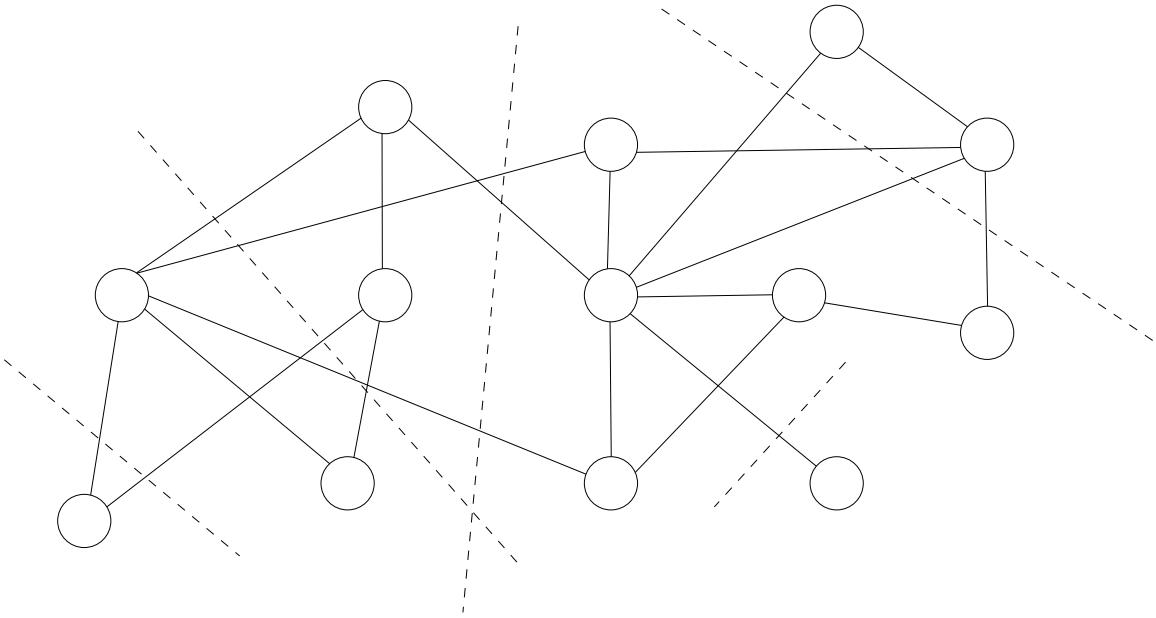
On the Complexity of the Normalized Cut Problem on $\mathcal{H}$-Free Graph Classes



Utrecht University

Thomas Nieboer
Master Thesis for Computing Science
Utrecht University
23 June 2023

Under the supervision of:
S. Pandey, MSc
Dr. E.J. van Leeuwen
Dr. J. Nederlof

Abstract

The Normalized Cut problem is a graph partitioning problem that is known to be NP-complete in general. Therefore, we look into the complexity of the Normalized Cut problem on certain $\mathcal{H}$-free graph classes. $\mathcal{H}$-free graphs are those which do not contain any graph of $\mathcal{H}$ as an induced subgraph, for any fixed set of graphs $\mathcal{H}$. We show that the Normalized Cut problem is NP-complete on claw-free, split, and complete graphs. Furthermore, we show that the Normalized Cut problem with unweighted edges is strongly NP-complete in general. On the other hand, we show that the partition with minimum *normalized cut value* has two connected *components* and use this property to construct polynomial-time algorithms on certain $\mathcal{H}$-free graph classes. We show that we can solve the Normalized Cut problem on forests in linear time and on outerplanar graphs in quadratic time. Furthermore, we show that we can solve the Normalized Cut problem with unweighted edges on cactus and cluster graphs in linear time. Lastly, we observe that there exists an $O(\log(n))$-approximation algorithm for another graph partitioning problem to which we can reduce the Normalized Cut problem. Therefore, we have an $O(\log(n))$-approximation algorithm for the Normalized Cut problem.

Contents

1 Introduction **3**

2 The Normalized Cut problem **5**

3 Related work **9**

4 Properties of the Normalized Cut **14**

5 Complexity of the Normalized Cut problem on $\mathcal{H}$-free graph classes **17**
5.1 NP-complete on claw-free graphs 18
5.2 NP-complete on split graphs 23
5.3 NP-complete on complete graphs 28

6 Strong NP-completeness of the Normalized Cut problem with unweighted edges **35**

7 Polynomial-time algorithms for the Normalized Cut problem on $\mathcal{H}$-free graph classes **42**
7.1 Linear time algorithm for forests 42
7.2 Quadratic time algorithm for outerplanar graphs 43

8 Polynomial-time algorithms for the Normalized Cut problem with unweighted edges on $\mathcal{H}$-free graph classes **48**
8.1 Linear time algorithm for cluster graphs 48
8.2 Linear time algorithm for cactus graphs 49

9 Approximation algorithm for the Normalized Cut problem **52**

10 Conclusion **52**

1 Introduction

The goal of graph partitioning is to divide a graph into multiple *components*. Graph partitioning has been widely researched (see e.g. [2, 18, 10, 20, 17, 12]). A reason for this is its wide applicability. For example, it is used for image segmentation [20], object and character recognition, information retrieval, data mining [11], VLSI circuit design [3] and for all kinds of networks such as road networks and social (media) networks [5].

A graph can be partitioned into *components* by removing its edges or vertices such that there are no edges connecting the *components*. The set of removed edges or vertices is called the *cut* of a graph. Which edges or vertices are removed to partition the graph is determined by the objective of the graph partitioning problem being studied. In this thesis, we only partition graphs by removing edges and we assume graphs to be undirected and edge-weighted unless explicitly stated otherwise.

One of the best known graph partitioning problems is the Minimum Cut problem. The Minimum Cut problem takes as input a graph and asks for a *cut* with the lowest *cut value* i.e. lowest sum of edge-weights of the edges in the cut. The Minimum Cut problem can result in one *component* with a small set of isolated vertices and one *component* with all other vertices [20]. Often it is desired, however, that the *components* are roughly 'equal'. Luckily, there are graph partitioning problems with an objective to partition the graph into roughly 'equal' *components*.

The Normalized Cut problem has the objective to minimize the *cut value* while also balancing the total sum of edge-weights of the edges in the *components*. See Definition 4 in Section 2 for a precise definition. The Normalized Cut problem finds application in image segmentation [20], pattern recognition [24], and community detection [21]. In Figure 1 we see an example of a graph with its edge-weights for which the Minimum Cut problem is solved and in Figure 2 we see the same graph for which the Normalized Cut problem is solved. We see that the solution of the Minimum Cut problem has 'unequal' *components* while the solution of the Normalized Cut problem has more 'equal' *components*.

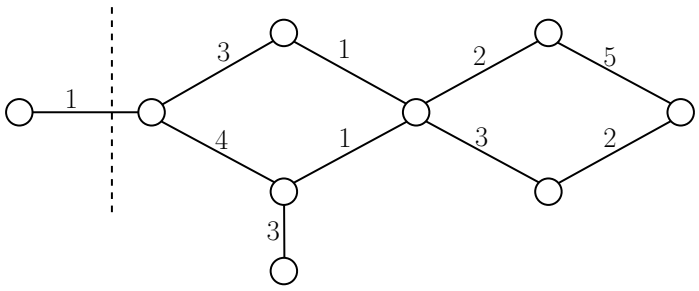


Figure 1: Solution of the Minimum Cut problem of a graph

Shi and Malik [20] showed that the Normalized Cut problem is NP-complete. To better understand the complexity of the Normalized Cut problem we will look at $\mathcal{H}$-free graph classes. Let $\mathcal{H}$ be a set of graphs. An $\mathcal{H}$-free graph is a graph that does not contain

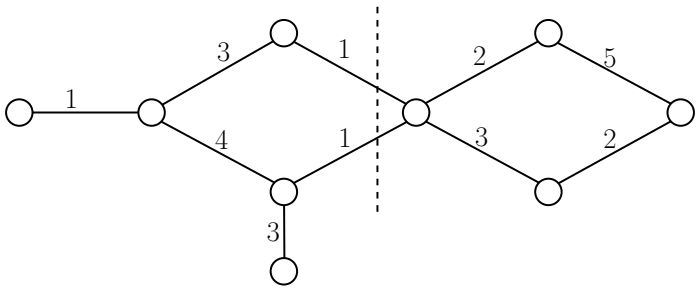


Figure 2: Solution of the NORMALIZED CUT problem of a graph

any graph from $\mathcal{H}$ as an induced subgraph. A graph class is $\mathcal{H}$-free if every graph G in the class is $\mathcal{H}$-free. Several well-known $\mathcal{H}$-free graph classes are forests, claw-free graphs, split graphs, and cluster graphs.

The rest of this thesis is organized in the following way. In Section 2, we define the NORMALIZED CUT problem and some variations of the NORMALIZED CUT problem. Furthermore, we show previous work on the NORMALIZED CUT problem and on the variations of the NORMALIZED CUT problem. Lastly, at the end of Section 2, we give an overview of the results we present in this thesis. In Section 3 we define some other graph partitioning problems and show previous work on these other graph partitioning problems.

In Section 5, we show that the NORMALIZED CUT problem is NP-complete on claw-free, split, and complete graphs. On the other hand, in Section 7, we show that we can solve the NORMALIZED CUT problem on forests in linear time and on outerplanar graphs in quadratic time.

In Section 6 we look into the complexity of the NORMALIZED CUT problem with unweighted edges. We show that the NORMALIZED CUT problem with unweighted edges is strongly NP-complete. On the other hand, in Section 8, we show that we can solve the NORMALIZED CUT problem with unweighted edges in linear time on cluster and cactus graphs.

In Section 4 we answer an open research question asked in the thesis of Vincken [25], namely if a partition with minimum *normalized cut value* has exactly two connected *components*. We show that it is indeed the case that the partition with minimum *normalized cut value* has exactly two connected *components* and use this property in the algorithms mentioned above.

In Section 9, we observe that there exists an $O(\log(n))$-approximation algorithm for another graph partitioning problem to which we can reduce the NORMALIZED CUT problem. Therefore, we have an $O(\log(n))$-approximation algorithm for the NORMALIZED CUT problem.

Lastly, in Section 10, we finish this thesis with a conclusion and some open research questions for the NORMALIZED CUT problem.

2 The Normalized Cut problem

In this section, we formally define graph partitioning, the NORMALIZED CUT problem, and some variations of the NORMALIZED CUT problem. Furthermore, we show previous work on the NORMALIZED CUT problem and its variations. Lastly, at the end of this section, we give an overview of the results we present in this thesis.

Graph partitioning is the partition of the set of vertices V of a graph $G = (V, E)$ into mutually disjoint non-empty vertex sets $V = S_1 \cup S_2 \cdots \cup S_k$ by removing the edges crossing these vertex sets. These vertex sets are called *components* and the set of removed edges is called the *cut* of the partition $\{S_1, S_2, \ldots, S_k\}$ of G. In this thesis, we assume that a *cut* of a graph $G = (V, E)$ is a partition of the vertices V into two *components* $\emptyset \subset S \subset V$ and $\bar{S} = V \setminus S$. If a graph is partitioned into $k > 2$ *components* $V = S_1 \cup S_2 \cdots \cup S_k$, it is called a *k-cut* of a graph. We assume every graph to be undirected and edge-weighted. If the graph is also vertex-weighted or if the edges do not have weights, it is mentioned explicitly. Furthermore, the weights of the edges and vertices of a graph are always strictly positive.

Before we can define the NORMALIZED CUT problem and its variations, we first have to define the *cut value* of a partition. The *cut value* of a partition $\{S, \bar{S}\}$ of a graph $G = (V, E)$ is defined as the sum of the edge-weights of the edges in the *cut*:

$$\mathrm{cut}(S, \bar{S}) = \sum_{\substack{v \in S \\ u \in \bar{S} \\ (u,v) \in E}} w(v, u). \tag{1}$$

The *k-cut value* of a partition $\{S_1, S_2, \ldots, S_k\}$ of a graph $G = (V, E)$ is defined as the sum of the edge-weights of the edges in the *cut*:

$$\mathrm{cut_k}(S_1, S_2, \ldots, S_k) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \sum_{\substack{v \in S_i \\ u \in S_j \\ (u,v) \in E}} w(v, u). \tag{2}$$

Note that for $k = 2$, the *k-cut value* is equal to the *cut value* of a partition $\{S, \bar{S}\}$ of a graph G.

Furthermore, we need to define the *volume* of a *component* S. The *volume* of a *component* S is defined as the sum of the edge-weights of the edges with one vertex-endpoint in *component* S and the other vertex-endpoint in the total set of vertices V of a graph $G = (V, E)$:

$$\mathrm{vol}(S, V) = \sum_{\substack{v \in S \\ u \in V \\ (v,u) \in E}} w(v, u). \tag{3}$$

Note that an edge-weight $w(v, u)$ of an edge $(v, u) \in E$ with vertex-endpoints v and u within the vertex set S is counted twice since $S \subseteq V$.

Now we can define the Normalized Cut problem. The Normalized Cut problem takes as input a graph G and a real number N and asks if there exists a partition $\{S, \bar{S}\}$ of G with a *normalized cut value* of at most N. The *normalized cut value* of a partition $\{S, \bar{S}\}$ is defined as:

$$\mathrm{NCut}(S, \bar{S}) = \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(S, V)} + \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(\bar{S}, V)}. \tag{4}$$

The *normalized cut value* of a partition $\{S, \bar{S}\}$ can also be rewritten into a second form:

$$\mathrm{NCut}(S, \bar{S}) = \frac{\mathrm{cut}(S, \bar{S})(\mathrm{vol}(S, V) + \mathrm{vol}(\bar{S}, V))}{\mathrm{vol}(S, V) \cdot \mathrm{vol}(\bar{S}, V)}. \tag{5}$$

Note that the second term of the numerator is not dependent on the partition. It is exactly two times the total edge-weight of the edges of graph G which is the *volume* $\mathrm{vol}(V, V)$ of V.

The Normalized Cut problem was first introduced in a paper by Shi and Malik [20] in which they also showed that the Normalized Cut problem is NP-complete. Furthermore, they showed that even for bipartite planar graphs the Normalized Cut problem is NP-complete. Vincken [25] showed in her thesis that the Normalized Cut problem is FPT parameterized by the vertex cover combined with the total sum of edge-weights of a graph as well as parameterized by treewidth combined with the total sum of edge-weights of a graph.

In this thesis, we introduce a new variant of the Normalized Cut problem, namely the Simplified Normalized Cut problem. The Simplified Normalized Cut problem takes as input a graph G and a real number N and asks if there exists a partition $\{S, \bar{S}\}$ of G with a *simplified normalized cut value* of at most N. The *simplified normalized cut value* of a partition $\{S, \bar{S}\}$ is defined as:

$$\mathrm{SNCut}(S, \bar{S}) = \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(S, V) \cdot \mathrm{vol}(\bar{S}, V)}. \tag{6}$$

Note that we can reduce the NORMALIZED CUT problem to the SIMPLIFIED NORMALIZED CUT problem since $\mathrm{NCut}(S, \bar{S}) = \mathrm{SNCut}(S, \bar{S}) \cdot (\mathrm{vol}(S, V) + \mathrm{vol}(\bar{S}, V))$ for a partition $\{S, \bar{S}\}$ and $(\mathrm{vol}(S, V) + \mathrm{vol}(\bar{S}, V))$ is not dependent on $\{S, \bar{S}\}$ because it is always equal to $\mathrm{vol}(V, V)$.

The NORMALIZED k-CUT problem takes as input a graph G, a natural number k, and a real number N and asks if there exists a partition $\{S_1, S_2, \ldots, S_k\}$ of G with a *normalized k-cut value* of at most N. The *normalized k-cut value* of a partition $\{S_1, S_2, \ldots, S_k\}$ is defined as:

$$\mathrm{NCut_k}(S_1, S_2, \ldots, S_k) = \sum_{1 \leq i \leq k} \frac{\mathrm{cut}(S_i, \bar{S}_i)}{\mathrm{vol}(S_i, V)}. \tag{7}$$

Since the NORMALIZED CUT problem is a special case of the NORMALIZED k-CUT problem, namely for $k = 2$, the NORMALIZED k-CUT problem is also NP-complete.

There are a few more variations of the NORMALIZED CUT problem. The MAX NORMALIZED k-CUT problem takes as input a graph G, a natural number k, and a real number N and asks if there exists a partition $\{S_1, S_2, \ldots, S_k\}$ of G with a *max normalized k-cut value* of at most N. The *max normalized k-cut value* of a partition $\{S_1, S_2, \ldots, S_k\}$ is defined as:

$$\mathrm{MaxNCut_k}(S_1, S_2, \ldots, S_k) = \max_{1 \leq i \leq k} \frac{\mathrm{cut}(S_i, \bar{S}_i)}{\mathrm{vol}(S_i, V)}. \tag{8}$$

The MAX NORMALIZED k-CUT problem is NP-complete, since for $k = 2$ the MAX NORMALIZED k-CUT problem is equal to the NP-complete CONDUCTANCE problem, introduced in Section 3. Furthermore, Daneshgar and Javadi [6] showed that even for weighted trees the MAX NORMALIZED k-CUT problem is NP-complete.

The following two variations of the NORMALIZED CUT problem are on vertex-weighted graphs. The MEAN VERTEX-WEIGHTED NORMALIZED k-CUT problem takes as input a graph G, a natural number k, and a real number N and asks if there exists a partition $\{S_1, S_2, \ldots, S_k\}$ of G with a *mean vertex-weighted normalized k-cut value* of at most N. The *mean vertex-weighted normalized k-cut value* of a partition $\{S_1, S_2, \ldots, S_k\}$ is defined as:

$$\mathrm{MeanNCut_{kw}}(S_1, S_2, \ldots, S_k) = \frac{1}{k} \sum_{1 \leq i \leq k} \frac{\mathrm{cut}(S_i, \bar{S}_i)}{w(S_i)}. \tag{9}$$

Note that the NP-complete NORMALIZED k-CUT problem can be reduced to the MEAN VERTEX-WEIGHTED NORMALIZED k-CUT problem by giving every vertex $v \in V$ of a graph $G = (V, E)$ the weight of the sum of the weights of its incident edges:

$$w(v) = \sum_{(v,u)\in E} w(v,u). \tag{10}$$

Therefore, the MEAN VERTEX-WEIGHTED NORMALIZED k-CUT problem is NP-complete.

The MAX VERTEX-WEIGHTED NORMALIZED k-CUT problem takes as input a graph G, a natural number k, and a real number N and asks if there exists a partition $\{S_1, S_2, \ldots, S_k\}$ of G with a *max vertex-weighted normalized k-cut value* of at most N. The *max vertex-weighted normalized k-cut value* of a partition $\{S_1, S_2, \ldots, S_k\}$ is defined as:

$$\text{MaxNCut}_{\text{kw}}(S_1, S_2, \ldots, S_k) = \max_{1\leq i\leq k} \frac{\text{cut}(S_i, \bar{S}_i)}{w(S_i)}. \tag{11}$$

Note that the NP-complete MAX NORMALIZED k-CUT problem can be reduced to the MAX VERTEX-WEIGHTED NORMALIZED k-CUT problem by giving every vertex $v \in V$ of a graph $G = (V, E)$ the weight of the sum of the weights of its incident edges:

$$w(v) = \sum_{(v,u)\in E} w(v,u). \tag{12}$$

Therefore, the MAX VERTEX-WEIGHTED NORMALIZED k-CUT problem is NP-complete.

