Integer Programming Models for Detecting Graph Bipartitions with Structural Requirements*

Chrysafis Vogiatzis†
Industrial and Manufacturing Engineering
North Dakota State University, Fargo, ND, USA
chrysafis.vogiatzis@ndsu.edu

Jose L. Walteros
Department of Industrial and Systems Engineering,
University at Buffalo
josewalt@buffalo.edu

Abstract

The graph bipartitioning problem consists of dividing a graph into two disjoint subgraphs, such that each node is highly similar to others in the same subgraph, but also different from members of the other subgraph, according to some homogeneity criterion. This problem has received significant attention over the last few years because of its applicability in areas as diverse as data classification, image segmentation, and social network analysis. In this paper we study a variation of the graph bipartitioning problem in which, in addition to considering homogeneity criteria for generating the partition, we also ensure that one of the subgraphs satisfies a set of predefined structural properties—i.e., such a subgraph is required to induce a given motif. We focus our attention on imposing structural constraints that force one of the subgraphs to induce stars, cliques, and clique relaxations (quasi-cliques) and discuss some specific applications for such particular cases. We tackle this problem by modeling it as a general fractional programming optimization problem and study several solution approaches. Moreover, we discuss additional algorithmic enhancements to tackle some of the aforementioned cases, and provide two greedy algorithms for the specific cases of induced cliques and stars, showing the approximation ratio for induced stars. Finally, we test the quality of our approach by solving a collection of several real-life and randomly generated instances with various configurations, analyzing the benefits of the proposed models, as well as possible further extensions.
Keywords normalized cut, clique relaxations, graph partitioning, data classification

1 Introduction

Partitioning a graph into its important parts has been an ongoing study of both academicians and practitioners for decades (Kernighan and Lin, 1970, Fiduccia and Mattheyses, 1982, Karypis and Kumar, 1998, Newman and Girvan, 2004). The main objective of most graph partitioning problems is to identify an informal set of communities on a given graph, whose members share

*This work was supported in part by the National Science Foundation Award CMMI-1635611, "Operational Decision-Making for Reach Maximization of Incentive Programs that Influence Consumer Energy-Saving Behavior" and the Mathematical Modeling and Optimization Institute of the Air Force Research Lab.

†Corresponding author. Phone: 701-231-7286; Address: NDSU Dept. 2485, Fargo, ND 158108-6050.

common properties and are expected to behave in similar ways (Fortunato, 2010). Applications of graph partitioning problems find roots in numerous areas due to their ability to classify collections of items into homogeneous groups. To name a few examples, in the context of image segmentation, a partition over the graph representation of an image (i.e., a graph in which each pixel is given by a node and the edges between them describe a measure of similarity between the pixels' color) may constitute a set of shapes of similar tonalities (Deschamps and Cohen, 2001). Likewise, in the context of online social networks, a community resulting from a partition of the social network could represent collaborators, friends, neighbors, or a group of people that share some common interests, among others (Balasundaram et al., 2011). For a comprehensive survey on definitions and applications of community structures in graphs, we refer the interested reader to the surveys by Boccaletti et al. (2006) and Fortunato (2010).

In general, providing a thorough definition of what detecting informal communities (clusters) on a graph entails is rather challenging because of the broad number of "properties", "behaviors", or "structures" that can be used to define such communities (Yang and Leskovec, 2015). This broadness leads to multiple approaches for generating such graph communities depending on the desired criteria. These approaches are often categorized as: (1) density-based approaches and (2) cut-based approaches (Schaeffer, 2007). Among the most prevalent methods we highlight k-means (Schenker et al., 2003), spectral techniques (Von Luxburg, 2007, Schenker et al., 2003), random walkers (Deschamps and Cohen, 2001), Markov clustering (Enright et al., 2002), and bipartitioning approaches. We classify the contribution of this paper within the latter category, thus we limit the discussion to this literature.

Normalized cut problems have attracted significant scientific interest ever since their introduction, see Shi and Malik (2000). The goal of these problems is to partition a graph into two disjoint node sets, such that the cut between such sets (the weighted sum of all edges with exactly one endpoint in each partition) is minimized, while the association of each set (the weighted sum of all edges within the partitions) is maximized. Several variants of the normalized cut have been extensively studied over the last few decades (Cox et al., 1996, Hochbaum, 2010, 2013, Sharon et al., 2006), including, the *size-normalized cut*, and the *ratio regions*. Overall, most normalized cut problems are modeled as the minimization of a single ratio form or a double ratio form like the ones described in (1) and (2), where $f(S)$ and $g(S, \bar{S})$ represent association metrics over the partitions (i.e., node subsets S and its complement $\bar{S}$) and the corresponding cut (i.e., $(S, \bar{S})$), respectively. In Hochbaum (2010) several ratio regions and normalized cut problems were shown to be solvable in polynomial time.

$$z = \frac{f(S)}{g(S)} \tag{1}$$

$$z = \frac{f(S)}{g(S)} + \frac{f(\overline{S})}{g(\overline{S})} \tag{2}$$

Another important factor about the recognition of informal communities in a graph is their structure. It is well-known that certain communities often possess a specific structure, like forming a clique, a star, or a tree, or satisfy some specific properties, like being connected or having a low diameter. The problem of detecting a structured community on a graph can also be framed as a bipartition problem in which one of the resulting subgraphs is a community that satisfies the desired structural properties. In the context of graph partitioning, researchers typically resort to an either-or approach in which the partitions are generated either based on a cut criterion or on structural properties, but never by considering both of these type of conditions simultaneously. For an example, consider the graphs depicted in Figure 1, in which three different criteria are used to

generate bipartitions of the same graph. Figure 1(a) depicts a partition that arises from detecting a maximum clique, Figure 1(b) shows a partition obtained using the normalized cut criterion, and Figure 1(c) presents a partition that is generated considering both the normalized cut and the clique structure—i.e., a bipartition that produces a community that is loosely connected outside, densely connected within, and also forms a clique.

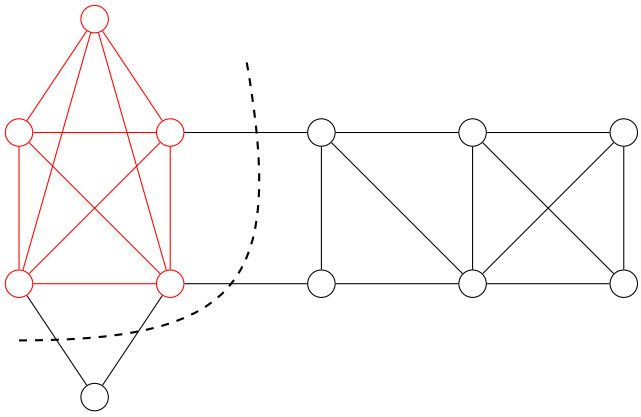
The intuition behind detecting a graph bipartition satisfying a structural requirement is that in graph partitioning we are commonly interested in detecting anomalies and outliers. Even though outliers are sometimes considered to be the result of random and isolated events, they often possess properties of interest that can be used for detecting and analyzing them. For a notable example, we refer the interested reader to the vast literature of Sybil node detection (Yu, 2011). In this problem, Sybil nodes (outliers) are considered to be fake users (nodes) that aim to infiltrate the honest areas of an online social network to perform illicit activities such as phishing and electronic spamming. To avoid detection, nodes in Sybil communities attempt to emulate the behavior of honest users. This is achieved by generating profiles that look similar to those of real users and that are also connected with other fake nodes, pretending to form a strong and real community. It is well documented that the subgraphs induced by Sybil nodes are often very dense, have quite a small diameter, and are generally loosely connected with the social network (Yu et al., 2006, Wei et al., 2012). Thus, identifying a normalized cut that also enforces a cohesiveness criterion could be a useful tool for detecting such abnormal communities.

Another application comes from the resiliency analysis of road infrastructure networks, where intersections and important landmarks are modeled with nodes, the road segments connecting them with the corresponding edges, and the weight associated with each edge provides a relative measure of the expected traffic on them. Infrastructure networks are known to be locally very dense but quite sparse globally and have been shown in practice to suffer from severe delays even in the presence of minor incidents (Berdica, 2002). In the graph partitioning context, identifying a partition that combines a normalized cut with a cohesiveness criterion can lead to identifying parts of the road network that are susceptible to disconnections and bigger delays under unpredictable road closures (due to accidents or unscheduled repairs), but could remain locally operational due to the internal cohesiveness.

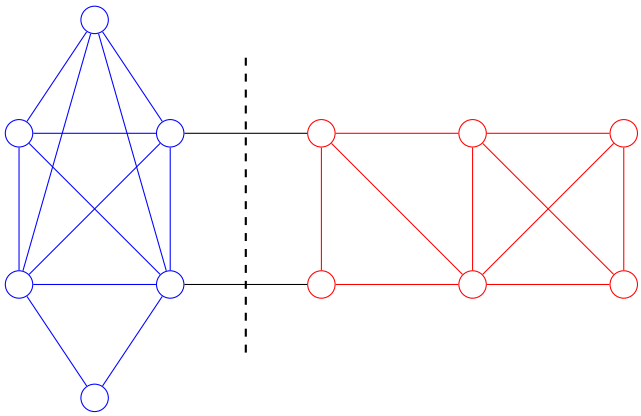
Last, community detection has been prominent in biological networks, more specifically, protein-protein interaction and metabolic networks. In protein-protein interaction networks (PPINs), detecting communities of interest between proteins has led to breakthroughs in the characterization of different protein complexes and functional modules (Chen and Yuan, 2006). We highlight the work by Pržulj et al. (2004), which has motivated many studies on the detection of specific types of induced subgraphs (typically smaller in size, with cardinalities ranging from 2 to 5 nodes) in such networks. We note here that most of those subgraphs are indeed of interest in our work: as an example, graphlets (i.e., size restricted induced subgraphs) 2, 8, and 29 are cliques and graphlets 1, 4, and 11 are induced stars, as per the indices used in Pržulj et al. (2004). Our proposed extension would allow for the detection of clusters following such structures in PPINs.

In this paper, we propose a general type of formulations to detect normalized cuts and some of its variants, considering additional *structural constraints* on one of the obtained clusters. We show that upon introduction of some of these constraints, even problems that are solvable in polynomial time are rendered $\mathcal{NP}$-hard. We complement the proposed formulations with further exact and heuristic approaches to improve their computational performance. In summary, the contributions of our work are as follows.

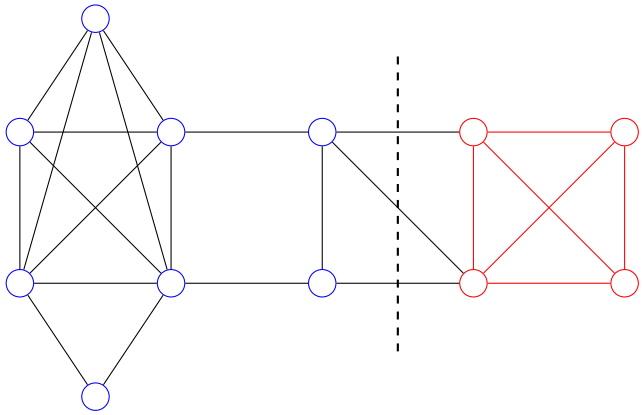
- We develop a framework for solving normalized cut problems with structural constraints on graphs.



(a) Maximum clique.



(b) Normalized cut (the two clusters are in blue and red).



(c) Combining the clique structure with the normalized cut.

Figure 1: Comparing the partitions obtained by finding the maximum clique versus using a normalized cut criterion.

- We derive the computational complexity of some of these new problems.
- We provide integer programming formulations for the problems.
- We propose multi-objective optimization techniques along with a combinatorial decomposition approach to solve the problems exactly.
- We propose two greedy algorithms and derive the worst case performance in the case of induced stars.
- We provide a computational study on how our approaches perform on a variety of real-life and synthetic networks.

In the next section (Section 2), we proceed to introduce the notation of the problem, our assumptions, and the problem definitions. Then, Section 3 presents all of our complexity results and derives the $\mathcal{NP}$-hardness of the studied problems. Following that, in Section 4, we show some indicative integer programming formulations. More specifically, we formulate our problems as fractional mathematical programs, provide their linearizations, and discuss a linear scalarization, and ϵ-constraint method to tackle them. In Section 5, we investigate a combinatorial decomposition approach, as well as two greedy algorithms to solving the problems. Section 6 compares our models and approaches in several real-life and synthetic networks. Last, in Section 7, we provide our concluding remarks and offer insight into future work in the field.

2 Problem Definition

Let $G = (V, E)$ be a simple, connected, undirected graph, where V is a set of n nodes and E is a set of m edges. We define $w : E \to \mathbb{R}^+$ to be a positive, non-trivial weight function over the edges of G. Given a node set $S \subseteq V$, we say that $w(S)$ represents the sum of the weights of the edges whose corresponding endpoints are in S (i.e., $w(S) = \sum_{(i,j)\in(S\times S)\cap E} w_{ij}$). A *cut-set* over a graph G is defined to be a collection of edges whose removal generates a *bipartition* (i.e., a *cut*) of node set V into a set S and its complement $\bar{S}$ (i.e., a pair of sets S and $\bar{S}$ for which $S \cup \bar{S} = V$ and $S \cap \bar{S} = \emptyset$). Given a cut $(S, \bar{S})$, we define $w(S, \bar{S})$ to be the sum of the weights over the cut-set (i.e., $w(S, \bar{S}) = \sum_{i,j\in(S\times\bar{S})\cap E} w_{ij}$). For any node set $S \subseteq V$, we define $\mathcal{N}(S)$ to be the open neighborhood of S, i.e. the set of nodes outside S that are adjacent to at least one node in S, $\mathcal{N}[S] = S \cup \mathcal{N}(S)$ as the closed neighborhood of S, respectively, and $G[S]$ to represent the subgraph of G induced by S; that is, the subgraph with node set $V(G[S]) = S$ and edge set $E(G[S]) = \{(i, j) \in E : i \in S, j \in S\}$.

