Approximate Kernels for Graph Contraction Problems

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Spoorthy Gunda

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

April, 2019

Supervisor: Soumen Maity and Saket Saurabh

Certificate

This is to certify that this dissertation entitled Approximate Kernels for Graph Contraction Problems towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Spoorthy Gunda at Institute of Mathematical Science under the supervision of Saket Saurabh, Professor, Theoretical Computer Science, and at Indian Institute of Science Education and Research under the supervision of Soumen Maity, Associate Professor, Department of Mathematics during the academic year 2018-2019.

Soumen Maity

Saket Saurabh

Committee:

Soumen Maity

Saket Saurabh

Neeldhara Misra

This thesis is dedicated to my friend.

Declaration

I hereby declare that the matter embodied in the report entitled Approximate Kernels for Graph Contraction Problems are the results of the work carried out by me at the Institute of Mathematical Sciences, Chennai under the supervision of Saket Saurabh and at Indian Institute of Science Education and Research, Pune, under the supervision of Soumen Maity and the same has not been submitted elsewhere for any other degree.

G.Spoorthy

Spoorthy Gunda

Acknowledgements

I would like to express my great appreciation to Dr. Soumen Maity for his constant support and guidance. I am thankful to him for introducing me to Prof. Saket Saurabh.

I would like to express my deep gratitude to Prof. Saket Saurabh for the opportunities and exposure provided to me at The Institute of Mathematical Sciences, Chennai. I am very thankful for the valuable suggestions and support during the project.

I thank Pratik Tale and Pallavi Jain for their time spent on the discussions. I am especially thankful to Pratik Tale for his constant help and support during the project.

Finally, I wish to thank my family and friends for their emotional support and encouragement.

Abstract

We study a few graph contraction problems Chordal Contraction, Clique Contraction and Split Contraction, from the viewpoint of Lossy Kernelization, which is a recently introduced framework to study NP-hard problems. H-Contraction problem is to find set of edges such that when contracted will give the graph H. We study approximate kernel for the minimization version of these problems by including some parameters.

Contents

Chapter 1

Introduction

Graph editing problems have been extensively studied in the literature. Basic graph editing operations include deleting vertices or edges, adding vertices or edges, and edge contraction. Given a graph G, graph editing problems ask if the given graph can be edited to get the desired property. Some of these problems can also be viewed as editing the graph so that the graph will not contain any induced subgraphs that are isomorphic to a given set of forbidden graphs. Many well-known problems can be viewed as graph editing problems. One such example is the VERTEX COVER problem, it can be viewed as deleting few vertices such that the remaining graph is independent or deleting few vertices such that it does not have any edges.

In recent years various graph editing problems with a restricted number of allowed edit operations, like CHORDAL EDITING [7, 11, 3], SPLIT EDITING [8], CLUSTER EDITING [6] were considered. When the editing is restricted to edge contraction, the problems are usually known to be very hard compared to the other editing operation. The initial results on NP-hardness of graph contraction problems were shown by Watanabe et al. [13, 12]. Recently parameterized complexity of graph contraction problems have received a lot of attention.

In Parameterized Complexity, the input has a parameter along with the problem instance, that captures the hardness of the problem. A central notion in this field is *fixed parameter tractable* (FPT) problems. A parameterized problem Π is said to be FPT if for a given instance (I, k), there is an algorithm that decides whether or not it is a yes instance of Π in time $f(k)|I|^{\mathcal{O}(1)}$ where f is some computable function of k. Every parameterized problem need not be fixed-parameter tractable for given parameter. The problems that are not fixed-parameter tractable are said to be W-hard. The class of W-hard problem can be further divided on the basis of hardness. The complexity classes are assumed to follow the containment FPT $\subsetneq$ W[1] $\subsetneq$ W[2] $\subseteq \ldots \subseteq$ W[P] $\subseteq$ XP.

Kernelization is another import notion of Parameterized Complexity. Kernelization algorithm for a parameterized problem Π is an algorithm that given an instance (I, k) of Q returns an equivalent instance (I', k') of Q in polynomial time, such that $|I'| + k \leq g(k)$, for some computable function g. An important result of this theory is that a problem Π is in FPT if and only if it admits a kernelization algorithm. H-Contraction problem and F-free contraction problem with a parameter k are defined as follows

H-Contraction	**Parameter:** Solution size k
Input: A graph G and H and an integer k	
Question: Does there exists k edges whose contraction results in the graph H?	

F- FREE CONTRACTION	**Parameter:** Solution size k
Input: A graph G, a family of graphs F and an integer k	
Question: Does there exists k edges whose contraction results in a graph that does not contain any induced subgraphs from H?	

Cai and Guo [2] have shown that F-free contraction parameterized by solution size is W-hard for all F that are three connected graphs except for complete graphs. Graph contraction problems are known to be W-hard even if the forbidden set is finite and bounded. In C_l-FREE CONTRACTION, problem the forbidden set is the cycle of length l, even for a fixed l when parameterized by solution size it has been proved to be W-hard by Lokshtanov et al. [9]. Following are a few of the results that are known on the parameterized complexity of graph contraction problems. The CHORDAL CONTRACTION problem is W[2]-hard when parameterized by the size of the solution [9]. CLIQUE CONTRACTION parameterized by the solution size can be solved in $\mathcal{O}(2^{7k} \cdot k^{2k+5})$, but admits no polynomial kernel unless $NP \subseteq CoNP/poly$ [2]. SPLIT CONTRACTION parameterized by the size of minimum vertex cover ℓ, does not have any algorithm with running time $2^{o(\ell^2)} \cdot |I|^{\mathcal{O}(1)}$ under the standard ETH hypothesis [1]. SPLIT CONTRACTION is W[1]-hard when parameterized by the size of solution [1].

Lossy Kernelization is a new framework developed by Lokshtanov et al. [10] to study NP-hard optimization problems. In Lossy Kernelization the question posed is different from that of the question posed in Parameterized complexity. We look for optimization solution instead of the existence of the solution. FPT-approximation algorithm for a parameterized optimization problem with parameter k, is an algorithm that finds approximate solution in time $f(k)n^{\mathcal{O}(1)}$, where f is some computable function.

Approximate Kernelization is the main concept of Lossy Kernelization. A problem Π with an input instance (I, k) is said to admit approximate kernel of size $g(k)$, if there exists a polynomial time running algorithm that outputs an instance (I', k') with $|I'| + k' \leq g(k)$ such that c-approximate solution for (I', k') can be lifted to $b \cdot c$-approximate solution for (I, k). One of the main results of this theory is the equivalence between FPT approximation and approximate kernel.

Under the Lossy Kernelization set up, for input graph G, H-Contraction problem is to find a minimum set of edges to be contracted so that the resulting graph is H. Formally the minimization version of H-Contraction with parameter being the size of the solution is defined as follows.

$$\mathrm{HC}(G, k, F) = \begin{cases} \infty & \textit{if } G/F \textit{ is not isomorphic to } H \\ min \{|F|, k+1\} & \text{otherwise.} \end{cases}$$

The optimum value of the solution is defined as $\mathrm{OPT}_{\mathrm{HC}}(G, k) = \min\limits_{F \subseteq E(G)} \mathrm{HC}(G, k, F)$. We consider that any solution of size greater than k is equally bad, so all the solutions of size greater than k are given the value $k + 1$. But if there is a solution of size at most k, we are interested in knowing the minimum value it can attain.

Our goal is to find α-approximate solution for graph contraction problems for some small $\alpha \geq 1$ with running time $f(k) \cdot n^{\mathcal{O}(1)}$, where n is the number of vertices in input graph. In other words if $|F^*|$ is the value of the optimum solution we will try to find a solution of size at most $\alpha|F^*|$. This can also be achieved by finding α-approximate kernels

1.1 Scope of the thesis

In Chapter 3, we formally define the framework of our study: Lossy Kernelization in detail. In this thesis, we study a few graph contraction problems from Lossy Kernelization point of view. In particular we study α-approximate kernels for Chordal Contraction, Clique Contraction and Split Contraction problems.

In Chapter 4, we prove that Chordal Contraction parameterized by solution size do not admit any polynomial size approximate kernel unless NP$\subseteq$CoNP/Poly.

In Chapter 5, we show that Clique Contraction, parameterized by solution size admits α-approximate kernel of polynomial size for every $\alpha > 1$.

In Chapter 6, we show that Split Contraction parameterized by solution size and maximum independent size, admits approximate kernels of polynomial size.

Chapter 2

Preliminaries

2.1 Graph Theory

In this thesis, we consider only simple and undirected graphs with finite number of vertices. Vertices of the graph G are represented by $V(G)$ and edges by $E(G)$. Two vertices u and v are said to be adjacent if there exists an edge uv between them and the vertices u, v are called the endpoints of the edge. The set of vertices that are adjacent to a vertex u are called its neighbours and is represented by $N(u)$. Let A be the set of vertices then $N(A)$ is union of neighbours of all the vertices in A.

A graph H is called a *subgraph* of a graph G, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A graph H is called an *induced subgraph* of graph G, if $V(H) \subseteq V(G)$ and $E(H)$ is set of all the edges of G whose both the endpoints are in $V(H)$. For a set of vertices $A \subset V(G)$ the induced subgraph on A is represented by $G[A]$. A *Spanning subgraph* H is a subgraph with $V(H) = V(G)$.

A *Path* P is a graph with $V(P) = \{v_1, v_2, ...v_n\}$ and $E(P) = \{v_i v_{i+1}; 1 \leq i \leq n-1\}$. A graph G is called connected if there exists a path between any two vertices of G.

A *Cycle* C is a graph with $V(C) = \{v_1, v_2, ...v_n\}$ and $E(C) = \{v_i v_{i+1}; 1 \leq i \leq n-1\} \cup \{v_n v_1\}$. A graph which does not contain any cycle as a subgraph is called an *acyclic* graph. A connected acyclic graph is called a *tree*. In a tree T, if $|V(T)| = n$ then $|E(T)| = n - 1$. A *spanning tree* of a graph G is a spanning subgraph that is connected and acyclic.