The NORMALIZED CUT problem and the NORMALIZED CUT problem with unweighted edges are investigated in Sections 4, 5, 6, 7, 8 and 9. In Section 4 we give some propositions about the NORMALIZED CUT problem and prove an interesting property of the partition with the minimum *normalized cut value*. We show that a partition with minimum *normalized cut value* has exactly two connected *components*. In Section 5 we investigate the complexity of the NORMALIZED CUT problem on a variety of $\mathcal{H}$-free graphs. First, in Subsection 5.1, we present a proof of the NP-completeness of the NORMALIZED CUT problem on $K_{1,3}$-free or claw-free graphs. Then, in Subsection 5.2 we present a proof of the NP-completeness of the NORMALIZED CUT problem on $\{2P_2, C_4, C_5\}$-free graphs or split graphs. Finally, in Subsection 5.3, we present a proof of the NP-completeness of the NORMALIZED CUT problem on complete graphs. In Section 6 we show that the NORMALIZED CUT problem with unweighted edges on general graphs is strongly NP-complete. In Section 7 we present two algorithms that solve the NORMALIZED CUT problem in polynomial-time. In Subsection 7.1 we present an algorithm that solves the NORMALIZED CUT problem on forests in linear time, and in Subsection 7.2 we present an algorithm that solves the NORMALIZED CUT problem on outerplanar graphs in quadratic time. Furthermore, in Section 8 we present two algorithms that solve the NORMALIZED CUT problem with unweighted edges in polynomial-time. In Subsection 8.1 we present an algorithm that solves the NORMALIZED CUT problem with unweighted edges on cluster graphs in linear time, and in Subsection

8.3 we present an algorithm that solves the Normalized Cut problem with unweighted edges on cactus graphs in linear time. Lastly, in Section 9, we observe that there exists an $O(\log(n))$-approximation algorithm for another graph partitioning problem to which we can reduce the Normalized Cut problem. Therefore, we have an $O(\log(n))$-approximation algorithm for the Normalized Cut problem.

We give an overview of the Theorems presented in this thesis[1]:

Theorem 4.6. *The partition of a connected graph with minimum normalized cut value has exactly two connected components.*

Theorem 5.1. *The* Normalized Cut *problem is NP-complete on claw-free graphs.*

Theorem 5.2. *The* Normalized Cut *problem is NP-complete on split graphs.*

Theorem 5.3. *The* Normalized Cut *problem is NP-complete on complete graphs.*

Theorem 6.1. *The* Normalized Cut *problem is strongly NP-complete on general graphs with unweighted edges.*

Theorem 7.1. *The* Normalized Cut *problem is solvable in linear time on forests.*

Theorem 7.2. *The* Normalized Cut *problem is solvable in quadratic time on outerplanar graphs.*

Theorem 8.1. *The* Normalized Cut *problem is solvable in linear time on cluster graphs with unweighted edges.*

Theorem 8.3. *The* Normalized Cut *problem is solvable in linear time on cactus graphs with unweighted edges.*

Theorem 9.1. *The* Normalized Cut *problem has an $O(\log(n))$-approximation algorithm.*

3 Related work

In this section, we define some other graph partitioning problems and show previous work on these graph partitioning problems.

The best known graph partitioning problem is the Minimum Cut problem. The Minimum Cut problem takes as input a graph G and a real number N and asks if there exists a partition $\{S, \bar{S}\}$ of G with a *cut value* of at most N. The Minimum Cut problem of a graph can be solved in polynomial-time. A simple algorithm that can solve the Minimum Cut

[1]Note that Theorem 5.3 implies Theorems 5.1 and 5.2. However, the proofs of Theorems 5.1 and 5.2 are still noteworthy. The proof of Theorem 5.1 uses an entirely different graph construction than the graph construction used for the proof of Theorem 5.3. The proof of Theorem 5.2 is a good stepping stone for the proof of Theorem 5.3

problem of a graph $G = (V, E)$ in $O(|V||E| + |V|^2 \log(|V|))$ time is the Stoer-Wagner algorithm [22]. Note that there is a wealth of further algorithms, some quite recent. See e.g. the algorithm with an $O(|E| \log^2 |V|)$ running time by Gawrychowski, Mozes, and Weimann [8].

The MINIMUM k-CUT problem takes as input a graph $G = (V, E)$, a natural number k, and a real number N and asks if there exists a partition $\{S_1, S_2, \ldots, S_k\}$ of G with a *k-cut value* of at most N. In contrary to the MINIMUM CUT problem, the MINIMUM k-CUT problem is NP-complete as shown by Goldschmidt and Hochbaum [10].

The counterpart of the MINIMUM CUT problem is called the MAXIMUM CUT problem. The MAXIMUM CUT problem takes as input a graph G and a real number N and asks if there exists a partition $\{S, \bar{S}\}$ of G with a *cut value* of at least N. The MAXIMUM CUT problem is one of the famous 21 NP-complete problems of Karp [14]. Furthermore, the MAXIMUM CUT problem with unweighted edges is shown to be strongly NP-complete by Garey et al. [7]. The best known approximation algorithm for solving the MAXIMUM CUT problem is presented by Williams and Goemans [9] and has an approximation ratio of ≈ 0.878. This approximation ratio is also optimal if the Unique Games Conjecture is true as shown by Khot et al. [15].

The MAXIMUM k-CUT problem takes as input a graph $G = (V, E)$, a natural number k, and a real number N and asks if there exists a partition $\{S_1, S_2, \ldots, S_k\}$ of G with a *k-cut value* of at least N. Since the MAXIMUM CUT problem is a special case of the MAXIMUM k-CUT problem, namely when $k = 2$, the MAXIMUM k-CUT problem is also NP-complete.

The EDGE EXPANSION problem takes as input a graph G and a real number N and asks if there exists a partition $\{S, \bar{S}\}$ of G with an *edge expansion* of at most N. The *edge expansion* of a partition $\{S, \bar{S}\}$ is defined as:

$$\Phi(S, \bar{S}) = \frac{\text{cut}(S, \bar{S})}{\min(|S|, |\bar{S}|)}. \tag{13}$$

A solution of the EDGE EXPANSION problem for a graph G is sometimes called the Cheeger number/constant or isoperimetric number of G. The EDGE EXPANSION problem with unweighted edges is NP-complete as shown by Mohar [18] and Kaibel [13], which implies NP-completeness of the EDGE EXPANSION problem. The best known approximation algorithm for solving the EDGE EXPANSION problem with unweighted edges is presented by Arora et al. [2] and has an approximation ratio of $O(\sqrt{\log(n)})$.

The k-WAY EDGE EXPANSION problem takes as input a graph G, a natural number k, and a real number N and asks if there exists a partition $\{S_1, S_2, \ldots, S_k\}$ of G with a *k-way edge expansion value* of at most N. The *k-way edge expansion value* of a partition $\{S_1, S_2, \ldots, S_k\}$ is defined as:

$$\Phi(S_1, S_2, \ldots, S_k) = \max_{1 \leq i \leq k} \Phi(S_i, V \setminus S_i). \tag{14}$$

Since the Edge Expansion problem is a special case of the k-way Edge Expansion problem, namely for $k = 2$, the k-way Edge Expansion problem is also NP-complete.

A variant of the Edge Expansion problem is the k-Small Set Expansion problem. The k-Small Set Expansion problem takes as input a graph G, a natural number k, and a real number N and asks if there exists a partition $\{S, \bar{S}\}$ of G with an *edge expansion value* of at most N and where $|S|$ or $|\bar{S}|$ is of size at most k. Note that we can reduce the k-Small Set Expansion problem to the Edge Expansion problem by defining $k = |V|/2$. The k-Small Set Expansion problem with unweighted edges is NP-complete as shown by Raghavendra and Steurer [19]. Javadi and Nikabadi [12] showed that the k-Small Set Expansion problem is fixed-parameter tractable (FPT) parameterized by the treewidth of the graph. They also showed that the k-Small Set Expansion problem is FPT parameterized by the vertex cover of the graph. Furthermore, they showed that the k-Small Set Expansion problem with unweighted edges is $W[1]$-hard for parameter k.

For the next graph partitioning problem, the Quotient Cut problem, we first need to define the *vertex-weighted edge expansion value* for vertex-weighted graphs. For vertex-weighted graphs, the *vertex-weighted edge expansion value* uses the total vertex-weight of the vertices in a *component* S:

$$w(S) = \sum_{v \in S} w(v). \tag{15}$$

For vertex-weighted graphs the *vertex-weighted edge expansion value* of a *component* S_i of a partition $\{S_1, S_2, \ldots, S_k\}$ of a graph $G = (V, E)$ is defined as:

$$\Phi_w(S, \bar{S}) = \frac{\text{cut}(S, \bar{S})}{\min(w(S), w(\bar{S}))}. \tag{16}$$

The Quotient Cut problem resembles the Edge Expansion problem for vertex-weighted graphs. The Quotient Cut problem takes as input a vertex-weighted graph G and a real number N and asks if there exists a partition $\{S, \bar{S}\}$ of G with a *vertex-weighted edge expansion value* of at most N. Since the Edge Expansion problem is a special case of the Quotient Cut problem, namely with unweighted vertices, the Quotient Cut problem is also NP-complete.

The next graph partitioning problem is the Sparsest Cut problem. Officially we should call it the Uniform Sparsest Cut problem since the Sparsest Cut problem is a more

general problem than the Uniform Sparsest Cut problem. However, in literature the Uniform Sparsest Cut problem is often abbreviated to the Sparsest Cut problem [4, 16]. Therefore, from now on we will also abbreviate the Uniform Sparsest Cut problem to the Sparsest Cut problem. Furthermore, note that the Sparsest Cut problem is sometimes defined as the Edge Expansion problem [12]. The Sparsest Cut problem takes as input a graph G and a real number N and asks if there exists a partition $\{S, \bar{S}\}$ of G with a *density cut value* of at most N. The *density cut value* of a partition $\{S, \bar{S}\}$ is defined as:

$$\text{DensCut}(S, \bar{S}) = \frac{\text{cut}(S, \bar{S})}{|S| \cdot |\bar{S}|}. \tag{17}$$

The Sparsest Cut problem is NP-complete as shown by Matula and Shahrokhi [17]. Later, Bonsma et al. [4] showed that the Sparsest Cut problem with unweighted edges is strongly NP-complete. Furthermore, they showed that the Sparsest Cut problem of a graph is FPT parameterized by the treewidth of the graph. They also showed that a partition with minimum *density cut value* always results in two connected *components*. Using this connectivity property of the two *components* they constructed algorithms that solve the Sparsest Cut problem of a cactus graph with unweighted edges and a unit interval graph with unweighted edges in linear time. Furthermore, they used this connectivity property to solve the Sparsest Cut problem of an outerplanar graph in quadratic time. The best known approximation algorithm for solving the Sparsest Cut problem with unweighted edges is presented by Arora et al. [2] and has an approximation ratio of $O(\sqrt{\log(n)})$.

There is also a vertex-weighted variant of the Sparsest Cut problem where instead of the product of the sizes of the *components*, the product of the sums of the weights of the vertices in the *components* is taken. The vertex-weighted Sparsest Cut problem takes as input a vertex-weighted graph G and a real number N and asks if there exists a partition $\{S, \bar{S}\}$ of G with a *vertex-weighted density cut value* of at most N. The *vertex-weighted density cut value* of a partition $\{S, \bar{S}\}$ is defined as:

$$\text{DensCut}(S, \bar{S}) = \frac{\text{cut}(S, \bar{S})}{w(S) \cdot w(\bar{S})}. \tag{18}$$

Note that we can reduce the Normalized Cut problem to the vertex-weighted Sparsest Cut problem by giving every vertex the weight of the sum of the edge-weights of the edges connected to this vertex:

$$w(v) = \sum_{(v,u) \in E} w(v, u). \tag{19}$$

Now the *vertex-weighted density cut value* of a partition is equal to the second form of the *normalized cut value* if we ignore the second term in the numerator of the *normalized*

cut value which is not dependent on the partition. Since the SPARSEST CUT problem is a special case of the vertex-weighted SPARSEST CUT problem, the vertex-weighted SPARSEST CUT problem is also NP-complete. Furthermore, Leighton and Rao [16] show in their paper that the vertex-weighted SPARSEST CUT problem has an $O(\log(n))$-approximation algorithm.

The following graph partitioning problem uses the *volume* of a *component* S. In Section 2 we already saw the definition of the *volume* of a *component* S. Nevertheless, we restate the definition here. The *volume* of a *component* S is defined as the sum of the edge-weights of the edges with one vertex-endpoint in *component* S and the other vertex-endpoint in the total set of vertices V of the graph $G = (V, E)$:

$$\text{vol}(S, V) = \sum_{\substack{v \in S \\ u \in V \\ (v,u) \in E}} w(v, u). \tag{20}$$

Note that an edge-weight $w(v, u)$ of an edge $(v, u) \in E$ with vertex-endpoints v and u within the vertex set S is counted twice since $S \subseteq V$.

The CONDUCTANCE problem takes as input a graph G and a real number N and asks if there exists a partition $\{S, \bar{S}\}$ of G with a *conductance* of at most N. The *conductance* of a partition $\{S, \bar{S}\}$ is defined as:

$$\text{CondCut}(S, \bar{S}) = \frac{\text{cut}(S, \bar{S})}{\min(\text{vol}(S, V), \text{vol}(\bar{S}, V))}. \tag{21}$$

Note that the CONDUCTANCE problem can be reduced to the QUOTIENT CUT problem by giving every vertex $v \in V$ of a graph $G = (V, E)$ the weight of the sum of the weights of its edges:

$$w(v) = \sum_{(v,u) \in E} w(v, u). \tag{22}$$

Furthermore, note that the MAX NORMALIZED k-CUT problem introduced in Section 2 is equal to the CONDUCTANCE problem for $k = 2$. The CONDUCTANCE problem with unweighted edges is NP-complete as shown by Sima and Schaeffer [26]. Since the CONDUCTANCE problem with unweighted edges is a special case of the CONDUCTANCE problem, the CONDUCTANCE problem is also NP-complete. The best known approximation algorithm for the CONDUCTANCE problem with unweighted edges is presented by Arora et al. [2] and has an approximation ratio of $O(\sqrt{\log(n)})$.

4 Properties of the Normalized Cut

In this section, we show two Propositions 4.1 and 4.2 regarding the *normalized cut value* of a partition. After that, we show an interesting property that a partition with minimum *normalized cut value* has exactly two connected *components*.

Recall the second form of the *normalized cut value* of a partition $\{S, \bar{S}\}$ of a graph G defined in Section 2:

$$\mathrm{NCut}(S, \bar{S}) = \frac{\mathrm{cut}(S, \bar{S}) \cdot (\mathrm{vol}(S, V) + \mathrm{vol}(\bar{S}, V))}{\mathrm{vol}(S, V) \cdot \mathrm{vol}(\bar{S}, V)}. \tag{23}$$

Proposition 4.1. *Among all partitions of a graph with the same cut value, the one for which the volumes differ the least has the lowest normalized cut value.*

Proof. We look at the second form of the *normalized cut value* and see that if the *cut value* is equal among all partitions, the numerator is also equal. The denominator is maximized if the product of the *volumes* is maximized. Since the sum of the two *volumes* of every partition is equal to the *volume* of the whole graph under consideration, this product is maximized when the *volumes* differ the least in value. □

Proposition 4.2. *Among all partitions of a graph where the difference between the volumes is equal, the one for which the cut value is lowest has the lowest normalized cut value.*

Proof. First note that the sum of the two *volumes* of every partition is equal to the *volume* of the whole graph under consideration. We look at the second form of the *normalized cut value* and see that if the difference between the *volumes* is equal among all partitions, the denominator is also equal. Therefore, we look at the numerator which is minimized if the *cut value* is minimized. □

Now we will prove an important property of a partition of a graph with minimum *normalized cut value*. Theorem 4.6 states that the partition of a connected graph with minimum *normalized cut value* has exactly two connected *components*. We will see in Sections 7 and 8 that this property is useful for creating polynomial-time algorithms on some $\mathcal{H}$-free graph classes. The proofs of Theorem 4.6 and corresponding Proposition 4.3 and Lemmas 4.4 and 4.5 are based on the proof of connected *components* for the partition with minimum *density cut value* by Bonsma et al. [4].

We start with some new definitions. First, we define the *simplified normalized cut value*, defined in Section 2, for pairs of vertex sets that do not necessarily partition V of a graph $G = (V, E)$. For a graph $G = (V, E)$ and two vertex sets $S, T \subseteq V$ with $S \cap T = \emptyset$, $S \neq \emptyset$ and $T \neq \emptyset$ we denote by $s(S, T)$ the *simplified normalized cut value* of vertex sets S and T, now allowing $S \cup T \neq V$:

$$s(S,T) = \frac{\text{cut}(S,T)}{\text{vol}(S,V)\cdot\text{vol}(T,V)}. \tag{24}$$

Furthermore, we define the excess of a pair S,T as $e(S,T) = s(S,T) - z$, where z is the minimum *simplified normalized cut value* of G. Note that, if S and T are disjoint and $S \cup T = V$, then $e(S,T) \geq 0$ and this is an equality if $\{S,T\}$ is a partition with a minimum *simplified normalized cut value*.

With these new definitions, we show for non-empty disjoint vertex sets $S, T_1, T_2 \subseteq V$ of a connected graph $G = (V,E)$ that $s(S, T_1 \cup T_2)$ is a weighted average of $s(S,T_1)$ and $s(S,T_2)$ and that $e(S, T_1 \cup T_2)$ is a weighted average of $e(S,T_1)$ and $e(S,T_2)$.