In this paper we use the term *structure* when referring to a subgraph that has a generic predefined set of characteristics. For instance, structures of interest in this work are cliques and stars. A *clique* is defined to be a subset of nodes S such that the subgraph induced by S is complete (i.e., $S \times S \subseteq E$). A *star* is a subgraph of G given by a set of nodes S composed of a hub node denoted $h(S)$ and a set of leaf nodes $l(S) \subseteq \mathcal{N}(h(S))$. For the sake of convenience, we will use the term star in reference to both the subgraph and its set of nodes S, unless further clarification is required. A clique S is said to be maximal if it is not a subset of any other clique in G. Similarly, a star S is maximal if $l(S) = \mathcal{N}_G(h(S))$ (i.e., it comprises the hub $h(S)$ and all of its neighbors). A star is also said to be induced if its leaf nodes are non-adjacent. The size of a clique is equal to the number of its nodes and the size of a star is given by the number of its leaves. Thus, we will say that a single node is a clique of size one or a star of size zero. Cliques and stars are commonly denoted by the letters C and S, respectively. However, since in this paper both cliques and stars are the result of

graph bipartitions given by cuts generically denoted by $(S, \bar{S})$, we will maintain consistency and use S to describe such partitions, independently of whether they form cliques or stars and clear any ambiguity by context.

Of scientific interest are also *clique relaxations* that require a subgraph to remain cohesive and with a low diameter, but allow for some edges to be absent. A γ-*quasi-clique* (or γ-*clique*), for $\gamma \in [0, 1]$, is a subset of nodes S such that S is connected and the number of edges in the subgraph induced by S contains a fraction of γ or more of all the possible connections; that is, $|E(G[S])| \geq \gamma \binom{|V(G[S])|}{2}$.

In this context, we are tackling fractional program as in **FP** in (3)

$$(\mathbf{FP}) : \min \quad \frac{f(S)}{g(S)} \tag{3a}$$

$$s.t.$$

$$S \in \mathcal{S}, \tag{3b}$$

$$S \subseteq V, \tag{3c}$$

where the objective function in (3a) aims to minimize the ratio of two convex functions on the set S, whereas (3b) and (3c) ensure that S is a member of a family of structures $\mathcal{S}$ and also a subset of the node set of the original graph G. Observe here that for many of the interesting structures mentioned herein, $\mathcal{S}$ can be of exponential in size (e.g., the number of maximal cliques in a graph can be remarkably large (Eppstein et al., 2010)).

Functions $f(\cdot)$ and $g(\cdot)$, with domains $V \to \mathbb{R}^+$ or $E \to \mathbb{R}^+$, can be selected from a series of function families on the nodes V. Some indicative function families are provided in (4)–(5).

$$f(S) = \begin{cases} w(S, \bar{S}) \\ |S, \bar{S}| \\ w(N(S)) \end{cases} \tag{4}$$

$$g(S) = \begin{cases} w(S) \\ |S| \\ \min\{|S|, |\bar{S}|\}. \end{cases} \tag{5}$$

We note here that the typical normalized cut problem is given in graphs with one type of weights and corresponds to the double ratio form given by (2), where $f(S) = f(\bar{S}) = w(S, \bar{S})$ and $g(S) = w(S), g(\bar{S}) = w(\bar{S})$. Furthermore, consider $f(S) = w(S, \bar{S})$ and $g(S) = w(S)$, and let $\mathcal{S}$ be the set of all connected node sets such that $|S| \leq |V|/2$. Then, (3a)–(3c) becomes the *normalized cut′* problem (that is, the single form of the normalized cut problem as presented by Sharon et al. (2006)). Other known normalized cut variants that can be constructed in our setup include the ratio regions problem (when $f(S) = w(S, \bar{S})$ and $g(S) = |S|$), or the Cheeger constant problem (Cheeger, 1969), after selecting $f(S) = w(S, \bar{S})$ and $g(S) = \min\{|S|, |\bar{S}|\}$.

Throughout the remainder of the paper, we will stay consistent with the literature, referring to single form ratio problems as normalized cut′, as opposed to double form ratio problems, which would be referred to as normalized cut. The decision versions of these problems will be written in small capital letters as Normalized Cut′ and Normalized Cut, respectively.

In this work, we focus on three problems, namely the *clique*, *star*, and γ-*quasi-clique normalized cut′* problems. This is achieved by selecting $f(S) = w(S, \bar{S}), g(S) = w(S)$, and $\mathcal{S}$ to be the set of cliques, induced stars, and γ-quasi-cliques, respectively. Nevertheless, the proposed formulations are suitable to tackle other variations different ratios and structures, by simply applying minor modifications.

3 Computational Complexity

We begin by providing a proof that the normalized cut′ problem, even in the absence of structural constraints, always produces a connected node set S, under the assumption that $f(S)$ and $g(S)$ are nonnegative.

Proposition 1. *Under the assumption that $f(\cdot)$ and $g(\cdot)$ are nonnegative, additive map functions, there always exists an optimal solution $(S, \bar{S})$ to the normalized cut′ problem for which S is connected.*

Proof. First of all, we note that there always exists an optimal solution to this problem, as it is feasible for simple, connected graphs. Assume for a contradiction and without loss of generality that there exists an optimal solution $S = S_1 \cup S_2$, where S_1 and S_2 are two disconnected components, i.e., $(S_1 \times S_2) \cap E = \emptyset$, and S_1 is connected. Further assume no connected optimal solution exists. Observe that, by assumption, $\frac{f(S)}{g(S)} = \frac{f(S_1)+f(S_2)}{g(S_1)+g(S_2)}$, $\frac{f(S)}{g(S)} < \frac{f(S_1)}{g(S_1)}$, and $\frac{f(S)}{g(S)} < \frac{f(S_2)}{g(S_2)}$. Thus we have:

$$\frac{f(S_1)+f(S_2)}{g(S_1)+g(S_2)} < \frac{f(S_1)}{g(S_1)} \tag{6}$$

and

$$\frac{f(S_1)+f(S_2)}{g(S_1)+g(S_2)} < \frac{f(S_2)}{g(S_2)} \tag{7}$$

We can differentiate between three cases: (a) $\frac{f(S_1)}{g(S_1)} = \frac{f(S_2)}{g(S_2)}$, (b) $\frac{f(S_1)}{g(S_1)} > \frac{f(S_2)}{g(S_2)}$, and (c) $\frac{f(S_1)}{g(S_1)} < \frac{f(S_2)}{g(S_2)}$. In the first case, it is easy to see that if $\frac{f(S_1)}{g(S_1)} = \frac{f(S_2)}{g(S_2)}$, that implies $\frac{f(S_1)+f(S_2)}{g(S_1)+g(S_2)} = \frac{f(S_1)}{g(S_1)} = \frac{f(S_2)}{g(S_2)}$, in which cases S has the same objective function value as the connected S_1.

Now, let us assume that $\frac{f(S_1)}{g(S_1)} > \frac{f(S_2)}{g(S_2)}$. From equation (7), we have

$$\begin{aligned}
&\frac{f(S_1)+f(S_2)}{g(S_1)+g(S_2)} < \frac{f(S_2)}{g(S_2)} \implies \\
&f(S_1)\cdot g(S_2) + f(S_2)\cdot g(S_2) < f(S_2)\cdot g(S_1) + f(S_2)\cdot g(S_2) \implies \\
&f(S_1)\cdot g(S_2) < f(S_2)\cdot g(S_1) \implies \frac{f(S_1)}{g(S_1)} < \frac{f(S_2)}{g(S_2)},
\end{aligned} \tag{8}$$

which clearly contradicts the assumption. For the last case, it suffices to do the same with $\frac{f(S_2)}{g(S_2)} > \frac{f(S_1)}{g(S_1)}$. Thus, if there is a disconnected optimal solution, there must also be a connected solution with the same value. □

As seen before, for some of the single form ratio problems that we discuss herein, there exist readily available algorithms that obtain a solution in polynomial time. However, whenever we require the obtained node set S to satisfy some structure constraints (e.g., require S to form a complete subgraph or clique), the problem can be shown to be $\mathcal{NP}$-hard.

Definition 1. Clique Normalized Cut′ *(Decision version)*
Given a graph $G(V, E)$, a weight function $w(\cdot)$ on the edges of the graph, and a real number ℓ, does there exist a clique $S \subseteq V$ such that $\frac{w(S,\bar{S})}{w(S)} \leq \ell$?

Theorem 1. Clique Normalized Cut′ *is $\mathcal{NP}$-complete.*

Proof. The problem is clearly in $\mathcal{NP}$ as it is easy to verify in polynomial time that both S forms a clique and $\frac{w(S,\bar{S})}{w(S)} \leq \ell$.

Let us consider the decision version of CLIQUE with an instance $< G, k >$. This problem is well-known to be $\mathcal{NP}$-complete (Garey and Johnson, 1979). We construct a graph G' as follows. Let $V' = V \cup \{s'\}$, and let $E' = E \cup \{\bigcup_{i \in V}(s', i)\}$. An example of the gadget, for better presentation, is given in Figure 2. Also let

$$w_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \\ M, & \text{otherwise.} \end{cases}$$

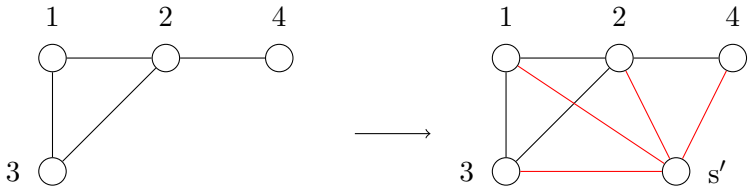


Figure 2: Example of the gadget for transforming a graph G with 4 nodes of a CLIQUE instance to G' of a CLIQUE NORMALIZED CUT′ instance,

where M is used as a "big M", and it is assumed to be bigger than n^2. Further, let

$$\ell = \frac{(n-k)M + k(n-k)}{\frac{k(k-1)}{2} + kM}.$$

We will show that every CLIQUE instance $< G, k >$ maps into a CLIQUE NORMALIZED CUT′ instance $< G', w, \ell >$. First, assume that there exists a clique $S \subseteq V$ such that $|S| \geq k$. Then, it is easy to see that the clique $S' = S \cup \{s'\} \subseteq V'$ has $w(S', \bar{S}') \leq (n-k)M + k(n-k)$, as in the upper limit each of the nodes in the clique is connected to every other of the $n-k$ nodes in the original graph. Moreover, $w(S') = \frac{k(k-1)}{2} + kM$, and hence $\frac{w(S',\bar{S}')}{w(S')} \leq \ell$.

Second, let us assume that there exists no clique $S \subseteq V$ such that $|S| \geq k$, and yet there exists a clique $S' \subseteq V'$ such that $\frac{w(S',\bar{S}')}{w(S')} \leq \ell$. Let the largest clique in G be $\widetilde{S}$ of size $q < k$. We distinguish between two cases: either $s' \in S'$ or $s' \notin S'$. For the first case, observe that we cannot have $|S'| > k$, as that would imply that there exists a clique of size at least k in the original graph. Hence, we have that $w(S', \bar{S}') \geq (n-q)M > (n-k)M + k(n-k)$, and $w(S') \leq \frac{q(q-1)}{2} + qM < \frac{k(k-1)}{2} + kM$, which implies that $\frac{w(S',\bar{S}')}{w(S')} > \ell$. For the latter case, where $s' \notin S'$, we have that $w(S', \overline{S'}) \geq q \cdot M$ and that $w(S') = q \cdot (q-1)/2$. Therefore, we get that

$$\frac{w(S', \overline{S'})}{w(S')} \geq \frac{q \cdot M}{q \cdot (q+1)/2} = \frac{2M}{q-1} > \ell.$$

The last inequality follows by construction and the selection of M to be big enough. This concludes the proof.

□

Definition 2. γ-QUASI-CLIQUE NORMALIZED CUT′ *(Decision version)*
Given a graph $G(V, E)$, a weight function $w(\cdot)$ on the edges of the graph, and real numbers ℓ, γ, does there exist a γ-quasi-clique $S \subseteq V$ such that $\frac{w(S,\bar{S})}{w(S)} \leq \ell$?

Theorem 2. γ-QUASI-CLIQUE NORMALIZED CUT′ *is $\mathcal{NP}$-complete.*

Proof. The gadget employed to reduce Clique to Clique Normalized Cut′ can be employed to reduce γ-Quasi-Clique to γ-Quasi-Clique Normalized Cut′, and as such this problem is also $\mathcal{NP}$-complete. □

Definition 3. Star Normalized Cut′ *(Decision version)*
Given a graph $G(V, E)$, a weight function $w(\cdot)$ on the edges of the graph, and a real number ℓ, does there exist an induced star $S \subseteq V$ such that $\frac{w(S,\bar{S})}{w(S)} \leq \ell$?