Complement of a graph G is $\tilde{G}$ such that $V(G) = V(\tilde{G})$, and two vertices are adjacent in $\tilde{G}$ if and only if they are non-adjacent in G.

A *clique* is a simple graph such that there is an edge between every pair of vertices. A set of vertices I is said to be independent if all the vertices of I are non-adjacent. A clique is a complement of an independent graph and vice versa.

A *Split graph* G is a simple graph whose vertices can be partitioned into two sets such that one set induces a clique and the other an independent set. A graph G is said to be split graph if and only if G does not have C_4, C_5 or $2K_2$ as an induced subgraph. Here C_4, C_5 are cycles of length 4 and 5 respectively and a $2K_2$ is a graph on 4 vertices with two disjoint edges. The set $\{C_4, C_5, 2K_2\}$ is called the forbidden set for a split graph. A clique is a split graph with at most one vertex in the independent set.

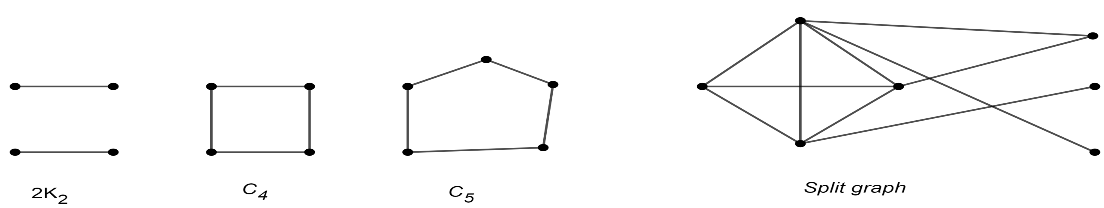


Figure 2.1: Forbidden graphs of split graph and an example of a split graph.

A *Chordal graph* is a simple graph with no induced cycles of length at least 4. A chordal graph has an induced cycle of length at most 3, so it is also called a triangulated graph. All split graphs are chordal graphs but all chordal graphs are not split graphs.

Vertex Cover of a graph G is a set of vertices such that every edge of the graph is incident on at least one vertex of this set. Let X be the vertex cover of a graph G, since every edge has at least one endpoint in X, $G \setminus X$ is an independent set i.e., there is no edge with both the endpoints in $V(G) \setminus X$.

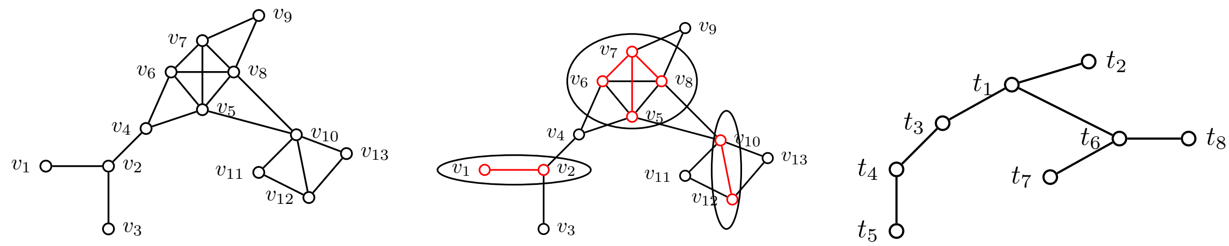
2.1.1 Contraction

Edge contraction is a basic operation on graphs, where the two endpoints of an edge $e = uv$ are replaced by a single vertex x, such that $N(x) = N(u) \cup N(v)$. The resultant graph obtained by contracting the edge e is represented by $G/\{e\}$. Let F be set of edges of G, then G/F represents the graph obtained by contracting all the edges of F, iteratively in G.

We say that G can be contracted to a graph H, if there exists a set of edges F such that $G/F = H$. Equivalently G can be contracted to a graph H, if there exists an onto function $\varphi : V(G) \rightarrow V(H)$, with $W(h) = \{v \in V(G) : \varphi(v) = h\}$ for all $h \in H$ such that

1. $G[W(h)]$ is a connected graph for every $h \in H$.
2. Two vertices $h, h' \in V(H)$ are adjacent if and only if there exists a vertex from $W(h)$ and $W(h')$ that are adjacent.
3. $\mathcal{W} = \{W(h) \;\forall\; h \in H\}$ is a partition of $V(G)$.

Refer to the Fig 2.2. $\mathcal{W}$ is said to be a H-witness structure of G and the sets $W(h)$ are called witness sets. The set of edges F is obtained by taking the union of the edges corresponding to spanning trees of all the witness sets. We call the witness sets with more than one vertex as big witness sets.



(a) Input Graph (b) Graph Partition (c) Resultant graph

Figure 2.2

Proposition 2.1.1. *If a graph H can be obtained from a graph G by contracting at most k edges, then for $\mathcal{W}$ a H-witness structure of G, the following conditions are satisfied*

1. *The total number of vertices in G that are contained in the big witness sets is bounded by $2k$.*
2. *For all $h \in H$, $|W(h)| \leq k + 1$.*
3. *The number of big witness sets is not more than k.*

2.2 Approximation Algorithms

NP is set of decision making problems, that can be solved in polynomial time. In other words, for a problem Π if there exists an algorithm to check if a string s is a correct solution for Π in polynomial time, then it is in NP.

Example 1 Given a subset of vertices of a graph G, in polynomial time it can be checked if these vertices are independent.

Example 2 In set cover problem the input instance consists of a Universe U and a family of subsets S, the goal is to find sets from S whose union will be U. In the decision version along with the instance few sets are also given, and one can in polynomial time find if the union of these sets is U and hence Set Cover problem is in NP.

A problem Π is said to be **NP-hard**, if there exists a polynomial time reduction from every problem of NP to Π. A problem is said to be **NP-complete** if it is in NP and NP-hard. Example of NP-complete problem is Vertex Cover, Set Cover.

Approximation algorithms is framework to study NP-hard problems. In approximation algorithms we try to find a solution for an optimization problem. But since most of these problems are NP-hard, optimum solution cannot be found in polynomial time. So the goal is to find a solution that is close to optimal solution in polynomial time. An optimization problem can be a minimization or a maximization problem. Examples of optimization problems are finding a minimum vertex cover of a graph or finding a maximum independent set of a graph. For an instance I of the problem Π, we denote the value of the optimum solution by $OPT_{\Pi}(I)$

For every instance I of a minimization problem Π, if there exists an algorithm $\mathcal{A}$ that outputs a solution with value $\mathcal{A}(I)$, such that $\dfrac{\mathcal{A}(I)}{OPT_{\Pi}(I)} \leq \alpha_{\mathcal{A}}$, then $\alpha_{\mathcal{A}}$ is called the approximate ratio of the algorithm $\mathcal{A}$. The solution outputted by the algorithm is called $\alpha_{\mathcal{A}}$-approximate solution. Observe that $\alpha_{\mathcal{A}}$ is always greater than or equal to 1.
In other words for a minimization problem the approximate ratio of a algorithm is defined as

$$\alpha_{\mathcal{A}} = \max_{I} \frac{\mathcal{A}(I)}{OPT_{\Pi}(I)}.$$

Similarly for a maximization problem the approximate ratio of a algorithm is defined as

$$\frac{1}{\alpha_{\mathcal{A}}} = \max_{I} \frac{\mathcal{A}(I)}{OPT_{\Pi}(I)}.$$

Vertex Cover: Given a graph G, we have to minimize $|S|$ such that $S \subseteq V(G)$ and S covers all the edges of G. We will give an algorithm that outputs a 2-approximate solution for every graph G. For an input graph G, pick an edge arbitrarily and put both the endpoints of this edge in a set X and delete these vertices from G, continue this process in the remaining graph until there are no edges left. The set X will be a 2-approximate vertex cover of G. The set X is a vertex cover of G as it covers all the edges of the graph. Since at least one endpoint of each edge must be part of vertex cover we get $OPT(G) \geq \frac{|X|}{2}$, hence X is a 2-approximate solution for vertex cover.

This algorithm can be generalized to find approximate solutions for any vertex deletion problem with bounded and finite forbidden set. Examples of one such problem is SPLIT VERTEX DELETION.

A problem Π is said to admit *Polynomial time approximation scheme* (**PTAS**), if there exists a family of α-approximate algorithms for every $\alpha > 1$ such that the running time is polynomial and can depend on α.

In approximation algorithms the goal is to come up with algorithms with efficient running time and to achieve good approximate ratio.

All problems need not have polynomial time α-approximate algorithm, for any constant $\alpha > 1$. Example for one such problem is TRAVELLING SALESMEN PROBLEM [4].

2.3 Parameterized complexity

For the problems that do not admit a polynomial time algorithm, we introduce a parameter and try to find an algorithms whose running time is polynomial in the instance size but some function in the parameter. We formally define parameterized problem and also define the notion of hardness in this section. For a more detailed explanation and concepts on Parameterized Complexity refer to the book Parameterized Algorithms by Cygan et al. [5].

Definition 1. *A parameterized problem is a language $L \subseteq \Sigma^* \times N$, where Σ is a fixed, finite alphabet. For an instance $(x, k) \in \Sigma^* \times N$, k is the parameter.*

Usually, structural property of the instance can be inferred from the parameter. Most commonly used parameter is the size of the solution. Example for parameterized problem is given a graph G, to check if the graph has a clique of size at least k.

