

PARAMETERIZED TRACTABILITY AND
KERNELIZATION
OF PROBLEMS ON UNIT DISK GRAPHS

by

Navid Imani

M.Sc., Simon Fraser University, 2008

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

© Navid Imani 2013

SIMON FRASER UNIVERSITY

Fall 2013

All rights reserved.

However, in accordance with the *Copyright Act of Canada*, this work may be reproduced without authorization under the conditions for “Fair Dealing.” Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.

APPROVAL

Name: Navid Imani
Degree: Doctor of Philosophy
Title of Thesis: Parameterized Tractability and Kernelization of Problems on Unit Disk Graphs

Examining Committee: Dr. Ramesh Krishnamurti
Chair

Dr. Qian-Ping Gu, Professor, Computing Science
Simon Fraser University
Senior Supervisor

Dr. Pavol Hell, Professor, Computing Science
Simon Fraser University
Supervisor

Dr. Jiangchuan Liu, Associate Professor, Computing
Science
Simon Fraser University
SFU Examiner

Dr. Guochuan Zhang, Professor of Computer Science,
Zhejiang University
External Examiner

Date Approved: _____

Partial Copyright Licence



The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the non-exclusive, royalty-free right to include a digital copy of this thesis, project or extended essay[s] and associated supplemental files ("Work") (title[s] below) in Summit, the Institutional Research Repository at SFU. SFU may also make copies of the Work for purposes of a scholarly or research nature; for users of the SFU Library; or in response to a request from another library, or educational institution, on SFU's own behalf or for one of its users. Distribution may be in any form.

The author has further agreed that SFU may keep more than one copy of the Work for purposes of back-up and security; and that SFU may, without changing the content, translate, if technically possible, the Work to any medium or format for the purpose of preserving the Work and facilitating the exercise of SFU's rights under this licence.

It is understood that copying, publication, or public performance of the Work for commercial purposes shall not be allowed without the author's written permission.

While granting the above uses to SFU, the author retains copyright ownership and moral rights in the Work, and may deal with the copyright in the Work in any way consistent with the terms of this licence, including the right to change the Work for subsequent purposes, including editing and publishing the Work in whole or in part, and licensing the content to other parties as the author may desire.

The author represents and warrants that he/she has the right to grant the rights contained in this licence and that the Work does not, to the best of the author's knowledge, infringe upon anyone's copyright. The author has obtained written copyright permission, where required, for the use of any third-party copyrighted material contained in the Work. The author represents and warrants that the Work is his/her own original work and that he/she has not previously assigned or relinquished the rights conferred in this licence.

Simon Fraser University Library
Burnaby, British Columbia, Canada

revised Fall 2013

Abstract

Unit disk graphs are used extensively in the field of networks in order to model the infrastructure of ad hoc wireless communication networks. Development of efficient algorithms for problems on unit disk graphs has therefore been a hot topic largely driven by the practical applications. This is while there are no general frameworks known in the literature for obtaining parameterized algorithms or kernelization results on UDGs. CLIQUE PARTITION is one of the Richard Karp's original 21 NP-hard problems and is proved to be an extremely useful tool for obtaining solutions for other fundamental combinatorial problems such as DOMINATING SET, UDG REALIZATION and FACILITY LOCATION. We make the following contributions in this thesis: (1) We develop a novel framework for obtaining parameterized and FPT algorithms for CLIQUE PARTITION and related problems on UDGs. (2) We bridge the complexity gap between the UDG and precision UDG classes through introducing a new useful subclass, namely quasi-precision UDGs, with interesting properties. We further describe (relaxed)-quasi-precision UDGs by proving structural properties for the subclass and classify the cases when the problems do not admit FPT algorithms under our framework. (3) We propose a new approach for constructing polynomial approximation schemes (PTAS) for MINIMUM CLIQUE PARTITION on UDGs which results in significant computational speed-up comparing to the best previously known PTAS for the problem. (4) We introduce first-time data reduction rules for the problems of CLIQUE PARTITION and WEIGHTED CLIQUE PARTITION and enhanced data reduction rules for CLIQUE COVER on UDGs. (5) We develop a framework for obtaining linear-size kernels on quasi-precision UDGs and provide the sufficient criteria for the problems under which the framework applies.

To My Parents – Soheila & Hooshang.

Acknowledgments

I would like to express my gratitude to my senior supervisor Dr. Qian-ping Gu. Without his support and guidance, this thesis would have not been possible. I would like to thank Dr. Pavol Hell for being the supervisor during my Ph.D. studies and for his many precious remarks on my thesis. My gratitude also goes to Dr. Jiangchuan Liu for acting as internal examiner for my Ph.D. defence and to Dr. Guochuan Zhang for being the external examiner of my defence.

Contents

Approval	ii
Partial Copyright License	iii
Abstract	iv
Dedication	v
Acknowledgments	vi
Contents	vii
List of Figures	xi
1 Introduction	1
1.1 Parameterized Algorithms on UDGs	2
1.2 Contributions of the Thesis	2
1.2.1 A Parameterization Framework for CLIQUE PARTITION and Related Problems on UDGs	3
1.2.2 Improved Polynomial Approximation Scheme for MINIMUM CLIQUE PARTITION	5
1.2.3 A Kernelization Framework for Problems on Quasi-precision UDGs . .	6
1.3 Thesis Outline	6
2 Background and Related Work	8
2.1 Definitions and Notations	8
2.2 Data Reduction	12

2.2.1	VERTEX COVER and Related Problems	12
2.2.2	DOMINATING SET	13
2.2.3	CONNECTED DOMINATING SET	14
2.3	Separators, Minors and Bidimensionality	14
2.3.1	Background	14
2.3.2	Bidimensionality Framework	15
2.3.3	Contraction Decomposition	16
2.3.4	(Meta)-Kernelization	20
2.3.5	Kernels for Problems on H -Minor-Free Graphs	22
2.4	Kernels for Degenerate Graphs and UDGs	24
2.4.1	Background	24
2.4.2	Degenerate Graphs	25
2.4.3	Unit Disk Graphs	29
3	Algorithms for CLIQUE PARTITION on UDGs	36
3.1	Introduction	36
3.2	Definitions and Notations	37
3.3	Arbitrary UDGs	38
3.3.1	A Parameterized Algorithm	47
3.4	Data Reduction	49
3.5	Summary	53
4	FPT Algorithms on Precision & Quasi-precision UDGs	54
4.1	Introduction	54
4.2	Precision UDGs	55
4.2.1	An FPT Algorithm on Precision UDGs	61
4.3	Quasi-precision UDGs	62
4.3.1	An FPT Algorithm on Quasi-precision UDGs	68
4.4	Structure of Quasi-precision UDGs	68
4.5	Summary	74
5	Improved PTAS for MINIMUM CLIQUE PARTITION	75
5.1	Introduction	75
5.2	Plane Decomposition	76

5.3	Exact Solution for UDG with Restricted Geometry	79
5.3.1	Dumitrescu and Pach's Approach	79
5.3.2	Using Convex Regions	82
5.4	Summary	84
6	Parameterized Algorithms for Other Problems	85
6.1	CLIQUE COVER	85
6.1.1	Data Reduction	86
6.1.2	A Parameterized Algorithm on Arbitrary UDGs	89
6.1.3	Analysis	89
6.1.4	Fixed-parameter Algorithms on Precision and Quasi-precision UDGs	92
6.2	WEIGHTED CLIQUE PARTITION	93
6.2.1	Data Reduction	95
6.2.2	Analysis	97
6.2.3	Fixed-parameter Algorithms	98
6.3	Summary and Future Work	99
7	Kernelization Framework on Quasi-precision UDGs	101
7.1	Introduction	101
7.2	Data Reduction	102
7.3	Analysis	104
7.4	Application	106
7.4.1	DOMINATING SET	106
7.4.2	CONNECTED DOMINATING SET	107
7.4.3	INDEPENDENT SET	108
7.4.4	CLIQUE COVER	109
7.4.5	CLIQUE PARTITION	110
7.5	Summary	113
8	Conclusion and Future Work	114
8.1	Summary of Contributions	114
8.2	Future Work	116

Appendix A Proofs	118
A.1 Proof of Formula (3.3) for calculating $ q_{xz}q_{yz} $	118
Bibliography	120
Index	130

List of Figures

Chapter 2

2.1	Two disks of radius 1 centered at points u and v . The area where the vertices of $\Lambda_{v,w}$ appear is highlighted in darker color.	31
2.2	Two alternative unit disk representations of K_3 . The representation on the left does not satisfy the Helly property.	32

Chapter 3

3.1	An arbitrary UDG on 4 vertices, its corresponding unit disk representation and region coverage graph.	39
3.2	An arbitrary placement of disks ζ_x, ζ_y and ζ_z with centers x, y and z such that Helly property does not hold. Points q_{xy}, q_{yz} and q_{xz} are marked on the boundary of $\Omega = \zeta_x \cap \zeta_y \cap \zeta_z$	42
3.3	Coverage of a square of length $\sqrt{2}$, denoted $S_{\sqrt{2}}$, by 4 disks of radius $1/2$. Line segments L_u and L_l have the length $\sqrt{3}$	43
3.4	Points $\{p_1, \dots, p_4\}$ are originally placed at the four corners of an axis-aligned $1/2 \times \sqrt{3}/2$ square (drawn in dashed black). The corresponding disks cover an area including a $1 \times \sqrt{3}$ rectangle (drawn in solid blue). As $ q_{xy}q_{xz} $ decreases, the horizontally aligned pairs (also the vertically aligned pairs) of points are shifted $\epsilon/2$ units towards (respectively $\epsilon'/2$ units away from) each other horizontally (respectively vertically), such that their corresponding disks cover an area including a $(1 + 2\epsilon') \times (\sqrt{3} + 2\epsilon)$ rectangle (drawn in solid red). . . .	46

Chapter 4

4.1 A circle of radius $\frac{1}{2} + \epsilon$ and a set of points distributed inside the circle close to its boundary. Every pair of points that form an independent set in UDG are connected with a dashed blue line in the illustration. 56

4.2 \mathcal{D}_δ is depicted as a disk with red boundary in the center of the figure. The filled ring around \mathcal{D}_δ shows annulus Ψ . The dashed red line in Ψ is the possible location of points for a scenario with maximum number of regions in \mathcal{D}_δ 57

4.3 A hexagonal mesh is placed over the plane. Hexagons with face distance 1, 2 and 3 from p_0 are colored red, green and blue respectively. 59

4.4 A possible decomposition of the points of a UDG to dense islands. The respective graph is (λ, α) -quasi-precision for some value of λ and α 63

4.5 Graph $K_{i,i}$ is depicted on the left side. The edges highlighted in red form a perfect matching in $K_{i,i}$. Clique-prism \bar{L}_i corresponding to the selected matching in $K_{i,i}$ is pictured on the right side. 72

Chapter 5

5.1 A hexagonal mesh with edge length 1/2 is placed over a circle of radius 3. Every such a hexagon can be inscribed in a circle of radius 1/2. 80

Chapter 6

6.1 A weighted graph on n vertices consisting of two cliques and a perfect matching. Choosing the two cliques of size $n/2$ as the partition would results in a solution of weight $2^{\frac{n}{2}+1}$ while the optimal solution for Weighted Clique Partition is the set of $n/2$ matchings. 94

Chapter 1

Introduction

Computation is an inevitable part of nowadays technological artifacts. Although the use of algorithms as systematic ways of performing a series of actions has been known long before the invention of electronic devices, the formal design of algorithms became prominently inevitable with the development of theory of computation.

The classical theory of complexity attributes the hardness of a problem as a function of its input size only. Hence, under the assumption that $P \neq NP$, there are numerous natural problems for which super-polynomial running-time is inevitable when complexity is measured in terms of the input size. It is while some inputs for such problems consist of sections that are easy to deal with and those that are more difficult. This sometimes makes the problem computable in a time that is polynomial in the input size and exponential or worse in some parameter k . Hence, if k is fixed at a small value and the growth of the function over k is relatively small then such problems can still be considered “tractable” despite their traditional classification as “intractable”. Parameterized complexity hence can be regarded as two-dimensional generalization of P vs. NP where, while estimating the running-time in addition to the overall input size n , the effects of a secondary measurement that captures additional problem-relevant information, is also taken into account.

An effective approach in fixed-parameter algorithmics is that before starting a cost-intensive exact algorithm to solve a fixed-parameter tractable problem characterized by a parameter k , a polynomial-time pre-processing data reduction phase is executed to shrink the input data of size n to a smaller instance. This is done through applying a set of reduction rules

to the input instance. The solution for the original input then can be reconstructed in polynomial time in n using a solution for the shrunk instance. The shrunk instance is called *problem kernel*. It is then hoped that the size of the problem kernel is upper-bounded by a polynomial in k , independent of n . The process is then called kernelization.

1.1 Parameterized Algorithms on UDGs

Unit disk graphs have been used extensively in the field of networks in order to model the infrastructure of ad hoc wireless communication networks. In this scenario, node locations are modeled as points in the plane, and the area within which a signal from one node can be received by other nodes is modeled as a circle. Driven by the applications, the design of practical algorithms on UDGs is of paramount importance in real-life applications.

However, the unit disk graph setting is often too relaxed to admit straight-forward algorithmic results. For example, the size of cliques (and hence the Euler genus) in UDGs is not bounded and therefore many of the techniques for bounded genus and H -minor-free graphs cannot be directly applied to UDGs. In reality, there are no general frameworks known in the literature to obtain kernelization or to design parameterized algorithms for problems on UDGs. However, one might exploit the specific sparsity properties to obtain improved results for specific problems.

We try to fill this gap in research by introducing novel techniques and frameworks that can be adopted to obtain parameterized algorithms and kernelization results for a range of problems on UDGs. It is however important to notice that for many classical NP-hard problems, the class of unit disk graphs is still too “wild” to admit kernelization or FPT results [9]. Therefore, in an attempt to explore the boundaries of fixed-parameter tractability, it deems necessary to restrict our focus to subclasses of UDGs that exhibit more attractive geometrical properties. Yet, to adhere to generality, we require them to include the whole class of UDGs when the choice of the parameter is unrestricted.

1.2 Contributions of the Thesis

In this section, we provide an outline of the problems that are tackled in this thesis. For each problem considered here, we point out the state-of-the-art results and our contribution in that context. In what follows, we assume that all input graphs that are fed to the algorithms

are simple (without self-loops or multi-edges) and undirected and are assumed to be on n vertices and m edges. We refer to the parameter in the decision version (of the standard parameterization) of the problem by k . When specified as a unit disk graph, we assume that the input is provided as a set of points (centers of disks) in the plane described by their Cartesian coordinates.