Theorem 3. Star Normalized Cut′ *is $\mathcal{NP}$-complete.*

Proof. First of all, the problem is in $\mathcal{NP}$, as it can be verified in polynomial time that S forms a star and that $\frac{w(S,\bar{S})}{w(S)} \leq \ell$.

We proceed to show the Independent Set problem can be reduced to the Star Normalized Cut′ problem. Consider an instance of Independent Set $< G, k >$, well-known to be $\mathcal{NP}$-complete (Garey and Johnson, 1979). We use the same gadget as before and construct $\hat{G}$ with node set $\hat{V}$ and edge set $\hat{E}$ defined the exact same way. We further use the same weight function on the edges of $\hat{G}$. We now let ℓ be equal to $\frac{(n-k)M+k(n-k)}{kM}$.

Let us assume that G has an independent set of size k. Then the star S' consisting of node s' as its center and the nodes in the independent set of G form a star (otherwise, the nodes would not form an independent set). Furthermore, we have that $w(S', \bar{S}') \leq (n-k)M + k(n-k)$, as every node not in the independent set is connected to s' and there can be up to $k(n-k)$ edges connecting the k leaves of S' to the rest of the $n-k$ nodes not in S' in $\hat{G}$. Last, clearly S' will have $w(S') = kM$, leading to $\frac{w(S,\bar{S})}{w(S)} \leq \frac{(n-k)M+k(n-k)}{kM} = \ell$.

For the opposite direction, assume that there is no independent set of size k in G, and yet there exists S' in $\hat{G}$ such that it forms a star, and $\frac{w(S,\bar{S})}{w(S)} \leq \ell$. We distinguish two cases: (i) $s' \notin S'$, and (ii) $s' \in S'$.

Case (i): $s' \notin S'$. Let S' consist of $\bar{k}$ leaves. If s' does not belong in the star, then we have that $w(S') = \bar{k}$, as there is no edge of weight M in the star. Moreover, we have that $w(S', \bar{S}') \geq \bar{k}M$, as at least $\bar{k}$ nodes are connected to s' and, as such, belong to the cut. That leads to:

$$\frac{w(S', \bar{S}')}{w(S')} \geq \frac{\bar{k}M}{\bar{k}} = M > \ell$$

Case (ii): $s' \in S'$. By construction, s' is connected to every node in G: as such, a star containing s' has to either be centered at s' or contain only two nodes, in which case s' can be viewed either as a center or a leaf. Let the star S' have $\bar{k}$ leaves. We have that $\bar{k} < k$, otherwise the leaves of the star form an independent set of size k. We can calculate that the weight of the star $(w(S'))$ is $\bar{k}M$, whereas the cut $(w(S', \bar{S}'))$ is at least equal to $(n-\bar{k})M$. That leads to

$$\frac{w(S', \bar{S}')}{w(S')} \geq \frac{(n-\bar{k})M}{\bar{k}M}.$$

However we also know (by assumption) that

$$\frac{w(S', \bar{S}')}{w(S')} \leq \ell,$$

implying that

$$\frac{(n-\bar{k})M}{kM} \leq \ell \implies \frac{(n-\bar{k})M}{\bar{k}M} \leq \frac{(n-k)M + k(n-k)}{kM} \implies$$

$$\implies \frac{(n-\bar{k})M}{\bar{k}M} \leq \frac{(n-k)M}{kM} \implies$$
$$\implies \frac{n-\bar{k}}{\bar{k}} \leq \frac{n-k}{k} \implies \bar{k} \geq k$$

If that is the case, though, the $\bar{k}$ leaves of S' also form an independent set of size $\bar{k}$, which contradicts the fact that there exists no independent set of size k. This finishes the proof.

□

4 Formulations

In this section, we discuss mixed-integer programming formulations for the problem of detecting a normalized cut′, such that one of the two partitions satisfies a specific structural constraint. We have already established that the general form of the problem can be represented as in (3). We focus here on the typical normalized cut′ problem, and use as structural constraints the *clique* and *star* structures. Of course, problems of detecting different structures can be formulated by substituting the appropriate constraint(s).

4.1 Clique Normalized Cut′

Once more, we select here for exposition purposes that $f(S) = w(S, \bar{S})$ and $g(S) = w(S)$, and we assume that S is required to form a clique in a simple, undirected graph $G(V, E)$. We introduce the following binary variables:

$$x_i = \begin{cases} 1, & \text{if } i \text{ is in the clique } S \subseteq V \\ 0, & \text{otherwise.} \end{cases}$$
$$y_{ij} = \begin{cases} 1, & \text{if both } i, j \in S \\ 0, & \text{otherwise.} \end{cases}$$
$$z_{ij} = \begin{cases} 1, & \text{if } i \in S \text{ and } j \in \bar{S} \\ 0, & \text{otherwise.} \end{cases}$$

We can thus obtain the integer programming formulation presented in (9)

$$\min \quad \frac{\sum_{(i,j)\in E} w_{ij} z_{ij}}{\sum_{(i,j)\in E} w_{ij} y_{ij}} \tag{9a}$$
$$s.t. \quad x_i + x_j \leq 1, \qquad \forall (i,j) \notin E \tag{9b}$$
$$y_{ij} \leq x_i, \qquad \forall (i,j) \in E \tag{9c}$$
$$y_{ij} \leq x_j, \qquad \forall (i,j) \in E \tag{9d}$$
$$z_{ij} \geq x_i - x_j, \qquad \forall (i,j) \in E \tag{9e}$$
$$z_{ij} \geq x_j - x_i, \qquad \forall (i,j) \in E \tag{9f}$$
$$x_i \in \{0,1\}, \qquad \forall i \in V \tag{9g}$$
$$y_{ij}, z_{ij} \in \{0,1\}, \qquad \forall (i,j) \in E, \tag{9h}$$

where the objective function (9a) captures the selected functions in the ratio ($f(S) = w(S, \bar{S})$, $g(S) = w(S)$), while constraint (9b) is the typical clique constraint where two nodes i, j are not

allowed to belong to S unless they are connected by an edge. Constraints (9c) and (9d) define variables y_{ij} to ensure they are equal to 1 whenever i and j are both in clique S. Similarly, constraints (9e) and (9f) ensure the proper definition of variables z_{ij}. Last, (9g) and (9h) restrict the decision variables to be binary.

It is interesting to note that constraints (9c) and (9d) are derived by the linearization of the constraint $y_{ij} = x_i x_j$. Observe though that the third linearization constraint ($y_{ij} \geq x_i + x_j - 1$) is unnecessary due to the fact that variables y_{ij} appear in the denominator of the objective function and, as such, are maximized. Last, constraints (9h), that enforce the binary nature of variables y_{ij} and z_{ij} can be relaxed. This is formally stated in Proposition 2.

Proposition 2. *Let $\mathcal{R}$ be a relaxed formulation (9), where variables y_{ij} and z_{ij} are allowed to be continuous. Then, there always exists an optimal solution to the problem, where y_{ij} and z_{ij} are binary.*

Proof. From constraints (9c) and (9d), we get that y_{ij} is positive, if and only if both $x_i = 1$ and $x_j = 1$. However, if that is the case, then y_{ij} has to be equal to 1 in an optimal solution, as the bigger it gets, the smaller the objective function. Similarly, from constraints (9e) and (9f), we obtain that z_{ij} can be strictly smaller than 1 if x_i and x_j are equal. Once more, if that is the case, then z_{ij} cannot be optimal unless it is equal to 0, as the smaller it gets, the smaller the objective function value, too. □

4.2 Star Normalized Cut′

Let us introduce two new variables, while redefining the variables x_i from before. The definitions appear below.

$$x_i^\ell = \begin{cases} 1, & \text{if } i \text{ is a leaf of the star } S \\ 0, & \text{otherwise.} \end{cases}$$

$$x_i^c = \begin{cases} 1, & \text{if } i \text{ is the center of the star } S \\ 0, & \text{otherwise.} \end{cases}$$

$$x_i = \begin{cases} 1, & \text{if } i \text{ is in the star } S \subseteq V \\ 0, & \text{otherwise.} \end{cases}$$

The formulation can now be written as in (10).

$$\min \quad \frac{\sum_{(i,j)\in E} w_{ij} z_{ij}}{\sum_{(i,j)\in E} w_{ij} y_{ij}} \tag{10a}$$

$$\begin{aligned} s.t. \quad & x_i^\ell + x_j^\ell \leq 1, && \forall (i,j) \in E \\ & x_i^\ell \leq \sum_{j \in N(i)} x_j^c, && \forall i \in V \qquad (10b) \\ & \sum_{i \in V} x_i^c = 1 \\ & x_i \leq x_i^\ell + x_i^c, && \forall i \in V \qquad (10c) \\ & (9c) - (9g) && (10d) \\ & x_i^\ell, x_i^c \in \{0,1\}, && \forall i \in V \qquad (10e) \end{aligned}$$

Here, the objective function is the same as in the clique version. However, in this case, (10b) contain three families of constraints: the first one signals that two nodes that are connected by an edge cannot both serve as leaves of the star, the next one enforces that a node cannot be a leaf unless it is connected to the center, and last, there needs to be exactly one center node selected in the graph. Furthermore, constraints (10c) are included so as to allow us to define variables y_{ij} and z_{ij} for the edges following the same constraints as before. Last, exactly as in Proposition 2, the binary nature of variables y_{ij} and z_{ij} is enforced by the model and the corresponding restriction (9h) is omitted.

4.3 Solution Approaches

To deal with the fractional programming problems formulated here, we can use one of the following techniques:

- **Linearization**

  The formulations presented so far are all fractional programs, and as such, pose significant computational challenges (Wu, 1997). One of the approaches that are typically used to tackle these problems is to transform the ratio by introducing a variable v, defined as $\frac{1}{g(S)}$ and then *linearize* the bilinear terms of the resulting problem (11).

$$\begin{aligned} \min \quad & vf(S) && \text{(11a)} \\ s.t. \quad & S \in \mathcal{S}, && \text{(11b)} \\ & vg(S) = 1, && \text{(11c)} \\ & S \subseteq V && \text{(11d)} \end{aligned}$$

  The linearization of both the objective function (11a) and constraints (11c) is however costly in terms of both the number of variables and the number of constraints that need to be added, particularly when graph G is dense. An improved linearization scheme for general fractional programming problems can be found in Borrero et al. (2016).

- **Linear Scalarization**

  Another approach that is common when presented with conflicting objectives, is to formulate the problem as a single objective optimization problem and appropriately scale the objective functions. The solutions to the new optimization problems are Pareto optimal solutions to the original one (Hwang et al., 1979). In our case, this means formulating the single ratio problem shown before as in (12), where $\lambda \geq 0$.

$$\begin{aligned} \min \quad & f(S) - \lambda g(S) && \text{(12a)} \\ s.t. \quad & S \in \mathcal{S}, && \text{(12b)} \\ & S \subseteq V && \text{(12c)} \end{aligned}$$

- **ϵ-constraint method**

  Finally, the well-known ϵ-constraint method is a viable approach for tackling problems in multiobjective optimization (Miettinen, 1999). In our context, we can rewrite the ratio problem as a *minimization* (resp. *maximization*) of a single objective function, while imposing a

lower (resp. *upper*) bound on the other objective. Using the above, our formulation can be given in terms of the programs in (13) and (14).

$$\begin{aligned} \min \quad & f(S) && \text{(13a)} \\ s.t. \quad & g(S) \geq \epsilon, && \text{(13b)} \\ & S \in \mathcal{S}, && \text{(13c)} \\ & S \subseteq V && \text{(13d)} \end{aligned} \qquad \begin{aligned} \max \quad & g(S) && \text{(14a)} \\ s.t. \quad & f(S) \leq \epsilon, && \text{(14b)} \\ & S \in \mathcal{S}, && \text{(14c)} \\ & S \subseteq V && \text{(14d)} \end{aligned}$$

One of the challenges that arise when using the linear scalarization and the ϵ-constraint methods is finding suitable values for λ and ϵ (Hochbaum, 2010). This is because the quality of the obtained solutions is often highly sensitive to these values. To produce high quality solutions, it is often required to perform several trial tests with different candidate values until reaching close to optimality. This can be achieved through binary search. It is also worth mentioning that in the case of linear functions in the ratio of the objective, we can apply a linear transformation following the work of Charnes and Cooper (1962), similarly to the above ϵ-constraint method.

5 Other Techniques and Further Enhancements

In this section we investigate two greedy algorithms for solving the clique and star normalized cut$'$ problems, as well as a combinatorial decomposition approach to tackle the former over highly sparse instances.

5.1 Greedy Algorithms

The greedy algorithms described in this section are construction heuristics that iteratively grow node set S based on the score given by an improvement metric. At each iteration, the algorithm adds to S the node with the largest score, until no further node produces an improvement. We will further show that for some of the metrics we propose, the greedy algorithms yield an approximation guarantee.

We begin with a greedy algorithm for the induced star normalized cut$'$ problem. The algorithm starts with a given node k which is set as the center of the star S and then it selects greedily the leaves from $\mathcal{N}(k)$. To find a solution for the star normalized cut$'$, the process can be repeated n times, starting each time with a different node in $i \in V$ as the center of S and then selecting the overall best.