Definition 2. *A parameterized problem $L \subseteq \Sigma^* \times N$, is said to be* ***fixed parameter tractable (FPT)****, if there exists a an algorithm $\mathcal{A}$ that correctly decides whether a given instance $(x, k) \in L$, whose running time is $f(k) \cdot |(x, k)|^c$ for a computable function $f : \mathbb{N} \to \mathbb{N}$, and a constant c.*

Definition 3. *A parameterized problem $L \subseteq \Sigma^* \times N$, is said to be* ***slice-wise polynomial (XP)****, if there exists a an algorithm $\mathcal{A}$ that correctly decides whether a given instance $(x, k) \in L$, whose running is $f(k) \cdot |(x, k)|^{g(k)}$ for a computable functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$.*

Set of all fixed-parameter tractable problems are represented by FPT complexity class. Similar to NP-completeness theory, there is also a lower bound theory for parameterized algorithms. All problems need not be FPT. Parameterized reduction is defined in order to obtain lower bounds for problems. There are different levels of hardness like W[1], W[2],... in parameterized complexity. We assume that FPT $\subsetneq$ W[1] $\subsetneq$ W[2] $\subseteq \ldots \subseteq$ W[P] $\subseteq$ XP which is stronger assumption than $NP \neq P$. We will see that Chordal Contraction and Split Contraction when parameterized by solution size are W-hard.

Definition 4. *For two parameterized problems A, B. A* ***parameterized reduction*** *is a algorithms that outputs an instance (x', k') of B from an instance (x, k) of A such that*

- *the running time of the algorithm is $f(k) \cdot |x|^{\mathcal{O}(1)}$ for some computable function f.*
- *(x, k) is yes instance of A if and only if (x', k') is yes instance of B.*
- *$k' \leq g(k)$ for some computable function g*

Theorem 2.3.1. *[5] If there is parameterized reduction from A to B and B is in FPT then A is also in FPT.*

Definition 5. *A* ***kernelization algorithm*** *for a paremeterized problem Π is an algorithm that given an instance (x, k) of Q returns an equivalent instance (x', k') of Q in polynomial time such that $|x'| + k \leq g(k)$, for some computable function g.*

Theorem 2.3.2. *[5] A parameterized problem Π is said to admit polynomial kernelization if and only if Π is fixed-parameter tractable.*

Chapter 3

Lossy Kernelization

Lossy Kernelization is a new framework for studying NP-Hard problems. Similar to approximation algorithms this theory deals with optimization problems along with some parameter. In Section 2.3, we have seen basics of Parameterized complexity theory, there the main concept was Kernelization. Parameterized Complexity theory has paved a way to give more insight on NP-hard problems. But the drawback of the theory was that kernelization do not work well when combined with heuristics. In order to address this issue Lokshtanov et al. [10], in their paper established a new framework of study called *Approximate Kernelization*. Informally, approximate kernelization can be described as reducing an input instance (I, k) to an instance (I', k'), such that for $c, \alpha > 1$, a c-approximate solution for (I', k') can be lifted to $c\alpha$-approximate solution for (I, k). In this chapter we will look at some basics of Approximate Kernelization. Definitions in this chapter are taken from the paper Lossy Kernelization by Lokshtanov et al. [10].

3.1 Parametrized Optimization Problem

We will first define parameterized optimization problem formally.

Definition 6. *A parameterized optimization problem* Π *is a computable function*

$$\Pi : \Sigma^* \times \mathbb{N} \times \Sigma^* \to \mathbb{R} \cup \{\pm\infty\}.$$

With the instances of Π *being the pairs* $(I,k) \in \Sigma^* \times \mathbb{N}$ *and a solution to* (I,k) *is simply a string* $s \in \Sigma^*$ *such that* $|s| \leq |I| + k$.

For an instance (I,k) of the problem Π value of the solution s is given by $\Pi(I,k,s)$. Optimum value for the instance is defined as $OPT_\Pi(I,k) = \max_{s\in\Sigma^*} \Pi(I,k,s)$ for a maximization problem Π, and $OPT_\Pi(I,k) = \min_{s\in\Sigma^*} \Pi(I,k,s)$ for a minimization problem Π. Look at the following example for how parameterized optimization version of Vertex Cover problem with solution size as parameter can be defined

$$VC(G,k,X) = \begin{cases} \infty & \text{if X is not a vertex cover of G} \\ \min\{|X|, k+1\} & \text{otherwise.} \end{cases}$$

Definition 7. *A fixed parameter tractable* α*-approximation algorithm for a parameterized optimization problem* Π *is an algorithm* $\mathcal{A}$ *that takes an input instance* (I,k) *and outputs a solution* s *in* $f(k) \cdot |I|^{O(1)}$ *time, such that if* Π *is a minimization problem then* $\Pi(I,k,s) \leq \alpha \cdot OPT_\Pi(I,k)$ *, and if* Π *is a maximization problem then* $\alpha \cdot \Pi(I,k,s) \geq OPT_\Pi(I,k)$.

3.2 Approximate kernel

Given an instance of the problem, one can preprocess the instance and transform it to some smaller instance either by solving it partially or by deleting the obvious cases. We will try to do some preprocessing for obtaining approximate solutions for NP-hard problem. Informally, the smaller instance obtained by preprocessing is called kernel. One could apply some brute-force techniques to get the solution for kernel and later lift it to get the solution for the original one. We formally define the notion of approximate kernel in this section.

Definition 8. *Let* $\alpha \geq 1$ *be a real number and* Π *be a parameterized optimization problem. An* α***-approximate polynomial-time preprocessing algorithm*** *is a pair of polynomial-time algorithms, namely a reduction algorithm and a solution lifting algorithm, such that the following properties are satisfied*

- *Given an instance* (I,k) *of* Π*, the reduction algorithm outputs an instance* (I',k') *of* Π *called the reduced instance.*

- *Given the instances* (I, k) *and* (I', k') *of* Π*, and a solution* s' *to the reduced instance* (I', k')*, the solution lifting algorithm outputs a solution* s *to* (I, k) *such that for a parameterized minimization problem* Π

$$\frac{\Pi(I,k,s)}{\text{OPT}(I,k)} \leq \alpha \cdot \frac{\Pi(I',k',s')}{\text{OPT}(I',k')},$$

and for a parameterized maximization problem Π

$$\alpha \cdot \frac{\Pi(I,k,s)}{\text{OPT}(I,k)} \geq \frac{\Pi(I',k',s')}{\text{OPT}(I',k')}.$$

In the definition 8, solution lifting algorithm does not demand for an optimum solution for the reduced instance, some c-approximate solution for the reduced instance will suffice and the algorithm will output $c\alpha$-approximate solution for the original instance.

Definition 9. *An* α***-approximate kernel*** *for a parameterized optimization problem* Π*, and for* $\alpha \geq 1$*, is an* α*-approximate polynomial-time preprocessing algorithm, such that the size of the reduced instance,* $|I'| + k'$ *is upper bounded by a computable function* $g : \mathbb{N} \to \mathbb{N}$*, for all* $I \in \Sigma^*$.

If the size of the reduced instance is bounded by a polynomial function g, then we say that the problem Π admits a polynomial kernel.
In the next proposition we establish the equivalence between FPT approximation and approximate kernelization. This is one of the important result in Lossy kernelization.

Proposition 3.2.1. *[10] For a paramemeterized optimization problem* Π *and* $\forall \alpha \geq 1$, Π *admits fixed parameter tractable* α*-approximation algorithm if and only* Π *has an* α*-approximate kernel.*

Similar to the notion of PTAS for approximation we have the notion of PSAKS on approximate kernelization defined as follows

Definition 10. *A polynomial-size approximate kernelization scheme **(PSAKS)** for* Π *is a family of* α*-approximate polynomial kernelization algorithms for each* $\alpha > 1$.

If a parameterized optimization problem Π has α-approximate kernels for all $\alpha > 1$, then we say that Π admits PSAKS.

3.3 Lower Bounds for Approximate Kernels

Similar to notion of W-hardness in parameterized complexity, we will define lower bounds for approximate kernelization, and show that few problems cannot have α-approximate kernel of polynomial size. Alike to the notion of reduction to prove NP-hardness of a problem, we will define the notion of α-approximate polynomial parameter transformation to prove the lower bounds for approximate kernelization.

Definition 11. *Let $\alpha \geq 1$ be a real number. Let Π and Π' be two parameterized optimization problems. An α-approximate polynomial parameter transformation (α-**appt**) $\mathcal{A}$ from Π to Π' is a pair of polynomial time algorithms, namely reduction algorithm $\mathcal{R}_\mathcal{A}$ and a solution lifting algorithm. Given as input an instance (I, k) of Π the reduction algorithm outputs an instance (I', k') of Π'. The solution lifting algorithm takes as input an instance (I, k) of Π, the output instance (I', k') of Π', and a solution s' to the instance I' and outputs a solution s' to (I, k) such that if Π is minimization problem*

$$\frac{\Pi(I,k,s)}{OPT_\Pi(I,k)} \leq \alpha \cdot \frac{\Pi'(I',k',s')}{OPT_{\Pi'}(I',k')}$$

and if Π is a maximization problem then

$$\alpha \cdot \frac{\Pi(I,k,s)}{OPT_\Pi(I,k)} \geq \frac{\Pi'(I',k',s')}{OPT_{\Pi'}(I',k')}$$

In approximate kernelization, we reduce an instance to smaller instance of the same problem. Similarly, compression is reducing the instance of one problem to smaller instance of some other problem. We will now formally define approximate compression.

Definition 12. *For parameterized optimization problems Π and Π', an α-approximate compression from Π to Π' is an α-appt from from Π to Π' such that the instance outputted by reduction algorithm, $|I'| + k'$ is upper bounded by a computable function $g : \mathbb{N} \to \mathbb{N}$, for all $I \in \Sigma^*$.*

If there is an α-appt from problem Π to Π', and if Π' admits an α-approximate polynomial kernel then Π admits an α'-approximate polynomial compression for some $\alpha' > 1$. Hence, if a problem Π does not have α-approximate polynomial compression for any $\alpha > 1$, one can find reduction from Π to other problems in order to find the lower bounds.

In the paper [10], the authors have proved the lower bound results for finding the LONGEST PATHS in the graph parameterized by solution size and finding minimum set cover parameterized by the universe size.

Set cover problem parameterized by universe size is defined as follows

$$\mathrm{SC}/n((S,U),|U|,F) = \begin{cases} \min\{|F|, k+1\} & \text{if } F \text{ is a set cover} \\ \infty & \text{otherwise} \end{cases}$$

Theorem 3.3.1. *[10] Set cover problem parameterized by universe size does not have α-approximate compression of polynomial size for any $\alpha \geq 1$, unless* NP $\subseteq$ coNP/POLY.