Proposition 4.3. *Suppose we have a graph $G = (V,E)$. For any non-empty disjoint vertex sets $S, T_1, T_2 \subseteq V$, we have:*

$$s(S, T_1 \cup T_2) = \frac{s(S,T_1)\cdot\text{vol}(T_1,V) + s(S,T_2)\cdot\text{vol}(T_2,V)}{\text{vol}(T_1,V)+\text{vol}(T_2,V)}, \tag{25}$$

$$e(S, T_1 \cup T_2) = \frac{e(S,T_1)\cdot\text{vol}(T_1,V) + e(S,T_2)\cdot\text{vol}(T_2,V)}{\text{vol}(T_1,V)+\text{vol}(T_2,V)}. \tag{26}$$

Proof. We rewrite the definition of $s(S, T_1 \cup T_2)$:

$$\begin{aligned}
s(S, T_1 \cup T_2) &= \frac{\text{cut}(S, T_1 \cup T_2)}{\text{vol}(S,V)\cdot(\text{vol}(T_1,V)+\text{vol}(T_2,V))} \\
&= \frac{\text{cut}(S,T_1)+\text{cut}(S,T_2)}{\text{vol}(S,V)\cdot(\text{vol}(T_1,V)+\text{vol}(T_2,V))} \\
&= \frac{\frac{\text{cut}(S,T_1)}{\text{vol}(S,V)\cdot\text{vol}(T_1,V)}\cdot\text{vol}(T_1,V) + \frac{\text{cut}(S,T_2)}{\text{vol}(S,V)\cdot\text{vol}(T_2,V)}\cdot\text{vol}(T_2,V)}{\text{vol}(T_1,V)+\text{vol}(T_2,V)} \\
&= \frac{s(S,T_1)\cdot\text{vol}(T_1,V) + s(S,T_2)\cdot\text{vol}(T_2,V)}{\text{vol}(T_1,V)+\text{vol}(T_2,V)}.
\end{aligned} \tag{27}$$

Now we use this result to rewrite the definition of $e(S, T_1 \cup T_2)$:

$$
\begin{aligned}
e(S, T_1 \cup T_2) &= \frac{s(S,T_1)\cdot \mathrm{vol}(T_1,V) + s(S,T_2)\cdot \mathrm{vol}(T_2,V)}{\mathrm{vol}(T_1,V)+\mathrm{vol}(T_2,V)} - z \\
&= \frac{s(S,T_1)\cdot \mathrm{vol}(T_1,V) + s(S,T_2)\cdot \mathrm{vol}(T_2,V)}{\mathrm{vol}(T_1,V)+\mathrm{vol}(T_2,V)} - z\cdot \frac{\mathrm{vol}(T_1,V)+\mathrm{vol}(T_2,V)}{\mathrm{vol}(T_1,V)+\mathrm{vol}(T_2,V)} \\
&= \frac{s(S,T_1)\cdot \mathrm{vol}(T_1,V) + s(S,T_2)\cdot \mathrm{vol}(T_2,V) - z\cdot(\mathrm{vol}(T_1,V)+\mathrm{vol}(T_2,V))}{\mathrm{vol}(T_1,V)+\mathrm{vol}(T_2,V)} \\
&= \frac{(s(S,T_1)-z)\cdot \mathrm{vol}(T_1,V) + (s(S,T_2)-z)\cdot \mathrm{vol}(T_2,V)}{\mathrm{vol}(T_1,V)+\mathrm{vol}(T_2,V)} \\
&= \frac{e(S,T_1)\cdot \mathrm{vol}(T_1,V) + e(S,T_2)\cdot \mathrm{vol}(T_2,V)}{\mathrm{vol}(T_1,V)+\mathrm{vol}(T_2,V)}.
\end{aligned} \tag{28}
$$

□

We now use Proposition 4.3 to show that if partition $\{S, T_1 \cup T_2\}$ of graph G is a partition with minimum *simplified normalized cut value* $\mathrm{SNCut}(S, T_1 \cup T_2) = z$ then $e(T_1, T_2) \geq 0$.

Lemma 4.4. *If partition $\{S, T_1 \cup T_2\}$ is a partition with minimum simplified normalized cut value of a graph G, then $e(T_1, T_2) \geq 0$. If $e(T_1, T_2) = 0$, then partitions $\{S \cup T_1, T_2\}$ and $\{S \cup T_2, T_1\}$ are also partitions of G with minimum simplified normalized cut value.*

Proof. According to Proposition 4.3 and because $\{S, T_1 \cup T_2\}$ is a partition with minimum *simplified normalized cut value* we have:

$$
e(S, T_1 \cup T_2) = \frac{e(S,T_1)\cdot \mathrm{vol}(T_1,V) + e(S,T_2)\cdot \mathrm{vol}(T_2,V)}{\mathrm{vol}(T_1,V)+\mathrm{vol}(T_2,V)} = 0, \tag{29}
$$

$$
e(T_1, S \cup T_2) = \frac{e(T_1,S)\cdot \mathrm{vol}(S,V) + e(T_1,T_2)\cdot \mathrm{vol}(T_2,V)}{\mathrm{vol}(S,V)+\mathrm{vol}(T_2,V)} \geq 0, \tag{30}
$$

$$
e(T_2, S \cup T_1) = \frac{e(T_2,S)\cdot \mathrm{vol}(S,V) + e(T_2,T_1)\cdot \mathrm{vol}(T_1,V)}{\mathrm{vol}(S,V)+\mathrm{vol}(T_1,V)} \geq 0. \tag{31}
$$

Recall that edge-weights are always positive. In Equation 29 we see that $e(S, T_1 \cup T_2)$ is a weighted average of $e(S, T_1)$ and $e(S, T_2)$. Therefore, we have that either $e(S, T_1) \leq 0$ and $e(S, T_2) \geq 0$ or $e(S, T_1) \geq 0$ and $e(S, T_2) \leq 0$. Suppose that $e(T_1, T_2) < 0$. First assume that $e(S, T_1) \leq 0$ and $e(S, T_2) \geq 0$. Now if we look at Equation 30 we see that $e(T_1, S \cup T_2)$ is a weighted average of $e(T_1, S)$ and $e(T_1, T_2)$. Therefore, since $e(T_1, T_2) < 0$ and $e(T_1, S) \leq 0$,

it follows that $e(T_1, S \cup T_2) < 0$. This is a contradiction, since this means that partition $\{T_1, S \cup T_2\}$ has a lower *simplified normalized cut value* than $\{S, T_1 \cup T_2\}$, which has a minimum *simplified normalized cut value*. Now assume that $e(S, T_1) \geq 0$ and $e(S, T_2) \leq 0$. Now if we look at Equation 31 we see that $e(T_2, S \cup T_1)$ is a weighted average of $e(T_2, S)$ and $e(T_2, T_1)$. Therefore, since $e(T_2, T_1) < 0$ and $e(T_2, S) \leq 0$, we have $e(T_2, S \cup T_1) < 0$. This is a contradiction, since this means that partition $\{T_2, S \cup T_1\}$ has a lower *simplified normalized cut value* than $\{S, T_1 \cup T_2\}$, which has a minimum *simplified normalized cut value*. Therefore, it must be that $e(T_1, T_2) \geq 0$. This proves the first part of the lemma.

For the second part of the lemma, if $e(T_1, T_2) = 0$, then we have that neither $e(S, T_1) < 0$ nor $e(S, T_2) < 0$, because partition $\{T_1, S \cup T_2\}$ or $\{T_2, S \cup T_1\}$ will have a lower *simplified normalized cut value* than $\{S, T_1 \cup T_2\}$. However, it still holds that either $e(S, T_1) \leq 0$ and $e(S, T_2) \geq 0$ or $e(S, T_1) \geq 0$ and $e(S, T_2) \leq 0$. Furthermore, since $e(S, T_1 \cup T_2) = 0$ is a weighted average of $e(S, T_1)$ and $e(S, T_2)$, it must be that $e(S, T_1) = e(S, T_2) = 0$. Therefore, $e(S, T_1 \cup T_2) = e(T_1, S \cup T_2) = e(T_2, S \cup T_1) = 0$ and partitions $\{T_1, S \cup T_2\}$ and $\{T_2, S \cup T_1\}$ also have a minimum *simplified normalized cut value*. □

Lemma 4.5. *The partition of a connected graph with minimum simplified normalized cut value has exactly two connected components.*

Proof. Assume that partition $\{S, T\}$ is a partition of a graph with minimum *simplified normalized cut value*. Now assume w.l.o.g. that T is not connected and can be partitioned into two subsets of vertices T_1 and T_2 such that there are no edges between T_1 and T_2 and thus they are disconnected. Then $s(T_1, T_2) = 0$ and $e(T_1, T_2) < 0$, since it is $s(T_1, T_2)$ subtracted by the minimum *simplified normalized cut value*, contradicting Lemma 4.4. Therefore, it must be that S is connected. By symmetry, S is also connected. □

Theorem 4.6. *The partition of a connected graph with minimum normalized cut value has exactly two connected components.*

Proof. Lemma 4.5 proves that a partition $\{S, T\}$ of a graph G with minimum *simplified normalized cut value* has connected *components* S and T. In Section 2 we saw that the Normalized Cut problem can be reduced to the Simplified Normalized Cut problem. This means that since $\{S, T\}$ has a minimum *simplified normalized cut value*, $\{S, T\}$ also has a minimum *normalized cut value*. Therefore, we conclude that a partition of a graph with minimum *normalized cut value* has exactly two connected *components*. □

5 Complexity of the Normalized Cut problem on $\mathcal{H}$-free graph classes

In this section, the complexity of the Normalized Cut problem on $\mathcal{H}$-free graph classes is investigated. In the thesis of Vincken [25] the NP-completeness of the Normalized Cut problem on a so-called *diamond graph* is proven. In Figure 3 the *diamond graph* is shown.

The *diamond graph* is $\mathcal{H}$-free for the following non-exhaustive set of graphs $\mathcal{H}$:

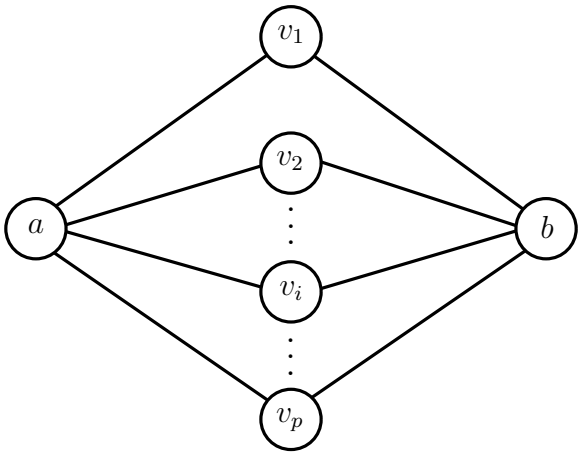


Figure 3: Diamond graph

1. P_t for $t \geq 4$
2. ($P_2 + sP_1$ for a constant s)
3. C_t for $t \geq 3$ and $t \neq 4$

Therefore, the Normalized Cut problem is NP-complete on the class of graphs that are $\mathcal{H}$-free for this set of graphs $\mathcal{H}$. In Subsections 5.1, 5.2, and 5.3 we show that the Normalized Cut problem is NP-complete on claw-free, split, and complete graphs respectively.

5.1 NP-complete on claw-free graphs

An important set of graphs that are not present in the set of graphs $\mathcal{H}$ for which the *diamond graph* is $\mathcal{H}$-free, are the $K_{1,t}$ graphs. Therefore, to show that the Normalized Cut problem is NP-complete on $K_{1,t}$-free graphs for a given t we need other NP-completeness proofs with graph constructions that are $K_{1,t}$-free. Luckily, a useful property of $K_{1,t}$-free graphs is that if we can prove for some t that the Normalized Cut problem is NP-complete, then the Normalized Cut problem is also NP-complete on $K_{1,l}$-free graphs for $l > t$. The reason for this is if you have a $K_{1,l}$ graph we can just remove vertices from the set of l vertices until we have a $K_{1,t}$ graph. Therefore, a proof of NP-completeness of the Normalized Cut problem on $K_{1,3}$-free graphs would imply NP-completeness of $K_{1,t}$-free graphs with $t \geq 3$. $K_{1,3}$-free graphs are also called claw-free graphs.

For the proof of NP-completeness of the Normalized Cut problem on claw-free graphs, we use a reduction from another NP-complete problem, the Partition problem. The Partition problem is defined in Definition 5.1.

Definition 5.1 (PARTITION problem)**.** *For a given multiset $\mathcal{S}$ of $p > 1$ natural numbers $\{x_1, x_2, \ldots, x_p\}$ with total sum $\sum_{i=1}^{p} x_i = 2D$, is it possible to partition $\mathcal{S}$ into two multisets $\mathcal{S}_1$ and $\mathcal{S}_2$ where $\sum_{x_i \in \mathcal{S}_1} x_i = D$ and $\sum_{x_i \in \mathcal{S}_2} x_i = D$?*

Note that assuming $p > 1$ is reasonable since $p = 1$ is a trivial NO-instance.

The proof of NP-completeness of the NORMALIZED CUT problem on claw-free graphs is based on the proof of NP-completeness of the NORMALIZED CUT problem on grids by Shi and Malik [20].

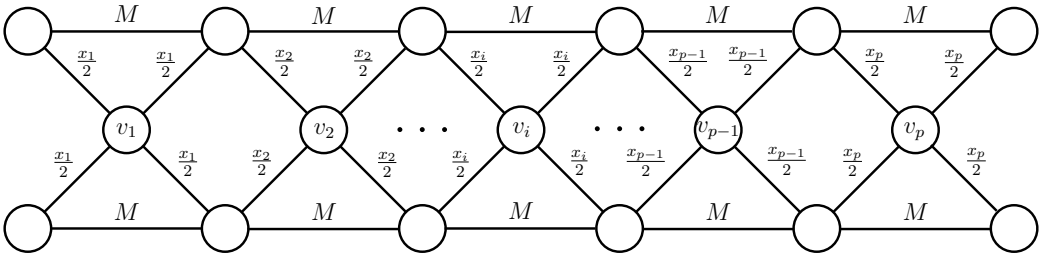


Figure 4: Claw-free graph for the reduction of Theorem 5.1

Theorem 5.1. *The* NORMALIZED CUT *problem is NP-complete on claw-free graphs.*

Proof. First note that the NORMALIZED CUT problem on claw-free graphs is in NP. We use a reduction from the PARTITION problem to prove that the NORMALIZED CUT problem on claw-free graphs is NP-complete. For the graph construction we create a claw-free graph, which can be found in Figure 4. The relation with the PARTITION problem is that for every natural number $x_i \in \mathcal{S}$ a cross-like structure is added to the graph. The top and bottom edges of this structure have a weight of $M = 2D$ and the cross edges all have a value of $x_i/2$. We call the vertices in the middle of the cross structures $P = \{v_1, v_2, \ldots, v_i, \ldots, v_{p-1}, v_p\}$. Furthermore, we call the set of $p + 1$ vertices at the top of the graph T and we call the set of $p + 1$ vertices at the bottom of the graph B. Lastly, we call the set of edges with weight $x_i/2$ at the top of the crosses Q and the set of edges with weight $x_i/2$ at the bottom of the crosses R.

We show that there exists a partition with *normalized cut value* $\leq N$ if and only if we have a YES-instance for the PARTITION problem. We define N as:

$$N = \frac{2D(4pM + 8D)}{(2pM + 4D) \cdot (2pM + 4D)}. \tag{32}$$

Assume we have a YES-instance for the PARTITION problem. This means that there is a partition of $\mathcal{S}$ into $\mathcal{S}_1$ and $\mathcal{S}_2$ where $\sum_{x_i \in \mathcal{S}_1} = D$ and $\sum_{x_i \in \mathcal{S}_2} = D$. A partition $\{S, \bar{S}\}$ with

normalized cut value $\mathrm{NCut}(S, \bar{S}) \leq N$ of the claw-free graph G in Figure 4 cuts for every $x_i \in \mathcal{S}_1$ the top two cross edges $\{t_{i1}, t_{i2}\} \in Q$ and for every $x_i \in \mathcal{S}_2$ the bottom two cross edges $\{b_{i1}, b_{i2}\} \in R$. Since $\mathcal{S}_1 \cup \mathcal{S}_2 = \mathcal{S}$, the *cut value* $\mathrm{cut}(S, \bar{S})$ is $\sum_{x_i \in \mathcal{S}} 2 \cdot x_i/2 = 2D$.

We claim that the *volume* $\mathrm{vol}(S, V)$ consists of $\mathrm{cut}(S, \bar{S}) = 2D$ and two times the edge-weights of the edges in S. The edges that lie completely in S are the p edges between the vertices of T at the top of the graph and the top two cross edges $\{t_{i1}, t_{i2}\} \in Q$ of the crosses where the bottom two cross edges $\{b_{i1}, b_{i2}\} \in R$ are in the *cut*. The p edges have a total weight of pM. The edges $\{b_{i1}, b_{i2}\} \in R$ are in the *cut* when $x_i \in \mathcal{S}_2$ and therefore have a weight of $\sum_{x_i \in \mathcal{S}_2} = D$. Since $t_{i1} + t_{i2} = b_{i1} + b_{i2}$ the total weight of the edges $\{t_{i1}, t_{i2}\} \in Q$ that lie completely in S is D. For the *volume*, we need to count the edge-weights of the edges completely in the *component* twice. Therefore, we count the total edge-weight of the edges in *component* S twice which is $2D + 2pM$. Together with *cut value* $\mathrm{cut}(S, \bar{S}) = 2D$ we have $\mathrm{vol}(S, V) = 2D + 2D + 2pM = 2pM + 4D$. Similar, we can show that $\mathrm{vol}(\bar{S}, V) = 2pM + 4D$.

The *normalized cut value* $\mathrm{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ is:

$$\mathrm{NCut}(S, \bar{S}) = \frac{2D(4pM + 8D)}{(2pM + 4D) \cdot (2pM + 4D)} \leq N. \tag{33}$$

Conversely, assume that we have a partition $\{S, \bar{S}\}$ with *normalized cut value* $\mathrm{NCut}(S, \bar{S}) \leq N$. We need to show that we have a YES-instance for the Partition problem. We distinguish three possible ways to partition the claw-free graph G in Figure 4. The first way we can partition G is defined by its *cut*, which has at least one edge with weight M in it. The first partition therefore covers all partitions that have one or more edges with weight M in their cut. The second way we can partition G has some subset of vertices $P_a \subseteq P$ in S, and all other vertices T, B, and $P \setminus P_a$ in $\bar{S}$. The third way we can partition G has a *cut* from left to right, cutting either the top two edges $\{t_{i1}, t_{i2}\} \in T$ of each cross or the bottom two edges $\{b_{i1}, b_{i2}\} \in B$ of each cross. Note that these last two partitions have a *cut* where either no edges, the two top edges, the two bottom edges, or all four edges around a v_i are in the *cut*. Therefore, these last two partitions cover all possible partitions where there is no edge with weight M in the *cut*, precisely all partitions we did not cover with the first way we can partition the graph. We investigate the three ways we can partition G to see if they can have $\mathrm{NCut}(S, \bar{S}) \leq N$:

1. The first way we can partition G, is a partition $\{S, \bar{S}\}$ that has a *cut* where one or more edges with weight M are in the *cut*. Without loss of generality assume that it is an edge a between two vertices of T. a is part of a cycle with length three in the graph, where the other two edges of this cycle are two cross edges $\{t_{i1}, t_{i2}\} \in Q$. Cutting through a cycle can only be achieved by cutting at least two edges. Therefore, at least one of the two cross edges t_{i1} or t_{i2} with weight $x_i/2$ must also be in the *cut*. The *cut value* $\mathrm{cut}(S, \bar{S})$ is therefore at least $M + x_i/2$. Since the smallest possible

cut value is known, Proposition 4.1 tells us that $\{S, \bar{S}\}$ achieves its lowest *normalized cut value* $\text{NCut}(S, \bar{S})$ if the *volumes* differ the least in value and are thus equal: $\text{vol}(S, V) = \text{vol}(\bar{S}, V)$. Therefore, $\{S, \bar{S}\}$ with $\text{vol}(S, V) = \text{vol}(\bar{S}, V)$ has the lowest *normalized cut value* and both *volumes* have a value of exactly half the total *volume* of the graph $2pM + 4D$. The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ is at least:

$$\text{NCut}(S, \bar{S}) \geq \frac{(M + \frac{x_i}{2}) \cdot (4pM + 8D)}{(2pM + 4D) \cdot (2pM + 4D)}. \tag{34}$$

Since $M = 2D$ the following inequality holds:

$$\text{NCut}(S, \bar{S}) \geq \frac{(2D + \frac{x_i}{2}) \cdot (4pM + 8D)}{(2pM + 4D) \cdot (2pM + 4D)} > \frac{2D(4pM + 8D)}{(2pM + 4D) \cdot (2pM + 4D)} = N. \tag{35}$$

Therefore, $\{S, \bar{S}\}$ does not have $\text{NCut}(S, \bar{S}) \leq N$.