1.2.1 A Parameterization Framework for CLIQUE PARTITION and Related Problems on UDGs

We introduce a framework for parameterization of clique covering problems¹ on UDGs. CLIQUE PARTITION asks whether the vertices of a graph can be partitioned into a certain number of cliques.

1) Our first exact parameterized algorithm for CLIQUE PARTITION on arbitrary UDGs relies on a novel idea and a geometric theorem that demonstrates the use of convex regions for guessing clique partitions. This results in an algorithm of running-time $\mathcal{O}(n^{6k+2})$ on arbitrary UDGs which significantly improves on the previously known algorithms for the problem on UDGs. The only previously known singly-exponential algorithm for the problem has running-time of $\mathcal{O}(n^{80q})$ and is restricted to UDGs whose points are within a square of known size [54]. Here q is an upper bound on the number of cliques in a partition.

2) Studying fixed-parameter tractability of CLIQUE PARTITION, we design first-time data reduction rules for the problem on arbitrary UDGs. A novel geometric technique along with a detailed analysis of the properties of the reduced graph with respect to those reduction rules results in fixed-parameter algorithms of running-times $\mathcal{O}(2^{(3k+1)\log k + 12k\log \frac{1}{\epsilon} + 30k}n + mn)$ and $2^{\mathcal{O}(k\log \frac{k}{\lambda})}n + \mathcal{O}(mn)$ for CLIQUE PARTITION on the classes of ϵ -precision and (λ, α) -(relaxed)-quasi-precision UDGs with $\alpha \leq 1/2$, respectively. To the best of our knowledge this is the first fixed-parameter algorithm for the problem on a UDG subclass.

Finally, we fine-tune our approach to apply to a number of other interesting problems. Given a graph with positive weights assigned to its vertices, WEIGHTED CLIQUE PARTITION asks whether there exists a clique partition of at most a certain weight for a graph. The

¹By “clique covering” problems, we specifically refer to CLIQUE PARTITION, CLIQUE COVER and WEIGHTED CLIQUE PARTITION, although the tools presented in this work can be technically applied to other related problems.

weight function for a partition is defined as the sum over the maximum weight of vertices of each partition.

3) Studying the problem of WEIGHTED CLIQUE PARTITION, we are able to design parameterized algorithms with the same running-time as the ones for CLIQUE PARTITION on the discussed classes of UDGs. We further consider extending our framework to the problem of CLIQUE COVER which asks whether the edges of a graph can be partitioned into a certain number of (not necessarily disjoint) cliques. When the input is provided as an arbitrary UDG, we obtain a parameterized algorithm of running-time $\mathcal{O}(n^{6k+3})$ for CLIQUE COVER.

4) We also manage to design improved reduction rules for the problem and use it to derive FPT algorithms for CLIQUE COVER on the classes of ϵ -precision UDGs and (λ, α) - (relaxed)-quasi-precision UDGs of running-times $\mathcal{O}(2^{(3k+1)\log k + 12k \log \frac{1}{\epsilon} + 24k} m + mn)$ and $2^{\mathcal{O}(k \log \frac{k}{\lambda})} m + \mathcal{O}(mn)$, respectively. Our framework further applies when the reduced graph (rather than the input instance) belongs to the discussed subclasses of UDGs.

Next, we discuss a few key components that are essential in development of our parameterization framework and are of independent theoretical and practical interest.

Quasi-precision UDGs

It is known that the complexity of many graph optimization problems remains NP-hard even when restricted to the inputs that can be realized as UDGs. Some of these problems are 3-COLORING, FEEDBACK VERTEX SET, VERTEX COVER, CONNECTED VERTEX COVER, CYCLE PACKING and CLIQUE PARTITION [26]. Similarly, many such problems are believed not to admit polynomial-size kernelization or fixed-parameter algorithms on UDGs. In an attempt to study the problems on more tractable classes of UDGs, ϵ -precision [5] and bounded-area UDGs [78] were introduced in the literature. Yet, in many cases such subclasses appear too trivial. For example, ϵ -precision UDGs immediately translate to graphs of bounded degree (See the discussion in Section 4.2). On the other hand, bounded area UDGs are known to have bounded clique partition size [78] and as a result, a number of problems such as k -COLORING and HAMILTONIAN CIRCUIT are easily solvable on UDGs of bounded area.

In order to get the best of both worlds, we introduce (α, λ) -quasi-precision UDGs as UDGs which are only partially λ -precision, yet their imprecision components have bounded area. We show that (α, λ) -quasi-precision UDGs are *ubiquitous* in the UDG class in the

sense that any UDG is quasi-precision for some values of λ and α . In particular, precision UDGs and bounded-area UDGs are derived as trivial cases of quasi-precision class. Quasi-precision UDGs are also *non-trivial*, meaning that in their general setting none of the known graph parameters are proved to be bounded when considered on the class.

A) We demonstrate how the kernelization and parameterization frameworks obtained in our work can be extended to the class of (relaxed)-quasi-precision UDGs.

B) We further study the structure of quasi-precision UDGs and prove structural obstructions for the parameterization of problems under our framework. In particular, we show that UDGs which do not admit FPT algorithms under our framework include as subgraph a loosely-connected co-bipartite graph.

Data Reduction Rules for (WEIGHTED) CLIQUE PARTITION/COVER

Data reduction is an effective technique in kernelization and construction of parameterized algorithms and hence the design of reduction rules that can safely reduce the size of input to smaller instance in polynomial time is of independent theoretical and practical interest.

Considering the inputs provided as UDGs, we design first-time data reduction rules for the problems of CLIQUE PARTITION, WEIGHTED CLIQUE PARTITION and enhanced data reduction rules for CLIQUE COVER which in time $\mathcal{O}(mn)$ shrink the input to a reduced instance. When the input is assumed to be restricted to precision and quasi-precision UDGs, our detailed analysis can prove that the size of the reduced instance is indeed bounded by a function which is only *linear* in the parameter k .

1.2.2 Improved Polynomial Approximation Scheme for MINIMUM CLIQUE PARTITION

Studying the optimization version of CLIQUE PARTITION, we design randomized $(1 + \epsilon)$ -approximation algorithms for MINIMUM CLIQUE PARTITION on arbitrary UDGs. Our first PTAS is based on an adoption of the technique of [54] and a new packing argument and has an improved running-time of $n^{(176/\epsilon)^2 + \mathcal{O}(1/\epsilon)}$.

Furthermore, we propose a novel approach for deriving PTAS which relies on the geometric theorem on coverage of cliques by convex regions. The new PTAS still runs in time $n^{\mathcal{O}(1/\epsilon^2)} \cdot (1/\epsilon^2)$, yet the hidden constant in big \mathcal{O} notation is significantly (at least 13 times) smaller than the fastest previously known PTAS. In practice, this allows our algorithm to

run in a reasonable time on instances of size $|I|^{13}$, if the previous algorithms could solve an instance I of size $|I|$.

1.2.3 A Kernelization Framework for Problems on Quasi-precision UDGs

Despite the vast application of unit disk graphs and paramount importance of effective parameterized algorithms for problems on real world inputs described as UDGs, there are almost no general tools or frameworks known in the literature that aim at design of polynomial kernels or (sub)-exponential parameterized algorithms on such inputs.

We describe a general framework for kernelization of problems on quasi-precision UDGs. Through the introduction of a simple generic data reduction rule that reduces the input instance in polynomial time to smaller instance and a detailed analysis, we prove that when λ is set as a parameter, polynomial-size kernels are attainable for numerous problems on a large class of (λ, α) -quasi-precision UDGs. Furthermore, we characterize the problems for which polynomial-size kernels are derivable through the application of this framework by providing sufficient criteria for the correctness of the generic reduction rule. The problems covered in this framework include many classical NP-hard problems for which no positive kernelization results were known beforehand on any non-trivial graph class. Among those are DOMINATING SET, CONNECTED DOMINATING SET, CLIQUE COVER, CLIQUE PARTITION and INDEPENDENT SET.

1.3 Thesis Outline

The rest of the thesis is organized as follows. In Chapter 2, we state the preliminary definitions and some notations that are used throughout the rest of thesis. There, we also review some of the previous works in the field of parameterized algorithms and kernelization and clarify the gap in research that this thesis addresses.

In Chapter 3, we study the problem of CLIQUE PARTITION on unit disk graphs. We describe a novel approach for obtaining parameterized results for the problem on arbitrary UDGs. Later, we introduce first-time reduction rules for the problem on arbitrary UDGs which reduces the problem to a smaller instance in polynomial time and provide a proof of correctness for the reduction rules.

In Chapter 4, we study the fixed-parameter tractability of the CLIQUE PARTITION problem on subclasses of UDG. In particular, we propose an approach for obtaining improved

fixed-parameter algorithms for the problem on precision UDGs using data reduction and detailed analysis. We define quasi-precision UDGs later in this chapter as a non-trivial subclass of UDGs and further strengthen the results for CLIQUE PARTITION to (relaxed)-quasi-precision UDGs by showing how the fixed-parameter results can be extended to that subclass. We further demonstrate the significance and usefulness of quasi-precision UDGs by proving structural obstructions for this subclass of UDGs.

In Chapter 5, we study the optimization version of the CLIQUE PARTITION problem, namely MINIMUM CLIQUE PARTITION, with respect to approximation when the input is a UDG representation of a graph in the plane. In particular, using a packing argument which was used in Chapter 3, we obtain simple improvements over the running-time of previous polynomial time approximation schemes (PTAS) for MINIMUM CLIQUE PARTITION. Later, we devise a novel approach for obtaining PTAS on the problem which results in a PTAS with significant running-time speed-up for MINIMUM CLIQUE PARTITION on UDGs.

In Chapter 6, we discuss parameterization of a few other related problems on UDGs. In particular, studying the problems of CLIQUE COVER and WEIGHTED CLIQUE PARTITION, we demonstrate how to derive results parallel to the ones obtained for CLIQUE PARTITION (in Chapter 3 and 4) for the above two problems on the previously discussed classes of UDGs.

In Chapter 7, we design a framework for kernelization of problems on quasi-precision UDGs. In particular, we establish criteria under which a problem would admit polynomial kernels on the class of quasi-precision UDGs. Having defined a generic data reduction rule, we perform an analysis to show that, if the problem satisfy the defined criteria, the size of the instance after performing the reduction rule is bounded by a polynomial in the parameter. We further demonstrate the importance and use of this framework by discussing its application to a few candidate problems.

We finally conclude the thesis in Chapter 8 and discuss possible future works and directions in this line of research.

Chapter 2

Background and Related Work

In this chapter, we review some of the tools and frameworks introduced in the parameterized algorithms literature for obtaining results on restricted classes of graphs, while the main focus is on the classes of graphs that admit small¹ problem kernels or fixed-parameter algorithms.

In Section 2.1, we put forth the definitions and notations that we adhere to throughout this thesis. Section 2.2 reviews the works in the field of data reduction by drawing a few case studies. We discuss the general frameworks that are introduced in the literature for obtaining parameterization and kernelization results in Section 2.3. Finally, in Section 2.4, we review the related work on sparse classes of graphs (degenerate graphs and unit disk graphs) in this line of research.

2.1 Definitions and Notations

Let $G = (V, E)$ be a graph. For a pair of vertices $u, v \in V$, an undirected edge between u and v in G is expressed by uv . A graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. Furthermore, if $E' = \{uv \in E \mid u, v \in V'\}$, then G' is called an *subgraph* of G induced by V' , denoted $G[V']$ alternatively. We refer the reader to a textbook on graph theory e.g. [112] for the other well-known terminologies. We denote by d_G , the diameter of G , that is the maximum distance between any two vertices in G . Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ on n vertices, G_1 is *isomorphic* to G_2 (written $G_1 \cong G_2$) if there is a bijection

¹By “small”, we often mean upper-bounded by a polynomial function of the parameter. However, this definition is very much dependent on the choice of the problem.

$f : V_1 \rightarrow V_2$ such that $\forall uv \in E_1, f(u)f(v) \in E_2$. We denote by \mathcal{G}_g , the class of all graphs embeddable in some surface of Euler genus g .

Given an edge $e = xy$ of a graph G , the graph $G \setminus e$ is the graph obtained from G after deleting the edge e . Similarly, the graph G/e is obtained from G after contracting the edge e , that is, after removal of edge e , the endpoints x and y are replaced by a new vertex v_{xy} which is adjacent to the old neighbors of x and y . Given a subgraph X of G , we denote by G/X the graph obtained from G after contracting every edge in X . Graph $H = G/X$ obtained after a sequence of edge-contractions is said to be a *contraction* of G and is denoted $H \leq_c G$. Given a graph G , a subgraph X of G and \mathcal{G} , a collection of subgraphs of G , we denote by \mathcal{G}/X the class of graphs obtained from \mathcal{G} after replacing every graph $H \in \mathcal{G}$ by H/X .

A graph H is a *minor* of a graph G if H is the contraction of some subgraph of G . The minor relation between G and H is denoted by $H \leq_m G$. We call edge deletion and contraction (resp. \setminus and $/$), the minor operations. A graph is *H -minor-free*, if it does not contain H as a minor. Similarly, a graph class \mathcal{G}_H is *H -minor-free* when all its members are H -minor-free.

Next, we review some of the standard definitions of parameterized complexity. We refer the interested reader to a text book on parameterized complexity e.g. [53] for a more formal discussion.

DEFINITION: [Parameterized problem]

A *parameterized problem* Π is a subset of $\Gamma^* \times \mathbb{N}$ for some finite alphabet Γ . An instance of a parameterized problem then consists of a pair (I, k) , where k is called the parameter. Assuming that k is given in unary, $k \leq |I|$. We denote by $\overline{\Pi}$ the set of all no-instances of Π .

DEFINITION: [FPT]

A *fixed-parameter algorithm* gives a solution to a problem with an input instance of size n and a parameter k in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable function f depending solely on k . The class of all problems for which a fixed-parameter algorithm exists is known as FPT. We also refer to the function $f(k) \cdot n^{\mathcal{O}(1)}$ as FPT-time.

