actual graph constructed with respect to different g is much smaller, with experiments shown in the next chapter.

4.2.1.2 Auxiliary graph construction

For a given input graph $G = (V, E)$, let $G' = (V', E')$ denote the graph where any $S \subset V$ induces a cut penalty of $g(|V|) - \beta g(|S|)$, which is positive since g non-decreasing and $|V| \geq |S|$. Since naturally, S would incur a cut penalty of $\textbf{cut}(S)$ on G, we combine G and G' into a new graph $G'' = (V \cup V', E \cup E') = (V \cup A_e, E \cup \mathcal{E})$ where A and $\mathcal{E}$ are auxiliary node and edge set generated by reductions above. Then for any $S \subseteq V$, the cut penalty induced by S on G is equivalent to $\textbf{cut}(S) + g(|V|) - \beta g(|S|)$, which shares the same minimizer with $\textbf{cut}(S) - \beta g(|S|)$ since $g(|V|)$ is a constant.

4.2.2 Localized algorithm

Analogous to section 4.1.2, given $G'' = (V \cup A_e, E \cup \mathcal{E})$ constructed above, similarly to the localization for the algorithm I, we merge all nodes in $\bar{R}$ with t by replacing $\bar{R} \cup t$ with a "supernode" t' where for every $u \in R$ the weight of (u, t) becomes $\sum_{v \in \bar{R}} w_{r,u}$. Therefore, $(V \cup A_e, E \cup \mathcal{E})$ becomes $(R \cup A_e, (E \cup \mathcal{E}) \backslash \{(u, v) | u, v \in R\})$. The correctness of this idea follows similarly to the localization in Algorithm I. Therefore, solving the minimum s-t cut problem with decreasing β works to minimize the objective with the local constraint. The detailed algorithm for the localized problem is as follows.

We can see that the algorithm shares the same idea of solving the minimum s-t cut problem with Algorithm 2, but using differently constructed auxiliary graphs. Algorithm 3 and Algorithm 2 share the same input, output, and every step of the flow-based algorithm except for the auxiliary graph. We could derive the following bound on the runtime with almost the same proof steps as Theorem 4, therefore the proof is omitted.

Theorem 8. *Algorithm 3 will need to solve the min-cut objective at most **cut(R)** times.*

4.2.3 Runtime

Similarly, a runtime bound can be derived based on the number of nodes and edges. It is dependent both on an existing flow-based algorithm and the function g in the objective. Let $T_{mf}(N, M)$

Algorithm 3 Minimizing generalized expansion ratio - algorithm II

Require: : G, R
  $G \leftarrow$ constructed above as $(R \cup A_e, (E \cup \mathcal{E}) \backslash \{(u,v)|u,v \in R\})$
  $\beta := \infty$
  $\beta_{new} \leftarrow \frac{\textbf{cut}(R)}{g(|R|)}$
  $S = R$
  while $\beta_{new} < \beta$ **do**
    $S_{best} \leftarrow S$
    $\beta \leftarrow \beta_{new}$
    $S \leftarrow$ argmin $\textbf{cut}(S) + g(|V|) - \beta g(|S|)$
    $\beta_{new} \leftarrow \frac{\textbf{cut}(S)}{g(|S|)}$
  end while
  return S_{best}

denote the time to solve a max-flow problem with N nodes and M edges. Dependent on the reduction technique used, the size of the final auxiliary graph is bounded by $\mathcal{O}(n)$ nodes and $\mathcal{O}(n^2 + n)$. Then, the time required to solve the max-flow problem in each iteration is $T_{mf}(\mathcal{O}(n), \mathcal{O}(n^2))$. A recent algorithm for the maximum s-t flow problem by van den Brand et al. would generate a runtime guarantee of $\tilde{O}(M + N^{1.5})$ where $M \in \mathcal{O}(n)$ and $N \in \mathcal{O}(n^2)$ [38].

5. NUMERICAL EXPERIMENTS

This chapter will introduce the experimental results of an important instance of the localized generalized expansion ratio problem below

$$\min_{S \subseteq R} \frac{\textbf{cut}(S)}{g(|S|)} \tag{5.1}$$

with $g(|S|) = |S|^\alpha$ where $\alpha \geq 0$, satisfying g being monotonically non-decreasing and convex.

5.1 Datasets

We mainly consider two datasets: Zachary's karate club is a small graph capturing the network of 34 members of a karate club [40]; Minnesota road network has 2642 nodes and 2 strongly connected components [41].