2. The second way we can partition G, is a partition $\{S, \bar{S}\}$ that has some subset of vertices $P_a \subseteq P$ in S, and all other vertices T, B, and $P \setminus P_a$ in $\bar{S}$. The *cut value* $\text{cut}(S, \bar{S})$ is:

$$\text{cut}(S, \bar{S}) = \sum_{v_i \in P_a} 4 \cdot \frac{x_i}{2} = \sum_{v_i \in P_a} 2x_i. \tag{36}$$

When we look at the *volume* $\text{vol}(S, V)$ we notice that there are no edges completely inside S and $\text{vol}(S, V)$ is therefore equal to $\text{cut}(S, \bar{S})$:

$$\text{vol}(S, V) = \text{cut}(S, \bar{S}) = \sum_{v_i \in P_a} 2x_i. \tag{37}$$

The other *volume* $\text{vol}(\bar{S}, V)$ is the total *volume* $\text{vol}(V, V)$ of G minus $\text{vol}(S, V)$ which is:

$$\text{vol}(\bar{S}, V) = 4pM + 8D - \sum_{v_i \in P_a} 2x_i. \tag{38}$$

The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ therefore is:

$$\mathrm{NCut}(S,\bar{S}) = \frac{(\sum\limits_{v_i \in P_a} 2x_i) \cdot (4pM + 8D)}{(\sum\limits_{v_i \in P_a} 2x_i) \cdot (4pM + 8D - \sum\limits_{v_i \in P_a} 2x_i)} > 1 > \frac{2D(4pM + 8D)}{(2pM + 4D) \cdot (2pM + 4D)} = N. \tag{39}$$

Therefore, $\{S, \bar{S}\}$ does not have a *normalized cut value* $\leq N$.

3. The third and last way we can partition G, is a partition $\{S, \bar{S}\}$ that has a *cut* from left to right, cutting either the top two edges $\{t_{i1}, t_{i2}\} \in Q$ of each cross or the bottom two edges $\{b_{i1}, b_{i2}\} \in R$ of each cross. Therefore, the set of vertices P is divided over the two *components* S and $\bar{S}$. Let $P_a \subseteq P$ be the vertices that are in *component* S and let $P_b = P \setminus P_a$ be the vertices that are in *component* $\bar{S}$. Notice that the *cut value* $\mathrm{cut}(S, \bar{S})$ is $\sum\limits_{v_i \in P} 2 \cdot x_i/2 = 2D$, because for each cross we put either $\{t_{i1}, t_{i2}\}$ or $\{b_{i1}, b_{i2}\}$ in the *cut*. The *volume* $\mathrm{vol}(S, V)$ of S is:

$$\mathrm{vol}(S, V) = 2 \sum_{v_i \in P_b} 2 \cdot \frac{x_i}{2} + 2D + 2pM. \tag{40}$$

Similarly the *volume* $\mathrm{vol}(\bar{S}, V)$ of $\bar{S}$ is:

$$\mathrm{vol}(\bar{S}, V) = 2 \sum_{v_i \in P_a} 2 \cdot \frac{x_i}{2} + 2D + 2pM. \tag{41}$$

The *normalized cut value* $\mathrm{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ therefore is:

$$\frac{2D(4pM + 8D)}{(2 \sum\limits_{v_i \in P_b} 2\frac{x_i}{2} + 2D + 2pM) \cdot (2 \sum\limits_{v_i \in P_a} 2\frac{x_i}{2} + 2D + 2pM)}. \tag{42}$$

Since the *cut value* is known, Proposition 4.1 tells us that $\{S, \bar{S}\}$ achieves the lowest *normalized cut value* when the *volumes* differ the least in value and are thus equal, $\mathrm{vol}(S, V) = \mathrm{vol}(\bar{S}, V)$. $\mathrm{vol}(S, V) = \mathrm{vol}(\bar{S}, V)$ can only happen when the following equality holds:

$$2 \sum_{v_i \in P_b} 2\frac{x_i}{2} = 2 \sum_{v_i \in P_a} 2\frac{x_i}{2} = 2D. \tag{43}$$

The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ is:

$$\frac{2D(4pM + 8D)}{(2pM + 4D) \cdot (2pM + 4D)} = N. \tag{44}$$

In this partition $\text{vol}(S, V)$ and $\text{vol}(\bar{S}, V)$ needed to be perfectly balanced. As we saw, this is only possible if Equality 43 holds. We can rewrite Equality 43 to:

$$\sum_{v_i \in P_b} x_i = \sum_{v_i \in P_a} x_i = D. \tag{45}$$

We see that the only way for Equality 45 to hold, is if the PARTITION problem is a YES-instance.

Since the NORMALIZED CUT problem on claw-free graphs is in NP and has a partition of claw-free graph G in Figure 4 with a *normalized cut value* $\leq N$ if and only if the PARTITION problem is a YES-instance, it is NP-complete. □

5.2 NP-complete on split graphs

Another important graph that is not present in the set of graphs $\mathcal{H}$ of which the *diamond graph* in Figure 3 is $\mathcal{H}$-free, is the C_4-free graph. We can make a small adjustment to the *diamond graph* to make it C_4-free. To make it C_4-free we add an edge e between vertices a and b and the resulting graph can be found in Figure 5. This $\mathcal{H}$-free graph is also a split graph which is defined as the $\mathcal{H}$-free graph for the set of graphs $H = \{2P_2, C_4, C_5\}$.

Theorem 5.2. *The* NORMALIZED CUT *problem is NP-complete on split graphs.*

Proof. The proof is similar to the proof of NP-completeness of the NORMALIZED CUT problem for graphs with a vertex cover number equal to two presented by Vincken [25]. From now on we refer to this proof as "the proof by Vincken". For the graph construction we first create a *diamond graph* in the same way as in the proof by Vincken which we can see in Figure 3. Let W be the total edge-weight of all edges in the *diamond graph*. Note that the *diamond graph* is constructed from an instance of the PARTITION problem in the proof by Vincken and therefore $W/2$ is the total sum of elements of the PARTITION problem. Now we add an edge e with weight $1/W$ between vertices a and b to get a split graph which we can see in Figure 5. Note that the graph construction is still a polynomial-time reduction from the PARTITION problem when we add edge e. We will see in the analysis of the *normalized cut values* of different partitions of G, that we have chosen the weight of e to be small enough, such that e with weight $1/W$ does not change the *normalized cut values* of the partitions significantly.

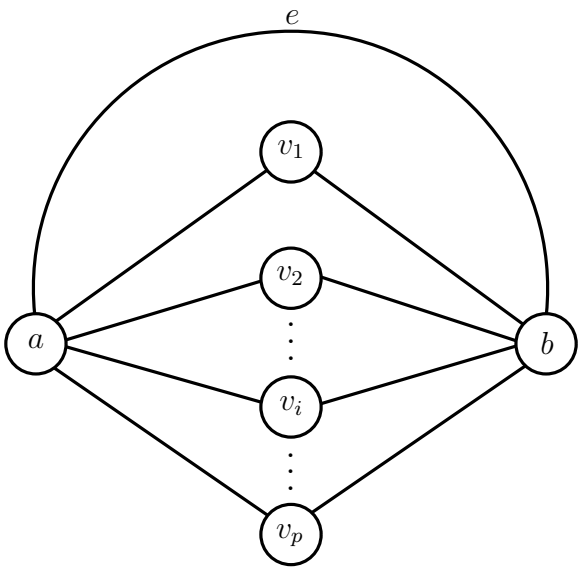


Figure 5: Split graph

First note that the NORMALIZED CUT problem on split graphs is in NP. We show that the PARTITION problem is a YES-instance if and only if we can find a partition of the split graph with a *normalized cut value* $\leq N$. We define N as:

$$N = \frac{(\frac{W}{2} + \frac{1}{W}) \cdot (2W + \frac{2}{W})}{(W + \frac{1}{W})^2} \tag{46}$$

If we have a YES-instance of the PARTITION problem, we create the same partition $\{S, \bar{S}\}$ as in the proof by Vincken. Notice that now edge e is extra in the *cut*. The *cut value* $\text{cut}(S, \bar{S})$ is:

$$\text{cut}(S, \bar{S}) = \frac{W}{2} + \frac{1}{W}. \tag{47}$$

The *volumes* $\text{vol}(S, V)$ and $\text{vol}(\bar{S}, V)$ also increase by $1/W$. The *volume* $\text{vol}(S, V)$ of S is:

$$\text{vol}(S, V) = W + \frac{1}{W}. \tag{48}$$

Similarly, the *volume* $\text{vol}(\bar{S}, V)$ of $\bar{S}$ is:

$$\text{vol}(\bar{S}, V) = W + \frac{1}{W}. \tag{49}$$

$\{S, \bar{S}\}$ therefore has a *normalized cut value* $\text{NCut}(S, \bar{S})$ of:

$$\text{NCut}(S, \bar{S}) = \frac{(\frac{W}{2} + \frac{1}{W}) \cdot (2W + \frac{2}{W})}{(W + \frac{1}{W}) \cdot (W + \frac{1}{W})} \leq N. \tag{50}$$

We conclude that if we have a YES-instance of the PARTITION problem, then we have a partition with *normalized cut value* $\leq N$.

Now the other way around, if we have a partition $\{S, \bar{S}\}$ with a *normalized cut value* $\leq N$ then the PARTITION problem is a YES-instance. We distinguish two possible partitions of the split graph. One where vertices a and b are in the same *component*, and one where they are in different *components*. We investigate the two partitions to see if they can have a *normalized cut value* $\text{NCut}(S, \bar{S}) \leq N$. We will show that if a and b are in the same component, the *normalized cut value* of such a partition cannot be $\leq N$. Furthermore, we will show that if a and b are in different *components*, the *normalized cut value* of such a partition can only be $\leq N$ for a specific partition that can be mapped to a YES-instance of the PARTITION problem.

1. The first partition $\{S, \bar{S}\}$ of split graph G has both vertices a and b in one *component*. W.l.o.g. assume that a and b are in *component* $\bar{S}$. Let $I \subseteq \{1, \ldots, n\}$, $\bar{I} = \{1, \ldots, n\} \setminus I$, $S = \{v_i | i \in I\}$ and $\bar{S} = V \setminus S$. We will show that $\{S, \bar{S}\}$ does not have a *normalized cut value* $\leq N$.

   As we see in the proof by Vincken, the s_i and t_i edges for $v_i \in S$ are in the *cut*. Therefore, the *cut value* $\text{cut}(S, \bar{S})$ is:

   $$\text{cut}(S, \bar{S}) = \sum_{i \in I} (w(s_i) + w(t_i)). \tag{51}$$

   The *volume* $\text{vol}(S, V)$ of S is equal to the *cut value* $\text{cut}(S, \bar{S})$ and therefore is:

   $$\text{vol}(S, V) = \sum_{i \in I} (w(s_i) + w(t_i)). \tag{52}$$

   Note that e lies completely in *component* $\bar{S}$ and therefore the edge-weight $1/W$ is added twice to the *volume* $\text{vol}(\bar{S}, V)$. Furthermore, the edge-weights of the s_i and t_i edges for $v_i \in \bar{S}$ are also added twice to the *volume* $\text{vol}(\bar{S}, V)$. Lastly, we need to add the *cut value* $\text{cut}(S, \bar{S})$ to the *volume* $\text{vol}(\bar{S}, V)$. The *volume* $\text{vol}(\bar{S}, V)$ of S therefore is:

$$\text{vol}(\bar{S}, V) = \sum_{i \in I}(w(s_i) + w(t_i)) + 2\sum_{i \in \bar{I}}(w(s_i) + w(t_i)) + \frac{2}{W}. \tag{53}$$

The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ therefore is:

$$\begin{aligned}\text{NCut}(S, \bar{S}) &\geq \frac{(\sum_{i \in I}(w(s_i) + w(t_i))) \cdot (2W + \frac{2}{W})}{(\sum_{i \in I}(w(s_i) + w(t_i))) \cdot (\sum_{i \in I}(w(s_i) + w(t_i)) + 2\sum_{i \in \bar{I}}(w(s_i) + w(t_i)) + \frac{2}{W})} \\ &= \frac{(2W + \frac{2}{W})}{\sum_{i \in I}(w(s_i) + w(t_i)) + 2\sum_{i \in \bar{I}}(w(s_i) + w(t_i)) + \frac{2}{W}}.\end{aligned} \tag{54}$$

We have $W = \sum_{i \in I}(w(s_i) + w(t_i)) + \sum_{i \in \bar{I}}(w(s_i) + w(t_i))$ and these terms are dependent on each-other. If we increase $\sum_{i \in I}(w(s_i) + w(t_i))$, the denominator decreases since it is one time $\sum_{i \in I}(w(s_i) + w(t_i))$ and two times $\sum_{i \in \bar{I}}(w(s_i) + w(t_i))$. Therefore, if we increase $\sum_{i \in I}(w(s_i) + w(t_i))$, we increase the *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$. To minimize $\text{NCut}(S, \bar{S})$, we therefore minimize $\sum_{i \in I}(w(s_i) + w(t_i))$.

Let edges s_i and t_i be the edges with the smallest weight apart from the weight of e. To minimize $\sum_{i \in I}(w(s_i) + w(t_i))$, we only put v_i in S and the rest of the vertices in $\bar{S}$. The way the split graph was constructed, means that the weights of s_i and t_i are equal to the smallest element of the PARTITION problem instance. Since the elements of a PARTITION problem instance are natural numbers, the weights of s_i and t_i are at least 1. We therefore assume that $w(s_i) = w(t_i) = 1$, since this minimizes the first term of the denominator and therefore minimizes the *normalized cut value* $\text{NCut}(S, \bar{S})$. The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ therefore is:

$$\text{NCut}(S, \bar{S}) = \frac{2W + \frac{2}{W}}{2W - 2 + \frac{2}{W}}. \tag{55}$$

We compare this Fraction 55 to N and see if it is $\leq N$:

$$\frac{2W + \frac{2}{W}}{2W - 2 + \frac{2}{W}} \leq N = \frac{(\frac{W}{2} + \frac{1}{W}) \cdot (2W + \frac{2}{W})}{(W + \frac{1}{W})^2} \tag{56}$$

We let WolframAlpha [1] solve Inequality 56 and see that it holds for $W \leq 0.56984$. An instance of the PARTITION problem must have at least two elements of a value of at least 1. For both elements we would add two edges to the split graph with value 1 and therefore the total weight W of G must be at least 4. Since we have that Inequality 56 only holds for $W \leq 0.56984$, $\{S, \bar{S}\}$ does not have $\text{NCut}(S, \bar{S}) \leq N$.

2. The second partition $\{S, \bar{S}\}$ of split graph G has vertices a and b in different *components*. W.l.o.g. assume that $a \in S$ and $b \in \bar{S}$. We will show that only for a specific partition we get a *normalized cut value* $\leq N$. Furthermore, this specific partition can be mapped to a YES-instance of the PARTITION problem.

   We see in the proof by Vincken that the s_i and t_i edges in the *cut* have a total edge-weight of $W/2$. Now edge e is extra in the *cut*. Therefore, the *cut value* $\text{cut}(S, \bar{S})$ is:

$$\text{cut}(S, \bar{S}) = \frac{W}{2} + \frac{1}{W}. \tag{57}$$

   The *volume* $\text{vol}(S, V)$ consists of the *cut value* $\text{cut}(S, \bar{S})$ and two times the edge-weights of the edges (a, v_i) where $v_i \in S$. The *volume* $\text{vol}(S, V)$ of S therefore is:

$$\text{vol}(S, V) = \frac{W}{2} + \frac{1}{W} + 2\sum_{v_i \in S}(w(s_i)). \tag{58}$$

   Similarly, the *volume* $\text{vol}(\bar{S}, V)$ consists of the *cut value* $\text{cut}(S, \bar{S})$ and two times the edge-weights of the edges (b, v_i) where $v_i \in \bar{S}$. The *volume* $\text{vol}(\bar{S}, V)$ of $\bar{S}$ therefore is:

$$\text{vol}(\bar{S}, V) = \frac{W}{2} + \frac{1}{W} + 2\sum_{v_i \in \bar{S}}(w(t_i)). \tag{59}$$

   The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ therefore is:

$$\text{NCut}(S, \bar{S}) = \frac{(\frac{W}{2} + \frac{1}{W}) \cdot (2W + \frac{2}{W})}{(\frac{W}{2} + \frac{1}{W} + \sum\limits_{v_i \in S} 2w(s_i)) \cdot (\frac{W}{2} + \frac{1}{W} + \sum\limits_{v_i \in \bar{S}} 2w(t_i))}. \tag{60}$$

   Since the *cut value* $\text{cut}(S, \bar{S})$ is known, we use Proposition 4.1 to see that $\{S, \bar{S}\}$ has the lowest *normalized cut value* if the *volumes* are equal, $\text{vol}(S, V) = \text{vol}(\bar{S}, V)$. To achieve $\text{vol}(S, V) = \text{vol}(\bar{S}, V)$ we need the following equality to hold:

$$\sum_{v_i \in S} 2w(s_i) = \sum_{v_i \in \bar{S}} 2w(t_i) = \frac{W}{2}. \tag{61}$$

The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ is:

$$\begin{aligned} \text{NCut}(S, \bar{S}) &= \frac{(\frac{W}{2} + \frac{1}{W}) \cdot (2W + \frac{2}{W})}{(\frac{W}{2} + \frac{1}{W} + \frac{W}{2}) \cdot (\frac{W}{2} + \frac{1}{W} + \frac{W}{2})} \\ &= \frac{(\frac{W}{2} + \frac{1}{W}) \cdot (2W + \frac{2}{W})}{(W + \frac{1}{W})^2} \\ &= N \end{aligned} \tag{62}$$

Therefore, $\text{NCut}(S, \bar{S}) \leq N$. Furthermore, if Equality 61 does not hold, then the *volumes* are not equal and according to Proposition 4.1 we will have a higher *normalized cut value*. Since $\text{NCut}(S, \bar{S})$ is actually equal to N, any increase in $\text{NCut}(S, \bar{S})$ will make sure that we do not have $\text{NCut}(S, \bar{S}) \leq N$. Therefore, if Equality 61 does not hold we do not have $\text{NCut}(S, \bar{S}) \leq N$. We conclude that Equality 61 needs to hold for $\{S, \bar{S}\}$ to have $\text{NCut}(S, \bar{S}) \leq N$ and this is exactly the case when the PARTITION problem is a YES-instance.

We conclude that the PARTITION problem is a YES-instance if and only if we can find a partition with a *normalized cut value* $\leq N$. □

5.3 NP-complete on complete graphs

Another important graph that is not present in the set of graphs $\mathcal{H}$ of which the *diamond graph* in Figure 3 is $\mathcal{H}$-free, is the P_3-free graph. P_3-free graphs are also called cluster graphs. A cluster graph is a graph that consists of a disjoint union of cliques. If a cluster graph consists of one clique, it is called a complete graph. We will prove that the NORMALIZED CUT problem is NP-complete on complete graphs. We can adjust the *diamond graph* in Figure 3 to become a complete graph by adding an edge between every pair of vertices that do not have an edge between them in the *diamond graph*. In specific we call the added edge between vertices a and b, e. The resulting graph can be found in Figure 6.