Like the hypothesis NP $\neq$ P; NP $\neq$ coNP/POLY is also a standard hypothesis.

Chapter 4

Chordal Contraction

In this chapter we formally define the parameterized optimization version of Chordal Contraction and then prove the lower bound results for the problem.

4.1 Introduction

The Chordal Contraction problem is to minimize the number of edges in a given graph G, such that the contraction of these edges will result in a chordal graph. The problem of Chordal Contraction parameterized by solution size, is to check if there exists an edge set of size at most k whose contraction will result in a chordal graph. Lokshtanov et al. [9] proved that Chordal Contraction is W-hard when parameterized by solution size. We study this problem in the lossy kernel set up. Formally the parameterized optimization version of Chordal Contraction is defined as follows

$$\mathrm{CC}(G, k, P) = \begin{cases} \infty & \text{if } G/P \text{ is not a chordal graph} \\ min\{|P|, k+1\} & \text{otherwise.} \end{cases}$$

We show that the Chordal Contraction parameterized by solution size does not have an α-approximate kernel of polynomial size. In order to get this lower bound result we first define C_4-Free Contraction problem and obtain lower bound results by showing a polynomial time reduction from the parameterized optimization version of Set Cover problem. Set Cover parameterized by the solution size does not have any $\alpha-$approximate compression of polynomial size under standard complexity theory hypothesis [10, Theorem 11].

4.2 (No) Lossy Kernel for Chordal Contraction

4.2.1 Lower Bound for C_4-Free Contraction

If a graph H has no induced cycle of length 4, we say that H is C_4-free graph. C_4-Free Contraction problem is to find minimum number of edges in given graph G, whose contraction will result in a graph H, such that H has no induced cycle of length 4. Formally the parameterized optimization version of C_4-Free Contraction is defined as follows.

$$\text{FC}(G, k, P) = \begin{cases} \infty & \text{if} G/P \text{is not a } C_4 \text{ free graph} \\ min\,\{|P|, k+1\} & \text{otherwise.} \end{cases}$$

Theorem 4.2.1. *There is a* 1*-approximate polynomial parameter transformation from Set Cover parameterized by solution size to* C_4-Free Contraction *parameterized by solution size.*

Refer to section 3.3 for the definition of approximate polynomial parameter transformation.

Proof. In order to prove this theorem we have to give a pair of polynomial time algorithms, reduction algorithm and a solution lifting algorithm, such that the reduction algorithm takes an instance $((S, U), k)$ of Set Cover and outputs an instance (G, k) of C_4-Free Contraction problem and if P is the solution to the instance of C_4-Free Contraction problem the solution lifting algorithm should output a solution F for the Set Cover instance such that

$$\frac{\text{SC}((S, U), k, F)}{\text{OPT}_{\text{SC}}((S, U), k)} \leq \frac{\text{FC}(G, k, P)}{\text{OPT}_{\text{FC}}(G, k)}.$$

Reduction Algorithm: Let the instance of set cover be $((S, U), k)$ with $U = \{x_1, x_2, ..., x_n\}$ and $S = \{S_1, S_2, ..., S_m\}$. Construct the instance (G, k) as follows. Add a special vertex g and for every set $S_i \in S$ add a corresponding vertex s_i, and for every element $x_i \in U$ add three vertices a_i , b_i and c_i. Look at the Figure 4.1.

Add the edges as follows

1. Make the set of vertices $\{s_i : 1 \leq i \leq m\}$ into a clique.
2. For all $1 \leq i \leq m$, and for all $1 \leq j \leq n$ add an edge from g to s_i, a_j and b_j .
3. For all $1 \leq j \leq n$, add an edge from c_j to a_j and b_j .
4. If $x_j \in S_i$ then add an edge between s_i and c_j .

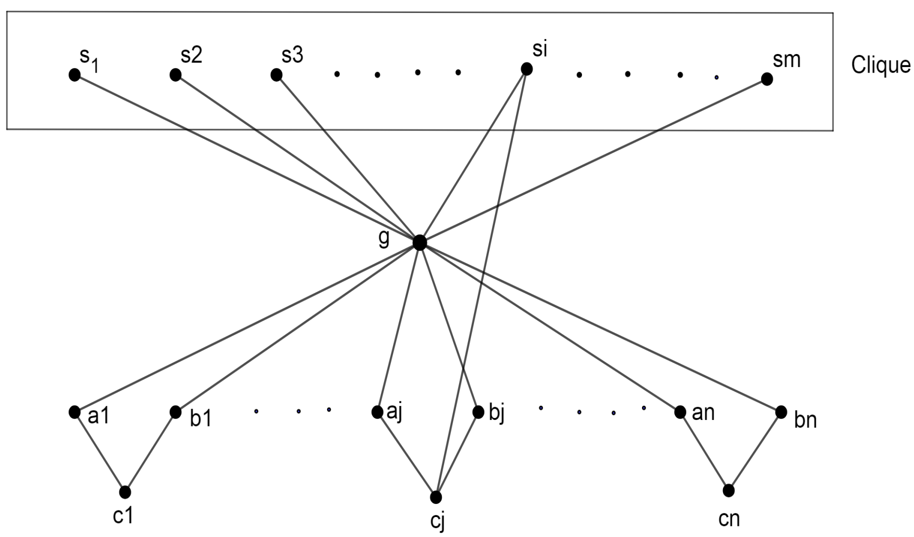


Figure 4.1: Construction for the reduction to an instance of C_4-free contraction.

Observation 4.3. *All the induced C_4 of the graph G are of the form $\{g, a_i, c_i, b_i\}$,$\{g, a_i, c_i, s_j\}$ or $\{g, b_i, c_i, s_j\}$.*

Proof. Since $G - \{g\}$ does not have any induced C_4 every induced C_4 must contain the vertex g. Let $T = \{s_i \; \forall \, 1 \leq i \leq\}$ be the vertex set $s_i \; \forall \, 1 \leq i \leq m$, $A = \{a_i \; \forall 1 \leq i \leq n\}$ and $B = \{b_i \; \forall \, 1 \leq i \leq n\}$ and $C = \{c_i \; \forall \, 1 \leq i \leq n\}$.

Suppose that there exists two vertices from T in the the cycle. Let the two vertices be s_i and s_j, they cannot be adjacent in the cycle since g is adjacent to both of them and hence does not form an induced C_4 and they cannot be diagonally opposite because they are adjacent. Therefore there can only exist at most one vertex form T in the induced C_4

First consider the cycle that does not have any vertex from T then the neighbours of g in the cycle will be from A and B. The vertices a_i and b_j have a common neighbour only if $i = j$ and no two vertices from A or B have common neighbours. Therefore, the only possibility of a cycle is $\{g, a_i, c_i, b_i\}$.

Now consider the cycle that has exactly one vertex from T say s_i, the neighbours of s_i other than $\{g\} \cup T$ are in C. Consider one such neighbour c_j, the only common neighbours of g and c_j are a_j and b_j. Therefore the only possible cycles are $\{g, a_i, c_i, s_j\}$ and $\{g, b_i, c_i, s_j\}$.

Solution Lifting Algorithm: Given a solution P of (G, k) the solution lifting algorithm constructs a solution F for $((S, U), k)$ as follows. For the edges of the form gs_i, and $c_j s_i$ add the set corresponding to $s_i \in S$ to F. If the edges are of the form ga_j, gb_j, $a_j c_j$, $b_j c_j$ add any set S_i such that $x_j \in S_i$. We will now argue that if G/P is C_4-free then, F is a set cover for (S, U). Suppose that there is an element x_j that has not been covered by any set in F, since the solution lifting algorithm would have added any set S_i such that $x_j \in S_i$ this implies P did not have any of the following edges ga_j, gb_j, $a_j c_j$, $b_j c_j$ and edge $x_j s_i$ for all $\{i : x_j \in S_i\}$. But if P did not have these edges in then the cycle $ga_i c_i b_i$ will be present in G/P, contradicting that G/P was C_4-free graph.

Corresponding to every edge in P we are adding one set to F so we obtain

$$\mathrm{SC}((S, U), k, F) = \mathrm{FC}(G, k, P) \tag{4.1}$$

Given an optimum solution F^* for $((S, U), k)$ for all $S_i \in F^*$ take the edges gs_i in P. Since F^* covers all the elements of U, contracting P will make kill all the induced cycles as c_i becomes adjacent with g. Hence G/P will be a C_4-free graph. Therefore we obtain the following

$$|F^*| = \mathrm{OPT}_{\mathrm{SC}}((S, U), k) = |P| \geq \mathrm{OPT}_{\mathrm{FC}}(G, k) \tag{4.2}$$

Combining 4.1 and 4.2 we get

$$\frac{\mathrm{SC}((S, U), k, F)}{\mathrm{OPT}_{\mathrm{SC}}((S, U), k)} \leq \frac{\mathrm{FC}(G, k, P)}{\mathrm{OPT}_{\mathrm{FC}}(G, k)}.$$

Hence this proves that that there is 1-approximate polynomial parameter transformation from SET COVER to C_4-FREE CONTRACTION.

Result:[10] For parameterized optimization problems A and B, if there is a α- approximate polynomial parameter transformation from A to B for some $\alpha \geq 1$ such that A does not admit approximate compression then B will also not admit any approximate compression. Using this result, the following corollary is the consequence of Theorem 4.2.1.

Corollary 4.3.1. C_4-FREE CONTRACTION *problem parameterized by solution size does not have any α-approximate compression of polynomial size for any $\alpha > 1$ unless* NP $\subseteq$ coNP/Poly.

4.3.1 Lower Bound for Chordal Contraction

Theorem 4.3.2. CHORDAL CONTRACTION *parameterized by solution size does not have any α-approximate compression of polynomial size for any $\alpha > 1$ unless* NP $\subseteq$ coNP/Poly.