Let $k \in V$ be the center of the induced star, and $\mathcal{I}$ the set of all nodes that can be added to the star; initially $\mathcal{I} = \mathcal{N}(k)$. Subsequently, assuming an induced star S centered at k has been built, we have that $\mathcal{I} = \{i \in \mathcal{N}(k) : (i,j) \notin E, \forall j \in S \setminus \{k\}\}$. Once more, let $f(S) = \sum_{i \in S} \sum_{j \notin S} w_{ij}$ and $g(S) = \sum_{i \in S} \sum_{j \in S} w_{ij}$. We also define here as Y_j the summation of all edge weights from node j to any node outside $\mathcal{I} \cup \{k\}$. Then, we have for every node $i \in \mathcal{I}$:

$$\frac{f(S \cup \{i\})}{g(S \cup \{i\})} = \frac{f(S) - w_{ik} + \sum\limits_{\{j \notin S, (i,j) \in E\}} w_{ij}}{g(S) + w_{ik}} = \frac{f(S) - w_{ik} + Y_i + \sum\limits_{\{j \in \mathcal{I}:(i,j) \in E\}} w_{ij}}{g(S) + w_{ik}}. \tag{15}$$

Observe that (15) shows that adding node i to S improves the objective function only if

$$\frac{-w_{ik} + Y_i + \sum\limits_{\{j \in \mathcal{I}:(i,j) \in E\}} w_{ij}}{w_{ik}} \leq \frac{f(S)}{g(S)}.$$

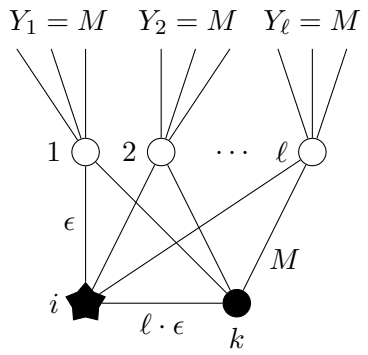


Figure 3: The worst case scenario for the simple greedy algorithm.

At each step then, we can easily evaluate the ratio $r_i = \frac{-w_{ik}+Y_i+\sum_{\{j\in\mathcal{I}:(i,j)\in E\}} w_{ij}}{w_{ik}}$. Our greedy approach, shown in Algorithm 1, selects at each step the node $\hat{i} = \arg\min_{i\in\mathcal{I}} r_i$, until there is no node $i \in \mathcal{I}$ such that $r_i \leq \frac{f(S)}{g(S)}$.

Algorithm 1: Greedy Star Normalized Cut′.

```
function GreedyNormalizedStar (k);
Input  : A node k ∈ V
Output: An induced star S centered at k
I ← N(k);
S ← {k};
continue ← true;
while continue do
    î ← arg min_{i∈I} r_i;
    if r_î ≤ f(S)/g(S) then
        S ← S ∪ {î};
        L = {j : (î, j) ∈ E};
        I ← I \ L;
    else
        continue ← false;
    end
end
return S
```

Unfortunately, the greedy algorithm using this particular ratio has no approximation guarantee. This is depicted in Figure 3, where the center of the star is k and the set $\mathcal{I}$ consists originally of nodes i and $1, 2, \ldots, \ell$. In the Figure, the optimal solution is the star centered at k having as leaves all nodes $j = 1, \ldots, \ell$ of value $(\sum_{j=1}^{\ell} w_{ij} + \sum_{j=1}^{\ell} Y_j)/\sum_{j=1}^{\ell} w_{jk}$, whereas the greedy solution would consist of the star with only one leaf (node i) with objective function value equal to $(\sum_{j=1}^{\ell} w_{ij} + \sum_{j=1}^{\ell} w_{kj})/w_{ik}$. Letting $Y_j = w_{jk} = M$, where M is assumed to be big enough, and also letting $w_{ik} = \ell \cdot \epsilon$ with $w_{ij} = \epsilon$ for $j = 1, \ldots, \ell$, results in the following ratio:

$$\frac{z_{greedy}}{z_{opt}} = \frac{\frac{\ell\cdot\epsilon+\ell\cdot M}{\ell\cdot\epsilon}}{\frac{\ell\cdot\epsilon+\ell\cdot\epsilon+\ell\cdot M}{\ell\cdot M}} \to M.$$

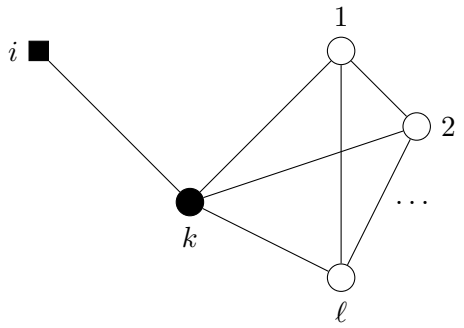


Figure 4: An example of how the simple ratio greedy algorithm fails in the case of cliques. In this case, starting from node k, node i is the best choice, having $r_i = -1 < r_j, j = 1, \ldots, \ell$. However, this selection would lead to a greedy solution of value $\frac{\sum_{j=1}^{\ell} w_{jk}}{w_{ik}}$ as compared to the optimal solution of value $\frac{w_{ik}}{\sum_{j=1}^{\ell} w_{jk}}$.

However, the greedy algorithm can be improved by considering not only the increase in the weight of the cut brought on by adding a new leaf, but also the increase in the weight of the cut from the nodes connected to the center that can no longer be leaves. Hence, we can introduce a new ratio

$$r_i = \frac{-w_{ik} + Y_i + \sum\limits_{\{j \in \mathcal{I}:(i,j)\in E\}} (w_{ij} + w_{jk})}{w_{ik}}. \tag{16}$$

The greedy algorithm can then be designed like Algorithm 1, with the ratio calculated as in (16). We can now show that this greedy algorithm is an $\mathcal{O}(\Delta)$-approximation algorithm, where Δ is the maximum degree of a node in the graph. To do that, we show that the greedy algorithm when provided a center node k with a degree of δ_k produces an $\mathcal{O}(\delta_k)$ approximation guarantee.

Theorem 4. *Let k be the center node of S, with a degree of δ_k. Then, Algorithm 1 using ratios*

$$r_i = \frac{-w_{ik} + Y_i + \sum\limits_{\{j \in \mathcal{I}:(i,j)\in E\}} (w_{ij} + w_{jk})}{w_{ik}}$$

for all $i \in V$ is an $\mathcal{O}(\delta_k)$-approximation algorithm.

The proof of Theorem 4 can be found in Appendix A.

We now continue with a greedy approach for the clique version. We first redefine the list of admissible candidates for the clique, considering a clique S has already been built, as $\mathcal{I} = \{i : (i,j) \in E, \forall j \in S\}$. Then, similarly to the star case, a simple ratio of

$$r_i = \frac{-\sum_{j\in S} w_{ij} + \sum_{j\notin S} w_{ij}}{\sum_{j\in S} w_{ij}}$$

can be used. However, selecting greedily according to that ratio fails to provide an approximation guarantee, as can be shown in Figure 4.

To bypass that issue, we will once more introduce a refined version of the ratio, similarly to the induced star case. To that extent, let us first define the sets $\mathcal{I}_i = \{j \in \mathcal{I} : (i,j) \in E\}$ and $\overline{\mathcal{I}_i} = \{j \in \mathcal{I} : (i,j) \notin E\}$ to keep track of the nodes that can potentially still belong to the clique

Algorithm 2: Greedy Clique Normalized Cut′.

function GreedyNormalizedClique (k);
Input : A node $k \in V$
Output: A clique S containing node k
$\mathcal{I} \leftarrow N(k)$;
$S \leftarrow \{k\}$;
$continue \leftarrow true$;
while *continue* **do**
  $\hat{i} \leftarrow \arg\min_{i \in \mathcal{I}} r_i$;
  if $r_{\hat{i}} \leq \frac{f(S)}{g(S)}$ **then**
    $S \leftarrow S \cup \{\hat{i}\}$;
    $L = \{j : (\hat{i}, j) \notin E\}$;
    $\mathcal{I} \leftarrow \mathcal{I} \setminus L$;
  else
    $continue \leftarrow false$;
  end
end
return S

and the nodes that are no longer plausible candidates after a node i has been entered, respectively. Now, assuming a clique S is being built starting from a node k, we can define:

$$r_i = \frac{-w_{ik} - \sum_{\ell \in \mathcal{I}_i} w_{i\ell} + \sum_{\ell \in \overline{\mathcal{I}_i}} w_{k\ell} + Y_i}{w_{ik} + \sum_{\ell \in \mathcal{I}_i} w_{i\ell}},$$

or, upon construction of a clique S, we can use:

$$r_i = \frac{-\sum_{\ell \in S} w_{i\ell} - \sum_{\ell \in \mathcal{I}_i} w_{i\ell} + \sum_{\ell \in S} \sum_{p \in \overline{\mathcal{I}_i}} w_{\ell p} + Y_i}{\sum_{\ell \in S} w_{i\ell} + \sum_{\ell \in \mathcal{I}_i} w_{i\ell}}. \tag{17}$$

In plain terms, this ratio captures the *potential* changes in the numerators and denominators by adding a node i. Seeing as not all nodes in $\mathcal{I}_i$ will be admissible in the clique (unless they form a complete subgraph themselves) this ratio can be viewed as an upper bound on the potential of a node. Assuming a clique S, the same logic as before as far as discarding nodes with a ratio bigger than $\frac{f(S)}{g(S)}$ carries here.

While this greedy approach with the ratio as defined above does not yield a theoretical bound of performance, it behaves very well in practice, as shown in Section 6. We now proceed to show that there can never exist a better approximation algorithm for this problem than $\mathcal{O}(n)$.

Let us go back to the gadget we used for reducing Clique to Clique Normalized Cut′ in Section 2. We will show that if clique normalized cut′ can be approximated then so can the maximum clique problem, which is known to be hard to approximate within $n^{1-\epsilon}$ (Håstad, 1999).

Lemma 1. *Clique normalized cut′ is inapproximable within a factor of* $\mathcal{O}(n^{1-\epsilon})$.

Proof. Consider again the gadget of Theorem 2, and let $\mathcal{A}$ be an algorithm that can approximate clique normalized cut$'$ within a factor of μ of $\mathcal{O}(n^{1-\epsilon})$. Note that, by construction, if S^* is the optimal solution to the clique normalized cut$'$ problem, then $S^* \setminus \{s'\}$ is the maximum clique in the original graph. We can further note that for any clique S in G' that does not include s', we have that $z(S) \geq z(S \cup \{s'\})$, and hence, wlog we can assume that the clique $\hat{S}$ returned by $\mathcal{A}$ contains s'.

We further note that $g(S) = w(S) = |S| \cdot M + \binom{|S|}{2}$ and seeing as the term involving M is significantly bigger, we have that $|S_1| \leq \mu |S_2| \iff g(S_1) \leq \mu g(S_2)$. Overall, then, we have that:

$$\frac{f(\hat{S})}{g(\hat{S})} \leq \mu \cdot \frac{f(S^*)}{g(S^*)} \implies g(S^*) \leq \mu \frac{f(S^*)}{f(\hat{S})} g(\hat{S}) \implies |S^*| \leq \mu \frac{f(S^*)}{f(\hat{S})} |\hat{S}|. \tag{18}$$

To prove the result, we need show that $f(S^*)/f(\hat{S}) \leq 1$. Remember that, as mentioned above, we can assume that $s' \in \hat{S}$. We distinguish between two cases:

1. $|\hat{S}| = |S^*|$: in that case, $g(\hat{S}) = g(S^*)$ by construction. Since, though, $f(S^*)/g(S^*) \leq f(\hat{S})/g(\hat{S})$, this implies that $f(S^*) \leq f(\hat{S})$, otherwise $\hat{S}$ is a better solution to the clique normalized cut$'$ problem than the optimal clique, S^*.

2. $|\hat{S}| < |S^*|$: then, by construction $f(\hat{S}) = (|V| - |\hat{S}|)M + o(1)$ while $f(S^*) = (|V| - |S^*|)M + o(1)$, and since $|V| - |\hat{S}| > |V| - |S^*|$, then also $f(S^*)/f(\hat{S}) < 1$.

Inequality (18), then, combined with the two cases, shows that $\mathcal{A}$ can be used to then approximate the maximum clique of G through the gadget of Theorem 2 within a factor of $\mathcal{O}(n^{1-\epsilon})$, which finally proves the theorem.

□

5.2 Clique Normalized Cut$'$ over Graphs with Low Degeneracy

In this section we describe a decomposition scheme that takes advantage of the graph's sparseness to reduce the computational burden of identifying clique normalized cut$'$ partitions. The proposed scheme is inspired by two algorithms designed to solve other clique-related problems: the maximum clique (Buchanan et al., 2014) and the maximal clique enumeration problem (Eppstein et al., 2010). We follow a similar approach and provide evidence that clique normalized cut$'$ over highly sparse graphs, such as many social networks (Verma et al., 2015), can be identified quite efficiently.

From the perspective of the normalized clique cut$'$ problem, as proven by Proposition 2, the main difficulty behind formulation (9) lies on the integrality of the variables that define set S. Interestingly, given the clique structural requirements that are imposed over set S (i.e., constraints (9b)), the size of S is bounded by the clique number (ω) of G (i.e., $|S| = \sum_{i \in V} x_i \leq \omega$). This observation leads to the fact that in any feasible solution to the clique normalized cut$'$ problem over a sparse graph, only a relative small proportion of binary variables will take the value of one.