5.2 Runtime

While the original problems on the entire graph are not tractable in practice, the two algorithms proposed by this thesis are both polynomial-time solvable. We test both algorithms on both the karate club and the Minnesota road network datasets with the same seed set at corresponding entries. All runtimes (in seconds) are averaged from five rounds. Table 5.1 gives the runtime of seed sets of 2 sizes on the karate club graph. While both algorithms are fast on the karate graph, Algorithm I generally outperform Algorithm II. An explanation is the time needed for auxiliary graph construction while the karate club graph is too small to suffer from runtime caused by the large auxiliary graphs.

Table 5.1: Runtime (s) in the 2nd row, and corresponding auxiliary graph size in the 3rd row

Algorithm I					Algorithm II			
seed set	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 2.0$	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 2.0$
$\|R_1\| = 5$	0.0042	0.0016	0.0018	0.0018	0.0080	0.0039	0.0013	0.0079
$\|R_2\| = 2$	0.0015	0.0029	0.0020	0.0019	0.0034	0.0021	0.0024	0.0041
$\|R_1\| = 5$	62	62	62	62	47	47	47	47
$\|R_2\| = 2$	65	65	65	65	50	50	50	50

Table 5.2: Runtime (s) in the 2nd row, and corresponding auxiliary graph size in the 3rd row

Algorithm I					Algorithm II			
seed set	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 2.0$	$\alpha = 1.1$	$\alpha = 1.3$	$\alpha = 1.5$	$\alpha = 2.0$
$\|R_1\| = 2$	8.7717	6.7575	6.8943	7.2734	59.1921	57.2558	61.5402	60.2531
$\|R_2\| = 50$	33.3645	7.7874	7.3141	6.8514	82.2321	79.4229	84.8938	76.2546
$\|R_1\| = 2$	5277	5277	5277	5277	3959	3959	3959	3959
$\|R_2\| = 50$	5229	5229	5229	5229	3911	3911	3911	3911

We run the same experiments on the road network graph with 2640 nodes as shown in Table 5.2. While Algorithm I still outperforms Algorithm II in runtime, we can also notice a significant difference in the size of the auxiliary graph. Therefore, if auxiliary graphs are pre-computed, Algorithm II is expected to be much faster. In fact, approximation techniques related to gadget reduction may be further used to improve the runtime, which will be introduced in the next chapter of future work.

$R \setminus \alpha$	1.1	1.2	1.3	1.4	1.5
{1,5,8,33,34}	1	24	29	29	29
{1,34}	5	32	32	32	32
{1,3,7,8,34}	15	21	21	29	29
{3,17,33,34}	14	14	14	22	24
{2,5,8,17,33,34}	1	22	22	28	28

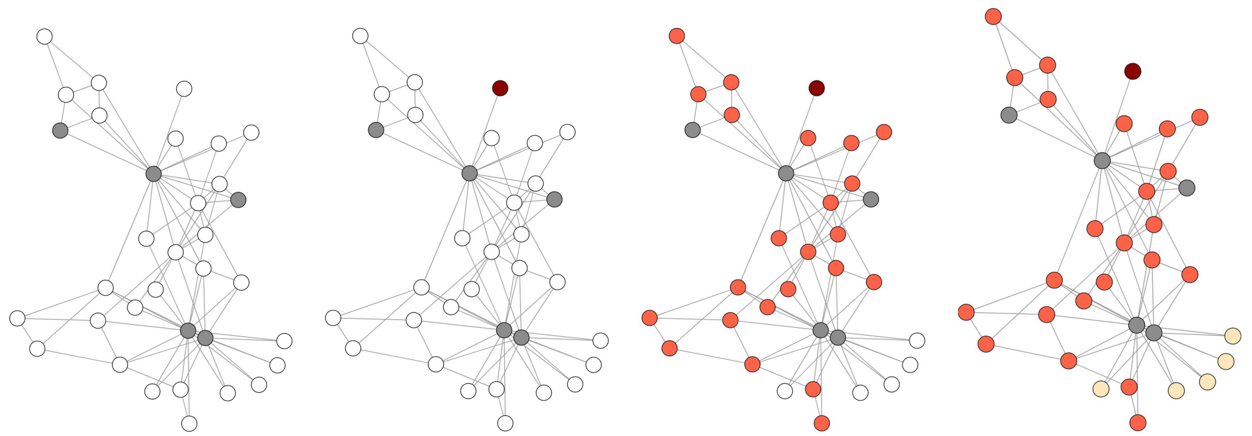
Table 5.3: Sizes of results from random seed sets on the Karate club graph

5.3 Relationship between parameter α and the resulting set size

First, we note that all results in this section are solutions to the generalized expansion ratio problem and solutions to other problems will be specified. For $g(|S|) = |S|^{\alpha}$, when g increases as α increases, the influences of the size of the resulting subgraph decrease by g being the denominator. Therefore, by varying α, we expect to manipulate the size of the subgraph returned. This is also demonstrated in the experimental results below on the Karate group where the entries are the sizes of the returning set.