In the parameterized complexity paradigm, the main complexity classes and the relation between them are described as follows:

$$\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \dots \subseteq \text{W}[P] \subseteq \text{XP}$$

The focus of this chapter is on the design of solutions for concrete computational problems. Hence, we avoid the formal definition of the classes and discuss only those that have compelling practical significance within the context of their associated complete problems. For this purpose, it usually suffices to only mention the relation of a given problem with the classes FPT, W[1] and (sometimes) XP. We refer the interested readers to a textbook on parameterized complexity (e.g. [53]) for a formal definition of the complexity classes.

It is worth noticing that k -MAX CLIQUE is a complete problem for W[1] class while another classical problem, k -DOMINATING SET is a complete problem for W[2]. The class XP consists of parameterized decision problems Π , such that for each instance (I, k) , it can be decided in $\mathcal{O}(f(k)|I|^{g(k)})$ time whether $(I, k) \in \Pi$, where f, g are computable functions depending on k only. That is, XP consists of parameterized decision problems which can be solved in polynomial time if the parameter k is considered as a constant.

DEFINITION: [**Kernelization**]

Let Π be a parameterized problem consisting of input pairs (I, k) , where I is the input instance and k is the parameter for I . Then, *kernelization* or *reduction to a problem kernel* is to map an instance (I, k) to a reduced instance (I', k') called *problem kernel* such that $k' \leq k$ and $|I'| \leq g(k)$ for some function g only depending on k , $(I, k) \in \Pi$ if and only if $(I', k') \in \Pi$, and the reduction is computable in polynomial time in the size of (I, k) . Here, $g(k)$ is called the *kernel size*.

DEFINITION: [**PTAS**]

A *polynomial time approximation scheme* (PTAS) is an algorithm which takes an instance I of an optimization and a parameter $\epsilon > 0$ and produces a solution that is within a factor $(1 + \epsilon)$ of being optimal, in a time polynomial in $|I|$ for any fixed ϵ . That is, the running-time has to be bounded by $\mathcal{O}(|I|^{f(1/\epsilon)})$ for a computable function f depending on ϵ only. An *efficient* PTAS (EPTAS) is the one that requires the running-time to be $\mathcal{O}(|I|^c)$ for a constant c independent of ϵ . However, the big- \mathcal{O} can still depend on ϵ arbitrarily.

Therefore, EPTAS is, in a sense, the closest to what practical applications hope to obtain for an NP-hard problem. Interested readers can refer to textbook on algorithms (e.g. [109]) for a detailed discussion on classical approximation.

The following theorem stresses the importance of existence of kernels the best as a practical tractability measure.

THEOREM 2.1.1. [61, Th. 1.32]

For a minimization (resp. maximization) problem $\overset{\circ}{\Pi}$, the parameterized problem denoted by Π is defined as the language consisting of all pairs (I, k) where I is an instance of $\overset{\circ}{\Pi}$ and $\text{OPT}(I) \geq k$ (resp. $\text{OPT}(I) \leq k$). If $\overset{\circ}{\Pi}$ has an EPTAS then $\Pi \in \text{FPT}$.

Thus, if a parameterized problem Π , naturally associated with an optimization problem $\overset{\circ}{\Pi}$, is hard for $W[1]$, then $\overset{\circ}{\Pi}$ cannot have an efficient PTAS unless $\text{FPT} = W[1]$.

Cai & Huang [20] introduced the concept of FP-approximation where given a decision problem and a parameter k , in FPT-time the approximation algorithm either answers that no optimal solution of value k exists or it produces a solution whose value is a ratio away from the parameter k . Approximation with respect to different choices of parameters have attracted a lot of attention in the literature. Marx [93] has provided a comprehensive survey of the topic.

DEFINITION: [Tree decomposition] [105, 106]

A tree decomposition of a graph $G = (V, E)$ is a pair (\mathcal{X}, T) where $T = (V_T, E_T)$ is a tree and $\mathcal{X} = \{X_i \mid i \in V_T\}$ is a collection of subsets of V with the following properties:

1. $\bigcup_{i \in V_T} X_i = V$,
2. for each edge $xy \in E$, $\{x, y\} \subseteq X_i$ for some $i \in V_T$,
3. for each $x \in V$ the set $\{i \mid x \in X_i\}$ induces a connected subtree of T .

The *width* of the tree decomposition (\mathcal{X}, T) is then defined as $\max_{i \in V_T} |X_i - 1|$. *Treewidth* of G is the minimum width of all tree decompositions of G and is denoted by $\text{tw}(G)$.

For example, the treewidth of a tree on at least 2 vertices is 1 and a complete graph on k vertices has the treewidth of $k - 1$. The usefulness of tree decomposition became more apparent in the late 1980s when it turned out that many algorithmic problems that are NP-complete for arbitrary graphs might be solvable efficiently for graphs of bounded treewidth through dynamic programming, using the tree-decompositions of these graphs [7, 13].

2.2 Data Reduction

Data reduction is a popular technique in the literature of fixed-parameter algorithms and kernelization. Pre-processing hard problems is not a new concept and it can be traced back to the very beginning of algorithm research. The concept of data reduction to a problem kernel was introduced by Downey and Fellows [53] for the first time to formalize reductions for parameterized complexity purposes. In a nutshell, data reduction is a technique to introduce a set of useful reduction rules that can be applied exhaustively to an input instance to shrink it to a smaller instance in polynomial time. A solution for the original input then should be able to be reconstructed in polynomial time using a solution for the reduced instance. It is hoped then that the size of the reduced instance can be bounded by a function depending on the parameter only.

The VERTEX COVER problem is perhaps one of the earliest problems studied with respect to data reduction [97]. Cai et al. [19] proved that every fixed-parameter tractable problem is kernelizable. The research on proving kernels of polynomial (linear) size using data reduction has been receiving much attention. Readers may refer to [14, 75] for a comprehensive survey of the topic.

2.2.1 VERTEX COVER and Related Problems

The ease and usefulness of data reduction is perhaps best demonstrated in its application to VERTEX COVER problem. The VERTEX COVER problem asks for the minimum cardinality of a set C of vertices in graph G of such that each edge of G is incident to at least one vertex in C . The parameterized version of the problem k -VERTEX COVER asks for a given parameter k whether $|C| \leq k$.

It is easy to get a kernel of size $2k$ vertices and $\Theta(k^2)$ edges for k -VERTEX COVER using a few simple reduction rules. This kernel is known to be optimal in size; this is proved by Dell et al. in [32] by showing that no kernel of size $\mathcal{O}(k^{2-\epsilon})$ is possible unless the polynomial hierarchy collapses to its third level ($PH = \Sigma_3^p$). However, Jansen recently showed in [80] that this might be due to the fact that k is almost always linear in n and therefore studied the problem with respect to a different parameter, namely feedback vertex set number, $FVS(G)$ and still obtained a cubic kernel for the VERTEX COVER problem. In contrast WEIGHTED VERTEX COVER does not admit a polynomial-size kernel unless the polynomial

hierarchy collapses [80].

Let C be a collection of subsets of a finite set S . A *hitting set* is a subset of S that has a nonempty intersection with every element of C . The HITTING SET problem is to decide whether there is a hitting set with cardinality at most k for a given (S, C) . The input pair can also be regarded as a hypergraph such that S and C correspond to the sets of vertices and hyperedges, respectively. In this sense, the HITTING SET is equivalent to the generalization of VERTEX COVER on hypergraphs. If the cardinality of the subsets in C is bounded by a fixed natural number d , then the problem is called d -HITTING SET and is known to be fixed-parameter tractable. Although this problem does not admit a polynomial-size kernel, if d is set to be a constant kernelization results are possible [81]. In particular, 3-HITTING SET is reducible to a kernel of size $\mathcal{O}(k^2)$ according to the work of Abu-Khazam [1].

2.2.2 DOMINATING SET

Given a graph G with vertex set $V(G)$, the DOMINATING SET problem asks for a minimum subset $D \subseteq V(G)$ of vertices such that every vertex in $V(G) \setminus D$ has a neighbor in D . The cardinality of a minimum dominating set of G is known as the *domination number* of G , denoted by $\gamma(G)$. The DOMINATING SET problem is an important NP-hard graph problem [67], which belongs to a broader class of domination and covering problems. From applications' point of view, dominating problems appear in numerous practical settings such as resource allocations and wireless networks [76, 77, 111]. The evident importance of DOMINATING SET problems is well-described by the enormous amount of research focusing on these problems. The algorithmic complexity of the domination and related problems are discussed in detail in the book of Haynes et al. [77].

In the parameterized setting, it is known that the DOMINATING SET problem on arbitrary graphs is $W[2]$ -complete, that is, not fixed-parameter tractable [53] but when restricted to planar graphs it is fixed-parameter tractable [2, 3]. The best known fixed-parameter algorithm for the DOMINATING SET problem on planar graphs runs in time $\mathcal{O}(2^{11.98\sqrt{\gamma(G)}} n^{\mathcal{O}(1)})$ [49]. Data reduction for the DOMINATING SET problem has received much attention [4, 24]. Indeed, dominating set is one of the first few problems to be challenged using data reduction. The landmark paper of Alber et al. [4] give data reduction rules which always reduce a planar graph of n vertices to a problem kernel of size $\mathcal{O}(\gamma(G))$ in $\mathcal{O}(n^3)$ time. This result was slightly improved later in [24].

2.2.3 CONNECTED DOMINATING SET

A vertex u is *dominated* by a vertex v in a graph G if either $u = v$ or $\{u, v\} \in E(G)$ (u is adjacent to v). A vertex v is dominated by a vertex set U , if either $v \in U$ or v is adjacent to at least one vertex of U . Similarly, a vertex set U' is dominated by a vertex set U , if every vertex of U' is dominated by U . If a set U' is dominated by a set U , we say U' is dominated by the vertices of U . A subset $D \subseteq V(G)$ is a dominating set of G if $V(G)$ is dominated by D . A connected dominating set (CDS) of G is a subset $D \subseteq V(G)$ such that D is a dominating set of G and the subgraph $G[D]$ is connected. The CONNECTED DOMINATING SET problem is then to find a minimum CDS D of G . The decision version of the problem CONNECTED k -DOMINATING SET asks given a graph G and a positive integer k , whether the size of minimum connected dominating set of G denoted $\gamma_c(G)$ is at most k [73].

It is not known whether CONNECTED DOMINATING SET is fixed-parameter tractable in arbitrary graphs but it becomes tractable when restricted to planar graphs [51], where it plays an important role in efficient routing in wireless networks [85]. Lokshantov et al. [88] showed a linear-size problem kernel of at most $3968187\gamma_c(G)$ vertices for CONNECTED DOMINATING SET on planar graphs. Gu & Imani [73] independently proved that on planar graphs, the problem admits a linear-size problem kernel of much smaller size (at most $413\gamma_c(G)$ vertices). Quite recently, Luo et al. [90] further shrink this upper bound to $130k$.

2.3 Separators, Minors and Bidimensionality

2.3.1 Background

In this section, we discuss the tools and frameworks for kernelization of problems on classes of graphs that are more general than the planar graphs. In particular, we consider graphs embeddable on a fixed surface (with a fixed genus) as a direct generalization of the planar graphs.

A larger class that we consider here is the graphs excluding a fixed graph as a minor. The theory of graph minors perhaps began with the Wagner's theorem [110] stating that a graph is planar if and only if it does not contain the complete graph K_5 nor the complete bipartite graph $K_{3,3}$ as a minor. Later, Robertson and Seymour showed that every family of graphs that is closed under minors can be defined by a finite set of forbidden minors

[104]. The latter statement particularly stresses the importance of studying the H -minor-free graphs in algorithmic research. Later, Demaine et al. [45] made it possible to obtain algorithmic results on this class of graphs by proposing a polynomial time algorithm for structural decomposition of H -minor-free graphs.

2.3.2 Bidimensionality Framework

The theory of bidimensionality developed in a series of papers [33–36, 38–44, 47] over the last decade providing a general framework for designing efficient fixed-parameter algorithms and approximation algorithms for NP-hard problems on large classes of graphs. This theory applies to graph problems that are *bidimensional* meaning that the value of solution for the $k \times k$ grid graph (respectively other dense graphs) grows with k , while the value of solution does not increase after performing a minor operation (contracting or deleting edges). Examples of such problems include FEEDBACK VERTEX SET, VERTEX COVER, MINIMUM MAXIMAL MATCHING, FACE COVER, DOMINATING SET, EDGE DOMINATING SET, r -DOMINATING SET, CONNECTED DOMINATING SET, CONNECTED EDGE DOMINATING SET, CONNECTED r -DOMINATING SET, and UNWEIGHTED TSP TOUR (a walk in the graph visiting all vertices) [42].

The main idea behind the bidimensionality theory is that the value of the bidimensional problems on a graph G is almost captured by the value of the problem on its biggest grid minor. More specifically, the treewidth in such problems is upper-bounded by a function of the problem’s solution value. By proving this bound, it is possible to design an (sub-)exponential algorithm in the solution size, when (sub-)exponential dynamic programming based on tree-decomposition is known for the problem. Also this framework can be used to yield PTASs for bidimensional problems [43, 62]. The results in this line includes PTASs for the WEIGHTED TRAVELING SALESMAN problem (WEIGHTED TSP) and for MINIMUM-WEIGHT c -EDGE-CONNECTED SUBMULTIGRAPH on bounded-genus graphs.

DEFINITION: [Minor bidimensional problem][39]

A parameterized problem Π is called *minor bidimensional* if

- The value of the parameter does not increase by a minor operation (deleting/contracting an edge) i.e. for any pair of graphs $H \leq_m G$ and an integer k , $(G, k) \in \Pi \Rightarrow (H, k) \in \Pi$, and

- The value of solution on a $(r \times r)$ -grid R , is at least δr^2 for some $\delta > 0$, i.e. $\exists \delta > 0$ such that $(R, k) \notin \Pi$ for $k \leq \delta r^2$.

DEFINITION: [**Contraction bidimensional problem**][39]

A parameterized problem Π is called *contraction bidimensional* if

- The value of the parameter does not increase by a contraction operation i.e. for any pair of graphs $H \leq_c G$ and an integer k , $(G, k) \in \Pi \Rightarrow (H, k) \in \Pi$, and
- The value of solution on triangulated folded $(r \times r)$ -grid Γ_r , is at least δr^2 for some $\delta > 0$ i.e. $\exists \delta > 0$ such that $(\Gamma_r, k) \notin \Pi$ for $k \leq \delta r^2$.

We simply call a problem Π *bidimensional* if Π is minor bidimensional or contraction bidimensional.