Theorem 5.3. *The* NORMALIZED CUT *problem is NP-complete on complete graphs.*

Proof. The proof is similar to the proof of NP-completeness of the NORMALIZED CUT problem for graphs with a vertex cover number equal to two presented by Vincken [25]. From now on we refer to this proof as "the proof by Vincken". For the graph construction we first create a *diamond graph* in the same way as in the proof by Vincken which we can see in Figure 3. Let W be the total edge-weight of all edges in the *diamond graph*. Note

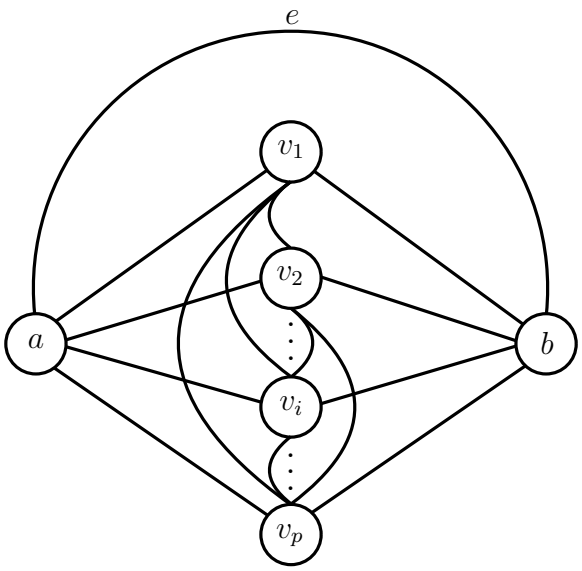


Figure 6: Complete graph

that the *diamond graph* is constructed from an instance of the PARTITION problem in the proof by Vincken and therefore $W/2$ is the total sum of elements of the PARTITION problem. We are going to add an edge e between vertices a and b and we add an edge between every pair of the p vertices $V \setminus \{a, b\}$ to get a complete graph which we can see in Figure 6. The total amount of edges we add is therefore $1 + (p(p-1))/2$. We give every edge that we add a weight ϵ of:

$$\epsilon = \frac{1}{(1 + (p(p-1))/2) \cdot W^2} \tag{63}$$

Therefore, if we sum all the edge-weights of the newly added edges we get a total edge-weight of:

$$(1 + (p(p-1))/2)) \cdot \frac{1}{(1 + (p(p-1))/2) \cdot W^2} = \frac{1}{W^2} \tag{64}$$

Note that the graph construction is still a polynomial-time reduction from the PARTITION problem when we add the $1 + (p(p-1))/2$ edges. We will see in the analysis of the *normalized cut values* of different partitions of G, that we have chosen the weight of the newly added edges to be small enough, such that the newly added edges with weight ϵ do not change the *normalized cut values* of the partitions significantly.

First note that the NORMALIZED CUT problem on complete graphs is in NP. We show that the PARTITION problem is a YES-instance if and only if we can find a partition of the complete graph with a *normalized cut value* $\leq N$. We define N as:

$$N = \frac{(\frac{W}{2} + \frac{1}{W^2}) \cdot (2W + \frac{2}{W^2})}{(\frac{2}{W^2} + W) \cdot W} \tag{65}$$

If we have a YES-instance of the PARTITION problem, we create the same partition $\{S, \bar{S}\}$ as in the proof by Vincken. We will show that even in the case where the newly added edges increase the *normalized cut value* $\mathrm{NCut}(\mathrm{S}, \bar{\mathrm{S}})$ as much as possible, we still have $\mathrm{NCut}(\mathrm{S}, \bar{\mathrm{S}}) \leq N$. We take the *cut value* as in the proof by Vincken and assume that the total edge-weight $\frac{1}{W^2}$ of the newly added edges is added to the *cut value*. The *cut value* $\mathrm{cut}(S, \bar{S})$ is:

$$\mathrm{cut}(S, \bar{S}) = \frac{W}{2} + \frac{1}{W^2}. \tag{66}$$

We also take the *volumes* as in the proof by Vincken. According to Proposition 4.1, for a given *cut value* the *normalized cut value* is the lowest if the *volumes* differ the least in value. Since the *volumes* were equal in the proof by Vinkcen, we add the extra *volume* $\frac{2}{W^2}$ from the edge-weights of the newly added edges to one *component* S to increase the difference between the *components* as much as possible. The *volume* $\mathrm{vol}(S, V)$ of S is:

$$\mathrm{vol}(S, V) = W + \frac{2}{W^2}. \tag{67}$$

The *volume* $\mathrm{vol}(\bar{S}, V)$ of $\bar{S}$ still is:

$$\mathrm{vol}(\bar{S}, V) = W. \tag{68}$$

The *normalized cut value* $\mathrm{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ therefore is:

$$\mathrm{NCut}(S, \bar{S}) \leq \frac{(\frac{W}{2} + \frac{1}{W^2}) \cdot (2W + \frac{2}{W^2})}{(W + \frac{2}{W^2}) \cdot W} \leq N. \tag{69}$$

We conclude that if we have a YES-instance of the PARTITION problem, then we have a partition with *normalized cut value* $\leq N$.

Now the other way around, if we have a partition $\{S, \bar{S}\}$ with a *normalized cut value* $\mathrm{NCut}(S, \bar{S}) \leq N$ then the PARTITION problem is a YES-instance. We distinguish two possible partitions of the complete graph. One where vertices a and b are in the same *component*, and one where they are in different *components*. We investigate the two partitions to see if they can have $\mathrm{NCut}(S, \bar{S}) \leq N$. We will show that if a and b are in the same component, the *normalized cut value* of such a partition cannot be $\leq N$. Furthermore, we will show that if a and b are in different *components*, the *normalized cut value* of such a partition can only be $\leq N$ for a specific partition that can be mapped to a YES-instance of the PARTITION problem.

1. The first partition $\{S, \bar{S}\}$ of complete graph G has both vertices a and b in one *component*. W.l.o.g. assume that a and b are in *component* $\bar{S}$. Let $I \subseteq \{1, \ldots, n\}$, $\bar{I} = \{1, \ldots, n\} \setminus I$, $S = \{v_i | i \in I\}$ and $\bar{S} = V \setminus S$. We will show that $\{S, \bar{S}\}$ does not have a *normalized cut value* $\leq N$, even with two assumptions that decrease the *normalized cut value* of $\{S, \bar{S}\}$ as much as possible. The first assumption is that the edge-weights of the newly added edges are not in the *cut value* $\text{cut}(S, \bar{S})$. This minimizes the *normalized cut value* because according to Proposition 4.2 among all partitions for which the difference between the *volumes* is equal, the *normalized cut value* is the lowest if the *cut value* is minimized. Since the assumption does not change the values of the *volumes*, we minimize the *normalized cut value* of $\{S, \bar{S}\}$. The second assumption is that we add the extra *volume* of the edge-weights of the newly added edges to the *volume* of the *component*, such that we minimize the difference between the the two *volumes* $\text{vol}(S, V)$ and $\text{vol}(\bar{S}, V)$. This minimizes the *normalized cut value* because according to Proposition 4.1 among all partition for which the *cut value* is equal, the *normalized cut value* is the lowest if the difference between the *volumes* is the lowest. Since the assumption does not change the *cut value*, we minimize the *normalized cut value* of $\{S, \bar{S}\}$.

   As we see in the proof by Vincken, the s_i and t_i edges for $v_i \in S$ are in the *cut*. We assume that there are no edge-weights of the newly added edges added to the *cut value*, to minimize the *normalized cut value* of $\{S, \bar{S}\}$. Therefore, the *cut value* $\text{cut}(S, \bar{S})$ is:

$$\text{cut}(S, \bar{S}) = \sum_{i \in I} (w(s_i) + w(t_i)). \tag{70}$$

   We first take the *volumes* as they are in the proof by Vincken. Then we add the extra *volume* $\frac{2}{W^2}$ from the edge-weights of the newly added edges to the *volume* such that the difference between the two *volumes* $\text{vol}(S, V)$ and $\text{vol}(\bar{S}, V)$ is minimized, to minimize the *normalized cut value* of $\{S, \bar{S}\}$. In the proof by Vincken the *volume* $\text{vol}(S, V)$ of S is equal to the *cut value*:

$$\text{vol}(S, V) = \sum_{i \in I} (w(s_i) + w(t_i)). \tag{71}$$

   In the proof by Vincken the *volume* $\text{vol}(\bar{S}, V)$ of $\bar{S}$ is:

$$\text{vol}(\bar{S}, V) = \sum_{i \in I} (w(s_i) + w(t_i)) + 2 \sum_{i \in \bar{I}} (w(s_i) + w(t_i)). \tag{72}$$

The extra *volume* from the edge-weights of the newly added edges is $\frac{2}{W^2}$. We have:

$$\frac{2}{W^2} < 2\sum_{i\in\bar{I}}(w(s_i) + w(t_i)), \tag{73}$$

and therefore we add all the extra *volume* $\frac{2}{W^2}$ to $\text{vol}(S, V)$:

$$\text{vol}(S, V) = \sum_{i\in I}(w(s_i) + w(t_i)) + \frac{2}{W^2}. \tag{74}$$

The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ therefore is:

$$\text{NCut}(S, \bar{S}) \geq \frac{(\sum_{i\in I}(w(s_i) + w(t_i))) \cdot (2W + \frac{2}{W^2})}{(\sum_{i\in I}(w(s_i) + w(t_i)) + \frac{2}{W^2}) \cdot (\sum_{i\in I}(w(s_i) + w(t_i)) + 2\sum_{i\in\bar{I}}(w(s_i) + w(t_i)))}. \tag{75}$$

We have $W = \sum_{i\in I}(w(s_i) + w(t_i)) + \sum_{i\in\bar{I}}(w(s_i) + w(t_i))$ and these terms are dependent on each-other. If we increase $\sum_{i\in I}(w(s_i) + w(t_i))$, the second term of the denominator decreases since it is one time $\sum_{i\in I}(w(s_i) + w(t_i))$ and two times $\sum_{i\in\bar{I}}(w(s_i) + w(t_i))$. Therefore, if we increase $\sum_{i\in I}(w(s_i) + w(t_i))$, we see that the numerator increases more in ratio than the denominator and $\text{NCut}(S, \bar{S})$ increases. To minimize $\text{NCut}(S, \bar{S})$, we therefore minimize $\sum_{i\in I}(w(s_i) + w(t_i))$.

Let edges s_i and t_i be the edges with the smallest weight apart from the weight of e and the other newly added edges. To minimize $\sum_{i\in I}(w(s_i) + w(t_i))$, we only put v_i in S and the rest of the vertices in $\bar{S}$. The way the complete graph was constructed, means that the edge-weights of s_i and t_i are equal to the smallest element of the PARTITION problem instance. Since the elements of a PARTITION problem instance are natural numbers, the edge-weights of s_i and t_i are at least 1. We therefore assume that $w(s_i) = w(t_i) = 1$, since this minimizes the first term of the denominator en therefore minimizes the *normalized cut value* $\text{NCut}(S, \bar{S})$. The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ therefore is:

$$\text{NCut}(S, \bar{S}) \geq \frac{2 \cdot (2W + \frac{2}{W^2})}{(2 + \frac{2}{W^2}) \cdot (2W - 2)} \tag{76}$$

We compare Fraction 76 to N and see if it is $\leq N$:

$$\frac{2 \cdot (2W + \frac{2}{W^2})}{(2 + \frac{2}{W^2}) \cdot (2W - 2)} \leq N = \frac{(\frac{W}{2} + \frac{1}{W^2}) \cdot (2W + \frac{2}{W^2})}{(\frac{2}{W^2} + W) \cdot W}. \tag{77}$$

We let WolframAlpha [1] solve Inequality 77 and see that it holds for $W < 1$. Note that for $W = 1$ we divide by 0. Furthermore, we get negative *volumes* for $W < 1$. Moreover, an instance of the PARTITION problem must have at least two elements of a value of at least 1. For both elements we would add two edges to the complete graph with value 1 and therefore the total weight W of G must be at least 4. Since we have that Inequality 77 only holds for $W < 1$, $\{S, \bar{S}\}$ does not have $\text{NCut}(S, \bar{S}) \leq N$.

2. The second partition $\{S, \bar{S}\}$ of complete graph G has vertices a and b in different *components*. W.l.o.g. assume that $a \in S$ and $b \in \bar{S}$. We will show that only for a specific partition $\{S, \bar{S}\}$ we get a *normalized cut value* $\leq N$. Furthermore, this specific partition can be mapped to a YES-instance of the PARTITION problem.

   We first look at the *normalized cut value* $\text{NCut}(S, \bar{S})$ of partition $\{S, \bar{S}\}$ as in the proof by Vincken:

$$\text{NCut}(S, \bar{S}) = \frac{\frac{W}{2} \cdot 2W}{(\frac{W}{2} + \sum_{v_i \in S} 2w(s_i)) \cdot (\frac{W}{2} + \sum_{v_i \in \bar{S}} 2w(t_i))}. \tag{78}$$

   Now we distinguish two cases. The first case is where we have the following equality:

$$\sum_{v_i \in S} 2w(s_i) = \sum_{v_i \in \bar{S}} 2w(t_i)) = \frac{W}{2}. \tag{79}$$

   Note that in the first case, the partition $\{S, \bar{S}\}$ can be mapped to a YES-instance of the PARTITION problem. The second case is where do not have Equality 79. Note that in the second case, the partition $\{S, \bar{S}\}$ cannot be mapped to a YES-instance of the PARTITION problem.

   For the first case we will show that even when we make the assumption that the edge-weights of the newly added edges increase the *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ as much as possible by being added to the *cut value* and by increasing the difference between the *volumes* as much as possible, we still have $\text{NCut}(S, \bar{S}) \leq N$.

The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ is:

$$\begin{aligned}\text{NCut}(S, \bar{S}) &\leq \frac{(\frac{W}{2} + \frac{1}{W^2}) \cdot (2W + \frac{2}{W^2})}{(\frac{W}{2} + \frac{W}{2} + \frac{2}{W^2}) \cdot (\frac{W}{2} + \frac{W}{2})} \\ &= \frac{(\frac{W}{2} + \frac{1}{W^2}) \cdot (2W + \frac{2}{W^2})}{(W + \frac{2}{W^2}) \cdot W} \\ &\leq N.\end{aligned} \tag{80}$$

Because of Equality 79 this case can be mapped to a YES-instance of the PARTITION problem.

Now we only need to show that when we do not have Equality 79, and thus we cannot map the partition $\{S, \bar{S}\}$ to a YES-instance of the PARTITION problem, that we have $\text{NCut}(S, \bar{S}) > N$. Therefore, the second case is where we do not have Equality 79 and have the following inequality:

$$\sum_{v_i \in S} 2w(s_i) \neq \sum_{v_i \in \bar{S}} 2w(t_i)) \neq \frac{W}{2}. \tag{81}$$

For the second case we will show that even when we make the assumption that the edge-weights of the newly added edges decrease the *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ as much as possible by not being in the *cut value* and by decreasing the difference between the *volumes* as much as possible, we still have $\text{NCut}(S, \bar{S}) > N$.

Because of Inequality 81, we do not have equal *volumes*. We assume that the difference between $\sum_{v_i \in S} 2w(s_i)$ and $\sum_{v_i \in \bar{S}} 2w(t_i))$ is minimal, since then the *volumes* differ the least and we have a lower *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$. We have $W = \sum_{i \in I}(w(s_i) + w(t_i)) + \sum_{i \in \bar{I}}(w(s_i) + w(t_i))$ and these terms are dependent on each-other. Therefore, to minimize the difference between the *volumes*, we assume that we have:

$$\sum_{i \in I}(w(s_i) + w(t_i)) = \frac{W}{2} - 1, \sum_{i \in \bar{I}}(w(s_i) + w(t_i)) = \frac{W}{2} + 1. \tag{82}$$

Since $\text{vol}(S, V)$ is now smaller than $\text{vol}(\bar{S}, V)$, we add the extra *volume* of the edge-weights of the newly added edges to $\text{vol}(S, V)$. The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ is:

$$\begin{aligned}\mathrm{NCut}(S,\bar{S}) &\geq \frac{\frac{W}{2}\cdot(2W+\frac{2}{W^2})}{(\frac{W}{2}+\frac{W}{2}-1+\frac{2}{W^2})\cdot(\frac{W}{2}+\frac{W}{2}+1)} \\ &= \frac{\frac{W}{2}\cdot(2W+\frac{2}{W^2})}{(W-1+\frac{2}{W^2})\cdot(W+1)}.\end{aligned} \tag{83}$$

We compare Fraction 83 to N and see if it is $\leq N$:

$$\frac{\frac{W}{2}\cdot(2W+\frac{2}{W^2})}{(W-1+\frac{2}{W^2})\cdot(W+1)} \leq N. \tag{84}$$

We let WolframAlpha [1] solve Inequality 84 and see that it holds for $W \leq 2.73205$. An instance of the Partition problem must have at least two elements of a value of at least 1. For both elements we would add two edges to the complete graph with value 1 and therefore the total weight W of G must be at least 4. Since we have that Inequality 84 only holds for $W < 1$, $\{S, \bar{S}\}$ does not have $\mathrm{NCut}(S,\bar{S}) \leq N$.

Therefore, if we have a partition with *normalized cut value* $\leq N$, then we have a YES-instance of the Partition problem.

We conclude that the Partition problem is a YES-instance if and only if we can find a partition with a *normalized cut value* $\leq N$. □

6 Strong NP-completeness of the Normalized Cut problem with unweighted edges

In this section, we show that the Normalized Cut problem with unweighted edges is strongly NP-complete.

Theorem 6.1. *The* Normalized Cut *problem is strongly NP-complete on general graphs with unweighted edges.*

Proof. First note that the Normalized Cut problem with unweighted edges is in NP. We use a reduction from the strongly NP-complete Sparsest Cut problem with unweighted edges [4] to prove that the Normalized Cut problem with unweighted edges is strongly NP-complete.

For a graph $G = (V, E)$ used in a Sparsest Cut problem instance, we construct a graph $G' = (V', E')$ which we will use for our Normalized Cut problem instance. We create a clique C_i of size n^3 with $n = |V|$ for every vertex $v_i \in V$. For every C_i we choose

two special vertices a_i and b_i. Let A be the set of all special vertices a_i and let B be the set of all special vertices b_i. Note that $|A| = |B| = |V|$. For every edge $(v_i, v_j) \in E$ we create an edge (a_i, a_j) and an edge (b_i, b_j), thereby creating two duplicates of G mapped on vertex sets A and B. Let $d(v_i)$ be the degree of a vertex $v_i \in V$. We choose $p = n - d(v_i) - 1$ distinct non-special vertices of the clique C_i representing v_i in G' and add a pendant vertex to all p vertices. Let $W = n/2(n^3(n^3 - 1) + 2n - 2)$ be the total edge-weight of all edges in G'. In Figure 7 we see an example graph G and in Figure 8 we see the graph G' constructed from G.