Proof. The same reduction algorithm and the solution lifting algorithm given to prove 4.2.1 will show 1-approximate polynomial parameter transformation from SET COVER to CHORDAL CONTRACTION. The solution for CHORDAL CONTRACTION will be the same as C_4-FREE CONTRACTION, as the graph obtained after the contraction does not have any induced cycles of length greater than 3.

Existence of α-approximate kernel of polynomial size for any $\alpha > 1$ for CHORDAL CONTRACTION will also prove the existence of α-approximate compression of polynomial size for some $\alpha > 1$ for SET COVER problem parameterized by solution size, which is a contradiction. Hence for any $\alpha > 1$, CHORDAL CONTRACTIONparameterized by solution size does not admit an α-approximate kernel of polynomial size.

Theorem 4.2.1 can be generalised to obtain lower bound results for C_l-FREE CONTRACTION for $l \geq 4$.

Theorem 4.3.3. *For $l \geq 4$,* C_l-FREE CONTRACTION *problem parameterized by solution size does not have any α-approximate kernel of polynomial size for any $\alpha \geq 1$*

Proof. We give a 1-approximate polynomial parameter transformation from SET COVER problem. The reduction algorithm is slight modification to the reduction algorithm mentioned in proof of 4.2.1. Instead of adding three vertices corresponding to every element x_i

of universe add $l-1$ vertices. Add edges between these $l-1$ vertices and the vertex g to form a cycle of length l. Make s_i adjacent to $\lfloor \frac{l}{2} \rfloor^{th}$ vertex from g in cycle, whenever the corresponding element is in S_i. This completes the reduction algorithm. The solution lifting algorithm is same as before. For any edge in P which are not incident on $\lfloor \frac{l}{2} \rfloor^{th}$ vertex from g, say c_i, add edge $c_i s_j$ for any set S_j which contains x_i

Hence, existence of a α-approximate kernel of polynomial size for any $\alpha \geq 1$ will show the existence of α-approximate compression for set cover problem which is a contradiction.

Chapter 5

Clique Contraction

In this chapter we define the parameterized optimization version of Clique Contraction problem and present a lossy kernel.

5.1 Introduction

In the Clique Contraction problem, given a graph G one has to find set of minimum number of edges F, such that G/F is a clique. The problem of Clique Contraction parameterized by solution size is defined as follows

Clique Contraction	**Parameter:** k
Input: A graph G and an integer k	
Question: Is G k-contractible to a clique?	

Informally, the problem is to check if there exists $F \subseteq E(G)$ such that $|F| \leq k$ and G/F is a clique. Independent works of Lokshtanov et al. [9] and Cai and Guo [2], prove that there is no polynomial kernel with solution size as parameter. In the paper by Lokshtanov et al. [9], they have presented a kernel of size $\mathcal{O}(4^k k)$ and Cai and Guo [2] have given a FPT algorithm. We study this problem under Lossy kernel set up. Formally the parameterized version of Clique Contraction is defined as follows

$$\textsc{ClC}(G, k, F) = \begin{cases} \infty & if\ G/F\ is\ not\ a\ clique \\ \min\{|F|, k+1\} & \text{otherwise} \end{cases}$$

We prove that given a graph G on n vertices, an integer k and an approximation parameter $\alpha > 1$, there is a polynomial time running algorithm that outputs a graph G' with $\mathcal{O}(k^{d+1})$ vertices and an integer k' such that for every $c > 1$, a c-approximate solution for (G', k') can be lifted into a $(c\alpha)$-approximate solution for (G, k) in $n^{\mathcal{O}(1)}$. Here $d = \lceil \frac{1}{\sqrt{\alpha - 1}} \rceil$.

5.2 Lossy kernel for Clique Contraction

We consider that the input graph is connected, because if the graph is not connected it cannot be contracted into a clique. Contracting the set of edges corresponding to the spanning tree of a graph always gives a single vertex, which is also a clique. CLIQUE VERTEX DELETION problem is to find the set of vertices in graph G, such that deleting the vertices will give a clique. Recall that the complement of a graph G is represented by $\tilde{G}$.

Lemma 5.2.1. *If G is k-contractible to a clique, then there exists a vertex cover of size at most $2k$ for $\tilde{G}$.*

Proof. Let F be the set of edges such that G/F is a clique and $|F| \leq k$. Consider G/F-witness structure $\mathcal{W}$ of G and let the set of vertices from the big witness sets be X. From Proposition 2.1.1, we can deduce that $|X| \leq 2k$. Since G/F is a clique, any two vertices from $V(G) \setminus X$ are adjacent. And so $\tilde{G} \setminus X$ is an independent set, this implies that X is a vertex cover of $\tilde{G}$ of size at most $2k$.

Lemma 5.2.2. *For a graph G and an integer k, if there does not exist set of vertices X such that $|X| \leq 4k$ and $G \setminus X$ is a clique then* $\text{OPT}_{\text{CLC}}(G, k) = k + 1$

Proof. A 2-approximate solution for vertex cover can be found in polynomial time (refer to Section **??**). For a given graph G, find 2-approximate vertex cover for $\tilde{G}$, let this be denoted by X. Observe that X will also be a 2-approximate solution of CLIQUE VERTEX DELETION for graph G. From the Lemma **??** if G is k-contractible to a clique, then there exists a 2-approximate solution X for CLIQUE VERTEX DELETION problem such that $|X| \leq 4k$.
If there does not exist X such that $|X| \leq 4k$ and $G \setminus X$ is clique, it implies that G cannot be contracted into clique using at most k edges. From the definition of optimization problem, for any set of edges F if G/F is a clique then the maximum value of $\text{CLC}(G, k, F)$ is $k + 1$, since there is no F with $|F| \leq k$, $\text{OPT}_{\text{CLC}}(G, k)$ will be $k + 1$.

In order to prove the main result of this chapter, we will need some preprocessing on the input instance as the following.

For a given graph G, find 2-approximate vertex cover X for $\tilde{G}$. Let $V(G) \setminus X = Y$, this implies that $G[Y]$ is a clique. For an $\alpha > 1$, let $\beta = \sqrt{\alpha}$, choose d such that, $d = \lceil \frac{1}{\beta - 1} \rceil$. If $|X| > 4k$ return $\text{OPT}_{\text{CLC}}(G, k) = k + 1$(Lemma **??**) and if $|X| \leq 4k$ then apply the reduction algorithm $\mathcal{R}_{\mathcal{A}}$ mentioned below.

Reduction Algorithm $\mathcal{R}_{\mathcal{A}}$:
Marking Scheme 1: For every $A \subseteq X$ $|A| \leq d$, mark a vertex u in Y, such that all the vertices in A are adjacent to u. If there is more than one such vertex, then arbitrarily mark one vertex.
Marking Scheme 2: For every subset A of X whose size is at most d, mark $2k+1$ common non-neighbours of A in Y. If the set has more than $2k+1$ non-neighbours in Y, arbitrarily mark any $2k+1$ of them. If it has less than $2k+1$ non-neighbours then mark all of them.

Apply the Marking Schemes 1 and 2. Delete all the unmarked vertices of Y. Let $Y' \subseteq Y$ be the marked vertices and let G' be graph induced on $X \cup Y'$, then return (G', k) as the reduced instance.

Lemma 5.2.3. *For an instance (G, k) of clique contraction problem, let (G', k) be the instance given by $\mathcal{R}_{\mathcal{A}}$. There exists a solution lifting algorithm such that given G', F' with G'/F' being a clique and $|F'| \leq k$, the output is F such that G/F is a clique and $|F| \leq \beta \cdot |F'|$.*

Proof. In order to prove this Lemma first we will give the solution lifting algorithm. Let $\mathcal{W}'$ be G'/F'-witness structure of G'. We give witness structure $\mathcal{W}$ of G from $\mathcal{W}'$.

Solution lifting Algorithm $\mathcal{S}_{\mathcal{A}}$:
For every vertex $y \in Y \setminus Y'$ add singleton witness sets $\{y\}$ to $\mathcal{W}'$. If there is witness set of size $> d$ such that intersection with Y is empty, then add a vertex from Y. Now make F the union of edges corresponding to spanning trees of all the witness sets in W.

The solution lifting algorithm $\mathcal{S}_{\mathcal{A}}$ added singleton witness sets, which do not contribute to any edges. And when a vertex is added to the witness set, the number of edges will be increased by one. Since whenever the algorithm $\mathcal{S}_{\mathcal{A}}$ added a vertex the size of the witness set was at least d, we get $|F| \leq \dfrac{d+1}{d}.|F'| = \beta.|F'|$.

Now consider that G'/F' is clique then we will prove that G/F is a clique. The solution lifting algorithm added single ton sets $\{y\}$ for all $y \in Y \setminus Y'$ to $\mathcal{W}'$. Since $G[Y]$ induces a clique, every singleton witness set that has been added will be adjacent to all the witness sets of $\mathcal{W}'$ whose intersection with Y is non-empty. And the witness sets in W with size $> d$ has a vertex from Y hence will induce a clique with newly added sets.
Consider that there is a witness set $W(a)$ of size at most d that is not adjacent with a newly added vertex say v, since v was unmarked and is a common non-neighbour of vertices in $W(a)$, by Marking Scheme 2 $W(a)$ has $2k+1$ non-neighbours in Y'. According to the proposition 2.1.1, at least one of the $2k+1$ vertices remains as a singleton set in G'/F'. This implies that G'/F' is not a clique, which is a contradiction. This proves that every witness set of size at most d is also adjacent with every singleton set added. Hence we proved G/F is a clique.