2.3.3 Contraction Decomposition

DEFINITION: [**Separator**]

Let $G = (V, E)$ be an undirected graph. A vertex set $S \subseteq V$ is called a *separator* for G , if S divides V into $A_1 \subseteq V$ and $A_2 \subseteq V$ such that:

- $A_1 \cup S \cup A_2 = V$, and
- No vertex in A_1 is adjacent to a vertex in A_2 , i.e. $N(A_1) = N(A_2) = S$.

The triple (A_1, S, A_2) is called a *separation* of G .

One can generalize the above definition to the case where the separator S divides V into ℓ subsets $\{A_1, \dots, A_\ell\}$ instead of two. In this case S is called an ℓ -separator for G . We call a separator S *small*, if $|S| = o(|V|)$.

There are a variety of different separator tools known in the literature partitioning the vertex set or the edge set of graphs into sets satisfying certain criteria. The most historic ones are perhaps the Lipton and Tarjan's vertex separator for planar graphs [87] followed by the Baker's separation theorem [8]. The main technique in these decompositions is to find relatively small cuts in the graph that minimize the interaction between the pieces. Among such results there are a few which make a connection between the size of the separator and the treewidth of the graph.

DEFINITION: [**Separator theorem**] [87]

An $f(\cdot)$ -separator theorem with constants $\alpha < 1, \beta > 0$ for a class \mathcal{G} of graphs closed under taking vertex-induced subgraphs, is a theorem of the following form: Let $G \in \mathcal{G}$ be a graph on n vertices, then there is a separation (A_1, S, A_2) of G such that:

- Neither A_1 nor A_2 contains more than αn vertices, and
- S contains no more than $\beta f(n)$ vertices.

Contraction decomposition states that the edges of every graph of bounded (Euler) genus can be partitioned into any prescribed number k of pieces such that contracting any piece results in a graph of bounded treewidth (where the bound depends on k). Demaine et al. [37] state the result formally as follows:

[**Contraction decomposition**]

For a fixed integer g , and any integer $k \geq 2$ and for every graph G of Euler genus at most g , the edges of G can be partitioned into k sets such that contracting any one of the sets results in a graph of treewidth $\mathcal{O}((g+1)^2 k)$. Furthermore, such a partition can be found in $\mathcal{O}((g+1)^{5/2} n^{3/2} \log n)$ time [37].

The main application of this method is its use as a tool for obtaining PTASs for contraction-closed problems (whose optimal solution only improves under contraction), a much more general class than minor-closed problems (whose optimal solution improves under taking minor) [37]. In particular, the framework obtained through contraction decomposition yields PTASs for the WEIGHTED TRAVELING SALESMAN problem and for MINIMUM-WEIGHT c -EDGE-CONNECTED SUBMULTIGRAPH on bounded-genus graphs.

The following result by Eppstein [55] relating the treewidth and diameter in a bounded-genus graph, is useful in understanding the idea of this technique.

[**Bounded local-treewidth**]

Let $G = (V, E)$ be a graph embedded in a surface of genus g and let $x_0 \in V$. If every vertex of G is at distance at most d from x_0 then $\mathbf{tw}(G) \leq 3d + 3$ if $g = 0$, and $\mathbf{tw}(G) = \mathcal{O}(g \cdot d)$ if $g \geq 1$ [55].

The first result discusses contraction decomposition of planar graphs [37].

[**Planar graph decomposition**]

If $w \geq 6k(q+3)$ for integers $k \geq 1, q \geq 0$, $\mathcal{C}_{k,q}^w$ contains the class of all planar graphs [37].

The above result can be extended to other surfaces by applying induction on the Euler genus. Cutting the graph along a simple non-contractible curve in G embedded in a surface with genus g would result in a graph with a smaller genus $g' < g$ for which the induction is assumed to be true.

[Bounded-genus decomposition]

Given any integers $k \geq 1, q \geq 0$, and $g \geq 0$, let $w = 120k(g + 1) \cdot (2g + q + 2)$.

Then $\mathcal{G}_g \subseteq \mathcal{C}_{k,q}^w$ [37].

The above result was further strengthened by Demaine et al. in [46] so that it applies to H -minor-free classes of graphs. They did so by proving a nicer decomposition for H -minor-free graphs which was a long standing open problem and is formalized as follows.

[H -minor-free decomposition]

For any fixed graph H , there is a constant c_H such that, for any integer $k \geq 1$, every H -minor-free graph G can have its edges partitioned into $k + 1$ color classes such that contracting any one of the color classes results in a graph of treewidth at most $c_H k$. Furthermore, such a partition can be found in polynomial time [46].

Applications

The above result can be used to obtain a variety of PTASs and exact algorithms for a number of problems on H -minor-free graphs. First, we provide a few definitions.

DEFINITION: $[(\delta, \alpha)$ -spanner]

Given a weighted graph (G, w) with weight w , a constant $\delta > 0$, G' – a (δ, α) -spanner for G – is a graph such that $\text{OPT}(G') \geq \alpha \cdot w(G')$, for some constant $\alpha > 0$ (possibly depending on δ) and any c -approximation to G' can be converted into a $(1 + \delta) \cdot c$ -approximation solution to G in polynomial time.

What follows provides a framework for obtaining PTAS on H -minor-free graphs using contraction decomposition, given that the problem satisfies the required criteria.

[PTAS via contraction decomposition]

Let Π be a minimization problem closed under contraction on weighted graphs, solvable in polynomial time on graphs of bounded treewidth. Furthermore, suppose the following criteria is true:

- There is a polynomial-time algorithm that given a weighted H -minor-free graph (G, w) and a constant $\delta > 0$, calculates an H -minor-free graph G' such that G' is (δ, α) -spanner of G .
- There is a polynomial-time algorithm that given a subset S of edges of a weighted graph (G, w) and given an optimal solution for G/S , constructs a solution for G of value at most $\text{OPT}(G/S) + \beta w(S)$ for some constant $\beta > 0$.

Then for any fixed minor H and any constant $0 < \epsilon < 1$, there is a polynomial-time $(1 + \epsilon)$ -approximation algorithm for problem Π on H -minor-free graphs. Furthermore, if α grows as a function of n , then the running-time becomes bounded by a polynomial multiplied by the cost of solving the problem on graphs of treewidth $\mathcal{O}(\alpha)$.

Next, we demonstrate the use of this framework by describing how it can be applied to a number of candidate problems.

k-CUT.

Instance: A graph $G = (V, E)$ and a weight function $f : E \rightarrow \mathbb{R}$.

Question: Find a minimum-weight set of edges whose removal would partition G into k connected components.

WEIGHTED TSP.

Instance: A graph $G = (V, E)$ and a weight function $f : E \rightarrow \mathbb{R}$.

Question: Find a Hamiltonian cycle such that the sum of weights on its edges is minimum.

The first result is obtained by the application of contraction decomposition to the problem of *k*-CUT. Notice that, since there is a vertex of degree at most $\mathcal{O}(|V(H)|\sqrt{\log |V(H)|})$ in an H -minor-free graph G , a k -cut in G can have a size at most $c_H k$ for some constant c_H depending on the minor H only. Now, one can use contraction decomposition and partition the edges to $c_H k + 1$ sets such that at least one of the sets has no intersection with the optimum solution. By guessing this set among the $c_H k + 1$ and contracting the edges, we

are left with a graph of bounded treewidth in time $2^{\tilde{O}(k)}n$.² This is formalized as follows.

PROPOSITION: There is an exact fixed-parameter algorithm that calculates k -CUT on H -minor-free graphs in time $2^{\tilde{O}(k)}n + n^{\mathcal{O}(1)}$ [46].

The existence of a spanning-subgraph is also promised in [71] as below.

Given an edge-weighted graph excluding K_h as a minor, there is a polynomial-time greedy algorithm to find a spanning subgraph approximating all shortest-path distances within a factor of $1 + \epsilon$, and with total edge-weight at most $\alpha = \mathcal{O}((h\sqrt{\log h} \cdot \log n)/\epsilon)$ times the weight of a minimum spanning tree.

The following result by Dorn et al. [50] addresses the problem of finding optimal-weight TSP in weighted H -minor free graphs.

Given a weighted graph G excluding H as a minor with $|V(H)| = h$ and $\mathbf{tw}(G) \leq k$, there is an exact algorithm of time $2^{f(h)k}n$ that finds an optimal-weight TSP for some function $f(h)$.

Now one can use the PTAS framework defined earlier, along with the spanner promised above to obtain a PTAS for WEIGHTED TSP.

PROPOSITION: WEIGHTED TSP admits a PTAS on H -minor-free graphs.

2.3.4 (Meta)-Kernelization

(Meta)-Kernelization theory states that all problems expressible in Counting Monadic Second Order (CMSO) logic satisfying a compactness property, admit a polynomial-size kernel on graphs of bounded genus, while all problems that have finite integer index and satisfy a weaker compactness condition admit a linear-size kernel on graphs of bounded genus [15].

DEFINITION: [**Radial distance**] [15]

Given a graph $G = (V, E)$ embedded in a surface Σ of Euler genus g , the *radial distance* between two vertices x and y is the minimum length of an alternating sequence of incident vertices and faces starting from x and ending in y . Given a $S \subseteq V$, we denote by $\mathfrak{R}_g^r(S)$ the set of all vertices in V with a radial distance at most r from some vertex in S .

²Here \tilde{O} means that there can be an extra log factor (rather than a constant) hidden in the Big-Oh notation.

DEFINITION: [**Compactness**] [15]

A parameterized problem $\Pi \subseteq \mathcal{G}_g \times \mathbb{N}$ is said to be *compact* if there is an integer r such that for all $(G, k) \in \Pi$, there is an embedding of G in a surface Σ of Euler genus g and a set $S \subseteq V$ with $|S| \leq k \cdot r$ such that $\mathfrak{R}_g^r(S) = V$.

Similarly, Π is called *quasi-compact* if there exists integer r such that for every $(G, k) \in \Pi$ there is an embedding of G in a surface Σ of genus g and a set $S \subseteq V$ of size at most $r \cdot k$ such that $\text{tw}(G \setminus \mathfrak{R}_g^r(S)) \leq r$.

DEFINITION: [**Counting monadic second order**]

Counting monadic second order, CMSO is an extension of monadic second order (MSO) with the addition of the following atomic formula: Let U denote a set X , then $\text{card}_{n,p}(U) = \text{true}$ if and only if $|X| \equiv n \pmod{p}$. For an introduction on monadic second order logic, please refer to [27–30].

DEFINITION: [**CMSO problem**] [15]

A p -MIN-CMSO problem $\Pi \subseteq \mathcal{G}_g \times \mathbb{N}$ is, given a graph $G = (V, E)$ and an integer k as input, to decide whether there is a vertex/edge set S of size at most k such that the CMSO-expressible predicate $P_\Pi(G, S)$ is satisfied. Similarly, in a p -EQ-CMSO problem, the size of S is required to be exactly k and in a p -MAX-CMSO problem, the size of S is required to be at least k . The *annotated* version $\dot{\Pi}$ of a p -MIN/EQ/MAX-CMSO problem, Π is defined as follows. A triple (G, Y, k) is given as input where G is a graph and $Y \subseteq V$ is a set of black vertices. In the annotated version of a p -MIN/EQ-CMSO graph problem, S is additionally required to be a subset of Y . For the annotated version of a p -MAX-CMSO graph problem, S is not required to be a subset of Y but instead of $|S| \leq k$, we require that $|S \cap Y| \leq k$.

The main result of this section is regarding the kernelization of $\dot{\Pi}$ for Π a p -MIN/EQ/MAX-CMSO problems.

[Kernelization of CMSO problems]

Let Π be a p -MIN/EQ/MAX-CMSO parameterized problem over the class of graphs embeddable in a surface of Euler genus g . Further, assume either Π or $\bar{\Pi}$ is compact. Then $\dot{\Pi}$ admits a cubic kernel if Π is a p -EQ-CMSO problem and a quadratic kernel if Π is a p -MIN/MAX-CMSO problem [15].

The existence of finite integer index for a problem is a stronger condition that can be used to prove the existence of smaller kernels for a problem. The next statement exploits this condition for kernelization of bounded-genus graphs.

[Kernelization of problems with finite integer index]

Suppose $\Pi \in \mathcal{G}_g \times \mathbb{N}$ has finite integer index and either Π or $\overline{\Pi}$ is quasi-compact.

Then, Π admits a linear-size kernel [15].

2.3.5 Kernels for Problems on H -Minor-Free Graphs

Bidimensionality, as discussed before, is best known as a framework for obtaining sub-exponential parameterized algorithms for specific problems (bidimensional problems) on H -minor-free graphs.

Given a graph $G = (V, E)$ and $S \subseteq V$, we define $\partial G(S)$ as the set of vertices in S that have a neighbor in $V \setminus S$. For a set $S \subseteq V$ the neighborhood of S is $N_G(S) = \partial G(V \setminus S)$.

DEFINITION: **[Protrusion]**[65]

Given a graph $G = (V, E)$, a set $X' \subseteq V$ is an r -protrusion of G if $|N(X')| \leq r$ and $\mathbf{tw}(G[X' \cup N(X')]) \leq r$. The vertex set $X = X' \cup N(X')$ is then called the *extended r -protrusion* of X' .

DEFINITION: **[Terminal graph]**[65]

A *terminal graph* is a triple (V, E, X) with (V, E) , an undirected graph and $X \subseteq V$, an ordered set of vertices called the set of *terminals*. Terminal graph (V, E, X) is called k -terminal graph, if $|X| = k$. If there are no edges between vertices in X , then the terminal graph is called *open*. The operation \oplus takes two k -terminal graphs G_1, G_2 and merge them into a graph $H = G_1 \oplus G_2$ by taking their disjoint union and identifying i th terminal of the first terminal graph with the i th terminal from the other terminal graph for $i = 1, \dots, k$ (multi-edges in H are then removed if any).

DEFINITION: **[Reduction rule]**[65]

A *reduction rule* $R : G_0 \xrightarrow{R} G_1$ is an ordered pair of open t -terminal graphs G_0, G_1 . The *application* of a reduction rule R is the operation that takes a graph H of form $H = H_0 \oplus H_2$ and replace it by the graph $H' = H_1 \oplus H_2$, where $H_0 \cong G_0$ and $H_1 \cong G_1$.

For a parameterized problem Π , a class \mathcal{G} , and a reduction rule $G_0 \rightarrow G_1$, we say that $G_0 \equiv_{\Pi} G_1$ if $\exists c > 0$ such that for all t -terminal graphs G_2 (for t the number of terminals in G_0 , likewise in G_1) and for all k , I) $G_0 \oplus G_2 \in \mathcal{G}$ iff $G_1 \oplus G_2 \in \mathcal{G}$, II) $(G_0 \oplus G_2, k + c) \in \Pi$ iff $(G_1 \oplus G_2, k) \in \Pi$.