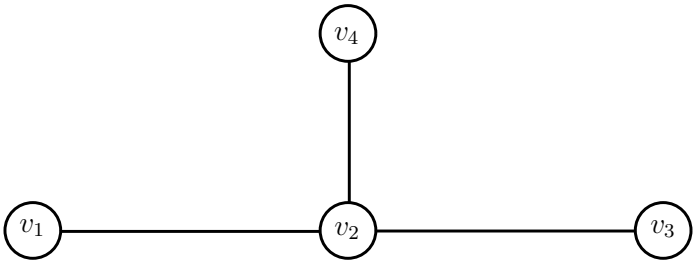


Figure 7: A graph G

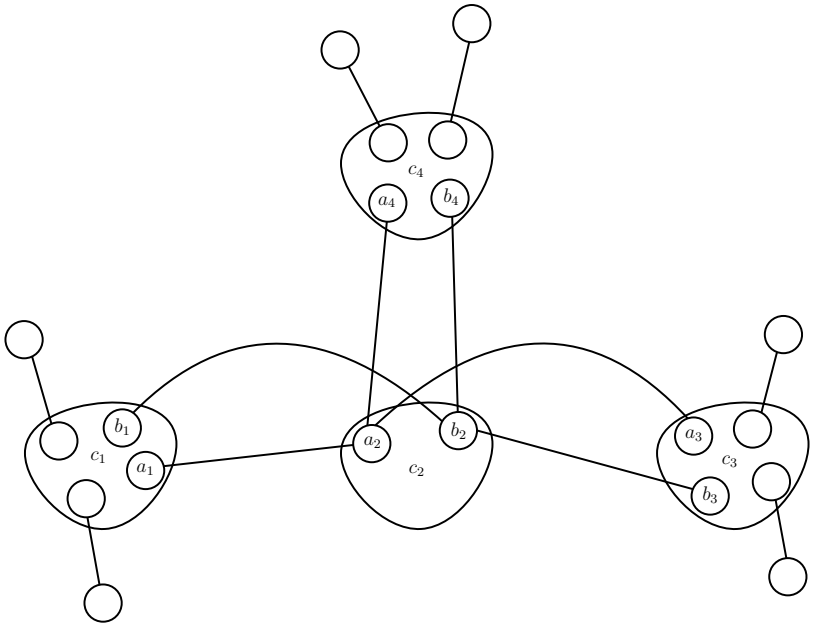


Figure 8: Graph G' constructed from G

Let W be the total number of edges in E':

$$W = |E'| = n \cdot \binom{n^3}{2} + \sum_{v \in V} n \cdot d(v) - 1 + \sum_{v \in V} d(v) = n \cdot \binom{n^3}{2} + n(n-1). \tag{85}$$

We define N' as:

$$N' = 2N \cdot \frac{2W}{(n^3(n^3-1)+2n-2)^2}. \tag{86}$$

We assume $0 \leq N \leq 1$, since a partition cannot have a *density cut value* < 0 and every possible partition of any graph has a *density cut value* ≤ 1. We show that there exists a partition of a graph $G' = (V', E')$ with *normalized cut value* $\leq N'$ if and only if there exists a partition of a graph $G = (V, E)$ with *density cut value* $\leq N$, where G' is the graph constructed from $G = (V, E)$.

Assume there exists a partition $\{S, \bar{S}\}$ of G with *density cut value* $\text{DensCut}(S, \bar{S}) \leq N$. We need to show that there exists a partition $\{S', \bar{S}'\}$ of G' with *normalized cut value* $\text{NCut}(S', \bar{S}') \leq N'$. For every vertex $v_i \in S$ we put all vertices in the clique C_i and the pendant vertices connected to C_i in S'. In the same way, for every vertex $v_i \in \bar{S}$ we put all vertices in the clique C_i and the pendant vertices connected to C_i in $\bar{S}'$. Now we have our partition $\{S', \bar{S}'\}$ of G' and analyse what the *normalized cut value* $\text{NCut}(S', \bar{S}')$ of $\{S', \bar{S}'\}$ is.

First, we look at what happens to the *volume* $\text{vol}(S', V')$ of the *component* S' when it comprises the vertices of a clique C_i and all the pendant vertices connected to C_i. All edges of C_i and the edges connecting the pendant vertices are fully inside S' and so we count them twice. We count the edges connecting the special vertices a_i and b_i to other cliques once since we also need to count these edges once for the other cliques they are connected to. The number of edges connecting the vertices in C_i is $n^3(n^3-1)/2$, so we add $2 \cdot n^3(n^3-1)/2 = n^3(n^3-1)$ to $\text{vol}(S', V')$. The way we chose the number of pendant vertices in the graph construction makes the amount we add to the $\text{vol}(S', V')$ for the edges connecting the pendant vertices $2(n - d(v_i) - 1)$. The number of edges connecting a_i and b_i to the other cliques is $2d(v_i)$. Therefore, if a clique C_i and all the pendant vertices connected to C_i is fully inside S' it adds a total of $n^3(n^3-1) + 2(n - d(v_i) - 1) + 2d(v_i) = n^3(n^3-1) + 2n - 2$ to $\text{vol}(S', V')$. Therefore, since there are $|S|$ vertices in S, we have $\text{vol}(S', V') = |S| \cdot (n^3(n^3-1)/2 + 2n - 2)$. Similarly, we have that $\text{vol}(\bar{S}', V') = |\bar{S}| \cdot (n^3(n^3-1) + 2n - 2)$.

Now we will look at the *cut value* $\text{cut}(S', \bar{S}')$ of $\{S', \bar{S}'\}$. For every edge (v_i, v_j) in the *cut* of partition $\{S, \bar{S}\}$ of G we now have (a_i, a_j) and (b_i, b_j) in the *cut* of partition $\{S', \bar{S}'\}$. Because of the way we constructed G' from G, we have $w((a_i, a_j)) = w((b_i, b_j)) = w((v_i, v_j))$. Therefore, the *cut value* $\text{cut}(S', \bar{S}')$ of $\{S', \bar{S}'\}$ is $2 \cdot \text{cut}(\text{S}, \bar{\text{S}})$.

Now we can compute the *normalized cut value* $\text{NCut}(S', \bar{S}')$ of $\{S', \bar{S}'\}$ as:

$$\begin{aligned}\mathrm{NCut}(S', \bar{S}') &= \frac{2\,\mathrm{cut}(\mathrm{S}, \bar{\mathrm{S}}) \cdot 2W}{(|S| \cdot (n^3(n^3-1)+2n-2)) \cdot (|\bar{S}| \cdot (n^3(n^3-1)+2n-2))} \\ &= \frac{2\,\mathrm{cut}(S, \bar{S})}{|S| \cdot |\bar{S}|} \cdot \frac{2W}{(n^3(n^3-1)+2n-2)^2} \\ &= 2\,\mathrm{DensCut}(S, \bar{S}) \cdot \frac{2W}{(n^3(n^3-1)+2n-2)^2}.\end{aligned} \tag{87}$$

Since we have $\mathrm{DensCut}(S, \bar{S}) \leq N$ we have:

$$\begin{aligned}\mathrm{NCut}(S', \bar{S}') &= 2\,\mathrm{DensCut}(S, \bar{S}) \cdot \frac{2W}{(n^3(n^3-1)+2n-2)^2} \\ &\leq 2N \cdot \frac{2W}{(n^3(n^3-1)+2n-2)^2} \\ &= N'.\end{aligned} \tag{88}$$

Therefore, we have a *normalized cut value* $\mathrm{NCut}(S', \bar{S}') \leq N'$.

Conversely, assume there exists a partition $\{S', \bar{S}'\}$ of a graph $G' = (V', E')$ with *normalized cut value* $\mathrm{NCut}(S', \bar{S}') \leq N'$. We need to show that we have a partition $\{S, \bar{S}\}$ of $G = (V, E)$ with *density cut value* $\mathrm{DensCut}(S, \bar{S}) \leq N$. We will find partition $\{S^*, \bar{S}^*\}$ with minimum *normalized cut value* for which we have $\mathrm{NCut}(S^*, \bar{S}^*) \leq \mathrm{NCut}(S', \bar{S}') \leq N'$. According to Theorem 4.6 a partition with minimum *normalized cut value* has exactly two connected *components*. There are three possible ways to partition G' that result in a partition that has exactly two connected *components*. The first way we can partition G' is by putting one pendant vertex in *component* S^* and all other vertices in *component* $\bar{S}^*$. The second way we can partition G' is by having a partition where for at least one clique the vertices are divided over S^* and $\bar{S}^*$. The third and last way we can partition G' is by having a partition where only edges between vertices in A and edges between vertices in B are in the *cut*. These three cases cover all partitions that result in exactly two connected *components*. An example of the three ways we can partition the graph construction that result in a partition with exactly two connected *components* can be found in Figure 9.

We now show that actually only the last case, where the partition only has edges between vertices in A and edges between vertices in B in the *cut*, can result in a partition with *normalized cut value* $\leq N'$.

1. The first case is where the partition consists of one *component* containing only one pendant vertex and the other *component* contains the rest of the vertices. The *normalized cut value* $\mathrm{NCut}(S^*, \bar{S}^*)$ of such a partition $\{S^*, \bar{S}^*\}$ is:

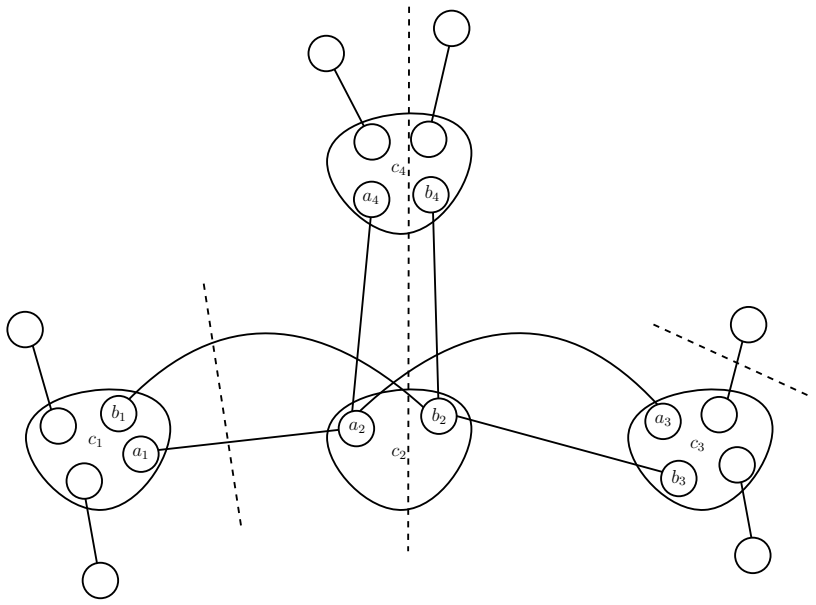


Figure 9: Graph construction G' with three partitions

$$\mathrm{NCut}(S^*, \bar{S}^*) = \frac{1 \cdot 2W}{1 \cdot (2W - 1)}. \tag{89}$$

Recall that $N \leq 1$ and $W = n/2(n^3(n^3-1)+2n-2)$. If we look at how N' is defined we see that N' gets larger when N is larger. If we can show that $\mathrm{NCut}(S^*, \bar{S}^*) > N'$ when $N = 1$, we know that $\{S^*, \bar{S}^*\}$ can never have a *normalized cut value* $\leq N'$. Therefore, we compare N' and $\mathrm{NCut}(S^*, \bar{S}^*)$ with $N = 1$ and have:

$$\begin{aligned} N' &\leq \frac{4W}{(n^3(n^3-1)+2n-2)^2} \\ &< \frac{2W}{n(n^3(n^3-1)+2n-2)-1} \\ &= \frac{1 \cdot 2W}{1 \cdot (2W-1)} \\ &= \mathrm{NCut}(S^*, \bar{S}^*). \end{aligned} \tag{90}$$

So we can conclude that a partition where we put one pendant vertex in one *component* and the other vertices in the other *components* never results in a partition with *normalized cut value* $\leq N'$.

2. The second case is where the partition divides the vertices of at least one clique over both *components*. Therefore, at least $n^3 - 1$ edges are added to the *cut* and the *cut value* $\text{cut}(S^*, \bar{S}^*)$ of such a partition $\{S^*, \bar{S}^*\}$ is at least $n^3 - 1$. According to Proposition 4.1 the lowest *normalized cut value* for a given *cut value* is when the *volumes* are equal. If we assume the *volumes* can be equal, the lowest possible *normalized cut value* $\text{NCut}(S^*, \bar{S}^*)$ of $\{S^*, \bar{S}^*\}$ is:

$$\begin{aligned} \text{NCut}(S^*, \bar{S}^*) &\geq \frac{(n^3 - 1) \cdot 2W}{W \cdot W} \\ &= \frac{(n^3 - 1) \cdot (n(n^3(n^3 - 1) + 2n - 2))}{(n(n^3(n^3 - 1) + 2n - 2))^2/4} \\ &= \frac{n^3 - 1}{(n(n^3(n^3 - 1) + 2n - 2))/4}. \end{aligned} \tag{91}$$

As we already saw, if we can show that $\text{NCut}(S^*, \bar{S}^*) > N'$ when $N = 1$, we know that $\{S^*, \bar{S}^*\}$ can never have a *normalized cut value* $\leq N'$. Therefore, we compare N' and $\text{NCut}(S^*, \bar{S}^*)$ with $N = 1$ and get:

$$\begin{aligned} N' &\leq \frac{4W}{(n^3(n^3 - 1) + 2n - 2)^2} \\ &= \frac{2n(n^3(n^3 - 1) + 2n - 2)}{(n^3(n^3 - 1) + 2n - 2)^2} \\ &= \frac{2n}{n^3(n^3 - 1) + 2n - 2} \\ &< \frac{n^3 - 1}{(n(n^3(n^3 - 1) + 2n - 2))/4} \leq \text{NCut}(S^*, \bar{S}^*). \end{aligned} \tag{92}$$

So we can conclude that a partition that divides the vertices of at least one clique over both *components* never results in a partition with *normalized cut value* $\leq N'$.

3. The third case is where the partition only has edges between vertices in A and edges between vertices in B in the *cut*. In such a partition $\{S^*, \bar{S}^*\}$ all vertices of a specific clique C_i and the pendant vertices connected to C_i are in one of the two *components*. Since there are n cliques, assume there are $1 \leq x \leq n$ cliques in *component* S^* and $n - x$ cliques in *component* $\bar{S}^*$. The *normalized cut value* $\text{NCut}(S^*, \bar{S}^*)$ of $\{S^*, \bar{S}^*\}$ is:

$$\begin{aligned}\mathrm{NCut}(S^*, \bar{S}^*) &= \frac{\mathrm{cut}(S^*, \bar{S}^*) \cdot 2W}{(x(n^3(n^3-1)+2n-2)) \cdot ((n-x) \cdot (n^3(n^3-1)+2n-2))} \\ &= \frac{\mathrm{cut}(S^*, \bar{S}^*)}{x(n-x)} \cdot \frac{2W}{((n^3(n^3-1)+2n-2))^2}.\end{aligned} \tag{93}$$

We map $\{S^*, \bar{S}^*\}$ on $G = (V, E)$, thereby creating a partition $\{S, \bar{S}\}$, by putting a vertex $v_i \in V$ in *component* S if all vertices of the clique C_i and the pendant vertices connected to C_i are in S^* and by putting a vertex $v_i \in V$ in *component* $\bar{S}$ if all vertices of the clique C_i and the pendant vertices connected to C_i are in $\bar{S}^*$. We have $\mathrm{cut}(S^*, \bar{S}^*) = 2\,\mathrm{cut}(S, \bar{S})$, $x = |S|$ and $(n - x) = |\bar{S}|$. Therefore, we have:

$$\begin{aligned}\mathrm{NCut}(S^*, \bar{S}^*) &= \frac{\mathrm{cut}(S^*, \bar{S}^*)}{x(n-x)} \cdot \frac{2W}{((n^3(n^3-1)+2n-2))^2} \\ &= 2\frac{\mathrm{cut}(S, \bar{S})}{|S||\bar{S}|} \cdot \frac{2W}{((n^3(n^3-1)+2n-2))^2} \\ &\leq N' \\ &= 2N \cdot \frac{2W}{((n^3(n^3-1)+2n-2))^2}.\end{aligned} \tag{94}$$

From Equation 94 we get the inequality:

$$2\frac{\mathrm{cut}(S, \bar{S})}{|S||\bar{S}|} \leq 2N. \tag{95}$$

Partition $\{S, \bar{S}\}$ has a *density cut value* $\mathrm{DensCut}(S, \bar{S})$ of:

$$\mathrm{DensCut}(S, \bar{S}) = \frac{\mathrm{cut}(S, \bar{S})}{|S||\bar{S}|} \leq N. \tag{96}$$

So we can conclude that there exists a partition of G with *density cut value* $\leq N$ if there exists a partition of G' with *normalized cut value* $\leq N'$.

We showed that there exists a partition of a graph G' with *normalized cut value* $\leq N'$ if and only if there exists a partition of a graph G with *density cut value* $\leq N$, concluding our proof. □

7 Polynomial-time algorithms for the Normalized Cut problem on $\mathcal{H}$-free graph classes

In this section, we present two polynomial-time algorithms for the Normalized Cut problem. First, in Subsection 7.1, we present a linear time algorithm for the Normalized Cut problem on cycle-free graphs i.e. forests. Then, in Subsection 7.2, we present a quadratic time algorithm for the Normalized Cut problem on outerplanar graphs.

7.1 Linear time algorithm for forests

In this subsection, we present a linear time algorithm for the Normalized Cut problem on forests.

Theorem 7.1. *The* Normalized Cut *problem is solvable in linear time on forests.*

Proof. We have an instance of the Normalized Cut problem of a forest $G = (V, E)$ and a real number N. First, we use the *Breadth-first search* algorithm to find out if G is connected. If G is not connected and consists of multiple disjoint trees, then we find a partition $\{S, \bar{S}\}$ with a minimum *normalized cut value* where we put all vertices of one tree in *component* S and the rest of the vertices in the other *component* $\bar{S}$. Since there are no edges between the *components*, the *cut value* $\text{cut}(S, \bar{S})$ is 0 which results in a *normalized cut value* $\text{NCut}(S, \bar{S})$ of 0. Recall that edge-weights are positive and therefore the *normalized cut value* of a partition can never be smaller than 0. Therefore, $\{S, \bar{S}\}$ has a minimum *normalized cut value* and we can check in constant time if $0 \leq N$. Since for forests $|E| \in O(n)$ the *Breadth-first search* algorithm takes linear time.

If G is connected, it is a single tree. Let W be the sum of the edge-weights of the edges in E. According to Theorem 4.6 the partition with minimum *normalized cut value* has exactly two connected *components*. Since G is a tree, there can only be one edge in the *cut* of a partition with minimum *normalized cut value*. Consider for an edge $e \in E$ with endpoints v and u the two subtrees $G_v = (V_v, E_v)$ and $G_u = (V_u, E_u)$ we get by removing e from G. If we know the total edge-weights $w(E_v)$ and $w(E_u)$, we can compute the *normalized cut value* $\text{NCut}(V_v, V_u)$ of partition $\{V_v, V_u\}$ in constant time as:

$$\text{NCut}(V_v, V_u) = \frac{w(e) \cdot 2W}{(2w(E_v) + w(e)) \cdot (2w(E_u) + w(e))}. \tag{97}$$

Therefore, we can compute for every edge $e \in E$ the *normalized cut value* of the partition with only e in the *cut* and determine if there is a partition that has a *normalized cut value* $\leq N$ in linear time. It rests us to show that we can compute $w(E_v)$ and $w(E_u)$ for every edge $e \in E$ in linear time.