To obtain a clique normalized cut$'$, we can naturally take advantage of techniques that work well in practice for solving maximum clique and other related problems. There are two challenges that must be taken into account to successfully adapt such techniques for the normalized cut$'$ context. First, identifying ω is also known to be $\mathcal{NP}$-hard, which means that the overall scope of these techniques is limited by the structure of the instances, or in this case the sparseness of the graph. Second, most clique problems, like maximum clique, often have as objective a non-decreasing function with respect to the size of the clique set S, which brings the advantage that

the search can be restricted to maximal cliques. On the other hand, given the fractional nature of the clique normalized cut$'$ objective, it is easy to see that a non-maximal clique S can be a unique optimal solution to the problem.

As mentioned before, we will restrict our attention to graphs that are highly sparse. A common way to account for the sparseness of a graph is by its degeneracy (also referred as the core number or the coloring number of a graph). It is well-known that many real-life graphs like social networks, hub-and-spoke transportation networks, citation networks, and protein interaction networks have low degeneracy (Chung, 2010).

Definition 4 (degeneracy (Lick and White, 1970))**.** *A graph is said to be d-degenerate if every (non-empty) subgraph has a node of degree at most d. The degeneracy of a graph is the smallest value of d such that it is d-degenerate.*

Since the size of the largest clique in a d-degenerate graph is at the most $d+1$, the size of the clique in any normalized clique cut$'$ is therefore limited by such a bound too ($|S| \leq d+1$). Furthermore, it is known that a d-degenerate graph admits an ordering $(v_1, v_2, \ldots, v_n)$ of its nodes such that each node in the ordering has at most d neighbors after it (Lick and White, 1970). The degeneracy, as well as such an ordering, can be found in $O(m)$ time by iteratively removing a node of minimum degree (Matula and Beck, 1983). As a reminder, m is assumed to be the number of edges in the considered graph.

The following proposition is a direct consequence of Lemma 1 in Buchanan et al. (2014) and it is the key for the proposed decomposition.

Proposition 3. *Let $(v_1, \ldots, v_n)$ be any node-ordering of V. Then,*

$$\min_{S \subset V} \frac{f(S)}{g(S)} = \min_{1 \leq i \leq n} \left\{ \min_{T \subseteq V_i} \frac{f(T \cup \{v_i\})}{g(T \cup \{v_i\})} \right\},$$

where $V_i = \mathcal{N}(v_i) \cap \{v_i, \ldots, v_n\}$.

Proof. Clearly, $\min_{S \subset V} \frac{f(S)}{g(S)} \leq \min_{T \subseteq V_i} \frac{f(T \cup \{v_i\})}{g(T \cup \{v_i\})}$ for any node $v_i \in V$. Now, let $S^* = \arg\min_{S \subset V} \frac{f(S)}{g(S)}$ be the clique in the optimal clique normalized cut$'$ partition and let $v_{i^*} \in S^*$ be its earliest node in the node-ordering. Then $S^* \setminus \{v_i\} \subseteq V_{i^*}$, implying that $\min_{T \subseteq V_{i^*}} \frac{f(T \cup \{v_i\})}{g(T \cup \{v_i\})} \leq \frac{f(S^*)}{g(S^*)}$. □

We now finally provide a decomposition algorithm for detecting a clique normalized cut$'$ partition on d-degenerate graphs in Algorithm 3. For a pictorial example, we direct the reader to Appendix B.

6 Computational Results

Our computational results aim to show the success of the decomposition scheme, and the quality of solutions obtained by the greedy algorithms for several synthetic and real-life networks. In this section, we first describe the experimental setup and the instances, and then provide a series of tables with our runtime and approximation ratio observations. We finally give an analysis of our observations.

Algorithm 3: A decomposition scheme for solving the clique normalized cut′ problem on d-degenerate graphs.

Data: A graph $G = (V, E)$
Result: A clique S that minimizes $\frac{f(S)}{g(S)}$

compute a degeneracy ordering $(v_1, \ldots, v_n)$ of G;
for $i = 1, \ldots, n$ **do**
  $V_i \leftarrow \mathcal{N}(v_i) \cap \{v_i, \ldots, v_n\}$;
  $S_i \leftarrow \{v_i\} \cup \arg\min_{T \subseteq V_i} \frac{f(T)}{g(T)}$; `// optimize (9) for` $G[\mathcal{N}[V_i]]$ `enforcing` $x_{v_i} = 1$
end
return $\min_{1 \leq i \leq n} \frac{f(S_i)}{g(S_i)}$;

6.1 Experimental setup

All results presented here were obtained on a computational cluster with eight nodes. Each node has a 6-core Intel Xeon E5-2643 v3 3.4GHz processor and 128 GB of RAM, while the operating system is Linux x86_64 CentOS 7.2. The formulations were modeled and solved using Gurobi 5.63 in C++, the decomposition scheme was implemented in C++ with each subproblem solved by Gurobi 5.63, and the greedy approaches were implemented in Python employing the network package of NetworkX (Hagberg et al., 2008). A time limit of 3 hours (i.e., 10,800 seconds) was selected for all instances. A description of the instances follows.

6.2 Instances

We opted for both real-world and randomly generated instances. For the synthetic networks, we selected to generate Erdös-Rényi networks (Erdös and Rényi, 1959), denoted by RND, Barabási-Albert networks (Albert and Barabási, 2002), denoted by BA, and Watts-Strogatz networks (Watts and Strogatz, 1998), denoted by WS. Instances were generated to have different network sizes, ranging from 30 to 1,000 nodes and 60 to 30,000 edges, with the number of edges being slightly different from configuration to configuration due to the generators. For each of the configurations we randomly generated three instances: the results reported are the average runtimes and approximation ratios obtained for each setup.

For the real-world networks, now, we selected some well-known benchmarks that have been prominently used in other computational studies. These are summarized briefly below:

- **Social networks instances from books, films, and pop culture:** `anna`, `david`, `huck`, and `jean` are all social networks obtained by the well-known books *Anna Karenina* by Leon Tolstoy, *David Copperfield* by Charles Dickens, *Adventures of Huckleberry Finn* by Mark Twain, and *Les Misérables* by Victor Hugo; `forrest-gump` and `titanic` are social networks obtained by these two popular movies; last, `Lindenstrasse` is a social network obtained by a German TV show.

- **Social networks commonly used as benchmarks:** `attiro` and `sanjuansur` are social networks of families in the Turrialba region in a rural area of Costa Rica (more specifically, the second one is of San Juan Sur) and was obtained by American sociologists in 1948 (Loomis et al., 1953); `krebs` is another well-studied social network portraying the ties of the 9/11 hijackers and their associates; `prison` is a social network of inmates; `mexican` is the

social network of prominent political figures in Mexico (Gil-Mendieta and Schmidt, 1996); `high-tech` is the relationship network of people employed in a high-tech company (Krackhardt, 1999); `jazz` lists the collaborations between jazz musicians (Gleiser and Danon, 2003); `karate` is the social network of a karate club (Zachary, 1977); `dolphins` is a network of the diverse associations of dolphins studied in Doubtful Sound, New Zealand (Lusseau et al., 2003); `sawmill` is a social network of ties in a small sawmill company (Michael and Massey, 1997).

- **Infrastructure networks:** `miles250` is the transportation network of U.S. cities in the 1940s; `ieeebus` is the well-known 118 Bus Power Flow Test Case of a U.S. Midwestern power subsystem as of 1962.

All smaller real-life instances were solved five times: the times reported is the average runtime experienced per instance.

6.3 Results

In Table 1 we present the computational runtime for the real-life instances, while the results for the Watts-Strogatz, Erdös-Renyi, Barabási-Albert instances are shown in Table 2. Specifically, we provide the runtime (in seconds) of the linearization (Exact), the decomposition approach presented in subsection 5.2 for cliques (Decomp), and the greedy approaches for both cliques and stars (Greedy). All instances are separated into smaller and larger scale networks. We also note here that we tested the performance of linear scalarization and the ϵ-constraint approaches (as described in subsection 4.3). However, they were consistently outperformed by the linearization of the ratio formulation and hence they are omitted from the result analysis.

From the results obtained, we can see that the greedy approaches are consistently faster. From the exact solution methods, the decomposition approach outperforms the linearization in most synthetic networks as the number of edges decreases. However, in denser synthetic networks, we can note that the decomposition approach is slower. In larger scale synthetic networks though, we note that the decomposition approach is consistently faster than the linearization. An interesting observation from the greedy approaches is obtained when focusing on larger scale networks. For stars, the greedy approach is more dependent on the size of the network, with density affecting little in the runtime. On the other hand, in the case of cliques, the runtime is affected by both number of nodes and density. Last, in all real-life instances, we note the same pattern: the decomposition outperforms the linearization in most instances. A notable example is the `jazz` instance, which is a denser network consisting of 198 vertices and 5,484 edges.

As far as the optimality gaps and approximation ratios are concerned, they are presented in Tables 3 and 4. Let us begin with the average and worst case approximation ratios as observed on the smaller benchmark instances and as contrasted with the maximum and average node degree in each graph. The first observation has to do with the fact that the worst case approximation guarantee is never tight: in every instance, and on both *clique* and *star* normalized cut$'$, the observed approximation ratio is well within the maximum degree of the graph. Moreover, on average, the greedy algorithm for *clique* normalized cut$'$ is impressively within a factor of 2 of the optimal solution. The same is unfortunately not true for the *star* normalized cut$'$; the ratio, in practice, for this version of the problem is almost always worse than the ratio obtained for the clique. This is further corroborated by our results in Table 4, where the clique normalized cut$'$ approximation is consistently (both on average and in the worst case) better than the approximation for the star.

In Table 4, we further note that both linearization and decomposition always find an optimal solution in all instances that are described as smaller scale. In larger scale instances, though, it is

easy to see that on average our proposed decomposition approach provides us with better quality solutions. The greedy approaches also perform well and provide high quality solutions in most smaller scale instances. It is when we move to larger scale instances that the greedy approaches behave worse, especially for the star normalized cut$'$ problem. Moreover, Table 5 reveals what fraction of instances were successfully solved to optimality for each of the approaches. As noted earlier, both the linearization and the decomposition techniques provide us with an optimal solution in all smaller scale instances, where also the greedy techniques perform adequately well. It is in larger scale instances though that the exact techniques begin running out of memory, while the greedy algorithms fail to consistently find an optimal solution.

Finally, to provide a specific discussion of the solutions obtained by our models over a real-life network, we describe the *clique* and *star normalized cut*$'$ partitions obtained over the Les Misérables social network. The network and both the resulting clique (in red) and star (in blue) are shown in Figure 5. Analyzing the partitions, we observe that the members of the clique are students and their friends. These characters are related to *Fantine*, one of the protagonists, through Tholomyés, the biological father of *Cosette*. The second group is even more tangentially related to the main plot. The induced star is centered at *Bishop Myriel*, a character that albeit his limited appearance time, has a large effect in *Jean Valjean's* redemption tale. The star consists of the bishop, relatives of the bishop (like Madame la Comtesse de Lo), and other characters that only appear in the beginning of the work (for example, Monsieur Geborand).

Table 1: Computational runtime (in seconds) for *clique* and *star normalized cut*$'$ on some known benchmark instances.

			Clique			Star	
Instance	$\|V\|$	$\|E\|$	Exact	Decomp	Greedy	Exact	Greedy
anna	138	493	1.76	2.44	1.45	4.72	1.53
attiro	59	128	0.44	0.09	0.43	2.29	1.30
david	87	406	1.21	1.77	1.21	2.75	2.2
dolphins	62	159	0.39	0.20	0.46	1.53	1.13
forrest-gump	94	271	0.80	0.44	0.84	8.39	1.91
high-tech	33	91	0.25	0.12	0.28	0.35	0.43
huck	69	297	0.55	0.76	0.53	7.66	2.41
ieeebus	118	179	1.19	0.11	1.07	2.83	2.03
jazz	198	2742	8.17	19.45	18.37	28.17	7.98
jean	77	254	0.67	0.32	0.70	2.65	1.43
karate	34	78	0.14	0.07	0.07	0.27	0.33
krebs	62	153	0.42	0.08	0.47	0.92	0.24
lindenstrasse	232	303	2.55	0.77	2.61	9.71	2.10
maxican	35	117	0.31	0.09	0.33	0.61	0.36
miles250	92	327	0.81	0.33	0.89	3.34	1.52
prison	67	142	0.46	0.38	0.34	1.80	0.97
sanjuansur	75	144	0.51	0.27	0.37	3.93	1.14
sawmill	36	62	0.16	0.04	0.23	0.19	0.25
titanic	70	299	0.76	0.83	0.67	1.63	0.95
email	1,133	10,902	615	257	92	7,200	933
ego-facebook	2,981	4,888	1,986	808	180	1,193	63
petster-friendships-hamster	18,58	12,534	3,419	992	275	10,435	947

Table 2: Computational runtime (in seconds) for *clique* and *star normalized cut*′ on Watts-Strogatz, Barabási-Albert and Erdös-Rényi synthetic networks.