DEFINITION: [Finite integer index][65]

A parameterized problem Π has finite integer index in a graph class \mathcal{G} , if for every t there exists a finite set \mathcal{S} of t -terminal graphs such that $\mathcal{S} \subseteq \mathcal{G}$ and for any t -terminal graph G_1 , there exists $G_2 \in \mathcal{S}$ such that $G_1 \equiv_{\Pi} G_2$. Such a set \mathcal{S} is called a *set of representatives* for (Π, t) .

Roughly speaking, a minimum set of representatives for Π over \mathcal{G} corresponds to the minimum set of reduction rules. Also note that \equiv_{Π} is an equivalence relation and hence it defines $|\mathcal{S}|$ equivalence classes over \mathcal{G} .

DEFINITION: [Separation property][43, 44]

A minor-bidimensional problem Π , has the *separation property* if for any graph G , given any vertex cut S , and an optimal solution OPT to G , for any union G' of some subset of connected components of $G \setminus S$, $|\text{OPT}(G')| - \mathcal{O}(|S|) \leq |\text{OPT} \cap G'| \leq |\text{OPT}(G')| + \mathcal{O}(|S|)$.

A contraction bidimensional problem has the *separation property*, if it satisfies the following: Given a graph G , let S be a vertex cut whose removal disconnects G into k connected components C_1, \dots, C_k .

For any $I \subseteq \{1, \dots, k\}$, let G_I be the graph obtained from G after contracting each component C_j in G for $j \notin I$ into a vertex in $N(C_j)$ with the lowest index. The optimal solution for G , OPT can be bounded in the following way: $|\text{OPT}(G_I)| - \mathcal{O}(|S|) \leq |\text{OPT} \cap G_I| \leq |\text{OPT}(G_I)| + \mathcal{O}(|S|)$. If Π has the separation property, we call Π *separable*.

Intuitively, the reduction rules for Π , look for extended γ -protrusion X of unbounded size $\Theta(\gamma n/k)$ and replace $G[X]$ with a graph $H \in \mathcal{S}$ of bounded size $c(\gamma, \Pi)$.

The following result extends the previous theorem to the H -minor-free graphs, provided that the problem is bidimensional and separable.

[Bidimensionality and Kernelization]

Every minor-bidimensional problem Π with the separation property and finite

integer index, has a linear-size kernel on graphs excluding some fixed graph as a minor. Every contraction bidimensional problem Π with the separation property and with finite integer index, has a linear-size kernel on graphs excluding some fixed apex graph as a minor [65].

2.4 Kernels for Degenerate Graphs and UDGs

2.4.1 Background

One of the most formal classifications of graphs into useful classes, is perhaps, based on the topology (more specifically genus) of the surface on which the graph is embeddable with no edges crossing. The more general classes of graphs such as apex-minor-free and H -minor-free classes of graphs are also a generalization of the same concept (since for each surface there is a finite set of excluded minors). Despite the fact that this classification is very well-defined and hence many useful frameworks are developed for obtaining algorithmic and complexity results, many of the problems that come from application fail to conform to any of the classes mentioned above.

Degenerate graphs and unit disk graphs are two large classes which are not minor-free or embeddable on a fixed-surface. For example, one can convert any graph into a 2-degenerate graphs by subdividing its edges (this does not change the genus). Similarly, in the case of unit disk graphs, the size of cliques (and hence the Euler genus) is not bounded. As a result, the previous techniques for bounded genus and H -minor-free graphs cannot be directly applied to these classes of graphs. In reality, there is no general framework known in the literature to deal with kernelization of problems on the classes of graphs that are not minor-closed. However, one might exploit the specific structural properties of degenerate graphs and unit disk graphs to obtain results for specific problems on these classes. Here, we often focus our attention on combinatorial problems for which no polynomial-size problem kernels are hoped in the class of general graphs and verify whether they admit kernels of polynomial-size on the restricted classes.

2.4.2 Degenerate Graphs

DEFINITION: A graph is *d-degenerate* if its every subgraph contains a vertex of degree at most d . Given a problem Π on general graphs we denote by *d-deg- Π* , the problem Π on the class of *d-degenerate* graphs.

It is known that planar graphs are 5-degenerate. Alon & Gutner [6] and later Golovach & Villanger [69] proved that DOMINATING SET and CONNECTED DOMINATING SET, which are $W[2]$ -hard in general graphs [53], become FPT when they are restricted to *d-degenerate* graphs. Very recently, Philip et al. [101] proved that DOMINATING SET is in FPT and admits a polynomial-size kernel on a class of graphs including the degenerate graphs: graphs excluding $K_{i,j}$ as a subgraph [92].

Lower Bound Kernelization

Degenerate graphs have particularly been studied with respect to negative results and lower bounds on the size of possible kernels. Cygan et al. [92] studied the hardness of kernelization for a few connectivity problems showing that unless $PH = \Sigma_3^p$, there do *not* exist polynomial-size kernels for CONNECTED DOMINATING SET, STEINER TREE, CONNECTED FEEDBACK VERTEX SET and CONNECTED ODD CYCLE TRANSVERSAL in *d-degenerate* graphs for $d \geq 2$.

There are two general approaches for obtaining lower bound kernelization results in the literature. The first approach is using a reduction from an NP-hard problem. We first need to define a parameterized variant of a polynomial reduction.

DEFINITION: Let P and Q be two parameterized problems. We say that P is *polynomial parameter reducible* to Q , written $P \leq_{Ptp} Q$, if there exists a polynomial time computable function $f : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \times \mathbb{N}$ and a polynomial p , such that, $\forall (x, k) \in \Sigma^* \times \mathbb{N}$ if $f((x, k)) = (x', k')$, we have $(x, k) \in P \iff (x', k') \in Q$ for $k' \leq p(k)$. The function f then is called a *polynomial parameter transformation* (PPT).

The following is a known result regarding the application of polynomial parameter transformation to polynomial size kernelization.

[Kernels under PPT]

Let P and Q be parameterized problems and \mathring{P} and \mathring{Q} be the unparameterized

versions of P and Q respectively. Further suppose that \mathring{P} is NP-hard and \mathring{Q} is in NP. Assume there is a polynomial parameter transformation from P to Q . Then if Q admits a polynomial-size kernel, so does P [92].

One can use the above theorem to prove lower bound on the size of kernels. More specifically, for a problem P not admitting a polynomial-size problem kernel and satisfying the theorem's criteria, and Q with unknown kernel lower bounds, one can find a polynomial parameter transform from P to Q to prove that Q does not have a polynomial-size kernel. We demonstrate this by showing reductions for two combinatorial problems.

CONNECTED VERTEX COVER.

Instance: A graph $G = (V, E)$ and a non-negative integer k .

Question: Is there a subset $S \subset V$ of vertices with $|S| \leq k$ such that $G[S]$ is connected and every edge $e \in E$, has an end vertex in S ?

CONNECTED ODD CYCLE TRANSVERSAL.

Instance: A graph $G = (V, E)$ and a non-negative integer k .

Question: Is there a subset $S \subset V$ of vertices with $|S| \leq k$ such that $G[S]$ is connected and $G[V \setminus S]$ is bipartite?

CONNECTED FEEDBACK VERTEX SET (CFVS).

Instance: A graph $G = (V, E)$ and a non-negative integer k .

Question: Is there a subset $S \subset V$ of vertices with $|S| \leq k$ such that $G[S]$ is connected and $G[V \setminus S]$ contains no cycles?

PROPOSITION: CONNECTED VERTEX COVER \leq_{Pt_p} 2-deg-CONNECTED ODD CYCLE TRANSVERSAL and CONNECTED VERTEX COVER \leq_{Pt_p} 2-deg-CONNECTED FEEDBACK VERTEX SET [92].

STEINER TREE.

Instance: A graph $G = (V, E)$, a set of terminals $T \subset V$ and an integer k .

Question: Is there a subset $S \subset V$ with $|S| \leq k$ such that $G[S \cup T]$ is connected?

PROPOSITION: STEINER TREE \leq_{Pt_p} 2-deg-STEINER TREE [92].

Finally, since by the result of [48], STEINER TREE and CONNECTED VERTEX COVER do not admit polynomial-size kernels on the class of general graphs, one can use the theorem regarding kernelization under PPT to obtain the following results.

STEINER TREE, CONNECTED FEEDBACK VERTEX SET and CONNECTED ODD CYCLE TRANSVERSAL do not admit a polynomial-size kernel on 2-degenerate graphs unless $PH = \Sigma_3^P$ [92].

Many of the results in this section are obtained using a polynomial parameter reduction from COLORFUL GRAPH MOTIF [92] which is defined as follows:

COLORFUL GRAPH MOTIF.

Instance: A graph $G = (V, E)$, an integer k and a function $g : V \rightarrow \{1, \dots, k\}$.

Question: Is there a connected subset of vertices, $S \subset V$ with $|S| = k$ such that $g|_S$ is bijective?

The second approach is using the composition theorem for obtaining lower bound kernelization.

DEFINITION: [**Compositional problem**]

A *composition algorithm* for a parameterized problem $Q \subset \Sigma^* \times \mathbb{N}$ is an algorithm that receives as input a sequence $(x_1, k), (x_2, k), \dots, (x_t, k)$ with $(x_i, k) \in \Sigma^* \times \mathbb{N}$ for $1 \leq i \leq t$, uses polynomial time $\sum_{i=1}^t |x_i| + k$ and outputs $(y, k') \in \Sigma^* \times \mathbb{N}$ with $(y, k') \in Q$ iff $\exists_{1 \leq i \leq t} (x_i, k) \in Q$ and k' a polynomial in k . A parameterized problem is then called *compositional* if there is a composition algorithm for it.

The following important result demonstrates the use of compositional problems for proving non-existence of polynomial size kernels.

[**Negative kernelization for compositional problems**]

Let Q be a compositional parameterized problem whose unparameterized version \mathring{Q} is NP-complete. Then, unless $PH = \Sigma_3^P$, there is no polynomial kernel for Q [92].

Using the above theorem, it is not hard to show that the COLORFUL GRAPH MOTIF does not admit a polynomial kernel on forests of maximum degree 3. Fellows et al. [58] already showed that this problem is NP-hard on this class of graphs. To show the kernel lower

bound, it is enough to take the disjoint union of graphs and the union of functions g as the composition algorithm to prove that the problem is compositional and use the theorem regarding kernelization of compositional problems immediately. Since forests are 1-degenerate graphs, we get:

PROPOSITION: 1-*deg*-COLORFUL GRAPH MOTIF does not admit a polynomial-size kernel unless $PH = \Sigma_3^p$ [92].

Now, we can use reductions from COLORFUL GRAPH MOTIF to CONNECTED DOMINATING SET and STEINER TREE to obtain negative results for the latter problems.

PROPOSITION: d -*deg*-COLORFUL GRAPH MOTIF \leq_{Pt_p} $(d + 1)$ -*deg*-CONNECTED DOMINATING SET and 2-*deg*-CONNECTED DOMINATING SET does not admit a polynomial-size kernel unless $PH = \Sigma_3^p$ [92].

PROPOSITION: d -*deg*-COLORFUL GRAPH MOTIF \leq_{Pt_p} $(d + 1)$ -*deg*-STEINER TREE and 2-*deg*-STEINER TREE does not admit a polynomial-size kernel unless $PH = \Sigma_3^p$ [92].

Positive results

The research on parameterized tractability of degenerate graphs is still in its early stages. In particular, the complexity of many problems have not yet been studied for the degenerate class of graphs. Therefore, here we mention briefly a few positive kernelization results in the field.

k -DOMINATING SET has been studied in the class of graphs that do not have a biclique $K_{i,j}$ as a subgraph [101] where polynomial-size kernels of $\mathcal{O}((j + 1)^{2(i+1)}k^{2i^2})$ vertices has been proven. Considering the fact that d -degenerate graphs do not have a $K_{d+1,d+1}$ as a subgraph, one can obtain a kernel for k -DOMINATING SET on degenerate graphs using the result of [101]. Since a d -degenerate graph on n vertices can have at most $n \cdot d$ edges, we have:

PROPOSITION: k -DOMINATING SET problem on d -degenerate graphs has a kernel of $\mathcal{O}((d + 2)^{2(d+2)}k^{(2d+1)^2})$ vertices and edges [101].

The MINIMUM FILL-IN problem is studied in the context of degenerate graphs and some other families of sparse graphs in [66].

MINIMUM FILL-IN.

Instance: A graph $G = (V, E)$, a non-negative integer k .

Question: Can G be triangulated by adding at most k edges?

The problem which is also known as CHORDAL GRAPH COMPLETION is NP-complete and is also proved to be in FPT due to an algorithm with running-time $\mathcal{O}(2^{\mathcal{O}(\sqrt{k} \log k)} + k^{mn})$ [64].

PROPOSITION: MINIMUM FILL-IN problem has a d -degenerate kernel of size $\mathcal{O}(k^{3/2})$ on d -degenerate graphs [66].

2.4.3 Unit Disk Graphs

Unit disk graphs (UDGs) can be defined in three equivalent ways:

- For n points in the plane, form a graph with n vertices corresponding to the n points, and an edge between two vertices if and only if the distance between the two points is at most 1.
- For n unit circles in the plane, form a graph with n vertices corresponding to the n circles, and an edge between two vertices if and only if one of the corresponding circles contains the other's center.
- For n circles of unit diameter in the plane, form a graph with n vertices corresponding to the n circles, and an edge between two vertices if and only if the two circles intersect.

Studying the algorithmic aspects of problems on UDGs has been an attractive topic of research in the past. The readers can refer to [9] for a recent survey. Although the class of unit disk graphs is very restricted, it is not closed under the minor operation and hence many of the algorithmic results applicable to minor-closed class of graphs do not immediately follow for UDGs. It is shown for example in [26] that the complexity of many NP-hard optimization problems on general graphs remain NP-hard in the unit disk graph setting. Some of these problems are 3-COLORING, FEEDBACK VERTEX SET, VERTEX COVER, CONNECTED VERTEX COVER, CYCLE PACKING and CLIQUE PARTITION. Breu & Kirkpatrick [18] showed that the problem of recognizing unit disk graphs is NP-hard, and thereby answered one of problems left open in [26].