Lemma 5.2.4. *For an instance (G, k) of clique contraction problem, let (G', k) be the instance given by $\mathcal{R}_{\mathcal{A}}$. If $\mathrm{OPT}_{\mathrm{CLC}}(G, k) \leq k$ then $\beta \cdot \mathrm{OPT}_{\mathrm{CLC}}(G, k) \geq \mathrm{OPT}_{\mathrm{CLC}}(G', k)$.*

Proof. Let F be a set of at most k edges in G such that $OPT(G, k) = ClC(G, k, F)$ and $\mathcal{W}$ be a G/F-witness structure of G. Since we are working with minimization problem, to prove the lemma it is sufficient to find a solution for G' which is of size $\beta \cdot |F|$. Recall that (X, Y) is a partition of $V(G)$ such that $G - X = G[Y]$ is a clique. Set of vertices marked by either of marking schemes is denoted by Y'.
At each step, we construct graph G^* from G by deleting one or more vertices in $Y \setminus Y'$. We also construct a set of edges F^* from F by replacing existing edges and/or adding extra edges to F. At any intermediate state, we ensure that G^*/F^* is a clique and the number of edges in F^* is at most $\beta \cdot |F|$.

To obtain G^* and F^*, we delete witness sets which are subsets of $Y \setminus Y'$ (Condition (1)) and modify the ones which intersect with $Y \setminus Y'$. Every witness set of later type intersects with Y' or X or both. We partition these big witness sets in $\mathcal{W}$ into two groups depending on whether they intersects X (Condition (2)) or not (Condition (3)). We modify witness sets which satisfy least indexed condition. If there exist no witness set which satisfy either of these three conditions then $Y \setminus Y'$ is an empty set and the lemma is vacuously true.

Condition (1): If there exists a witness set $W(t)$ in $\mathcal{W}$ which is a subset of $Y \setminus Y'$.

Construct G^* from G by deleting witness sets $W(t)$ in $\mathcal{W}$. Let $F^* = F$. Since a clique graph is closed under vertex deletion, G^*/F^* is a clique. We repeat this process until there exists a witness set which satisfy Condition (1).

At this stage we rename G^* to G and F^* to F for notational convenience.

Condition (2): If there exists a witness set $W(t)$ in $\mathcal{W}$ which contains vertices from $Y \setminus Y'$ but does not intersects X.

Note that $W(t)$ must intersects with Y'. Let the unmarked vertex i.e., vertex from $Y \setminus Y'$ be y. Let $W(a)$ be a witness set that contains a vertex from Y'. To obtain $\mathcal{W}^*$ merge the witness sets $W(t)$ and $W(a)$ and delete the vertex y. The merging can be done as there is at least one vertex from Y' in $W(t)$ that is adjacent to $W(a)$. $\mathcal{W}^*$ is a clique witness structure because all the witness sets except for $W(t)$ and $W(a)$ are unchanged and $W(a)$ was already adjacent to these witness sets so even after merging this set with $W(t) \setminus y$ the adjacency will remain. The size of F^* is same as that of F and G^*/F^* is a clique. We repeat this process until there exist a witness set which satisfy Condition (2).

At this stage we rename G^* to G and F^* to F for notational convenience.

Condition (3): If there exists a witness set $W(t)$ in $\mathcal{W}$ which contains vertices from $Y \setminus Y'$ and intersects X.

Let the vertex from $Y \setminus Y'$ in $W(t)$ be y. Partition $W(t) \setminus \{y\}$ into sets $W_1, W_2, \dots, W_p$ such that the number of vertices in W_i for $1 \leq i \leq p-1$, is exactly d and the number of vertices in W_p is at most d (Note that if the number of vertices is at most d then we have $p = 1$). Since y is unmarked, by Marking Scheme 1, there is a marked vertex, say y_i for $1 \leq i \leq p$, different from y that has been marked for each W_i. We assume that all vertices in $\{y_1, y_2, \dots, y_p\}$ are different to obtain the upper bound.

We construct F^* from F by following operation: Replace an edge xy in F by an edge xy_i for $1 \leq i \leq p-1$ and for every i such that $1 \leq i \leq p-2$ add an edge y_iy_{i+1}. We first argue about the cardinality of F^*. Note that we have added an extra edge corresponding to W_i for each $1 \leq i \leq p-1$. These sets are of size d. We did not add an extra edge corresponding to W_p whose cardinality may be smaller than d. This implies that we have added an extra edge for d edges in F. Moreover, since W_i's are pairwise disjoint, no edge in F can be part of two sets of edges corresponding to which new edge has been added. Hence size of F^* is at most $\frac{d+1}{d}|F| = \beta \cdot |F|$. We now argue that if G^* is obtained from G by deleting y then G^*/F^* is a clique. For $1 \leq i \leq p$, let $W(y_i)$ be the witness set containing y_i. Let Z be the union of $W(t) \setminus \{y\}$ and $W(y_i)$ for all $1 \leq i \leq p$. Let $\mathcal{W}^*$ be a witness structure of G^* obtained from $\mathcal{W}$ by removing $W(t), W(y_1), \ldots, W(y_t)$ and adding Z. Since all other witness sets remains same and we only replaced or added edges incident on vertices in $Z \cup \{y\}$, union of all spanning trees of witness sets in $\mathcal{W}^*$ is contained in F^*. Any two witness sets in $\mathcal{W}^*$ which are part of $\mathcal{W}$ are adjacent with each other. As Z contains $W(y_1)$, any witness set in $\mathcal{W}^*$ which is not contained in Z is adjacent with Z. Hence any two witness sets in $\mathcal{W}^*$ are adjacent with each other. This implies that G^*/F^* is a clique. We repeat this process until there exists a witness set which satisfy Condition (3). We argue that $|F^*| \leq \beta \cdot |F|$ even after repeating this process. Consider a witness set $W(t)$ in $\mathcal{W}$ which satisfy Condition (3) and which has been replaced by set Z. If Z does not intersect $Y \setminus Y'$ then it does not satisfy any condition and hence never been modified again. If it intersects $Y \setminus Y'$ then it also intersects Y' and hence satisfy Condition (2). This implies that any witness set in $\mathcal{W}$ is replaced by this process at most once. In other words, if an edge xy in F which has been replaced with edge xy_i before adding extra edge y_iy_{i+1} for some $1 \leq i \leq p$ then edge xy is never considered by the process again.

Any vertex in $Y \setminus Y'$ must be part of some some witness set in $\mathcal{W}$ and any witness set in $\mathcal{W}$ satisfies at least one of the conditions mentioned above. If there is no witness sets which satisfy any condition then $Y \setminus Y'$ is empty. This implies $G^* = G'$ and there exists a solution F^* of size $\beta \cdot |F|$. This concludes the proof of this lemma.

Theorem 5.2.5. *For any $\alpha > 1$,* Clique Contraction *parametrized by solution size admits α-approximate kernel of with $\mathcal{O}(k^{d+1})$ vertices.*

Proof. For an instance (G, k), and $\alpha > 1$, let $\beta = \sqrt{\alpha}$. We gave a reduction algorithm $\mathcal{R}_\mathcal{A}$ that outputs an instance (G', k) such that from the Lemma **??** we have

$$\beta \cdot \text{OPT}_{\text{CLC}}(G, k) \geq \text{OPT}_{\text{CLC}}(G', k). \tag{5.1}$$

And from Lemma 5.2.3, given a solution F' for (G', k), there is a solution lifting algorithm $\mathcal{S}_\mathcal{A}$ that outputs a solution F for (G, k), such that

$$\text{CLC}(G, k, F) = |F| \leq \beta \cdot |F'| = \beta \cdot \text{CLC}(G', k, F') \tag{5.2}$$

Combining 5.1 and 5.2 we get

$$\frac{\text{CLC}(G,k,F)}{\text{OPT}_{\text{CLC}}(G,k)} \leq \alpha \cdot \frac{\text{CLC}(G',k,F')}{\text{OPT}_{\text{CLC}}(G',k)}.$$

From the marking schemes mentioned in $\mathcal{R}_{\mathcal{A}}$, we get $|G'| = \sum_{1\leq i\leq d} \binom{4k}{i} \cdot (2k+2) = \mathcal{O}(k^{d+1})$ where $d = \lceil \frac{1}{(\beta-1)} \rceil$. Hence we proved the existence of α-approximate kernel of polynomial size for every $\alpha > 1$.

Chapter 6

Split Contraction

6.1 Introduction

Recall that a graph is called split graph if its vertices can be partitioned into two sets such that one set induces a clique and the other an independent set. A graph is said to be a split graph if it does not have any induced subgraphs that are isomorphic to C_4, C_5 or $2K_2$. In SPLIT CONTRACTION, given a graph G one has to find minimum number of edges whose contraction will result in split graph. In SPLIT CONTRACTION problem parameterized by solution size k, one has to check if the given graph G is k-contractible to split graph. Formally the parameterized problem is defined as follows.

SPLIT CONTRACTION	**Parameter:** k
Input: A graph G and an integer k	
Question: Is G k-contractible to a split graph?	

Even though the forbidden set of split graph is finite, SPLIT CONTRACTION is known to be W-hard. Saket et al.[1], proved that SPLIT CONTRACTION parameterized by solution size is $W[1]$-hard and when the parameter is vertex cover it is FPT. We study the problem in lossy kernel set up. Formally, the parameterized minimization of SPLIT CONTRACTION is defined as follows

$$\text{SC}(G, k, F) = \begin{cases} \infty & \text{if G/F is not a split graph} \\ \min\{|F|, k+1\} & \text{otherwise} \end{cases}$$

We will prove the existence of an approximate kernel for Split Contraction by taking two different set of parameters. First set of parameters are the size of the maximum independent set $|I|$ and solution size k and second set of the parameters are size of the maximum clique $|C|$ and solution size k.