		Watts-Strogatz					Barabási-Albert					Erdös-Rényi				
		Clique			Star		Clique			Star		Clique			Star	
n	m	Exact	Decomp	Greedy	Exact	Greedy	Exact	Decomp	Greedy	Exact	Greedy	Exact	Decomp	Greedy	Exact	Greedy
	60	0.21	0.23	0.25	0.24	0.21	0.25	0.25	0.27	0.53	0.39	0.37	0.35	0.29	0.44	0.38
30	90	0.39	0.35	0.29	0.33	0.24	0.38	0.36	0.31	0.68	0.45	0.50	0.37	0.38	0.51	0.42
	180	0.55	0.56	0.33	0.60	0.25	0.62	0.44	0.36	0.72	0.47	0.65	0.60	0.43	0.55	0.43
	80	0.24	0.25	0.30	0.25	0.23	0.31	0.25	0.27	0.55	0.38	0.44	0.35	0.36	0.56	0.37
40	120	0.52	0.45	0.42	0.37	0.28	0.60	0.37	0.33	0.68	0.42	0.51	0.43	0.39	0.58	0.41
	240	0.61	0.57	0.45	0.50	0.31	0.78	0.51	0.39	0.75	0.48	0.74	0.61	0.45	0.61	0.45
	120	0.32	0.27	0.33	0.34	0.30	0.41	0.33	0.31	0.62	0.41	0.51	0.36	0.38	0.53	0.35
60	180	0.55	0.48	0.47	0.60	0.41	0.61	0.47	0.44	0.75	0.47	0.57	0.47	0.42	0.61	0.42
	360	0.75	0.61	0.50	0.77	0.51	0.88	0.55	0.45	0.79	0.51	0.78	0.65	0.58	0.65	0.55
	140	0.36	0.33	0.35	0.34	0.30	0.44	0.35	0.34	0.77	0.43	0.55	0.41	0.40	0.58	0.43
70	210	0.68	0.63	0.67	0.62	0.55	0.82	0.51	0.45	0.81	0.50	0.73	0.51	0.44	0.63	0.53
	420	0.82	0.64	0.72	0.85	0.60	1.03	0.67	0.60	0.85	0.55	0.98	0.81	0.60	0.67	0.63
	200	0.51	0.45	0.44	0.41	0.37	0.52	0.37	0.35	0.89	0.48	0.63	0.58	0.55	0.78	0.52
100	300	0.78	0.67	0.66	0.80	0.58	0.88	0.55	0.46	0.90	0.55	0.88	0.66	0.56	0.89	0.56
	600	1.03	0.80	0.78	0.92	0.65	1.12	0.78	0.69	1.07	0.62	1.18	0.97	0.66	1.23	1.14
	5,000	197	78	53	603	117	3,600	259	247	4,913	322	2,967	655	503	8,897	947
500	10,000	919	196	80	1,892	116	4,734	693	329	10,800	356	4,107	1,591	507	10,800	1,013
	15,000	1,638	265	91	3,449	125	6,877	1,480	371	10,800	416	10,800	2,731	571	10,800	1,104
	10,000	10,329	2,330	637	10,800	2,103	7,104	2,293	936	8,190	684	10,800	10,800	1,001	10,800	1,132
1,000	20,000	10,800	6,784	802	10,800	2,311	10,800	2,936	1,007	10,800	682	10,800	4,693	1,782	10,800	2,516
	30,000	10,800	10,189	826	10,800	2,355	10,800	3,778	1,233	10,800	703	10,800	10,800	2,393	10,800	3,244

Table 3: Average and worst case approximation ratios for *clique* and *star normalized cut′* on some of the benchmark instances.

	Statistics		Clique		Star	
Instance	Maximum degree	Average degree	Worst case approximation	Average approximation	Worst case approximation	Average approximation
anna-138	70	7.38	6.56	1.15	9.25	2.02
attiro-60	14	4.28	2.02	1.07	1.95	1.15
david-87	81	9.41	6.17	1.09	4.44	1.33
dolphis-62	11	5.10	1.33	1.06	1.21	1.07
forrest-gump-94	88	5.76	1.54	1.11	2.33	1.42
high-tech-33	16	5.66	1.50	1.10	1.55	1.17
huck-69	53	6.56	2.85	1.07	3.13	1.22
ieeebus-118	9	3.04	1.25	1.03	1.17	1.09
jazz-198	100	27.78	9.72	1.92	3.55	2.31
jean-77	34	6.70	2.07	1.21	3.07	1.53
karate-34	17	4.36	1.47	1.04	1.35	1.10
krebs-62	20	4.90	1.80	1.06	2.28	1.48
lindenstrasse-232	13	2.59	1.51	1.05	1.74	1.17
maxican-35	17	6.73	1.47	1.07	1.51	1.16
miles250-92	16	7.13	1.33	1.06	1.25	1.05
prison-67	11	4.25	1.60	1.10	2.52	1.33
sanjuansur-75	12	3.83	1.82	1.06	1.78	1.15
sawmill-36	13	3.61	2.12	1.09	2.07	1.21
titanic-70	46	8.62	2.02	1.17	3.86	1.59

Table 4: Average and worst case optimality gap (in %) per instance type per approach.

	Average Optimality Gap					Worst Case Optimality Gap				
	Clique			Star		Clique			Star	
Instance type	Exact	Decomp	Greedy	Exact	Greedy	Exact	Decomp	Greedy	Exact	Greedy
WS Small	0.00	0.00	5.97	0.00	13.16	0.00	0.00	37.13	0.00	53.46
WS Big	13.09	0.00	28.20	20.31	42.07	33.33	0.00	54.66	51.17	71.50
ER Small	0.00	0.00	8.46	0.00	18.23	0.00	0.00	28.11	0.00	46.17
ER Big	7.51	4.33	37.26	23.61	47.33	33.85	22.50	66.33	42.25	106.81
BA Small	0.00	0.00	6.11	0.00	10.98	0.00	0.00	26.34	0.00	43.48
BA Big	9.10	8.75	23.41	18.50	39.02	34.68	35.75	60.75	47.71	98.87
Other Small	0.00	0.00	5.47	0.00	8.16	0.00	0.00	19.37	0.00	26.74
Other Big	0.00	0.00	7.33	0.00	9.97	0.00	0.00	26.29	0.00	29.30

7 Concluding Remarks

In this paper, we are interested in identifying "unbiased" communities in general graphs that are satisfying both cut and cohesiveness criteria. More specifically, we showed that the problem of combining a normalized cut′ criterion with clique, γ-quasi-clique, or star structures is $\mathcal{NP}$-hard. We then derived the fractional programs to detect such structures, and provided their linearization along with a linear scalarization and an ϵ-constraint approach to solve them.

Another main contribution has to do with the development of two greedy heuristic algorithms for solving the clique and star versions of the problem. Both approaches were shown to be approximation algorithms, and their worst case performance guarantees were obtained. A decomposition

Table 5: Average percentage of instances solved to optimality per instance type per approach.

	Clique			Star	
Instance type	Exact	Decomp	Greedy	Exact	Greedy
WS Small	100.00	100.00	55.56	100.00	46.67
WS Big	77.78	100.00	16.67	61.11	5.56
ER Small	100.00	100.00	45.45	100.00	31.82
ER Big	33.33	72.22	5.56	16.67	5.56
BA Small	100.00	100.00	58.52	100.00	45.19
BA Big	77.78	88.89	11.11	72.22	11.11
Other Small	100.00	100.00	59.09	100.00	40.91
Other Big	100.00	100.00	33.33	100.00	33.33

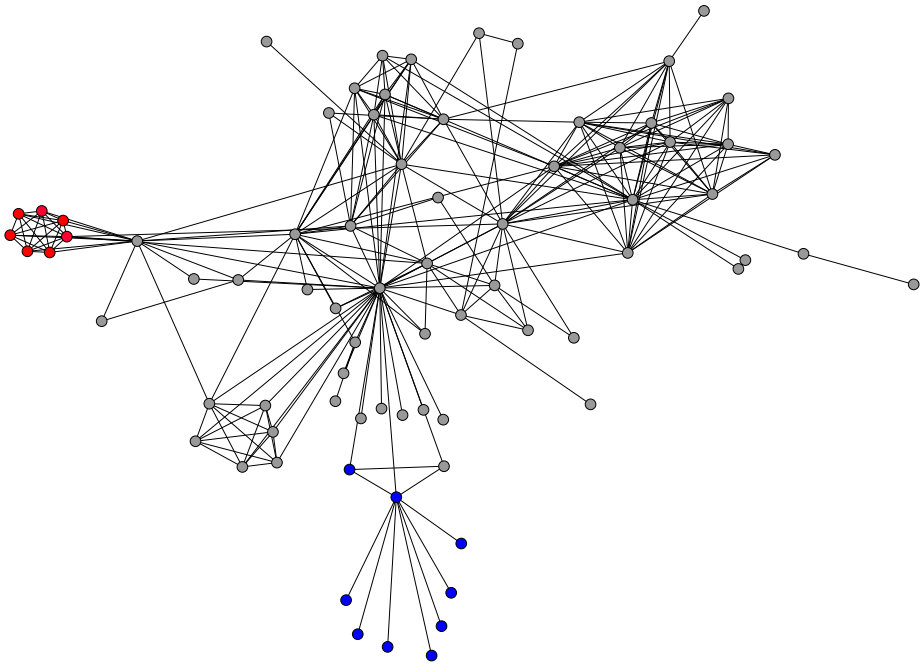


Figure 5: The clique and induced star produced with a normalized cut′ criterion on the `Les Misérables` social network. In red, the members of the clique are presented, while in blue we note the members of the star.

approach, shown to be very efficient in sparse networks, was also devised and implemented.

All of our approaches were put to the test in several synthetic and real-life networks. The computational results revealed that our approaches are efficient and can be used to solve the problems proposed here exactly and approximately.

Future work would look into incorporating more structural constraints and function families in the present framework. As mentioned before, this is a step towards "unbiased" community detection. A next step should investigate how to best relax the restrictive requirements of specific structures and allow for induced subgraphs that are similar, albeit not identical, to the desired structures.

Acknowledgments

This work was supported in part by the National Science Foundation Award CMMI-1635611, "Operational Decision-Making for Reach Maximization of Incentive Programs that Influence Consumer Energy-Saving Behavior". The authors would like to take this opportunity to thank Eduardo L. Pasiliao for his invaluable input during the inital stages of this paper, the Center for Computational Research (CCR) at the University at Buffalo for their computational support, and the two anonymous referees for their detailed and insightful comments, which helped us to significantly improve this paper.

References

Albert, R., Barabási, A. L., 2002. Statistical mechanics of complex networks. Reviews of modern physics 74 (1), 47.

Balasundaram, B., Butenko, S., Hicks, I. V., 2011. Clique relaxations in social network analysis: The maximum k-plex problem. Operations Research 59 (1), 133–142.

Berdica, K., 2002. An introduction to road vulnerability: what has been done, is done and should be done. Transport Policy 9 (2), 117 – 127.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-U., 2006. Complex networks: Structure and dynamics. Physics reports 424 (4), 175–308.

Borrero, J. S., Gillen, C., Prokopyev, O. A., 2016. A simple technique to improve linearized reformulations of fractional (hyperbolic) 0–1 programming problems. Operations Research Letters 44 (4), 479–486.

Buchanan, A., Walteros, J. L., Butenko, S., Pardalos, P. M., 2014. Solving maximum clique in sparse graphs: an $o(nm + n2^{d/4})$ algorithm for d-degenerate graphs. Optimization Letters 8 (5), 1611–1617.

Charnes, A., Cooper, W. W., 1962. Programming with linear fractional functionals. Naval Research Logistics (NRL) 9 (3-4), 181–186.

Cheeger, J., 1969. A lower bound for the smallest eigenvalue of the laplacian.

Chen, J., Yuan, B., 2006. Detecting functional modules in the yeast protein–protein interaction network. Bioinformatics 22 (18), 2283–2290.

Chung, F., 2010. Graph theory in the information age. Notices of the AMS 57 (6), 726–732.

Cox, I. J., Rao, S. B., Zhong, Y., 1996. "ratio regions": A technique for image segmentation. In: Pattern Recognition, 1996., Proceedings of the 13th International Conference on. Vol. 2. IEEE, pp. 557–564.

Deschamps, T., Cohen, L. D., 2001. Fast extraction of minimal paths in 3d images and applications to virtual endoscopy. Medical image analysis 5 (4), 281–299.

Enright, A. J., Van Dongen, S., Ouzounis, C. A., 2002. An efficient algorithm for large-scale detection of protein families. Nucleic acids research 30 (7), 1575–1584.

Eppstein, D., Löffler, M., Strash, D., 2010. Listing all maximal cliques in sparse graphs in near-optimal time. Algorithms and Computation, 403–414.

Erdös, P., Rényi, A., 1959. On random graphs, I. Publicationes Mathematicae (Debrecen) 6, 290–297.

Fiduccia, C. M., Mattheyses, R. M., 1982. A linear-time heuristic for improving network partitions. In: Design Automation, 1982. 19th Conference on. IEEE, pp. 175–181.

Fortunato, S., 2010. Community detection in graphs. Physics Reports 486 (3), 75–174.

Garey, M., Johnson, D., 1979. Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman and Co., New York.

Gil-Mendieta, J., Schmidt, S., 1996. The political network in mexico. Social Networks 18 (4), 355–381.