We repeat a procedure where we consider a leaf vertex v and remove v from the tree at the end of the procedure, creating a new tree. We repeat the procedure until the tree

consists of only one vertex. Consider a leaf vertex v. Let C be the possibly empty set of neighbors of v in the original graph G that are no longer in the current graph. Now $w(E_v) = \sum_{c \in C} (w(E_c) + w((v,c)))$ where $w((v,c))$ is the weight of edge (v,c). Note that $w(E_c)$ is already computed since otherwise v is not a leaf. Furthermore, $w(E_u) = W - w(e) - w(E_v)$. Now we subtract e with its endpoint v from the graph. When there is only one vertex left, we have computed the total edge-weights $w(E_v)$ and $w(E_u)$ for every edge $e \in E$ in linear time. □

7.2 Quadratic time algorithm for outerplanar graphs

In this subsection, we present a quadratic time algorithm for the Normalized Cut problem on outerplanar graphs. The algorithm is based on the algorithm for the Sparsest Cut problem on outerplanar graphs presented in the paper by Bonsma et al. [4].

Theorem 7.2. *The* Normalized Cut *problem is solvable in quadratic time on outerplanar graphs.*

Proof. An outerplanar graph consists of biconnected blocks, from now on called blocks, which are either single edges or cycles with chords. In Figure 10 we see an example of an outerplanar graph and its blocks. We can find the blocks of an outerplanar graph in linear time using the algorithm presented in the paper by Tarjan [23]. We denote with $V(B)$ the set of vertices of a block B and with $E(B)$ the set of edges of a block B. Vertices that are part of multiple blocks are called cut vertices. According to Theorem 4.6 a partition with minimum *normalized cut value* has exactly two connected *components*. Therefore, there is a partition with minimum *normalized cut value* that only has edges of one block in the *cut*.

We have an instance of the Normalized Cut problem of an outerplanar graph $G = (V, E)$ and a real number N. We present an algorithm that can find the partition with minimum *normalized cut value* of G in quadratic time. This algorithm requires two preprocessing steps:

1. The first preprocessing step is creating an adjacency matrix for every block B so that we can find in constant time if an edge $e = (v, u)$ with $v, u \in V(B)$ exists and what the weight $w(e)$ is. If the graph is one block, there are $n(n-1)/2$ pairs of vertices we need to consider. Otherwise, if the graph consists of multiple blocks, we do not have to consider the pairs of vertices that are not in the same block. Furthermore, note that we cannot have $v_i, v_j \in V(B_a)$ and $v_i, v_j \in V(B_b)$ for two cut vertices v_i and v_j and two blocks B_a and B_b, and a vertex that is not a cut vertex is part of only one block. Therefore, we never check a pair of vertices $v_i, v_j \in V$ more than once. Checking at most $n(n-1)/2$ pairs of vertices takes $O(n^2)$ time. Note that a diagonal entry of an adjacency matrix for a block B corresponds with the existence of an edge (v, v) for a vertex $v \in V(B)$. Since a vertex can be part of at most n blocks, we have at most n entries (v, v) for a vertex $v \in V$. Since there are n vertices we have at

most n^2 entries of this type, which takes $O(n^2)$ time to create. Therefore, the first preprocessing step takes $O(n^2)$ time.

2. The second preprocessing step computes for every block B and vertex $v \in V(B)$ the total edge-weight $f(v, B)$ of the connected subgraph that contains v in the graph $G' = G \setminus E(B)$. We first set $f(v, B) = 0$ for every vertex v in a block B that is not a cut vertex. Furthermore, to compute $f(v, B)$ for every cut vertex v we first initialize $f'(v) = 0$. Now we repeat a procedure in which we consider a block B with one cut vertex v and in which at the end of the procedure we contract B into v, creating a new outerplanar graph in which v might not be a cut vertex anymore. Note that there always exists a block B with just one cut vertex v. Let A be the possibly empty set of vertices that were originally cut vertices in B and are now not anymore. For every $u \in A$ we set $f(u, B) = f'(u)$. Let W be the sum of edge-weights of all edges E of the original graph $G = (V, E)$. We compute $f(v, B)$ as $W - w(E(B)) - \sum_{u \in A} f(u, B)$. Furthermore, we set $f'(v) = f'(v) + w(E(B)) + \sum_{u \in A} f(u, B)$. Lastly, we contract B into v. We repeat this procedure until there is just one block B left, for which we can compute $f(u, B)$ for every vertex u that was originally a cut vertex as $f(u, B) = f'(u)$. The procedure takes $O(l)$ time for a block with l edges, since $|E| \in O(n)$ the second preprocessing step takes $O(n)$ time. In Figure 10 we see an outerplanar graph and its blocks. As an example, for the highlighted block B, we have $f(v_1, B) = 12$.

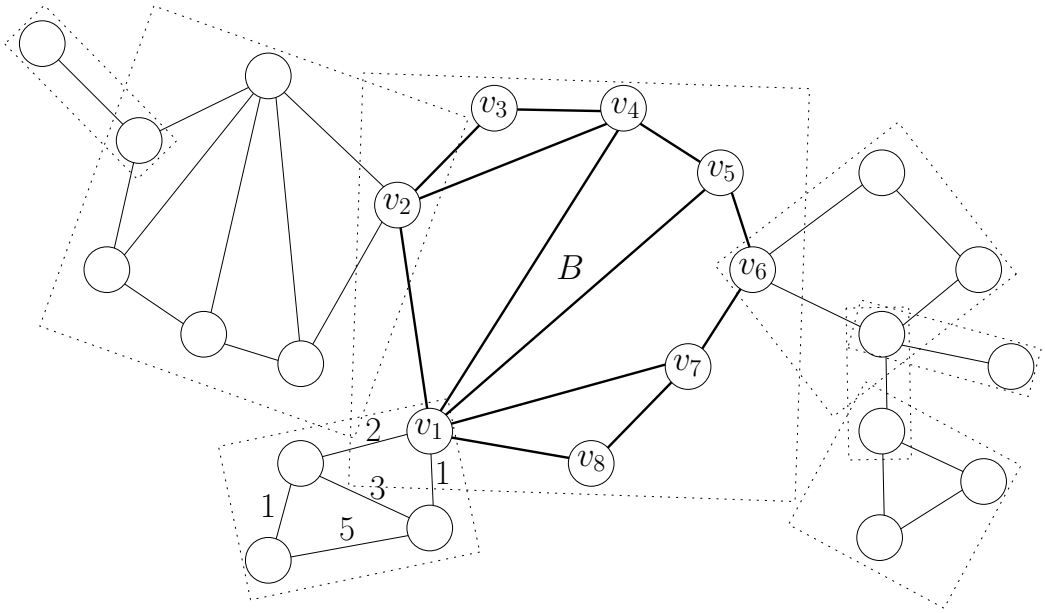


Figure 10: An outerplanar graph and its blocks

To find the partition with minimum *normalized cut value* we consider one block at a time

and compute the partition with minimum *normalized cut value* that only has edges of the current block in the *cut* and which has exactly two connected *components*. We then compare the *normalized cut value* of this partition to the partition with minimum *normalized cut value* among all partitions of the already considered blocks. If the minimum *normalized cut value* of the partition of the current block is lower, we set this partition as the new partition with minimum *normalized cut value* among all considered blocks. Once we have considered all blocks we check if the minimum *normalized cut value* among all blocks is $\leq N$ in constant time. Therefore, we only have to show that we can find the partition with minimum *normalized cut value* of a block with l edges in $O(l^2)$ time, thereby showing that we can find the partition with minimum *normalized cut value* among all blocks in $O(n^2)$ time since $|E| \in O(n)$.

We have two different types of blocks, a single edge or a cycle with chords. For a block B that is a single edge $e = (v, u)$ we compute the *normalized cut value* $\text{NCut}(S, \bar{S})$ of the partition $\{S, \bar{S}\}$ defined by its *cut* which only contains e in constant time as:

$$\text{NCut}(S, \bar{S}) = \frac{w(e) \cdot 2W}{(2f(v, B) + w(e)) \cdot (2f(u, B) + w(e))}. \tag{98}$$

For a block B that is a cycle with chords, we call the $k = |V(B)|$ vertices along the cycle $v_1, \ldots, v_k$. In Figure 10 we see an example of a block B and its vertices $v_1, \ldots, v_8$ along the cycle. Let $C_{x,y}$ be the set of vertices $\{v_x, \ldots, v_y\}$ for some $1 \leq x \leq y \leq k$. For two sets of vertices Q and R, let $w(Q, R)$ be the sum of the edge-weights of the edges between the sets of vertices Q and R. Now let $p_{i,j} = w(\{v_i\}, C_{1,j})$ where $w(\{v_i\}, C_{1,j})$ is the sum of the edge-weights of the edges between v_i and the set of vertices $C_{1,j}$. We have $p_{i,j} = p_{i,j-1}$ if edge (v_i, v_j) does not exist and $p_{i,j} = p_{i,j-1} + w((v_i, v_j))$ otherwise. We can retrieve the existence and weight of an edge in constant time from the adjacency matrix we created in the first preprocessing step. Furthermore, $p_{i,1} = 0$ if edge (v_i, v_1) does not exist and $p_{i,1} = w((v_i, v_1))$ otherwise.

If we would compute $p_{i,j}$ for every pair of vertices $v_i, v_j \in V$, it would take $O(n^2)$ time. We only compute $p_{i,j}$ for a subset of the pairs of vertices $v_i, v_j \in V$ and therefore computing all $p_{i,j}$ for every block also takes $O(n^2)$ time.

Let partition $\{S_{x,y}, \bar{S}_{x,y}\}$ be the partition that only contains edges between $C_{x,y}$ and $V(B) \setminus C_{x,y}$ in their *cut*. Now let A be all partitions $\{S_{x,y}, \bar{S}_{x,y}\}$ for any $1 \leq x \leq y \leq k$. Note that these partitions are all partitions that have exactly two connected *components* and only have edges of $E(B)$ in their *cut*. According to Theorem 4.6 a partition with minimum *normalized cut value* has exactly two connected *components*. Therefore, if there is a partition of G with minimum *normalized cut value* that only contains edges of $E(B)$ in its cut, it must be that this partition is in A.

Now we analyze how much time it takes to compute the *normalized cut values* of all partitions in A. We can think of the partitions in A as all partitions that have a unique pair of edges of the cycle of edges around B in their cut. If we can compute the *normalized cut value* of a partition in A in constant time, we can compute the *normalized cut values* of all partitions in A in $O(l^2)$ time where $l = |E(B)|$. Therefore, we can compute the *normalized cut values* of the set of partition A for every block B in $O(n^2)$ since $|E| \in o(n)$.

We first consider the partitions $\{S_{x,x}, \bar{S}_{x,x}\}$ defined by their *cuts* which only contains edges between $C_{x,x}$ and $V(B) \setminus C_{x,x}$ where $1 \leq x \leq k$. Note that from the vertices $V(B)$ of a block B, a *component* $S_{x,x}$ only contains vertex $x \in V(B)$ and possibly vertices from $V \setminus V(B)$. We can compute the *normalized cut value* $\mathrm{NCut}(S_{x,x}, \bar{S}_{x,x})$ of partition $\{S_{x,x}, \bar{S}_{x,x}\}$ as:

$$\mathrm{NCut}(S_{x,x}, \bar{S}_{x,x}) = \frac{p_{x,k} \cdot 2W}{(2f(v_x, B) + p_{x,k}) \cdot (2W - (2f(v_x, B) + p_{x,k}))}, \tag{99}$$

where $\mathrm{cut}(S_{x,x}, \bar{S}_{x,x}) = p_{x,k}$, $\mathrm{vol}(S_{x,x}, V) = 2f(v_x, B) + p_{x,k}$ and $\mathrm{vol}(\bar{S}_{x,x}, V) = 2W - (2f(v_x, B) + p_{x,k})$ for a partition $\{S_{x,x}, \bar{S}_{x,x}\}$. Note that we had computed all $p_{i,j}$ with $1 \leq i \leq k$ and $1 \leq j \leq k$ so we can retrieve $p_{i,j}$ in constant time. Furthermore, note that in the second preprocessing step we had already computed $f(v, B)$ for every vertex $v \in V$ and for every block B for which $v \in V(B)$, so we can retrieve $f(v, B)$ in constant time. Therefore, we can compute the *normalized cut value* $\mathrm{NCut}(S_{x,x}, \bar{S}_{x,x})$ of a partition $\{S_{x,x}, \bar{S}_{x,x}\}$ in constant time for $1 \leq x \leq k$.

After we have computed the *normalized cut values* of all partitions $\{S_{x,x}, \bar{S}_{x,x}\}$, we consider the partitions $\{S_{x,y}, \bar{S}_{x,y}\}$ for $x < y \leq k$ one by one in increasing order for y. Recall that a partition $\{S_{x,y}, \bar{S}_{x,y}\}$ only contains edges between $C_{x,y}$ and $V(B) \setminus C_{x,y}$ in its *cut*. Below in Equation 100 we compute the *cut value* $\mathrm{cut}(S_{x,y}, \bar{S}_{x,y})$ of partition $\{S_{x,y}, \bar{S}_{x,y}\}$ in constant time as follows. We first rewrite the sum of edge-weights $w(\{v_i\}, C_{a,b})$ of the edges between a vertex v_i and a vertex set $C_{a,b}$ to the form $w(\{v_i\}, C_{1,b}) - w(\{v_i\}, C_{1,a-1})$. Then for the sum of edge-weights $w(\{v_i\}, C_{1,d})$ we can substitute the corresponding $p_{i,d}$ value which we already computed beforehand in constant time. We compute the *cut value* $\mathrm{cut}(S_{x,y}, \bar{S}_{x,y})$ of $\{S_{x,y}, \bar{S}_{x,y}\}$ as:

$$\begin{aligned}
\mathrm{cut}(S_{x,y}, \bar{S}_{x,y}) &= \mathrm{cut}(S_{x,y-1}, \bar{S}_{x,y-1}) - w(\{v_y\}, C_{x,y-1}) + w(\{v_y\}, C_{1,x-1}) + w(\{v_y\}, C_{y+1,k}) \\
&= \mathrm{cut}(S_{x,y-1}, \bar{S}_{x,y-1}) - (w(\{v_y\}, C_{1,y-1}) - w(\{v_y\}, C_{1,x-1})) + w(\{v_y\}, C_{1,x-1}) \\
&+ (w(\{v_y\}, C_{1,k}) - w(\{v_y\}, C_{1,y})) \\
&= \mathrm{cut}(S_{x,y-1}, \bar{S}_{x,y-1}) - p_{y,y} + 2p_{y,x-1} + p_{y,k} - p_{y,y-1}.
\end{aligned} \tag{100}$$

Therefore, if we consider the partitions $\{S_{x,y}, \bar{S}_{x,y}\}$ with $x < y \leq k$ one by one in increasing order for y, we can compute the *cut value* $\text{cut}(S_{x,y}, \bar{S}_{x,y})$ of a partition $\{S_{x,y}, \bar{S}_{x,y}\}$ in constant time.

Now we consider the *volume* $\text{vol}(S_{x,y}, V)$ of a partition $\{S_{x,y}, \bar{S}_{x,y}\}$. Below in Equation 101 we compute $\text{vol}(S_{x,y}, V)$ of partition $\{S_{x,y}, \bar{S}_{x,y}\}$ in constant time as follows. We once again first rewrite the sum of edge-weights $w(\{v_i\}, C_{a,b})$ of the edges between a vertex v_i and a vertex set $C_{a,b}$ to the form $w(\{v_i\}, C_{1,b}) - w(\{v_i\}, C_{1,a-1})$. Then for the sum of edge-weights $w(\{v_i\}, C_{1,d})$ we can substitute the corresponding $p_{i,d}$ value which we already computed beforehand in constant time. Note that in Equation 101 we use the *cut values* $\text{cut}(S_{x,y}, \bar{S}_{x,y})$ and $\text{cut}(S_{x,y-1}, \bar{S}_{x,y-1})$, which we already computed in the previous step. Furthermore, note that in Equation 101 we use $f(v, B)$, which we computed in the second preprocessing step. We compute the *volume* $\text{vol}(S_{x,y}, V)$ of $S_{x,y}$ as:

$$\begin{aligned}
\text{vol}(S_{x,y}, V) &= \text{vol}(S_{x,y-1}, V) + 2w(\{v_y\}, C_{x,y-1}) - \text{cut}(S_{x,y-1}, \bar{S}_{x,y-1}) + \text{cut}(S_{x,y}, \bar{S}_{x,y}) + 2f(v_y, B) \\
&= \text{vol}(S_{x,y-1}, V) + 2(w(\{v_y\}, C_{1,y-1}) - w(\{v_y\}, C_{1,x-1})) - \text{cut}(S_{x,y-1}, \bar{S}_{x,y-1}) \\
&+ \text{cut}(S_{x,y}, \bar{S}_{x,y}) + 2f(v_y, B) \\
&= \text{vol}(S_{x,y-1}, V) + 2p_{y,y-1} - 2p_{y,x-1} - \text{cut}(S_{x,y-1}, \bar{S}_{x,y-1}) + \text{cut}(S_{x,y}, \bar{S}_{x,y}) + 2f(v_y, B).
\end{aligned} \tag{101}$$

Therefore, if we consider the partitions $\{S_{x,y}, \bar{S}_{x,y}\}$ with $x < y \leq k$ one by one in increasing order for y, we can compute the *volume* $\text{vol}(S_{x,y}, V)$ of a certain partition $\{S_{x,y}, \bar{S}_{x,y}\}$ in constant time.

Lastly, we can compute the *volume* $\text{vol}(\bar{S}_{x,y})$ of $\bar{S}_{x,y}$ in constant time as:

$$\text{vol}(\bar{S}_{x,y}, V) = 2W - \text{vol}(S_{x,y}, V). \tag{102}$$

We can compute the *normalized cut value* $\text{NCut}(S_{x,y}, \bar{S}_{x,y})$ of $\{S_{x,y}, \bar{S}_{x,y}\}$ in constant time as:

$$\text{NCut}(S_{x,y}, \bar{S}_{x,y}) = \frac{\text{cut}(S_{x,y}, \bar{S}_{x,y}) \cdot 2W}{\text{vol}(S_{x,y}, V) \cdot \text{vol}(\bar{S}_{x,y}, V)}. \tag{103}$$

Therefore, we can compute the *normalized cut value* of all partitions in A in $O(l^2)$ time with $l = |E(B)|$. We then check which partition in A has a minimum *normalized cut value*. Once we have achieved this for every block B of G we know which partition has a minimum *normalized cut value*. Since we have $|E| \in O(n)$ for an outerplanar graph, finding the partition with minimum *normalized cut value* takes $O(n^2)$ time. We then check if the partition with minimum *normalized cut value* is $\leq N$. We conclude that the NORMALIZED CUT problem is solvable in quadratic time for outerplanar graphs. □

8 Polynomial-time algorithms for the Normalized Cut problem with unweighted edges on $\mathcal{H}$-free graph classes

In this section, we present two polynomial-time algorithms for the Normalized Cut problem with unweighted edges. First, in Subsection 8.1, we present a linear time algorithm for the Normalized Cut problem with unweighted edges on cluster graphs. Then, in Subsection 8.2, we present a linear time algorithm for the Normalized Cut problem with unweighted edges on cactus graphs. Note that all algorithms in Section 7 also work on graphs with unweighted edges using the same amount of time.