Let Σ be the 2-dimensional Euclidean plane. For two points p_1 and p_2 in Σ , we denote by $\mathbf{dist}(p_1, p_2)$ the Euclidean distance between the two points. For a point v of Σ , we define $\mathcal{D}(v) = \{w | \mathbf{dist}(v, w) \leq 1/2\}$ to be the disk with radius $1/2$ and center v , and $\partial\mathcal{D}(v) = \{w | \mathbf{dist}(v, w) = 1/2\}$ to be the boundary of $\mathcal{D}(v)$. Given an input graph G , we consider a unit disk representation of G where disks of radius $1/2$ are placed in the plane centered at every vertex of $V(G)$. In this case, we alternatively refer to $V(G)$ as point set $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$ with XY-coordinates, corresponding to the centers of disks. When we want to refer to a disk of radius $1/2$ centered at a vertex $v \in V(G)$, we denote it by $\mathcal{D}(v)$. Yet, whenever the center of the disk is unimportant in the context or the disk is not centered at a vertex of G , we refer to the disk by \mathcal{D}_r where r is the radius of the disk.

Although, unit disk graphs are not in general planar, as an intersection graph of some geometrical objects (disks) in the plane they exhibit some planar properties. The following result describes structural properties for UDGs.

A UDG has no induced subgraphs isomorphic to $K_{2,3}$ or $K_{1,6}$. [91].

The above theorem implies that the size of a maximum independent set in the neighborhood of a vertex v , $G[N(v)]$, is at most 6. However, this can be proved to be smaller if the vertex is located on the convex hull of the point set. Particularly, Marathe et al. [91] prove the following:

Let G be a unit disk graph and v be a vertex in G with the the smallest X-coordinate. The size of a maximum independent set in $G[N(v)]$ is at most 3 [91].

One can use the above results, for example to argue a lower bound on the size of cliques based on the maximum degree of a UDGs.

PROPOSITION: A unit disk graph G with maximum vertex degree Δ contains a clique of size at least $\lceil \Delta/6 \rceil + 1$ [91].

Given a pair of vertices v and w in a unit disk graph G with $\mathbf{dist}(v, w) \leq 1$, let us denote by $\mathcal{D}_v \cap \mathcal{D}_w$ the intersection of two disks of radius 1 centered at v and w and by $\Lambda_{v,w}$, the subset of vertices of G that are in $\mathcal{D}_v \cap \mathcal{D}_w$. See Figure 2.4.3.

Let C be the vertex set for a maximum-sized clique in $G = (V, E)$, then $C \subseteq \Lambda_{v,w}$ for some $v, w \in V$ with $\mathbf{dist}(v, w) \leq 1$ [26].

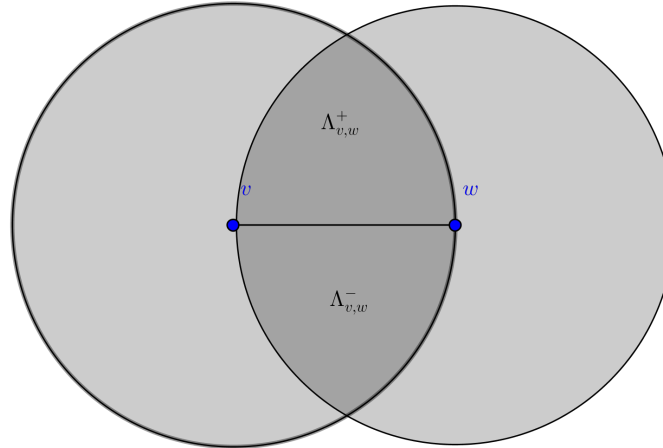


Figure 2.1: Two disks of radius 1 centered at points u and v . The area where the vertices of $\Lambda_{v,w}$ appear is highlighted in darker color.

Draw a straight line from v to w dividing $\mathcal{D}_v \cap \mathcal{D}_w$ into two halves. Let us denote by $\Lambda_{v,w}^-$ and $\Lambda_{v,w}^+$ the set of vertices in each of the halves. Then for each of the two regions, every two of vertices in that region have pairwise Euclidean distance of at most 1. Therefore, the graph induced by $\Lambda_{v,w}^-$ (likewise $\Lambda_{v,w}^+$) is a clique. Thus, one can conclude the following:

PROPOSITION 2.4.1. *The subgraph of G induced by $\Lambda_{v,w}$ is the complement of a bipartite graph.*

DEFINITION 2.4.1. [**Helly property**][26]

An intersection graph on a set of geometrical objects is said to satisfy the Helly property if for every number of k objects intersecting pairwise their k -fold intersection is a non-empty set.

The set of unit disks in the plane does not necessarily satisfy the Helly property. In particular, a set of 3 unit disks can be placed in the plane such that they pairwise intersect each other however the mutual intersection of the 3 unit disks be empty. See Figure 2.4.3 for an example of the case. Both drawings are valid unit disk representations of K_3 . Yet, only the

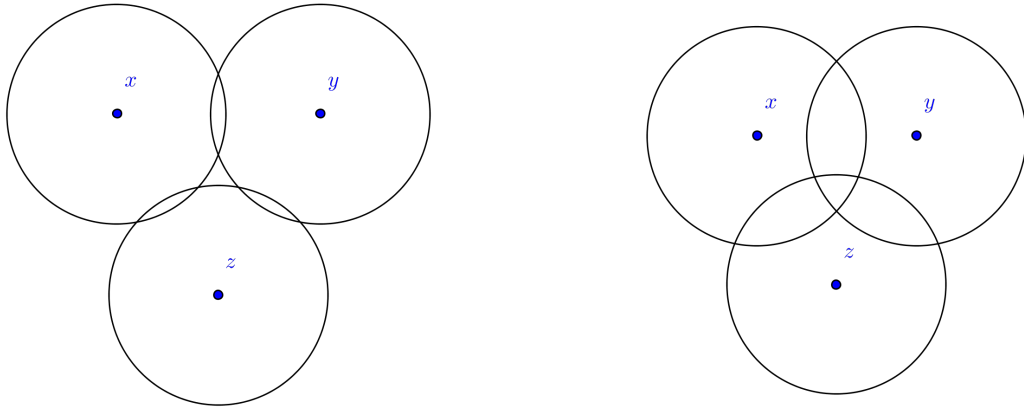


Figure 2.2: Two alternative unit disk representations of K_3 . The representation on the left does not satisfy the Helly property.

representation on the right satisfies the Helly property.

CONNECTED VERTEX COVER.

Instance: A graph $G = (V, E)$ and a non-negative integer k .

Question: Is there a subset $S \subset V$ of vertices with $|S| \leq k$ such that $G[S]$ is connected and every edge $e \in E$, has an end vertex in S ?

When restricted to the planar graphs the problem is known to have a kernel with $14k$ vertices according to the work of Guo and Niedermeier [74]. On the contrary, CONNECTED VERTEX COVER does not admit a polynomial-size kernel in the general graphs unless the polynomial hierarchy collapses [48]. It is not hard to see that for a connected unit disk graph with n vertices, the size of a connected vertex cover is at least $n/12$ [113]. This leads to a trivial kernel of at most $12k$ vertices for the problem on the UDG class [79].

RED-BLUE DOMINATING SET.

Instance: A graph $G = (V, E)$, a non-negative integer k , a partition of the vertices into Red and Blue color classes: $R \subseteq V$ and $B \subseteq V$, with $R \cap B = \emptyset$ and $R \cup B = V$.

Question: Is there a subset $S \subseteq R$ with $|S| \leq k$ of red vertices such that every blue vertex $b \in B$ is adjacent to *at least* one vertex in S ?

The RED-BLUE DOMINATING SET on unit disk graphs is not easier than k -DOMINATING SET on the same class of graphs when parameterized by the size of the solution set. Indeed there is an easy reduction from the regular k DOMINATING SET to RED-BLUE DOMINATING SET problem by making a copy of every vertex and coloring one red and the original vertex blue. This makes RED-BLUE DOMINATING SET also $W[1]$ -hard on UDGs when the problem is parameterized by k . However, [79] shows that the situation is different when RED-BLUE DOMINATING SET is parameterized by the size of the smaller color class in which case the problem admits a polynomial-size kernel on unit disk graphs. This is in sharp contrast with the result of Dom et al. [48], which proves that the problem does not admit a polynomial-size kernel on general graphs when parameterized by $|R| + k$ or $|B| + k$ unless the polynomial hierarchy collapses.

Reduction rules.

- If there are vertices $v, w \in R$ such that $N(v) \cap B \subseteq N(w) \cap B$, delete v .
- If there are vertices $v, w \in B$ such that $N(v) \cap R \subseteq N(w) \cap R$, delete w .

To bound the size of the reduced graph, then one can use the following result by Jansen et al. [79].

Let G be a unit disk graph whose vertices are partitioned into two sets X and Y . If for all $v, w \in X$ it holds that $N(v) \cap Y \not\subseteq N(w) \cap Y$, then $|X| = \mathcal{O}(|Y|^2)$ [79].

PROPOSITION: RED-BLUE DOMINATING SET has a kernel with $\mathcal{O}(\min(|R|, |B|)^2)$ vertices on unit disk graphs.

H -MATCHING.

Instance: A Graph G , a connected graph H , an integer k .

Question: Does G contains at least k vertex-disjoint subgraphs that are isomorphic to H ?

For simplicity, let us denote $|V(H)|$ by h . In the class of general graph, H -MATCHING can be reduced to a kernel of size $\mathcal{O}(k^{h-1})$ [95]. Yet, there is a linear-size kernel for the problem on minor-free graphs according to the recent works of [65]. H -MATCHING on unit disk graphs has a kernel with $\mathcal{O}(k)$ vertices for every fixed H using the following argument of Jansen [79].

Reduction rule.

Delete all vertices in G that are not contained in an isomorphic copy of H .

Analysis.

Consider a maximal H -matching of G with vertex set S , containing k^* copies of H . Since the copies are disjoint, it is clear that k^*h vertices are included in the solution. Let $O = V \setminus S$ be the set of vertices not included in any of the copies of H chosen. Suppose a vertex $v \in V$ has more than $6(h-1)$ neighbors in O . Since G is a UDG, $G[N(v) \cap O]$ has a clique of size h and hence there is a copy of H in O , contradicting the maximality of the solution. Let d_H be the diameter of graph H , then for every vertex $v \in O$, v is at a distance at most d_H from some vertex $w \in S$. Using the bounded degree argument, the number of vertices in O with distance $i \leq d_h$ from some vertex in S is at most $6(h-1)^i |S|$. Adding this up we get $|V| = \mathcal{O}(k^*(6h)^{d_H})$. Therefore, for a solution of size k and a fixed H , after the reduction the number of vertices in graph is linear in k .

PROPOSITION: H -MATCHING admits a kernel of $\mathcal{O}((6h)^{d_H} \cdot k)$ vertices on unit disk graphs.

DEFINITION 2.4.2. [ϵ -precision][5]

A unit disk graph G with center of disks $P = \{p_1, \dots, p_n\}$ is called an ϵ -precision UDG, if for every pair of distinct points $p_i, p_j \in P$, the Euclidean distance between the two points $d(p_i, p_j)$ is at least ϵ . The class of such unit disk graphs are denoted UDG_ϵ .

The class of penny graphs (also known as unit coin graphs), are defined as UDG_1 , the 1-precision UDG.

Bounded-area UDGs

Intuitively, if the area of distribution of the centers of disks is known beforehand one can predict some density properties for UDG. Unit disk graphs whose vertices are located in a square-shaped region with fixed area α , are studied in [78]. When α is set as the parameter, there is a fixed-parameter algorithm for HAMILTONIAN CIRCUIT problem on unit

disk graphs. k -VERTEX COLORING is also proved to be in FPT when α and k are set as the parameter. Similarly, polynomial kernelization results can be obtained for the above problems with the bounded area restriction. On the other hand, two of the classical NP-hard problems, INDEPENDENT SET and DOMINATING SET remain in W[1] when given the bounded area restriction.

Let us denote the square-shaped region with fixed area α , by \mathcal{S} . We also denote by, $\text{UDG}_{\mathcal{S}}$, the class of unit disk graphs such that the vertices of the graph are placed in \mathcal{S} . Since the vertices of the graph inside any imaginary disk of diameter 1 has to induce a clique, it is not hard to see using a tiling argument that a unit disk graph on \mathcal{S} has a clique partition with bounded number of cliques depending on α only. This interesting property can be used to prove better results for specific problems on this class of graphs.

For example, Ito et al. [78] showed that if G is a unit disk graph on \mathcal{S} , then G has a clique-partition of size at most $p = 2\alpha + 2\sqrt{2\alpha} + 1 = 2\alpha + o(\alpha)$. Likewise, HAMILTONIAN CIRCUIT problem can be solved in time $\mathcal{O}(m + c^p p^2)$ assuming that G has m edges yet the constant c is large. Namely, $c = 18^{18}$. It is also possible to design an algorithm of running-time $\mathcal{O}(k^{k \cdot p})$ for solving k -COLORING on G [78].

On the other hand, when parameterized by α , there are fpt-reductions from α -INDEPENDENT SET to k -INDEPENDENT SET and from α -DOMINATING SET to k -DOMINATING SET making the problems as difficult as k -DOMINATING SET with respect to parameterized complexity.

PROPOSITION: DOMINATING SET and INDEPENDENT SET are in W[1] even on the class of $\text{UDG}_{\mathcal{S}}$ [78].

Chapter 3

Parameterized Algorithms for CLIQUE PARTITION on UDGs

In this chapter, we study the parameterization of CLIQUE PARTITION on UDGs.

CLIQUE PARTITION.

Instance: An undirected graph $G = (V, E)$, a non-negative integer k .

Question: Is there a set \mathcal{C} of cliques in G such that for every vertex $v \in V$, there exists at least a clique $c \in \mathcal{C}$ with $v \in c$ and $|\mathcal{C}| \leq k$?

3.1 Introduction

Beside its theoretical importance which we will discuss later in detail, CLIQUE PARTITION is proved to be a useful tool for solving other problems. For example, Pemmaraju et al. [99] describe how a clique partition with small number of cliques can be used to construct a large collection of mostly disjoint dominating sets. Small-sized clique partitions can also be used for calculating sparse spanners [86]. In the realm of unit disk graphs, there are techniques in the literature [100] that benefit from small-sized clique partitions as the main component of their algorithm for obtaining high-quality realizations of UDGs. Pandit et al. [98] show how CLIQUE PARTITION can be used to derive approximation algorithms for the FACILITY LOCATION problem on UDGs.