From Table 5.3 where several instances with randomly chosen seed set R's are presented, the localized expansion ratio problem with $g(|S|) = |S|^{\alpha}$ is solved with respect to a range of $alpha$. We can see that as α increases, the size of the resulting set increases accordingly. In Figure 5.1, we also present a specific case where as α increases, the entire graph is returned gradually as the result of the localized generalized expansion ratio problem. The newly recovered nodes with greater α are marked by lighter colors.

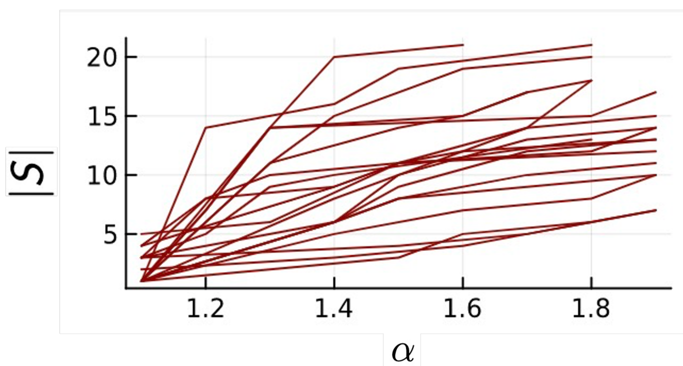
Trends in Table 5.3 are present throughout the solutions with random seed sets as shown in Figure 5.2 where each line represents a different seed set R. While the monotonicity of the set sizes is guaranteed, the change in set sizes is not always gradual as many lines are not smooth. In fact, the effects of changing the parameter α greatly depend on the seed set and structure of the input graph.



(a) R in white (b) $\alpha = 1.1$ (c) $\alpha = 1.2$ (d) $\alpha = 1.3$

Figure 5.1: Resulting sets with different α's

Figure 5.2: Graph size with respect to different α's



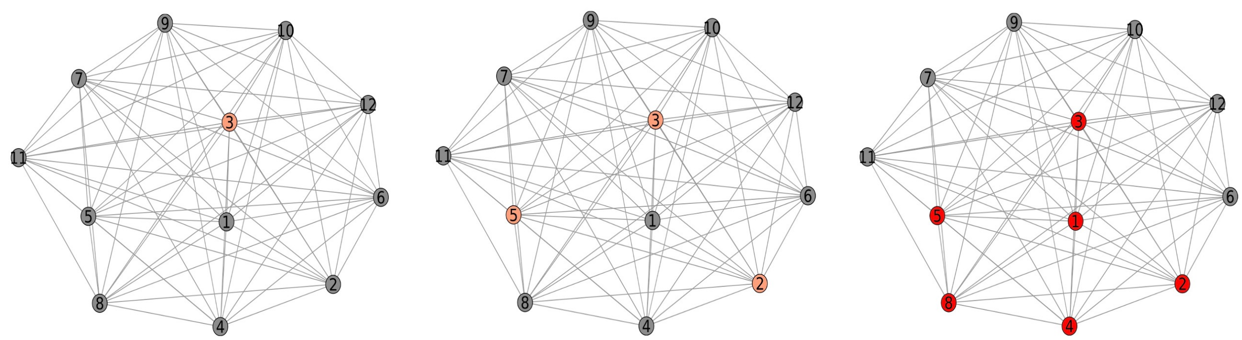
5.4 Connection between the generalized sparsest cut and the minimum bisection

As proved in Theorem 3, the generalized sparest cut problem shares the optimal set of solutions with the minimum bisection problem for large enough α. Therefore, the minimum bisection problem can be solved by the solution to the generalized sparsest cut problem with arbitrarily large α. Recall that the problems of the generalized sparest cut and the minimum bisection are

$$\min_{S\subseteq V} \frac{\textbf{cut}(S)}{g(|S|)} + \frac{\textbf{cut}(S)}{g(|\bar{S}|)} \qquad \text{and} \qquad \min_{S\subseteq V,\ |(|S|-|\bar{S}|)|\leq 1} \textbf{cut}(S).$$