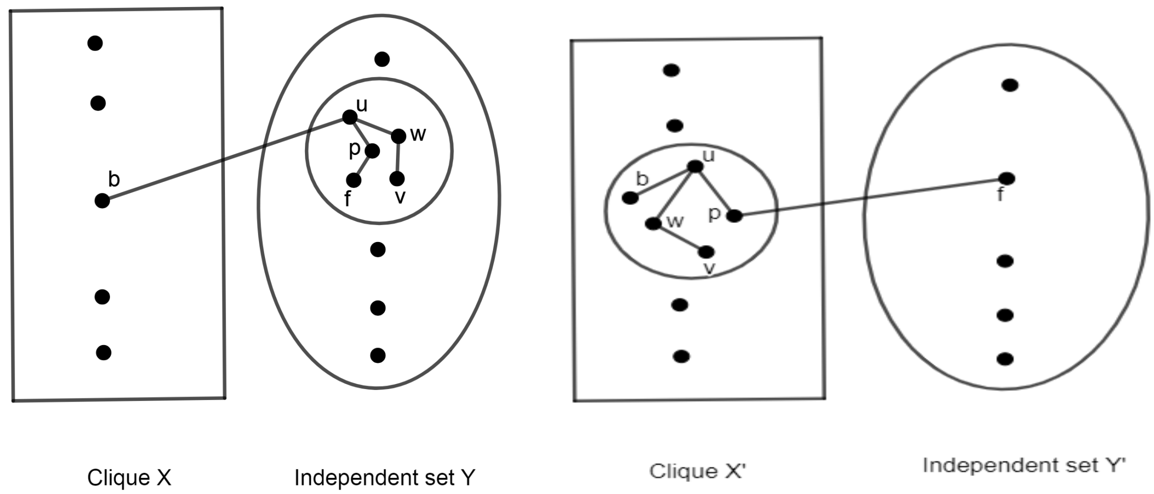
6.2 Lossy Kernel for Split Contraction

In this section we will see approximate kernels for Split Contraction for two different pairs of parameter. Before proving these results we will look at few observations that are crucial for the proofs.

For an instance (G, k) if $|G| \leq k+3$ then we can return the same instance as kernel, so for rest of this chapter we consider that size of the graph is at least $k+3$. For a graph G, Split Vertex Deletion problem is to find minimum set of vertices such that deletion of these vertices from the graph will result in a split graph. Since the size of forbidden set of split graph is upper bounded by 5, a 5-approximate solution for Split Vertex Deletion can be easily found in polynomial time.

Observation 6.3. *For a graph G, there exists an optimum solution for* Split Contraction *such that in the corresponding witness structure all the big witness sets will be in the partition that induces clique.*

Proof. Let F be a subset of edges such that G/F is a split graph and let $\mathcal{W}$ be a G/F-witness structure of G. Let X, Y be the partition of vertices of G/F such that X induces a clique and Y induces an independent set. Suppose there is a big witness set, say $W(a)$, corresponding to a vertex $a \in Y$. We will construct a new witness structure $\mathcal{W}^*$ by shifting few vertices of $W(a)$ to other witness sets. Since G is a connected graph there is at least one vertex, say $u \in W(a)$, such that u is adjacent to a witness set, say $W(b)$, for a vertex $b \in X$. There is a spanning tree T of $W(a)$ that is rooted at the vertex u, such that the edges of the trees are from F. Let f be a leaf vertex of T and p be its neighbour in T, then



(a) Split Graph with big witness set $W(a) = \{u, p, f, v, w\}$ on independent set side.

(b) Split Graph with big witness set $W^*(b) = \{b, u, p, v, w\}$.

Figure 6.1: Modification of the witness structure so that big witness sets lie on the clique.

delete the edge fp from F and add the edge ub to F. Refer to the Figure 6.1. Note that the number of edges in F^* and F are the same. Let the witness corresponding to vertices a, b in $\mathcal{W}^*$ be $W^*(a)$ and $W^*(b)$ respectively. Now we will argue that the graph G/F^* is also a split graph. The witness sets $W^*(b)$ and $W^*(a)$ will still be part of clique and independent sets respectively and other witness sets are not disturbed G^*/F^* be a split graph.

Lemma 6.3.1. *If a graph G is k-contractible to a split graph then G has a solution for* Split Vertex Deletion *of size at most $10k$.*

Proof. For the graph G, let F be the set of edges such that G/F is split graph and $|F| \leq k$ and let $\mathcal{W}$ be G/F-witness structure of G. If X is the union of all the vertices that are part of big witness sets then from Proposition 2.1.1 we get $|X| \leq 2k$. Note that X will be solution for Split Vertex deletion of G because even after deleting these witness sets the remaining graph will still induce a split graph. The existence of $2k$ size solution implies there exists 5-approximate solution for Split Vertex Deletion of size at most $10k$.

Lemma 6.3.2. *For an instance (G, k) of* Split Contraction, *if a 5-approximate solution for* Split Vertex Deletion *is not less than $10k$ then $\text{Opt}_{\text{SC}}(G, k) = k + 1$.*

Proof. From the definition of Split contraction, the maximum possible value any valid solution can have is $k + 1$. A graph always has a trivial solution of contracting all the edges of spanning tree. Therefore there always exists a solution for Split contraction of G whose value is $k+1$. From Lemma 6.3.1 if 5-approximate solution for Split Vertex Deletion is not less than $10k$, then G is not k-contractible to split graph. Therefore the value of the optimum solution will be $k+1$

6.3.1 Parameterized by solution size and independent set size

Given an instance (G, k) of Split Contraction, we will first do some preprocessing on the instance. Find 5-approximate solution of the instance for Split Vertex Deletion then from Lemma 6.3.2, if the solution is greater than $10k$ return $\text{Opt}_{\text{SC}}(G, k) = k+1$. Let S be the 5-approximate solution of Split Vertex Deletion such that $|S| \leq 10k$ and let $G \setminus S = (X, Y)$ such that $G[X]$ induces a clique and $G[Y]$ induces an independent set. For an $\alpha > 1$, set $\beta = \sqrt{\alpha}$ and $d = \lceil \frac{1}{\beta - 1} \rceil$ and apply the following reduction algorithm.

Reduction Algorithm $\mathcal{SCR}_{\mathcal{A}}$:
Marking Scheme 3: For every $A \subseteq S \cup Y$ such that $|A| \leq d$, mark a vertex say $x \in X$ such that $A \subseteq N(x)$ is connected. If there is more than one such vertex, then arbitrarily mark one. If there does not exits such a vertex x then do not mark anything.
Marking Scheme 4: For every $A \subseteq S \cup Y$ such that $|A| \leq d$, mark $2k+2$ common non-neighbours of A in X. If there are more than $2k+2$ common non-neighbours then mark any $2k+2$ of them, if the set has less than $2k+2$ common non-neighbours then mark all of them.
Let the vertices that have been marked by either of these two marking schemes be X'. Delete all the vertices from $X \setminus X'$, and return (G', k) as the reduced instance where $G' = G \setminus (X \setminus X')$.

Lemma 6.3.3. *For an instance (G, k) of* Split Contraction *problem, let (G', k) be the instance given by the reduction algorithm $\mathcal{SCR}_{\mathcal{A}}$. There exists a solution lifting algorithm such that given G', F' with G'/F' is a split graph and $|F'| \leq k$, the output is F such that G/F is a split graph and $|F| \leq \beta \cdot |F'|$.*

Proof. In order to prove this Lemma first we will give the solution lifting algorithm. Let $\mathcal{W}'$ be G'/F'-witness structure of G'. We give witness structure $\mathcal{W}$ of G from $\mathcal{W}'$.

Solution lifting Algorithm $\mathcal{SCS}_\mathcal{A}$:
For every vertex $x \in X \setminus X'$ add singleton witness sets $\{x\}$ to $\mathcal{W}'$. If there is witness set of size greater than d such that the intersection with X is empty, then add one neighbouring vertex from X' to that witness set. Now make F the union of edges corresponding to spanning trees of all the witness sets in W.

The solution lifting algorithm $\mathcal{SCS}_\mathcal{A}$ added singleton witness sets, which do not contribute to any edges. And when a vertex is added to the witness set, the number of edges will be increased by one. Since whenever the algorithm $\mathcal{SCS}_\mathcal{A}$ added a vertex the size of the witness set was at least d, we get $|F| \leq \dfrac{d+1}{d} \cdot F'| = \beta \cdot |F'|$.

Now consider that G'/F' is split graph then we will prove that G/F is a split graph. Let (P, Q) be the partition of vertices in G'/F' such that P induces a clique and Q induces independent set. From Lemma 6.3 we can assume that all the big witness sets correspond to vertices from P alone, this implies that at most one vertex from X' can be in Q. In order to prove that G/F is a split graph, it is enough to show that vertices in $PU(X \setminus X')$ induces a clique in G/F.

The solution lifting algorithm $\mathcal{SCS}_\mathcal{A}$ made sure that every witness set of size greater than d has a vertex from X'. So every vertex in $X \setminus X'$ will be adjacent to every witness set of size greater than d as $G[X]$ induced a clique in G.

Now consider that there exists a witness set $W(a)$ of size at most d in P such that it is not adjacent with a vertex say $x \in X \setminus X'$. This implies that $W(a)$ has vertices only from $S \cup Y$ and Marking Scheme 4 has marked $2k+2$ common non-neighbours of $W(a)$. From Proposition 2.1.1 at least two of these marked vertices say v, w will remain as a singleton witness set and at most one of the vertex say u can go to Q. This implies v is in P, but this is a contradiction as we assumed that P induces a clique. Hence we proved that every witness set of size at most d in P will be adjacent to every vertex $x \in X \setminus X'$.

Lemma 6.3.4. *For an instance (G, k) of* Split Contraction *problem, let (G', k) be the instance given by $\mathcal{SCR}_\mathcal{A}$. If $\mathrm{OPT}_{\mathrm{SC}}(G, k) \leq k$ then $\beta \cdot \mathrm{OPT}_{\mathrm{SC}}(G, k) \geq \mathrm{OPT}_{\mathrm{SC}}(G', k)$.*

Proof. Let F be a set of at most k edges in G such that $\mathrm{OPT}(G, k) = \mathrm{SC}(G, k, F)$ and $\mathcal{W}$ be a G/F-witness structure of G. Since we are working with minimization problem, to prove the lemma it is sufficient to find a solution for G' which is of size $\beta \cdot |F|$. Recall that (S, X, Y) is a partition of $V(G)$ such that $G - S = G[X \cup Y]$ is a split graph with X being a clique and Y an independent set. Set of vertices marked by either of marking schemes was denoted by X'.