Gleiser, P. M., Danon, L., 2003. Community structure in jazz. Advances in complex systems 6 (04), 565–573.

Hagberg, A., Swart, P., S Chult, D., 2008. Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Laboratory (LANL).

Håstad, J., 1999. Clique is hard to approximate withinn1- ε. Acta Mathematica 182 (1), 105–142.

Hochbaum, D. S., 2010. Polynomial time algorithms for ratio regions and a variant of normalized cut. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (5), 889–898.

Hochbaum, D. S., 2013. A polynomial time algorithm for rayleigh ratio on discrete variables: Replacing spectral techniques for expander ratio, normalized cut, and cheeger constant. Operations Research 61 (1), 184–198.

Hwang, C.-L., Masud, A. S. M., Paidy, S. R., Yoon, K. P., 1979. Multiple objective decision making, methods and applications: a state-of-the-art survey. Vol. 164. Springer Berlin.

Karypis, G., Kumar, V., 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on scientific Computing 20 (1), 359–392.

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

Krackhardt, D., 1999. The ties that torture: Simmelian tie analysis in organizations. Research in the Sociology of Organizations 16 (1), 183–210.

Lick, D. R., White, A. T., 1970. k-degenerate graphs. Canad. J. Math 22, 1082–1096.

Loomis, C. P., Clifford, J. O., RA Leonard, O., 1953. Turrialba: social systems and the introduction of change. Tech. rep., Free Press.

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., Dawson, S. M., 2003. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behavioral Ecology and Sociobiology 54 (4), 396–405.

Matula, D. W., Beck, L. L., 1983. Smallest-last ordering and clustering and graph coloring algorithms. Journal of the ACM 30 (3), 417–427.

Michael, J. H., Massey, J. G., 1997. Modeling the communication network in a sawmill. Forest Products Journal 47 (9), 25.

Miettinen, K., 1999. Nonlinear multiobjective optimization. Vol. 12. Springer.

Newman, M. E., Girvan, M., 2004. Finding and evaluating community structure in networks. Physical review E 69 (2), 026113.

Pržulj, N., Corneil, D. G., Jurisica, I., 2004. Modeling interactome: scale-free or geometric? Bioinformatics 20 (18), 3508–3515.

Schaeffer, S. E., 2007. Graph clustering. Computer science review 1 (1), 27–64.

Schenker, A., Last, M., Bunke, H., Kandel, A., 2003. Classification of web documents using a graph model. In: Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on. IEEE, pp. 240–244.

Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A., 2006. Hierarchy and adaptivity in segmenting visual scenes. Nature 442 (7104), 810–813.

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

Verma, A., Buchanan, A., Butenko, S., 2015. Solving the maximum clique and vertex coloring problems on very large sparse networks. INFORMS Journal on Computing 27 (1), 164–177.

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

Watts, D. J., Strogatz, S. H., 1998. Collective dynamics of small-world networks. Nature 393 (6684), 440–442.

Wei, W., Xu, F., Tan, C. C., Li, Q., 2012. Sybildefender: Defend against sybil attacks in large social networks. In: INFOCOM, 2012 Proceedings IEEE. IEEE, pp. 1951–1959.

Wu, T.-H., 1997. A note on a global approach for general 0–1 fractional programming. European Journal of Operational Research 101 (1), 220–223.

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

Yu, H., 2011. Sybil defenses via social networks: a tutorial and survey. ACM SIGACT News 42 (3), 80–101.

Yu, H., Kaminsky, M., Gibbons, P. B., Flaxman, A., 2006. Sybilguard: defending against sybil attacks via social networks. In: ACM SIGCOMM Computer Communication Review. Vol. 36. ACM, pp. 267–278.

Zachary, W. W., 1977. An information flow model for conflict and fission in small groups. Journal of anthropological research, 452–473.

A Appendix

Proof of Theorem 4. For this proof, we need to discuss the following statements first.

Remark 1. *Given two ratios $\frac{a}{b}, \frac{c}{d}$, such that $\frac{a}{b} \leq \frac{c}{d}$, and $a, b, c, d > 0$, we have that $\frac{a}{b} \leq \frac{a+c}{b+d} \leq \frac{c}{d}$.*

Claim 1. *If the nodes in $\mathcal{N}(k)$ induce an independent set (i.e., are not adjacent in G). Then, Algorithm 1 using the ratios r_i as in (16) produces the best star normalized cut' for the star partition centered at k.*

Proof. If the nodes in $\mathcal{N}(k)$ are not adjacent, the ratio for node $i \in \mathcal{N}(k)$ becomes $r_i = \frac{-w_{ik}+Y_i}{w_{ik}}$, which depends exclusively on i. Notice that the actual contribution for both the numerator and denominator when node i is added to S are indeed $-w_{ik} + Y_i$ and w_{ik}, respectively. □

Claim 2. *Let S_{greedy} be the induced star centered at k obtained by the greedy algorithm, and S_{opt} be the optimal solution. Furthermore, let i be the first node in the construction of the greedy algorithm that does not exist in the optimal solution. Then, there exists at least one node in the optimal solution that is adjacent to i.*

Proof. Assume for a contradiction that i is not adjacent to any node in S_{opt}. Then, since $r_i \leq r_j, \forall j \in \mathcal{I}$, we have that $r_i \leq \min_{\ell \in \mathcal{I}} r_\ell \leq f(S_{opt})/g(S_{opt})$ (see Remark 1), which implies that r_i remains equal to $\frac{-w_{ik} + \sum_{\ell \in \mathcal{I}:(i,\ell)\in E} (w_{i\ell}+w_{k\ell}) + Y_i}{w_{ik}} \leq \frac{f(S_{opt})}{g(S_{opt})}$ for s_{opt}. Thus, adding i to S_{opt} still forms an induced star with a better objective function value, contradicting the optimality of S_{opt}. □

Claim 3. *Without loss of generality, we can assume that a node i, which belongs to the greedy solution but not to the optimal star, is the first node added in the greedy algorithm.*

Proof. If i is not the first node to be added, this implies that there exists a series of nodes, $\hat{S}$ that belong to both the optimal star and the one obtained from the greedy algorithm. Since those appear in both solutions, they do not affect the approximation ratio. □

We are now ready to proceed with the approximation ratio proof. Define the following:

- $\mathcal{I} = \mathcal{N}(k)$ be the initial set of candidates;
- k be the center node;
- i be the first selection of the greedy that does not appear in the optimal solution;
- $\widetilde{S} = \{\ell : (i, \ell) \in E\}$ be the set of nodes that are discarded if i is added to the induced star;
- S^* be the optimal solution;
- $\hat{S}$ be the set of nodes that are not in the optimal solution, excluding i and the ones in $\widetilde{S}$. i.e., $\bar{S} \setminus \{\{i\} \cup \widetilde{S}\}$.
- Y_j be the summation of all the weights associated with the edges between j and all the nodes outside $\mathcal{I} \cup \{k\}$. i.e. $Y_j = \sum_{\{\ell \in V \setminus \{\mathcal{I} \cup \{k\}\}:(j,\ell)\in E\}} w_{j\ell}$

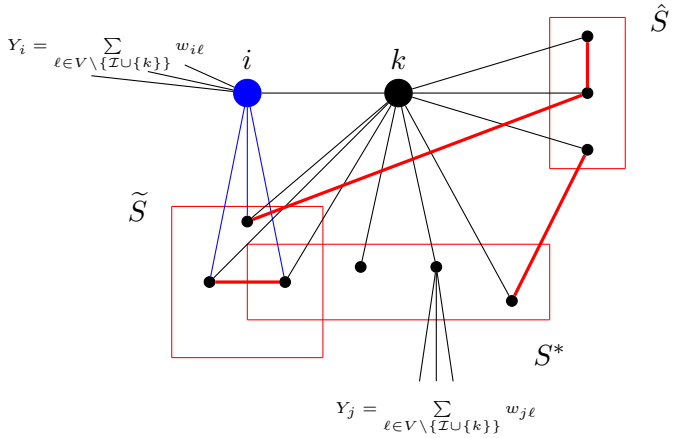


Figure 6: A small representation of the general case. We note k as the center, along with all of its adjacent nodes, which consist of the sets $\{i\}$ (in blue, along with its edges), $\widetilde{S}$ (along with its connections to i and with the other sets in red, including itself), S^* (along with its connections to other sets, but not itself, in red), and $\hat{S}$ (along with its connections to i and with the other sets in red, including itself).

We now describe the ratios used in the greedy algorithm for several nodes in the given graph (see Figure 6 for a graphical representation). For this purpose, recall that

$$r_j = \frac{-w_{jk} + Y_j + \sum\limits_{\{\ell \in \mathcal{I}:(j,\ell)\in E\}} (w_{j\ell} + w_{\ell k})}{w_{jk}}.$$

We distinguish the ratios of nodes in four sets of interest: $\{i\}, \widetilde{S} \setminus S^*, \widetilde{S} \cap S^*, S^* \setminus \widetilde{S}$. The main difference between such ratios lies in the third term of the numerator, as set $\mathcal{I} : (j,\ell) \in E\}$ is different for the nodes in those sets. Note that $\widetilde{S} \cap S^*$ is not an empty set because of Claim 2. Furthermore, based on Claim 3, we will assume that i is the first node to be added. i.e., the node with the smallest ratio among all candidates in $\mathcal{I}$. Such a ratio is as follows:

$$r_i = \frac{-w_{ik} + Y_i + \sum\limits_{\ell \in \widetilde{S} \setminus S^*} (w_{i\ell} + w_{k\ell}) + \sum\limits_{\ell \in S^* \cap \widetilde{S}} (w_{i\ell} + w_{k\ell})}{w_{ik}}. \tag{19}$$

For all nodes $j \in \widetilde{S} \setminus S^*$, we have:

$$r_j = \frac{-w_{jk} + Y_j + w_{ik} + w_{ij}}{w_{jk}} + \frac{\sum\limits_{\{\ell \in S^* \cap \widetilde{S}:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell})}{w_{jk}} + \frac{\sum\limits_{\{\ell \in S^* \setminus \widetilde{S}:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell}) + \sum\limits_{\{\ell \in \hat{S}:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell})}{w_{jk}} \tag{20}$$

For all nodes $j \in \widetilde{S} \cap S^*$, we have:

$$r_j = \frac{-w_{jk} + Y_j + w_{ik} + w_{ij}}{w_{jk}} + \frac{\sum\limits_{\{\ell \in S^* \setminus \widetilde{S}:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell}) + \sum\limits_{\{\ell \in \hat{S}:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell})}{w_{jk}} \tag{21}$$

For all nodes $j \in S^* \setminus \widetilde{S}$, we have:

$$r_j = \frac{-w_{jk} + Y_j + \sum\limits_{\{\ell \in \widetilde{S} \setminus S^*:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell}) + \sum\limits_{\{\ell \in \hat{S}:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell})}{w_{jk}} \tag{22}$$

Moreover, note that the greedy solution is at least as good as the solution containing i and k, exclusively (i.e., the greedy solution contains k, i, and potentially other nodes in $\mathcal{I}$ that may reduce the overall objective):

$$z_{greedy} \leq \frac{Y_i + \sum\limits_{\ell \in \widetilde{S} \setminus S^*} (w_{i\ell} + w_{k\ell}) + \sum\limits_{\ell \in S^* \cap \widetilde{S}} (w_{i\ell} + w_{k\ell}) + \sum\limits_{\ell \in S^* \setminus \widetilde{S}} w_{k\ell} + \sum\limits_{\ell \in \hat{S}} w_{k\ell}}{w_{ik}}. \tag{23}$$

Similarly, for the optimal solution the objective is:

$$z_{opt} = \frac{f(S^*)}{g(S^*)} = \frac{\sum\limits_{\ell \in S^*} Y_\ell + w_{ik} + \sum\limits_{\ell \in \widetilde{S} \cap S^*} w_{i\ell}}{\sum\limits_{\ell \in S^*} w_{k\ell}} + \frac{\sum\limits_{\ell \in S^*} \sum\limits_{\{p \in \widetilde{S} \setminus S^*:(\ell,p)\in E\}} w_{\ell p} + \sum\limits_{\ell \in S^*} \sum\limits_{\{p \in \hat{S}:(\ell,p)\in E\}} w_{\ell p} + \sum\limits_{p \in \widetilde{S} \setminus S^*} w_{kp} + \sum\limits_{p \in \hat{S}} w_{kp}}{\sum\limits_{\ell \in S^*} w_{k\ell}}. \tag{24}$$