	(a)	(b)	(c)
α	1.0	1.1	1.2
S	{3}	{2,3,5}	{1,2,3,4,5,8}
cut(S)	9	23	32
Sparsest Cut(S)	9.72	8.92	7.45

Table 5.4: Details of solution sets in Figure 5.3



(a) $\alpha = 1.0$ (b) $\alpha = 1.1$ (c) $\alpha = 1.2$

Figure 5.3: Generalized sparsest cut problem solutions with different α's

Although both problems are NP-hard to solve, solutions to smaller instances can be obtained by brute force. Below is an example set of solutions to the sparsest cut problem with increasing α where the third solution set exactly matches the solution of the minimum bisection problem. We can see that as α increases, the solution sets of two problems eventually match.

6. CONCLUSIONS AND FUTURE WORK

This thesis introduces a novel generalized graph clustering objective inspired by the graph expansion ratio problem. After a brief overview of existing graph optimization problems, this thesis establishes the connections between this newly introduced objective and existing objectives in graph clustering.

Regarding newly introduced objective, this thesis presents a local variant of the corresponding optimization problem. While optimization problems corresponding to existing objectives suffer from computational hardness, this thesis presents two polynomial-time solvable algorithms for the local variant of the optimization problem involving the new objective.

Using existing flow-based algorithms with theoretical runtime guarantees, the two proposed algorithms achieve the effects of changing the size of the resulting graph. Furthermore, the connection between the proposed generalized sparsest cut problem and the minimum bisection problem is also shown.

Although this thesis presents the performances and usage of the algorithms in exploring graphs, its efficient usage in graph clustering requires more research into how performances can be improved with respect to specific real-world graphs for clustering.

One major natural future direction is to explore the potential of the sparse reduction strategies used in Algorithm II where approximations can be possible with much faster runtime. Since the domain and the final output of the function is a discrete, approximation for a continuous extension may also result in the exact discrete solution. One minor future direction is to explore the steps of auxiliary graph construction in Algorithm II. Currently, the localization is after the main graph construction, and the graph size may be reduced by having localization before constructing the whole graph.

In addition, the proposed algorithms not being strongly local algorithms result in runtime limitations for large graphs in nature. One possible direction for future work is to explore the potential of other objectives with varying granularity achieved by generalized functions.

REFERENCES

[1] S. E. Schaeffer, “Survey: Graph clustering,” *Comput. Sci. Rev.*, vol. 1, p. 27–64, aug 2007.

[2] K. Fountoulakis, M. Liu, D. F. Gleich, and M. W. Mahoney, “Flow-based algorithms for improving clusters: A unifying framework, software, and performance,” *SIAM Review*, vol. 65, no. 1, pp. 59–143, 2023.

[3] S. Hoory and N. Linial, “Expander graphs and their applications,” *Bulletin of the American Mathematical Society*, vol. 43, pp. 439–561, 2006.

[4] A. Sinclair and M. Jerrum, “Approximate counting, uniform generation and rapidly mixing markov chains,” *Information and Computation*, vol. 82, no. 1, pp. 93–133, 1989.

[5] D. W. Matula and F. Shahrokhi, “Sparsest cuts and bottlenecks in graphs,” *Discrete Applied Mathematics*, vol. 27, no. 1, pp. 113–123, 1990.

[6] D. S. Hochbaum, “A polynomial time algorithm for rayleigh ratio on discrete variables: Replacing spectral techniques for expander ratio, normalized cut, and cheeger constant,” *Operations Research*, vol. 61, no. 1, pp. 184–198, 2013.

[7] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar, “On the hardness of approximating multicut and sparsest-cut,” in *20th Annual IEEE Conference on Computational Complexity (CCC’05)*, pp. 144–153, 2005.

[8] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, “A survey of kernel and spectral methods for clustering,” *Pattern Recognition*, vol. 41, no. 1, pp. 176–190, 2008.

[9] W. Ye, S. Goebl, C. Plant, and C. Böhm, “Fuse: Full spectral clustering,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, (New York, NY, USA), p. 1985–1994, Association for Computing Machinery, 2016.

[10] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[11] M. C. Nascimento and A. C. de Carvalho, "Spectral methods for graph clustering – a survey," *European Journal of Operational Research*, vol. 211, no. 2, pp. 221–231, 2011.

[12] K. Lang and S. Rao, "A flow-based method for improving the expansion or conductance of graph cuts," in *Integer Programming and Combinatorial Optimization* (D. Bienstock and G. Nemhauser, eds.), (Berlin, Heidelberg), pp. 325–337, Springer Berlin Heidelberg, 2004.