CLIQUE PARTITION is one of the Richard Karp's original 21 problems which were shown

to be NP-complete in his landmark paper [82]. On general graphs, CLIQUE PARTITION is equivalent to VERTEX COLORING of the complement graph and hence is not fixed-parameter tractable and is inapproximable within $n^{1-\delta}$ for any $\delta > 0$ [54]. Bhasker & Tariq [11] studied the number of cliques in a minimum clique partition and provided an upper bound of $\left\lfloor \frac{1+\sqrt{4n^2-4n-8m+1}}{2} \right\rfloor$ for this parameter which is also proved to be tight.

Supowit [107] first showed the NP-completeness of the problem on UDGs. Later, Cerioli et al. [22] proved that it remains NP-complete even when restricted to unit coin graphs, a subclass of ϵ -precision UDGs. Design of practical algorithms for CLIQUE PARTITION on UDGs has received a particular attention in the literature. There are PTAS algorithms for the problem on the class of UDGs due to the result of Dumitrescu & Pach in 2011 [54] and also the work of Pirwani & Salavatipour in 2010 [102]. The best known PTAS produces a $(1 + \delta)$ -approximate solution for the problem in $\mathcal{O}(n^{1/\delta^2})$ time [54] while the best approximation ratios for practical and randomized algorithms are 3 and 2.16 due to Cerioli et al. [22] and Dumitrescu & Pach [54] both running in $\mathcal{O}(n^2)$ time. One of only few classes of graphs for which CLIQUE PARTITION is known to be polynomially solvable is the class of ω -strip graphs [17] — when all points of a UDG lie in a parallel strip of width $\omega \leq \sqrt{3}/2$. Despite the importance of the problem, exact parameterized algorithms for CLIQUE PARTITION have not explicitly been studied in the literature. However, in [54], the authors describe an exact algorithm of running-time $\mathcal{O}(n^{80k})$ to calculate an optimal solution for the problem on a UDG restricted to a square of known width; here k is an upper bound on the number of cliques in an optimal solution.

Notice that CLIQUE PARTITION should not be confused with the slightly easier (and polynomially kernelizable) problem of (EDGE) CLIQUE PARTITION studied by Mujini & Rosamond [96] and Rees [103] where the goal is to partition the edges of a graph into a minimum number of cliques such that every edge is in *exactly* one clique in the partition.

3.2 Definitions and Notations

Let Σ be the 2-dimensional Euclidean plane. For two points p_1 and p_2 in Σ , we denote by $\mathbf{dist}(p_1, p_2)$ the Euclidean distance between the two points. For simplicity, when the presence of an edge or line segment is clear from the context, we alternatively refer to the length of the line segment (edge) p_1p_2 by $\|p_1p_2\|$. An *open region* on Σ is an arc-connected subset of Σ *homeomorphic* to the unit open disk $\{(x, y) | x^2 + y^2 < 1\}$. For an open region r ,

we denote by \bar{r} the closure of r , and by $\partial(r) = \bar{r} \setminus r$ the boundary of r . A region r is called *convex* if a straight line segment between any two points on the boundary of r is a subset of \bar{r} . For a point v of Σ , we define $\mathcal{D}(v) = \{w \mid \mathbf{dist}(v, w) \leq 1/2\}$ to be the disk with radius $1/2$ and center v , and $\partial\mathcal{D}(v) = \{w \mid \mathbf{dist}(v, w) = 1/2\}$ to be the boundary of $\mathcal{D}(v)$.

In what follows, we use G to refer to the unit disk graph (UDG) on n vertices and m edges. Given the input graph G , we consider a representation of G where disks of radius $1/2$ are centered at every vertex in $V(G)$ in the plane. In this case, we alternatively refer to $V(G)$ as point set $P = \{p_1, \dots, p_n\} \subset \Sigma$ with XY-coordinates, corresponding to the centers of disks. When we want to refer to a disk of radius $1/2$ centered at a vertex $v \in V(G)$, we denote it by $\mathcal{D}(v)$. Yet, whenever the center of the disk is unimportant in the context or the disk is not centered at a vertex of G , we refer to the disk by \mathcal{D}_r where r is the radius of the disk. Let $S \subseteq V(G)$ be a subset of vertices of G . We denote by $\Lambda_S = \cap_{v \in S} \mathcal{D}(v)$ and $\Phi_S = \cup_{v \in S} \mathcal{D}(v)$ the intersection and the union of disks of radii $1/2$ with centers in S , respectively. We define $Rg(S) = \Phi_S \setminus (\cup_{v \in S} \partial\mathcal{D}(v))$ to be the set of arc-connected regions obtained after removing from Φ_S , the boundary $\partial\mathcal{D}(v)$ of every $\mathcal{D}(v)$ with $v \in S$ and $Dg(S) = \{\{p\} \mid \{p\} = \Lambda_{S'}, S' \subseteq S\}$ to be the set of degenerate regions (points) generated by the single point intersection of disks corresponding to subsets of vertices in S . Finally, we define $R_S = Rg(S) \cup Dg(S)$. For a region $r \in R_S$, let $\mathcal{X}_G(r) = \{v \in V(G) \mid r \subseteq \mathcal{D}(v)\}$ be the set of all vertices of G such that the disks centered at them include r .

REMARK: Throughout the text while referring to the standard parameterization of a problem (k -II), if it is clear from the context, we remove the parameter k from the name of the problem. On the other hand, when we refer to the optimization version of the problem, we prefix the name with the relevant optimization term (MINIMUM-II or MAXIMUM-II).

REMARK: With a little abuse of the definition of region, we call each element of $Dg(S)$ a convex region. This is despite the fact that each element of $Dg(S)$ is a singleton set of one point.

3.3 Arbitrary UDGS

DEFINITION 3.3.1. [Region coverage graph]

The region coverage graph of G denoted by $\vec{G} = (\vec{V}, \vec{E})$ is defined as follows: Every region in $R_{V(G)}$ is represented as a vertex. Hence $\vec{V} = R_{V(G)}$. A directed edge $(r_1, r_2) \in \vec{E}$ is

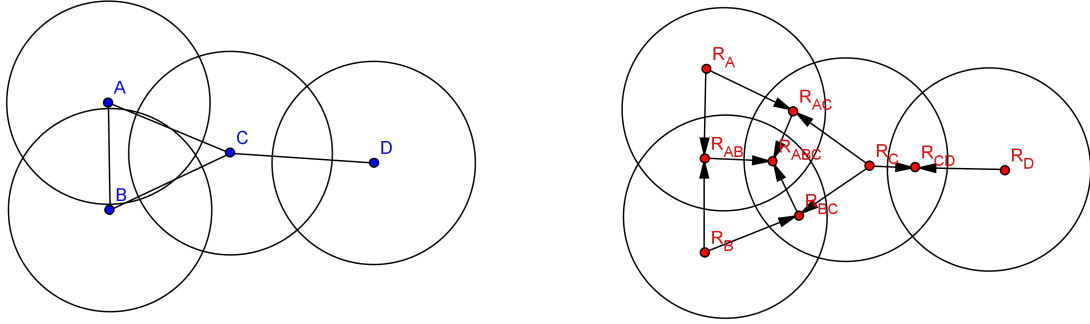


Figure 3.1: An arbitrary UDG on 4 vertices, its corresponding unit disk representation and region coverage graph.

present, if (1) $\bar{r}_1 \cap \bar{r}_2 \neq \emptyset$ and $\mathcal{X}_G(r_1) \subset \mathcal{X}_G(r_2)$, and (2) for any region $r \in \vec{V} \setminus \{r_1, r_2\}$ with $\mathcal{X}_G(r_1) \subset \mathcal{X}_G(r)$, $\mathcal{X}_G(r) \not\subset \mathcal{X}_G(r_2)$.

It is easy to observe that \vec{G} is a planar directed acyclic graph (DAG).

PROPOSITION 3.3.1. *For any UDG G , \vec{G} has at most $n(n - 1) + 1$ vertices and at most $3n(n - 1) - 3$ edges.*

Proof. We upper-bound $|\vec{V}|$ using an induction on the number of disks. Consider $m + 1$ disks in the plane and let us identify one of them (say the $(m + 1)$ th disk) by \mathcal{D} . Since all disks have the same radius, \mathcal{D} can cross the boundary of each of the remaining m disks at most twice. Notice that if more than two disks intersect at a single point then the number of regions would be smaller. Therefore, to upper-bound the number of regions, without loss of generality, we assume that at any intersection point at most two disks intersect. Let f_m be the total number of intersection points formed after intersecting the boundary of m such

disks. Then:

$$\begin{aligned} f_{m+1} &= f_m + 2m, \\ f_1 &= 0. \end{aligned}$$

Therefore,

$$f_n = n(n - 1).$$

Consider the planar multi-graph formed with the above intersection points as vertices and the connected curve on the boundary of the disks as edges. By the assumption that every vertex is created by intersection of exactly 2 disks, we conclude that the degree of every vertex is exactly 4. Hence, there are exactly $2f_n$ edges (arcs) in the graph. Then we can calculate the number of faces using the Euler formula for planar graphs as $f_n + 1$. The number of regions is therefore upper-bounded by $n(n - 1) + 1$. To upper-bound $|\vec{E}|$, let us first notice that although \vec{G} is a directed graph, it has no multiple edges. Therefore, $|\vec{E}|$ is equal to the maximum number of edges in a planar undirected graph which is upper-bounded by $3N - 6$ for a graph on N vertices. In the case of \vec{G} , we get this upper bound as $3n(n - 1) - 3$. \square

REMARK: A vertex v of \vec{G} is called a *sink* vertex if it has no outgoing edges. We denote by $\mathcal{S} \subset \vec{V}$ the set of all sink vertices in \vec{G} . A convex region $r \in R_{V(G)}$, corresponds to a sink vertex in \vec{G} and a clique in G . We call $G[\mathcal{X}_G(r)]$, a *especial clique* of G .

A recent paper of Fomin et al. [63] independently introduces a planar structure similar to the region coverage graph proposed here. However, the graph they define is undirected and unweighted and is used for a different purpose. In order to be able to use \vec{G} as a platform for calculating CLIQUE PARTITION, it is sufficient to show that for every maximal clique c in G , there can be at most a constant number of sink vertices in \vec{G} such that every vertex in c is included in at least one of the cliques corresponding to the sink vertices.

REMARK: The set of unit disks in the plane does not necessarily satisfy the Helly property (Definition 2.4.1). In particular, 3 unit disks can be placed in the plane such that they intersect pairwise, yet their mutual intersection is empty. With a little abuse of notation, we say a set of vertices (does not) satisfy the Helly property if the disks of radius $1/2$ centered at those vertices (does not) satisfy the Helly property.

LEMMA 3.3.1. *Every maximal clique c of a UDG G is included in the union of at most four especial cliques.*

Proof. We consider two cases. Let us first assume that any three vertices $\{x, y, z\} \subseteq V(c)$ satisfy the Helly property (Definition 2.4.1). Then by Helly's theorem [16, Problem 29, Intersecting Convex Sets: Helly's Theorem], the mutual intersection of set of disks of radius $1/2$ centered at $V(c)$, $\Lambda_{V(c)}$, is non-empty. In this case, one can simply find a special clique in G that contains c in the following way: Take an arbitrary point p in the mutual intersection $\Lambda_{V(c)}$. Find a region $r \in R_{V(G)}$ with $p \in \bar{r}$ and a convex region $s \in R_{V(G)}$ such that there is a path from vertex r to sink vertex s in \vec{G} . $G[\mathcal{X}_G(s)]$ is a especial clique in G containing c . For the proof argument to be comprehensive, here we also provide an alternative proof of Helly's theorem for disks in the plane. That is, we prove by induction that if for every subset $S \subseteq V(c)$ with $|S| = 3$, $\Lambda_S \neq \emptyset$, then $\Lambda_{V(c)} \neq \emptyset$.

For $|V(c)| = 3$, $\Lambda_{V(c)} \neq \emptyset$ by the Helly property. Assume that $\Lambda_{V(c)} \neq \emptyset$ for $|V(c)| = k-1$. We prove that $\Lambda_{V(c)} \neq \emptyset$ for $|V(c)| = k$. Let c' be a clique consisting of $k-1$ arbitrary vertices of c . By the induction hypothesis, $\Lambda_{V(c')} \neq \emptyset$. Let u be the vertex of $V(c) \setminus V(c')$. If $\mathcal{D}(u) \cap \Lambda_{V(c')} \neq \emptyset$ then $\Lambda_{V(c)} \neq \emptyset$. So we assume that $\mathcal{D}(u) \cap \Lambda_{V(c')} = \emptyset$. For vertex $v \in V(c')$ with $\partial\mathcal{D}(v) \cap \partial\Lambda_{V(c')} \neq \emptyset$, we call $\partial\mathcal{D}(v) \cap \partial\Lambda_{V(c')}$ a *boundary arc* of $\Lambda_{V(c')}$. Let $P = \{p | p \text{ is an intersection point of two boundary arcs of } \Lambda_{V(c')}\}$. Let q be a point of P such that the distance between q and u is minimum among all points of P . Assume that point q is formed by the intersection of the boundaries of disks centered at v and w . Since $\mathcal{D}(u) \cap \Lambda_{V(c')} = \emptyset$ and the distance between q and u is minimum for all points of P , $\mathcal{D}(u) \cap \mathcal{D}(v) \cap \mathcal{D}(w) = \emptyset$, a contradiction to the assumption that for every three vertices of $V(c)$ the Helly property holds.

Now let us assume that the Helly property does not hold for a triple $\{x, y, z\} \subseteq V(c)$. We prove that there are (not necessarily distinct) convex regions $r_1, \dots, r_4 \in R_{V(G)}$ such that $V(c) \subseteq \cup_{i=1}^4 \mathcal{X}_G(r_i)$. Let us denote by ζ_x, ζ_y and ζ_z , disks of radius 1 centered at x, y and z , respectively and let $\Omega = \zeta_x \cap \zeta_y \cap \zeta_z$ (See Figure 3.2). Since any disk $\mathcal{D}(u)$ with $u \in V(c) \setminus \{x, y, z\}$ has to intersect $\mathcal{D}(x), \mathcal{D}(y)$ and $\mathcal{D}(z)$, u has to lie in Ω .