By assumption, we have that $r_i \leq r_j, \forall j \in \mathcal{I}$ which include $\widetilde{S} \cap S^*$. Thus:

$$\begin{aligned}
&r_i \leq r_j \implies \\
&\frac{-w_{ik} + Y_i + \sum\limits_{\ell \in \widetilde{S} \setminus S^*} (w_{i\ell} + w_{k\ell}) + \sum\limits_{\ell \in S^* \cap \widetilde{S}} (w_{i\ell} + w_{k\ell})}{w_{ik}} \leq \\
&\frac{-w_{jk} + Y_j + w_{ik} + w_{ij} + \sum\limits_{\{\ell \in S^* \setminus \widetilde{S}:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell}) + \sum\limits_{\{\ell \in \hat{S}:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell})}{w_{jk}} \implies \\
&\frac{Y_i + \sum\limits_{\ell \in \widetilde{S} \setminus S^*} (w_{i\ell} + w_{k\ell}) + \sum\limits_{\ell \in S^* \cap \widetilde{S}} (w_{i\ell} + w_{k\ell})}{w_{ik}} \leq \qquad (25) \\
&\frac{Y_j + w_{ik} + w_{ij} + \sum\limits_{\{\ell \in S^* \setminus \widetilde{S}:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell}) + \sum\limits_{\{\ell \in \hat{S}:(j,\ell)\in E\}} (w_{j\ell} + w_{k\ell})}{w_{jk}} \implies \\
&\frac{Y_i + \sum\limits_{\ell \in \widetilde{S} \setminus S^*} (w_{i\ell} + w_{k\ell}) + \sum\limits_{\ell \in S^* \cap \widetilde{S}} (w_{i\ell} + w_{k\ell})}{w_{ik}} \leq \qquad (26)
\end{aligned}$$

$$\frac{\sum\limits_{\ell\in\widetilde{S}\cap S^*} Y_\ell + \sum\limits_{\ell\in\widetilde{S}\cap S^*} w_{ik} + \sum\limits_{\ell\in\widetilde{S}\cap S^*} w_{i\ell} + \sum\limits_{\ell\in\widetilde{S}\cap S^*} \sum\limits_{\{p\in S^*\setminus\widetilde{S}:(\ell,p)\in E\}} (w_{\ell p}+w_{kp}) + \sum\limits_{\ell\in\widetilde{S}\cap S^*} \sum\limits_{\{p\in\hat{S}:(\ell,p)\in E\}} (w_{\ell p}+w_{kp})}{\sum\limits_{\ell\in\widetilde{S}\cap S^*} w_{k\ell}}$$

In the above, equation (25) follows by separating the negative elements of each fraction and canceling them out, as in both cases the negative element in the numerator is equal to the denominator. For the second step in (26), we use the fact that given a set of fractions $\frac{\alpha_i}{\beta_i}$, for various i and for $\alpha_i > 0, \beta_i > 0$, if $\gamma \leq \frac{\alpha_i}{\beta_i}$, then $\gamma \leq \frac{\sum_i \alpha_i}{\sum_i \beta_i}$.

Performing the above operations for all nodes $j \in S^* \setminus \widetilde{S}$, we also get that:

$$\frac{Y_i + \sum\limits_{\ell\in\widetilde{S}\setminus S^*} (w_{i\ell}+w_{k\ell}) + \sum\limits_{\ell\in S^*\cap\widetilde{S}} (w_{i\ell}+w_{k\ell})}{w_{ik}} \leq$$
$$\frac{\sum\limits_{\ell\in S^*\setminus\widetilde{S}} Y_\ell + \sum\limits_{\ell\in S^*\setminus\widetilde{S}} \sum\limits_{\{p\in\widetilde{S}\setminus S^*:(\ell,p)\in E\}} (w_{\ell p}+w_{kp}) + \sum\limits_{\ell\in S^*\setminus\widetilde{S}} \sum\limits_{\{p\in\hat{S}:(\ell,p)\in E\}} (w_{\ell p}+w_{kp})}{\sum\limits_{\ell\in S^*\setminus\widetilde{S}} w_{k\ell}} \tag{27}$$

Then, adding up the right hand sides of both (26) and (27), we finally get:

$$\frac{Y_i + \sum\limits_{\ell\in\widetilde{S}\setminus S^*} (w_{i\ell}+w_{k\ell}) + \sum\limits_{\ell\in S^*\cap\widetilde{S}} (w_{i\ell}+w_{k\ell})}{w_{ik}} \leq$$
$$\frac{\sum\limits_{\ell\in S^*} Y_\ell + |\widetilde{S}\cap S^*| w_{ik} + \sum\limits_{\ell\in\widetilde{S}\cap S^*} w_{i\ell} + \sum\limits_{\ell\in S^*} \sum\limits_{\{p\in\widetilde{S}\setminus S^*:(\ell,p)\in E\}} w_{\ell p}}{\sum\limits_{\ell\in S^*} w_{k\ell}} +$$
$$\frac{\sum\limits_{\ell\in S^*} \sum\limits_{\{p\in\widetilde{S}\setminus S^*:(\ell,p)\in E\}} w_{kp} + \sum\limits_{\ell\in S^*} \sum\limits_{\{p\in\hat{S}:(\ell,p)\in E\}} w_{\ell p} + \sum\limits_{\ell\in S^*} \sum\limits_{\{p\in\hat{S}:(\ell,p)\in E\}} w_{kp}}{\sum\limits_{\ell\in S^*} w_{k\ell}} \tag{28}$$

Note now that the left hand side of (28) resembles the upper bound on the objective function value of the greedy soluciont as in (23), barring two terms that are missing from the numerator: $\sum\limits_{\ell\in S^*\setminus\widetilde{S}} w_{k\ell} + \sum\limits_{\ell\in\hat{S}} w_{k\ell}$. Consider the following remark:

Remark 2. *If $\frac{a}{b} \leq \frac{c}{d}$, and $d \leq \mu b$ for $\mu > 0$, then $\frac{a+x}{b} \leq \frac{c+\mu x}{d}$.*

In our case, we see that $\sum\limits_{\ell\in S^*} w_{k\ell} \leq \delta_k \cdot w_{ik}$: if not, then there exists some j in S^* such that $r_j < r_i$, contradicting our assumption for the greedy selection. This can be seen as a part of $\sum\limits_{\ell\in S^*} w_{k\ell}$ (the edge weights from k to every $\ell \in S^* \cap \widetilde{S}$) is in the numerator of r_i and, as soon as it becomes bigger than $\delta \cdot w_{ik}$, it leads to $r_i > r_j$, for some j. Hence, we finally have that:

$$\frac{Y_i + \sum\limits_{\ell\in\widetilde{S}\setminus S^*} (w_{i\ell}+w_{k\ell}) + \sum\limits_{\ell\in S^*\cap\widetilde{S}} (w_{i\ell}+w_{k\ell}) + \sum\limits_{\ell\in S^*\setminus\widetilde{S}} w_{k\ell} + \sum\limits_{\ell\in\hat{S}} w_{k\ell}}{w_{ik}} \leq$$

$$\frac{\sum_{\ell\in S^*} Y_\ell + |\widetilde{S}\cap S^*|w_{ik} + \sum_{\ell\in\widetilde{S}\cap S^*} w_{i\ell} + \sum_{\ell\in S^*}\sum_{\{p\in\widetilde{S}\setminus S^*:(\ell,p)\in E\}} w_{\ell p}}{\sum_{\ell\in S^*} w_{k\ell}} +$$
$$\frac{\sum_{\ell\in S^*}\sum_{\{p\in\widetilde{S}\setminus S^*:(\ell,p)\in E\}} w_{kp} + \sum_{\ell\in S^*}\sum_{\{p\in\hat{S}:(\ell,p)\in E\}} w_{\ell p} + \sum_{\ell\in S^*}\sum_{\{p\in\hat{S}:(\ell,p)\in E\}} w_{kp}}{\sum_{\ell\in S^*} w_{k\ell}} +$$
$$\frac{\delta_k\left(\sum_{\ell\in S^*\setminus\widetilde{S}} w_{k\ell} + \sum_{\ell\in\hat{S}} w_{k\ell}\right)}{\sum_{\ell\in S^*} w_{k\ell}} \tag{29}$$

$$z_{greedy} \leq \frac{\sum_{\ell\in S^*} Y_\ell + |\widetilde{S}\cap S^*|w_{ik} + \sum_{\ell\in\widetilde{S}\cap S^*} w_{i\ell} + \sum_{\ell\in S^*}\sum_{\{p\in\widetilde{S}\setminus S^*:(\ell,p)\in E\}} w_{\ell p} + |S^*|\sum_{p\in\widetilde{S}} w_{kp}}{\sum_{\ell\in S^*} w_{k\ell}} + \tag{30}$$
$$+ \frac{\delta_k\sum_{\ell\in S^*\setminus\widetilde{S}} w_{k\ell} + \sum_{\ell\in S^*}\sum_{\{p\in\hat{S}:(\ell,p)\in E\}} w_{\ell p} + (|S^*|+\delta_k)\sum_{p\in\hat{S}} w_{kp}}{\sum_{\ell\in S^*} w_{k\ell}} \leq \mathcal{O}(\delta_k)\cdot z_{opt}.$$

The ratio follows because the numerator of the right hand side is at most $\mathcal{O}(\delta_k)$ times the numerator of the optimal objective function, while the denominator is the same in both (24) and (30).

□

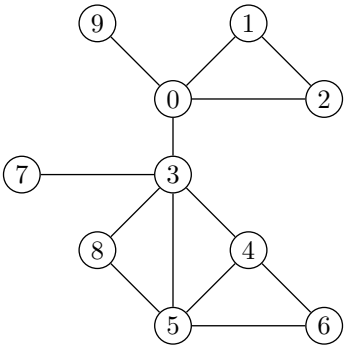
B Appendix

The following Figure presents an example of the decomposition approach described in Section 5.2 for solving the clique normalized cut′. In this example, we use the 10-node, 2-degenerate graph depicted in Figure 6(a), for which a valid degeneracy ordering, constructed as indicated by Matula and Beck (1983), is given in Figure 6(b). Notice that, each node has at the most two neighbors to its right in such ordering.

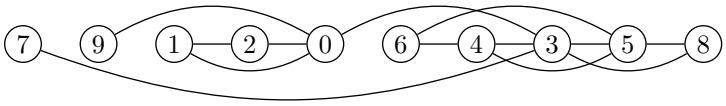
Figures 6(c)-(l) provide a graphical description of the n subproblems, which are listed according to the degeneracy ordering (i.e., the subproblems from steps (3)-(4) in Algorithm 3). In these subgraphs, we color with dark blue, the nodes that are forced to be part of each S_i (i.e., v_i); with light blue, the nodes that could potentially be part of S_i (i.e., the nodes in V_i); with dark red, the neighbors of v_i that are forced out of S_i (i.e., the nodes in $\mathcal{N}(i) \setminus V_i$); and, with light red, the nodes that cannot be part of S_i, but could potentially be part of $\mathcal{N}[S_i]$ (i.e., the nodes in $\mathcal{N}(V_i) \setminus v_i$). Any node that cannot be in S_i or $\mathcal{N}[S_i]$ is removed from the subgraph.

As for the edges, we remove the ones that cannot be in $E(G(S_i))$ or the $(S, \bar{S})$ cut (i.e., all edges that have endpoints removed from the subgraph, or whose two endpoints are colored light red), and color with red the edges that must be in the $(S, \bar{S})$ cut (i.e., edges connecting v_i with dark red nodes). The edges in black that are incident to v_i are edges that could be either in $E(G(S_i))$ or the $(S, \bar{S})$ cut depending on whether the light blue endpoint is in S_i. Finally, edges in black incident to light red nodes are edges that could potentially be in the $(S, \bar{S})$ cut, if their light blue endpoint is in S_i.

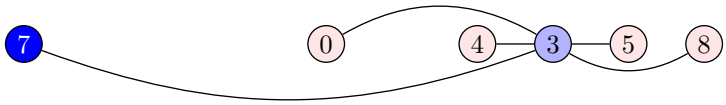
As mentioned in Section 5.2, the proposed decomposition requires solving n clique normalized cut′ subproblems that are significantly simpler than the original problem they stemmed from. In all these subproblems, the total number of binary variables $\mathbf{x}$ that are not implied by the decomposition (i.e., variables x_j for $j \in V_i$) is at the most $d = 2$.



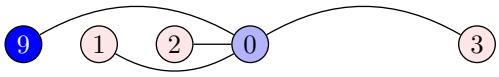
(a) Graph G



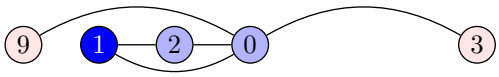
(b) Degeneracy ordering



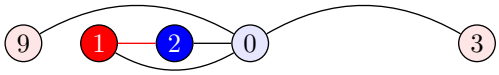
(c) Subproblem for node 7



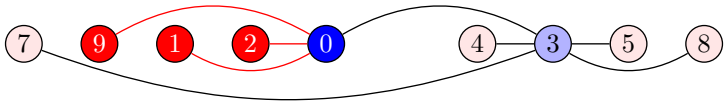
(d) Subproblem for node 9



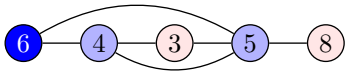
(e) Subproblem for node 1



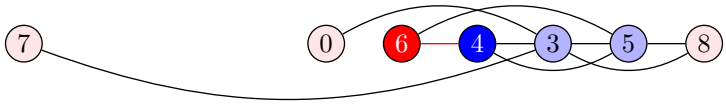
(f) Subproblem for node 2



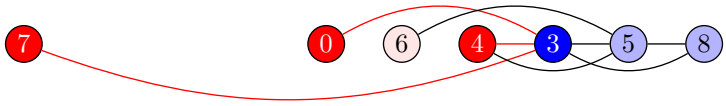
(g) Subproblem for node 0



(h) Subproblem for node 6



(i) Subproblem for node 4



(j) Subproblem for node 3

(k) Subproblem for node 5

(l) Subproblem for node 8

Figure 6: Graphical example of the decomposition for the clique normalized cut′ over a 10-node 2-degenerate graph.