[13] N. Alon and V. Milman, "1, isoperimetric inequalities for graphs, and superconcentrators," *Journal of Combinatorial Theory, Series B*, vol. 38, no. 1, pp. 73–88, 1985.

[14] R. M. Tanner, "Explicit concentrators from generalized n-gons," *SIAM Journal on Algebraic Discrete Methods*, vol. 5, no. 3, pp. 287–293, 1984.

[15] R. B. Boppana, "Eigenvalues and graph bisection: An average-case analysis," *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pp. 280–285, 1987.

[16] T. Leighton and S. Rao, "An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms," in *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, SFCS '88, (USA), p. 422–431, IEEE Computer Society, 1988.

[17] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," *J. ACM*, vol. 46, p. 787–832, nov 1999.

[18] S. Arora, S. Rao, and U. Vazirani, "Expander flows, geometric embeddings and graph partitioning," *J. ACM*, vol. 56, apr 2009.

[19] R. Khandekar, S. Rao, and U. Vazirani, "Graph partitioning using single commodity flows," *J. ACM*, vol. 56, jul 2009.

[20] L. Orecchia and Z. A. Zhu, *Flow-Based Algorithms for Local Graph Clustering*, pp. 1267–1286.

[21] K. Lang, M. Mahoney, and L. Orecchia, "Empirical evaluation of graph partitioning using spectral embeddings and flow," pp. 197–208, 06 2009.

[22] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 475–486, 2006.

[23] K. Fountoulakis, D. F. Gleich, and M. W. Mahoney, "A short introduction to local graph clustering methods and software," *ArXiv*, vol. abs/1810.07324, 2018.

[24] N. Veldt, D. Gleich, and M. Mahoney, "A simple and strongly-local flow-based method for cut improvement," in *Proceedings of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proceedings of Machine Learning Research*, (New York, New York, USA), pp. 1938–1947, PMLR, 20–22 Jun 2016.

[25] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, pp. 123 – 29, 2008.

[26] M. Ghaffari, *Local Graph Algorithms*, pp. 201–238. 2022.

[27] R. Andersen and K. J. Lang, "An algorithm for improving graph partitions," in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, (USA), p. 651–660, Society for Industrial and Applied Mathematics, 2008.

[28] N. Veldt, C. Klymko, and D. F. Gleich, *Flow-Based Local Graph Clustering with Better Seed Set Inclusion*, pp. 378–386.

[29] N. Veldt, A. R. Benson, and J. Kleinberg, "Hypergraph cuts with general splitting functions," *SIAM Review*, vol. 64, no. 3, pp. 650–685, 2022.

[30] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[31] D. Wagner and F. Wagner, "Between min cut and graph bisection," in *Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science*, MFCS '93, (Berlin, Heidelberg), p. 744–750, Springer-Verlag, 1993.

[32] A. R. Benson, J. M. Kleinberg, and N. Veldt, "Augmented sparsifiers for generalized hypergraph cuts," *ArXiv*, vol. abs/2007.08075, 2020.

[33] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

[34] C. M. Fiduccia and R. M. Mattheyses, "A linear-time heuristic for improving network partitions," in *Proceedings of the 19th Design Automation Conference*, DAC '82, p. 175–181, IEEE Press, 1982.

[35] M. Garey, D. Johnson, and L. Stockmeyer, "Some simplified np-complete graph problems," *Theoretical Computer Science*, vol. 1, no. 3, pp. 237–267, 1976.

[36] B. Mohar, "Isoperimetric numbers of graphs," *Journal of Combinatorial Theory, Series B*, vol. 47, no. 3, pp. 274–291, 1989.

[37] Y. Kawase and A. Miyauchi, "The densest subgraph problem with a convex/concave size function," *Algorithmica*, vol. 80, p. 3461–3480, dec 2018.

[38] J. van den Brand, Y. T. Lee, Y. P. Liu, T. Saranurak, A. Sidford, Z. Song, and D. Wang, "Minimum cost flows, mdps, and 1-regression in nearly linear time for dense instances," in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, (New York, NY, USA), p. 859–869, Association for Computing Machinery, 2021.

[39] N. Veldt, A. R. Benson, and J. Kleinberg, "Approximate decomposable submodular function minimization for cardinality-based components," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 3744–3756, Curran Associates, Inc., 2021.

[40] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[41] T. A. Davis and Y. Hu, “The university of florida sparse matrix collection,” *ACM Trans. Math. Softw.*, vol. 38, dec 2011.