8.1 Linear time algorithm for cluster graphs

In this subsection, we present a linear time algorithm for the Normalized Cut problem with unweighted edges on cluster graphs. As we saw in the description of cluster graphs in Section 5 they are defined by the disjoint union of cliques. We are going to prove that the Normalized Cut problem is solvable in linear time on cluster graphs with unweighted edges.

Theorem 8.1. *The* Normalized Cut *problem is solvable in linear time on cluster graphs with unweighted edges.*

Let $n = |V|$. Before we can prove Theorem 8.1 we need the following lemma:

Lemma 8.2. *All partitions of a complete graph with unweighted edges have a normalized cut value of* $n/(n-1)$.

Proof. In a complete graph G with n vertices, every vertex has degree $n-1$. Assume we have a partition $\{S, \bar{S}\}$ of G where $|S| = x$ and $|\bar{S}| = n - x$. We have $\text{cut}(S, \bar{S}) = x(n-x)$, $\text{vol}(S, V) = x(n-1)$ and $\text{vol}(\bar{S}, V) = (n-x) \cdot (n-1)$. The *normalized cut value* $\text{NCut}(S, \bar{S})$ of $\{S, \bar{S}\}$ is:

$$\text{NCut}(S, \bar{S}) = \frac{x(n-x) \cdot n(n-1)}{x(n-1) \cdot (n-x) \cdot (n-1)} = \frac{n}{n-1}. \tag{104}$$

Note that we need $x \neq 0$, $n \neq 0$, $x \neq n$ and $n \neq 1$. However, for a partition of a complete graph with at least two vertices this is always the case. Therefore, all partitions of a complete graph with unweighted edges have a *normalized cut value* of $n/(n-1)$. □

Now we will prove Theorem 8.1 using Lemma 8.2.

Proof. We have an instance of the Normalized Cut problem of a cluster graph with unweighted edges $G = (V, E)$ and a real number N. First, we want to find out if G is connected using the following algorithm. We start by choosing a random vertex v. We perform a breadth-first search from v. Let A be all vertices u for which the edge (v, u) exists and v itself i.e. $A = N(v) \cup v$. If $|A| = |V|$ we have a connected graph which is

a complete graph. Otherwise, if $|A| < |V|$, the graph is not connected and G consists of multiple disjoint cliques. First, we look at the case where G consists of multiple disjoint cliques. We find a partition $\{S, \bar{S}\}$ with minimum *normalized cut value* if we put all vertices of A in *component* S and $V \setminus A$ in the other *component* $\bar{S}$. Now there are no edges between the *components* and thus the *cut value* is 0 and therefore the *normalized cut value* is also 0. The *normalized cut value* of a partition of a graph can never be below 0 because the edge-weights are not allowed to be negative. Therefore, $\{S, \bar{S}\}$ has a minimum *normalized cut value* and we can check in constant time if $0 \leq N$.

Now if the graph is connected, it is a complete graph. By Lemma 8.2 every partition of a complete graph with unweighted edges has the same *normalized cut value*. Therefore, we can find a partition with minimum *normalized cut value* in linear time by putting $1 < x < n$ vertices in *component* S and the rest of the $n-x$ vertices in *component* $\bar{S}$. Since the *Breadth-first search* algorithm took linear time we found the partition with minimum *normalized cut value* in linear time. Now we only have to check if the *normalized cut value* of $\{S, \bar{S}\}$ is $\leq N$.

Therefore, the Normalized Cut problem is solvable in linear time on cluster graphs with unweighted edges. □

8.2 Linear time algorithm for cactus graphs

In this subsection, we show a linear time algorithm for the Normalized Cut problem on cactus graphs with unweighted edges. Note that in Subsection 7 we already showed a quadratic time algorithm for outerplanar graphs. Since cactus graphs are outerplanar graphs we could use the algorithm for outerplanar graphs to obtain a quadratic time algorithm for cactus graphs. However, we now show that for cactus graphs with unweighted edges, we can solve the Normalized Cut problem in linear time. The algorithm is based on the algorithm for the Sparsest Cut problem on cactus graphs with unweighted edges presented in the paper by Bonsma et al. [4].

Theorem 8.3. *The* Normalized Cut *problem is solvable in linear time on cactus graphs with unweighted edges.*

Proof. A cactus graph consists of biconnected blocks, from now on called blocks, which are either single edges or cycles. In Figure 11 we see an example of a cactus graph and its blocks. We can find the blocks of a cactus graph in linear time using the algorithm presented in the paper by Tarjan [23]. We denote with $V(B)$ the set of vertices of a block B and with $E(B)$ the set of edges of a block B. Vertices that are part of multiple blocks are called cut vertices. According to Theorem 4.6 a partition with minimum *normalized cut value* has exactly two connected *components*. Therefore, there is a partition with minimum *normalized cut value* that only has edges of one block in the *cut*.

We have an instance of the Normalized Cut problem of a cactus graph with unweighted edges $G = (V, E)$ and a real number N. We present an algorithm that can find

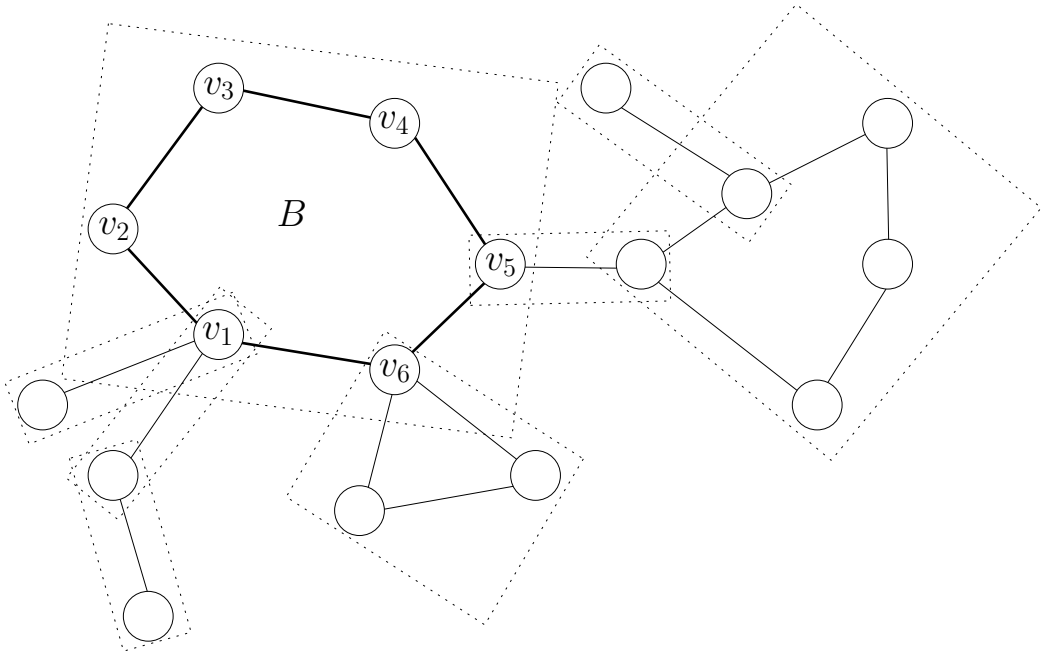


Figure 11: A cactus graph and its blocks

the partition with minimum *normalized cut value* of G in linear time. This algorithm requires one preprocessing step.

We compute for every block B and every vertex $v \in B$ the total number of edges $f(v, B)$ of the *component* that contains v in the graph $G' = G \setminus E(B)$. We use the same method presented in the algorithm for solving the NORMALIZED CUT problem on outerplanar graphs in Subsection 7.2. Therefore, this preprocessing step takes $O(n)$ time. In Figure 11 we see a cactus graph and its blocks. As an example, for the highlighted block B, we have $f(v_1, B) = 3$.

We have two different types of blocks, a single edge or a cycle. For a block B that is a single edge (v, u) we compute the *normalized cut value* $\mathrm{NCut}(S, \bar{S})$ of the partition $\{S, \bar{S}\}$ defined by its *cut* which only contains (v, u) in constant time as:

$$\mathrm{NCut}(S, \bar{S}) = \frac{2|E|}{(2f(v, B) + 1) \cdot (2f(u, B) + 1)}. \tag{105}$$

For a block B that is a cycle, we call the $k = |V(B)|$ vertices along the cycle $v_1, \ldots, v_k$. In Figure 11 we see an example of a block B and its vertices $v_1, \ldots, v_6$ along the cycle. According to Theorem 4.6 a partition with minimum *normalized cut value* has exactly two connected *components*. Therefore, the *cut value* of a partition of a block that is a cycle is

always two, otherwise, the partition does not have exactly two connected *components*. Let $C_{x,y}$ be the set of vertices $\{v_x, \ldots, v_y\}$ for some $1 \leq x \leq y \leq k$. Furthermore, let $\{S_{x,y}, \bar{S}_{x,y}\}$ be the partition defined by its *cut* which only contains edges between $C_{x,y}$ and $V(B) \setminus C_{x,y}$.

We want to find the partition with minimum *normalized cut value* among all partitions that only have edges from B in their *cut* and have exactly two connected *components*. Recall that the *cut value* $\text{cut}(S_{x,y}, \bar{S}_{x,y})$ is always 2. Proposition 4.1 tells us that we achieve the lowest *normalized cut value* for a given *cut value* if the *volumes* differ the least in value. Therefore, we first compute the *normalized cut value* of partition $\{S_{1,1}, \bar{S}_{1,1}\}$. Then we start a procedure where when $\text{vol}(S_{x,y}, V) \leq |E|$ we add vertex v_{y+1} to $S_{x,y}$ to increase $\text{vol}(S_{x,y}, V)$ and otherwise we remove vertex v_x from S to decrease $\text{vol}(S_{x,y}, V)$. Therefore, we compute the *normalized cut value* of partition $\{S_{x,y+1}, \bar{S}_{x,y+1}\}$ and otherwise we compute the *normalized cut value* of partition $\{S_{x+1,y}, \bar{S}_{x+1,y}\}$. We stop when either $x > y$ or $y > k$. Among the partitions for which we have computed the *normalized cut value*, there must be a partition with minimum *normalized cut value* since we consistently tried to minimize the difference between the *volumes* $\text{vol}(S, V)$ and $\text{vol}(\bar{S}, V)$. We can compute the *normalized cut value* $\text{NCut}(S_{x,x}, \bar{S}_{x,x})$ of $\{S_{x,x}, \bar{S}_{x,x}\}$ in constant time as:

$$\text{NCut}(S_{1,1}, \bar{S}_{1,1}) = \frac{2 \cdot 2|E|}{(2f(v_1, B) + 2) \cdot (2|E| - (2f(v_1, B) + 2))}. \tag{106}$$

We can compute the *normalized cut value* $\text{NCut}(S_{x+1,y}, \bar{S}_{x+1,y})$ of $\{S_{x+1,y}, \bar{S}_{x+1,y}\}$ in constant time using the known *volumes* $\text{vol}(S_{x,y}, V)$ and $\text{vol}(\bar{S}_{x,y}, V)$ as:

$$\text{NCut}(S_{x+1,y}, \bar{S}_{x+1,y}) = \frac{2 \cdot 2|E|}{(\text{vol}(S_{x,y}, V) - f(x, B) - 2) \cdot (2|E| - (\text{vol}(S_{x,y}, V) - f(x, B) - 2))}. \tag{107}$$

Here we removed $f(x, B)$ from $\text{vol}(S_{x+1,y}, V)$ since vertex x is not in S anymore and we removed 2 from $\text{vol}(S_{x+1,y}, V)$ for the edge between vertices x and $x + 1$ that is now in the *cut* instead of completely inside S. We can compute the *normalized cut value* $\text{NCut}(S_{x,y+1}, \bar{S}_{x,y+1})$ of $\{S_{x,y+1}, \bar{S}_{x,y+1}\}$ in constant time using the known *volumes* $\text{vol}(S_{x,y}, V)$ and $\text{vol}(\bar{S}_{x,y}, V)$ as:

$$\text{NCut}(S_{x,y+1}, \bar{S}_{x,y+1}) = \frac{2 \cdot 2|E|}{(\text{vol}(S_{x,y}, V) + f(y + 1, B) + 2) \cdot (2|E| - (\text{vol}(S_{x,y}, V) + f(y + 1, B) + 2))}. \tag{108}$$

Here we added $f(y+1, B)$ to $\text{vol}(S_{x,y+1}, V)$ since vertex $y+1$ is now in S and we added 2 to $\text{vol}(S_{,y+1}, V)$ for the edge between vertices y and $y + 1$ that is now inside S instead of in the *cut*.

The procedure takes at most $2l$ steps where $l = |E(B)|$ and therefore takes $O(l)$ time. Performing the procedure for every block that is a cycle takes $O(n)$ time since $|E| \in O(n)$. Once we have computed the partition with minimum *normalized cut value* for every block, we can check which partition among these partitions has the minimum *normalized cut value* and if it is $\leq N$. We conclude that the Normalized Cut problem is solvable in linear time for cactus graphs. □

9 Approximation algorithm for the Normalized Cut problem

In this section, we observe that there is an $O(\log(n))$-approximation algorithm for the Normalized Cut problem with $n = |V|$.

Theorem 9.1. *The* Normalized Cut *problem has an* $O(\log(n))$*-approximation algorithm.*

Proof. As we saw in Section 3, we can reduce the Normalized Cut problem to the vertex-weighted Sparest Cut problem. This reduction takes $O(|V| + |E|)$ time. After the reduction, we have an instance of the vertex-weighted Sparsest Cut problem and we can apply the approximation algorithm from Leighton and Rao [16]. □

10 Conclusion

In this thesis, we showed that the Normalized Cut problem is NP-complete on a variety of $\mathcal{H}$-free graph classes. Specifically, we showed that the Normalized Cut problem is NP-complete on claw-free, split, and complete graphs. Furthermore, we showed that the Normalized Cut problem with unweighted edges is strongly NP-complete. We showed an important property that the partition with minimum *normalized cut value* has two connected *components* and used this property to construct polynomial-time algorithms on certain $\mathcal{H}$-free graph classes. We showed that we can solve the Normalized Cut problem on forests in linear time and on outerplanar graphs in quadratic time. Furthermore, we showed that we can solve the Normalized Cut problem with unweighted edges on cluster graphs and cactus graphs in linear time. Lastly, we observed that there exists an $O(\log(n))$-approximation algorithm for another graph partitioning problem to which we can reduce the Normalized Cut problem. Therefore, we have an $O(\log(n))$-approximation algorithm for the Normalized Cut problem..

Since we showed the property that the partition with minimum *normalized cut value* has two connected *components*, it would be interesting to find out if we can generalize this property such that we can show that the partition with minimum *normalized k-cut value* has k connected *components*. Unfortunately, we cannot rewrite the *normalized k-cut value* to a second form in the same way we did for the *normalized cut value*. Therefore, we need to find another approach to try and generalize the property.

Another result of this thesis is that we showed that the Normalized Cut problem is NP-complete on complete graphs and that we can solve the Normalized Cut problem with unweighted edges on cluster graphs in linear time. Future work might direct their research towards finding for which other $\mathcal{H}$-free graph classes the Normalized Cut problem and the Normalized Cut problem with unweighted edges differ in complexity.

In this thesis we used a reduction from the Sparsest Cut problem with unweighted edges to show that the Normalized Cut problem with unweighted edges is strongly NP-complete in general. Therefore, we cannot hope to obtain a fully polynomial-time approximation scheme. However, it would be interesting to find out if the Normalized Cut problem obtains a polynomial-time approximation scheme or not. Furthermore, in Section 3 we saw that the Sparsest Cut problem has an $O(\sqrt{\log(n)})$-approximation algorithm [2]. Since the Normalized Cut problem and the Sparsest Cut problem have a lot of similarities, it would be interesting to try and adapt the $O(\sqrt{\log(n)})$-approximation algorithm of the Sparsest Cut problem and obtain an $O(\sqrt{\log(n)})$-approximation algorithm for the Normalized Cut problem.

References

[1] https://www.wolframalpha.com/.

[2] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 222–231. ACM, 2004.

[3] Christopher J. Augeri and Hesham H. Ali. New graph-based algorithms for partitioning VLSI circuits. In *Proceedings of the 2004 International Symposium on Circuits and Systems, ISCAS 2004, Vancouver, BC, Canada, May 23-26, 2004*, pages 521–524. IEEE, 2004.

[4] Paul S. Bonsma, Hajo Broersma, Viresh Patel, and Artem V. Pyatkin. The complexity of finding uniform sparsest cuts in various graph classes. *J. Discrete Algorithms*, 14:136–149, 2012.

[5] Aydin Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent advances in graph partitioning. In Lasse Kliemann and Peter Sanders, editors, *Algorithm Engineering - Selected Results and Surveys*, volume 9220 of *Lecture Notes in Computer Science*, pages 117–158. 2016.

[6] Amir Daneshgar and Ramin Javadi. On the complexity of isoperimetric problems on trees. *Discret. Appl. Math.*, 160(1-2):116–131, 2012.

[7] M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified np-complete graph problems. *Theor. Comput. Sci.*, 1(3):237–267, 1976.

[8] Pawel Gawrychowski, Shay Mozes, and Oren Weimann. Minimum cut in o(m $\log^2$ n) time. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 57:1–57:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[9] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[10] Olivier Goldschmidt and Dorit S. Hochbaum. Polynomial algorithm for the k-cut problem. In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 444–451. IEEE Computer Society, 1988.

[11] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[12] Ramin Javadi and Amir Nikabadi. On the parameterized complexity of sparsest cut and small-set expansion problems. *CoRR*, abs/1910.12353, 2019.

[13] Volker Kaibel. On the expansion of graphs of 0/1-polytopes. In Martin Grötschel, editor, *The Sharpest Cut*, volume 4 of *MPS-SIAM series on optimization*, pages 199–216. SIAM, 2004.

[14] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

[15] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, 2007.

[16] Frank Thomson Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

[17] David W. Matula and Farhad Shahrokhi. Sparsest cuts and bottlenecks in graphs. *Discret. Appl. Math.*, 27(1-2):113–123, 1990.

[18] Bojan Mohar. Isoperimetric numbers of graphs. *J. Comb. Theory, Ser. B*, 47(3):274–291, 1989.

[19] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on*

Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010, pages 755–764. ACM, 2010.

[20] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[21] Arlei Silva, Ambuj K. Singh, and Ananthram Swami. Spectral algorithms for temporal graph cuts. In Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis, editors, *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 519–528. ACM, 2018.

[22] Mechthild Stoer and Frank Wagner. A simple min-cut algorithm. *J. ACM*, 44(4):585–591, 1997.

[23] Robert Endre Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[24] Dijana Tolic, Nino Antulov-Fantulin, and Ivica Kopriva. A nonlinear orthogonal non-negative matrix factorization approach to subspace clustering. *Pattern Recognit.*, 82:40–55, 2018.

[25] Suzanne Vincken. Computation of normalized cuts of graphs. Master's thesis, Utrecht University, 2021.

[26] Jiří Šíma and Satu Elisa Schaeffer. On the np-completeness of some graph cluster measures. In Jirí Wiedermann, Gerard Tel, Jaroslav Pokorný, Mária Bieliková, and Julius Stuller, editors, *SOFSEM 2006: Theory and Practice of Computer Science, 32nd Conference on Current Trends in Theory and Practice of Computer Science, Merín, Czech Republic, January 21-27, 2006, Proceedings*, volume 3831 of *Lecture Notes in Computer Science*, pages 530–537. Springer, 2006.