At each step, we construct graph G^* from G by deleting one or more vertices in $X \setminus X'$. We also construct a set of edges F^* from F by replacing existing edges and/or adding extra edges to F. At any intermediate state, we ensure that G^*/F^* is a split graph and the number of edges in F^* is at most $\beta \cdot |F|$.

To obtain G^* and F^*, we delete witness sets which are subsets of $X \setminus X'$ (Condition (1)) and modify the ones which intersect with $X \setminus X'$. Every witness set of later type intersects with X' or $Y \cup S$ or both. We partition these big witness sets in $\mathcal{W}$ into two groups depending on whether they intersects X' (Condition (2)) or not (Condition (3)). We modify witness sets which satisfy least indexed condition. If there exist no witness set which satisfy either of these three conditions then $X \setminus X'$ is an empty set and the lemma is vacuously true.

Condition (1): If there exists a witness set $W(t)$ in $\mathcal{W}$ which is a subset of $X \setminus X'$.

Construct G^* from G by deleting witness sets $W(t)$ in $\mathcal{W}$. Let $F^* = F$. Since a split graph is closed under vertex deletion, G^*/F^* is a split graph. We repeat this process until there exists a witness set which satisfy Condition (1).

At this stage we rename G^* to G and F^* to F for notational convenience.

Condition (2): If there exists a witness set $W(t)$ in $\mathcal{W}$ which contains vertices from $X \setminus X'$ that does not intersects $S \cup Y$.

Note that $W(t)$ must intersects with X'. Let the vertex from $X \setminus X'$ in $W(t)$ be x. Let $W(a)$ be a witness set on the clique side that contains vertices(vertex) from X'. To obtain $\mathcal{W}^*$ merge the witness sets $W(t)$ and $W(a)$ and delete the vertex x. The merging can be done as there is at least one vertex from X' in $W(t)$ that is adjacent to $W(a)$. $\mathcal{W}^*$ is a split graph witness structure because all the witness sets except for $W(t)$ and $W(a)$ are unchanged and $W(a)$ was already adjacent to all the witness sets of clique, so even after merging this set with $W(t) \setminus \{x\}$ the adjacency with the clique vertices will remain. The size of F^* is same as that of F and G^*/F^* is a split graph. We repeat this process until there exist a witness set which satisfy Condition (2).

At this stage we rename G^* to G and F^* to F for notational convenience.

Condition (3): There exists a witness set $W(t)$ in $\mathcal{W}$ which contains vertices from $X \setminus X'$

and intersects $S \cup Y$.

Let the vertex from $X \setminus X'$ in $W(t)$ be x. Partition $W(t) \setminus \{x\}$ into sets $W_1, W_2, \ldots, W_p$ such that the number of vertices in W_i for $1 \leq i \leq p-1$, is exactly d and the number of vertices in W_p is at most d (Note that if the number of vertices is at most d then we have $p = 1$). Since x is unmarked, there is a marked vertex, say x_i for every $1 \leq i \leq p$, different from x that has been marked for each W_i by Marking Scheme 3. We assume that all vertices in $\{x_1, x_2, \ldots, x_p\}$ are different to obtain the upper bound.

We construct F^* from F by following operation: Replace an edge ux in F by an edge ux_i for every $1 \leq i \leq p$ and for every i such that $1 \leq i \leq p-1$ add an edge $x_i x_{i+1}$. We first argue about the cardinality of F^*. Note that we have added an extra edge corresponding to W_i for each $1 \leq i \leq p-1$. These sets are of size d. We did not add an extra edge corresponding to W_p whose cardinality might be smaller than d. This implies that we have added an extra edge for d edges in F. Moreover, since W_i's are pairwise disjoint, no edge in F can be part of two sets of edges corresponding to which new edge has been added. Hence size of F^* is at most $\frac{d+1}{d}|F| = \beta \cdot |F|$.

We now argue that if G^* is obtained from G by deleting x then G^*/F^* is a split. For $1 \leq i \leq p$, let $W(x_i)$ be the witness set containing x_i. Let Z be the union of $W(t) \setminus \{x\}$ and $W(x_i)$ for all $1 \leq i \leq p$. Let $\mathcal{W}^*$ be a witness structure of G^* obtained from $\mathcal{W}$ by removing $W(t), W(x_1), \ldots, W(x_p)$ and adding Z. Since all other witness sets remains same and we only replaced or added edges incident on vertices in $Z \cup \{x\}$, union of all spanning trees of witness sets in $\mathcal{W}^*$ is contained in F^*. Witness sets that induced the independent set of the split graph were unaffected. If we prove that the witness set Z is adjacent to all the witness sets that were inducing clique of the split graph then the proof will be done. Since Z has the witness sets $W(x_i)$, Z will be adjacent to all the witness sets that were part of clique. Hence we proved that G^*/F^* is a split graph.

We repeat this process until there exists a witness set which satisfy Condition (3). We argue that $|F^*| \leq \beta \cdot |F|$ even after repeating this process. Consider a witness set $W(t)$ in $\mathcal{W}$ which satisfy Condition (3) and which has been replaced by set Z. If Z does not intersect $X \setminus X'$ then it does not satisfy any condition and hence will never be modified again. If it intersects $X \setminus X'$ then it also intersects X' and hence satisfy Condition (2). This implies that any witness set in $\mathcal{W}$ is replaced by this process at most once. In other words, if an edge xy in F which has been replaced with edge ux_i before adding extra edge $x_i x_{i+1}$ for some $1 \leq i \leq p$ then edge ux is never considered by the process again.

Any vertex in $X \setminus X'$ must be part of some some witness set in $\mathcal{W}$ and any witness set in $\mathcal{W}$ satisfies at least one of the conditions mentioned above. If there is no witness set which satisfy any condition then $X \setminus X'$ is empty. This implies $G^* = G'$ and there exists a solution F^* of size $\beta \cdot |F|$. This concludes the proof of the lemma.

Theorem 6.3.5. *For any $\alpha > 1$, SPLIT CONTRACTION paramaterized by solution size k and maximum independent set size I admits α-approximate kernel with $\mathcal{O}((k + |I|)^{d+1})$ vertices where $d = \lceil \frac{1}{(\beta-1)} \rceil$.*

Proof. For an instance (G, k), and $\alpha > 1$, let $\beta = \sqrt{\alpha}$. We gave a reduction algorithm $\mathcal{SCR}_{\mathcal{A}}$ that outputs an instance (G', k) such that from the Lemma 6.3.4 we have

$$\beta.\text{OPT}_{\text{SC}}(G, k) \geq \text{OPT}_{\text{SC}}(G', k). \tag{6.1}$$

And from Lemma 6.3.3, given a solution F' for (G', k), there is a solution lifting algorithm $\mathcal{SCS}_{\mathcal{A}}$ that outputs a solution F for (G, k), such that

$$\text{SC}(G, k, F) = |F| \leq \beta \cdot |F'| = \beta \cdot \text{SC}(G', k, F') \tag{6.2}$$

Combining 6.1 and 6.2 we get

$$\frac{\text{SC}(G, k, F)}{\text{OPT}_{\text{SC}}(G, k)} \leq \alpha \cdot \frac{\text{SC}(G', k, F')}{\text{OPT}_{\text{SC}}(G', k)}.$$

From the marking schemes mentioned in $\mathcal{SCR}_{\mathcal{A}}$, we get $|V(G')| \leq |I| + 10k + \sum_{1 \leq i \leq d} \binom{10k+|I|}{i} \cdot (2k + 3) = \mathcal{O}(k^{d+1})$ where $d = \lceil \frac{1}{(\beta-1)} \rceil$ and $|I|$ is the size of maximum independent set of G. Hence we proved the existence of α-approximate kernel of polynomial size for every $\alpha > 1$.

Chapter 7

Conclusion

In this thesis we have proved that Clique Contraction parameterized by solution size admits polynomial a approximate kernel, where as Chordal Contraction parameterized by solution size do not admit any polynomial approximate kernel. We have also shown approximate kernels for Split Contraction when the parameters are maximum independent size and solution size. One can try to find approximate kernel for Split Contraction parameterized by solution size alone. As clique graphs are contained in split graphs and split graph are contained in chordal graphs. It will be very interesting if one can utilize the structure of the graph, generalize the results of these problems to differentiate all the graph contraction problems that admit approximate kernel of polynomial size.

Bibliography

[1] Akanksha Agrawal, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Split contraction: The untold story. In *STACS*, volume 66 of *LIPIcs*, pages 5:1–5:14, 2017.

[2] Leizhen Cai and Chengwei Guo. Contracting few edges to remove forbidden induced subgraphs. In *Parameterized and Exact Computation: 8th International Symposium, IPEC*, pages 97–109. Springer International Publishing, 2013.

[3] Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75(1):118–137, 2016.

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009.

[5] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer-Verlag, 2015.

[6] Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, and Yngve Villanger. Tight bounds for parameterized complexity of cluster editing with a small number of clusters. *Journal of Computer and System Sciences*, 80(7):1430–1447, 2014.

[7] Fedor V. Fomin and Yngve Villanger. Subexponential parameterized algorithm for minimum fill-in. *SIAM Journal on Computing*, 42(6):2197–2216, 2013.

[8] Esha Ghosh, Sudeshna Kolay, Mrinal Kumar, Pranabendu Misra, Fahad Panolan, Ashutosh Rai, and M. S. Ramanujan. Faster parameterized algorithms for deletion to split graphs. *Algorithmica*, 71(4):989–1006, 2015.

[9] Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. On the hardness of eliminating small induced subgraphs by contracting edges. In *IPEC*, pages 243–254, 2013.

[10] Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Lossy kernelization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*, pages 224–237, 2017.

[11] Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010.

[12] Toshimasa Watanabe, Tadashi Ae, and Akira Nakamura. On the removal of forbidden graphs by edge-deletion or by edge-contraction. *Discrete Applied Mathematics*, 3(2):151–153, 1981.

[13] Toshimasa Watanabe, Tadashi Ae, and Akira Nakamura. On the np-hardness of edge-deletion and -contraction problems. *Discrete Applied Mathematics*, 6(1):63–78, 1983.