Next we show that if one can cover Ω with at most t arbitrary disks of radius $1/2$ in the plane, for t a constant integer, then the clique c is contained in at most t especial cliques. Let $\{p_1, \dots, p_t\}$ be arbitrary points in the plane and let $\mathcal{D}(p_i)$, $1 \leq i \leq t$, denote a

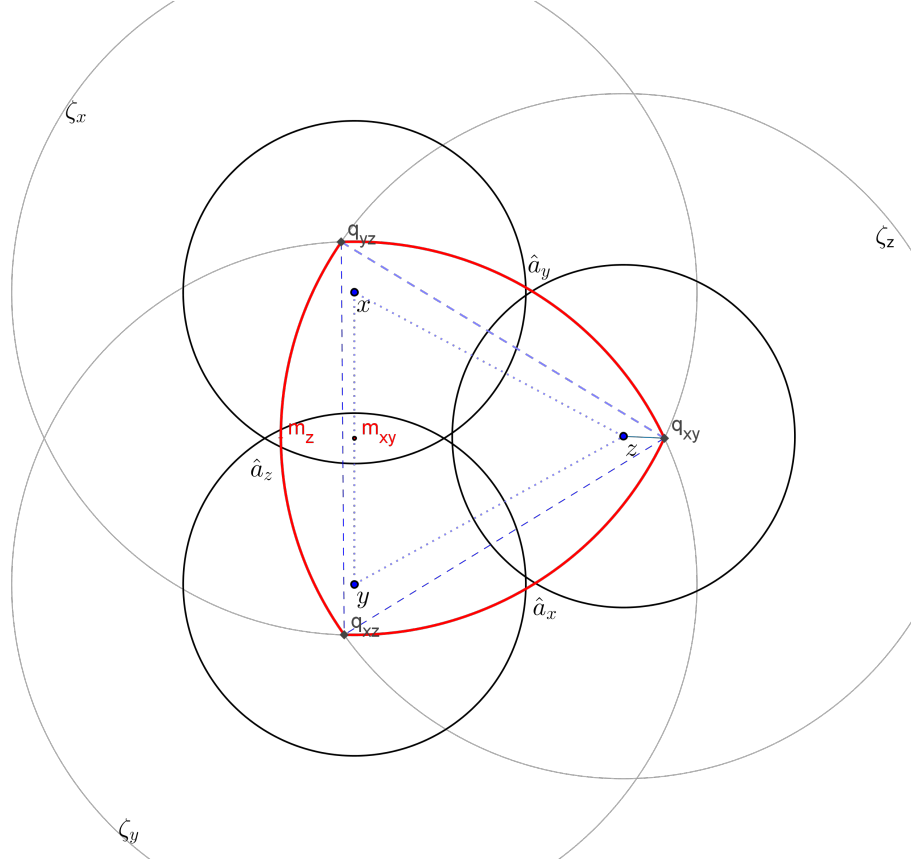


Figure 3.2: An arbitrary placement of disks ζ_x, ζ_y and ζ_z with centers x, y and z such that Helly property does not hold. Points q_{xy}, q_{yz} and q_{xz} are marked on the boundary of $\Omega = \zeta_x \cap \zeta_y \cap \zeta_z$.

disk of radius $1/2$ centered at point p_i . If $\Omega \subseteq \cup_{i=1}^t \mathcal{D}(p_i)$, then for every $u \in V(c)$, there is a point $p \in \{p_1, \dots, p_t\}$ such that $\mathbf{dist}(u, p) \leq 1/2$. Without loss of generality, assume that $p = p_1$. Consider a region r_1 in $R_{V(G)}$ such that $p \in \bar{r}_1$. Since $\mathbf{dist}(u, p) \leq 1/2$, $u \in \mathcal{X}_G(r_1)$. The region r_1 might not be convex but one can traverse the edges in \vec{G} , starting from vertex r_1 to reach a sink vertex. Let s_1 be a sink vertex in \vec{V} reachable from r_1 in \vec{G} (hence $s_1 = r_1$ if r_1 is convex). The region corresponding to s_1 is convex. Also, since $\mathcal{X}_G(r_1) \subseteq \mathcal{X}_G(s_1)$, it also holds that $u \in \mathcal{X}_G(s_1)$. In a similar way, we can find a convex region s_i if $p = p_i$ for $1 < i \leq t$. Therefore, we can find t especial cliques $G[\mathcal{X}_G(s_i)]$ for $1 \leq i \leq t$ such that $V(c) \subseteq \cup_{i=1}^t \mathcal{X}_G(s_i)$. Clearly, the condition above holds for some value of t .

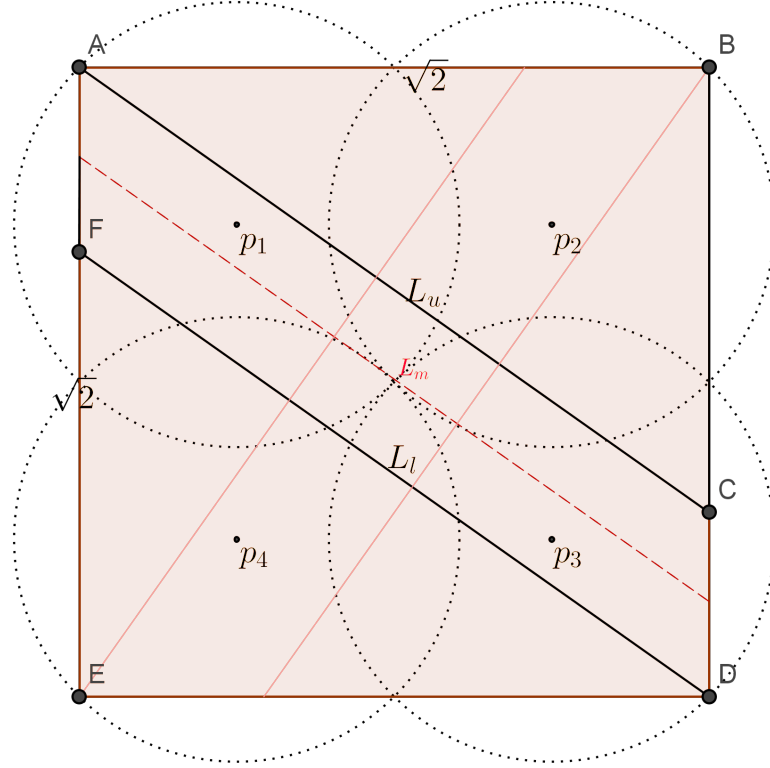


Figure 3.3: Coverage of a square of length $\sqrt{2}$, denoted $S_{\sqrt{2}}$, by 4 disks of radius $1/2$. Line segments L_u and L_l have the length $\sqrt{3}$.

Finally, we prove that there exist four points p_1, p_2, p_3, p_4 such that $\Omega \subseteq \cup_{i=1}^4 \mathcal{D}(p_i)$. To do so, we assume that the points p_1, p_2, p_3, p_4 are placed at the 4 corners of a square of width $\sqrt{2}/2$. Notice that with the above placement, 4 disks of radius $1/2$ centered at p_i , $1 \leq i \leq 4$, cover a square of width $\sqrt{2}$ (See Figure 3.3). Let us denote by $S_{\sqrt{2}}$, this square after possible translations and rotations in the plane (The position of $S_{\sqrt{2}}$ is determined by the coordinates of p_1, p_2, p_3, p_4). Therefore, to complete the proof, we show that one can place $S_{\sqrt{2}}$ such that $\Omega \subset S_{\sqrt{2}}$. Let us denote by q_{xy} the intersection point of the circles of radius 1 at x and y that lies on the boundary of Ω . That is, $q_{xy} = \partial\zeta_x \cap \partial\zeta_y \cap \zeta_z$. Similarly, we define $q_{yz} = \partial\zeta_y \cap \partial\zeta_z \cap \zeta_x$ and $q_{xz} = \partial\zeta_x \cap \partial\zeta_z \cap \zeta_y$. See Figure 3.2. Let us also denote by \hat{a}_x the arc on the boundary of Ω between q_{xy} and q_{xz} . Similarly, we define \hat{a}_y and \hat{a}_z .

Without loss of generality, assume that xy is the shortest among $\{xz, yz, xy\}$. Let $\sigma = \|xy\|$. Notice that if σ is close to zero because the Helly property does not hold for $\{x, y, z\}$, the distance of z from both x and y is close to 1.

Let m_z be the point on \hat{a}_z with maximum distance from q_{xy} (See Figure 3.2) and let $h_z = \|q_{xy}m_z\|$. To show that $\Omega \subset S_{\sqrt{2}}$, we upper-bound the values of h_z and $\|q_{xz}q_{yz}\|$. $\|q_{xz}q_{yz}\|$ is maximized when $x = y$. In this case, $\|q_{xz}q_{yz}\| = \sqrt{3}$ and h_z has the global minimum value of 1. For $\|xy\| > 0$, as the distance between x and y increases, the value of $\|q_{xz}q_{yz}\|$ decreases and the value of h_z increases. To obtain an upper bound on h_z , notice that h_z is maximized when $\|xz\|$ (alternatively $\|yz\|$) is minimized. Yet, since $\sigma = \|xy\|$ is assumed to be at most $\|xz\|$ (alternatively $\|yz\|$), h_z is maximized when $\|xy\| = \|xz\| = \|yz\| = \sqrt{3}/2$ (This is obtained from the disks in Figure 3.2 by pulling the centers $\{x, y, z\}$ closer together so that they mutually intersect at a single point). In which case, the value of h_z can be easily calculated as $\sqrt{13}/4 + 1/4 < \sqrt{3}$. To see a proof, observe that in this configuration, both z and q_{xy} are on the perpendicular bisector of xy . Let m_{xy} be the middle point of line segment xy . The length of line segment $q_{xy}z$ is calculated as follows:

$$\|q_{xy}z\| = \|q_{xy}m_{xy}\| - \|zm_{xy}\| = \sqrt{1 - 3/16} - \sqrt{3/4 - 3/16} = \sqrt{13}/4 - 3/4. \quad (3.1)$$

Since the triangle $\triangle xyz$ is equilateral, triangle $\triangle q_{xy}q_{yz}q_{xz}$ is similar to $\triangle xyz$ and the perpendicular to the side xy is also a perpendicular to line segment $q_{xz}q_{yz}$. Finally, since the distance of z from any point of \hat{a}_z is exactly 1, we obtain $h_z = 1 + \|q_{xy}z\| = \sqrt{13}/4 + 1/4 \approx 1.15$. See Figure 3.2 for details.

Consider a line segment from the upper-left corner of $S_{\sqrt{2}}$ (point A in Figure 3.3) to a point 1 unit distance below the upper-right corner of the square on its boundary (point C in Figure 3.3). Call it L_u . Observe that $\|L_u\| = \sqrt{3}$. We also define L_l as the line segment between the lower-right corner of $S_{\sqrt{2}}$ (point D in Figure 3.3) and a point 1 unit distance above the lower-left corner of the square on its boundary (point F in Figure 3.3). Clearly, $\|L_l\| = \sqrt{3}$ and L_u and L_l are parallel. Notice that one can slide L_u downwards till it reaches L_l . Indeed, there are infinitely many such parallel line segments. Let us denote the set of all these parallel line segments by \mathcal{L} . We also define \mathcal{L}^\perp as the set of line segments perpendicular to line segments in \mathcal{L} and have one end point on the upper side (AB

in Figure 3.3) and the other end point on the lower side (ED in Figure 3.3) of $S_{\sqrt{2}}$. Clearly, $\forall L^\perp \in \mathcal{L}^\perp, \|L^\perp\| = \sqrt{3}$.

For any given configuration of the points $\{x, y, z\}$, we place Ω in $S_{\sqrt{2}}$ by positioning the line segment $q_{xz}q_{yz}$ on a line segment $L_m \in \mathcal{L}$ with the middle point of L_m matching the middle point of $q_{xz}q_{yz}$ such that Ω is placed entirely in $S_{\sqrt{2}}$. In particular, when $x = y$, this is possible since $h_z = 1$ while the length of any $L^\perp \in \mathcal{L}^\perp$ is $\sqrt{3}$. As σ increases, we gradually slide L_m towards L_l so that q_{xy} is placed inside $S_{\sqrt{2}}$. Notice that one can always find a line segment $L_m \in \mathcal{L}$ satisfying the situation. This is because the perpendicular distance of the upper side of $S_{\sqrt{2}}$ from the center of L_l is calculated as ≈ 1.11 while the perpendicular distance of q_{xy} from the line segment $q_{xz}q_{yz}$ is at most $3\|q_{xy}z\|/2 + \|zq_{xy}\| < 3(\sqrt{13} - 1)/8 \approx 0.97$. Therefore, L_m can be chosen such that it is located above L_l in the drawing and hence $L_m \in \mathcal{L}$. This brings us to the conclusion that regardless of the position of $\{x, y, z\}$, one can ensure that $\Omega \subset S_{\sqrt{2}}$. Finally, knowing that $S_{\sqrt{2}} \subset \cup_{i=1}^4 \mathcal{D}(p_i)$, we get $\Omega \subset \cup_{i=1}^4 \mathcal{D}(p_i)$.

We also provide an alternative proof which considers coverage of Ω by a rectangle whose width and height change dynamically based on the shape of Ω . Initially, the four disks are centered at the four corners of a rectangle of width $w = \sqrt{3}/2$ and height $h = 1/2$. Let us denote this rectangle by $\text{Rec}_{w,h}$. The four disks in this configuration then cover a square of width $2w = \sqrt{3}$ and height $2h = 1$. If we assume that $\sigma = 0$, then $\|q_{xz}q_{yz}\| = \sqrt{3}$ and $h_z = 1$. Therefore Ω is easily covered by the rectangle of width $2w$ and height $2h$. As the position of vertices $\{x, y, z\}$ changes, we move the position of disks centered at the corners of rectangle such that the four disks cover a rectangle of longer height and shorter width. In particular, if the width of the covered rectangle is decreased by a value ϵ (this happens when the center of disks on a common horizontal side are each pushed $\epsilon/4$ distance closer towards each other), the center of disks on vertical sides can be each pulled $\epsilon'/4$ units away from each other. Therefore, the new rectangle would be $\text{Rec}_{w+\epsilon/2, h+\epsilon'/2}$. See Figure 3.4. In that configuration, for a rectangle $\text{Rec}_{2w+\epsilon, 2h+\epsilon'}$ is covered by the four disk, one can calculate ϵ' easily as:

$$\epsilon' = \sqrt{4 - (\sqrt{3} - \epsilon)^2}. \quad (3.2)$$

Now let us assume that $\sigma > 0$. In which case, by the proof provided in the Appendix A,

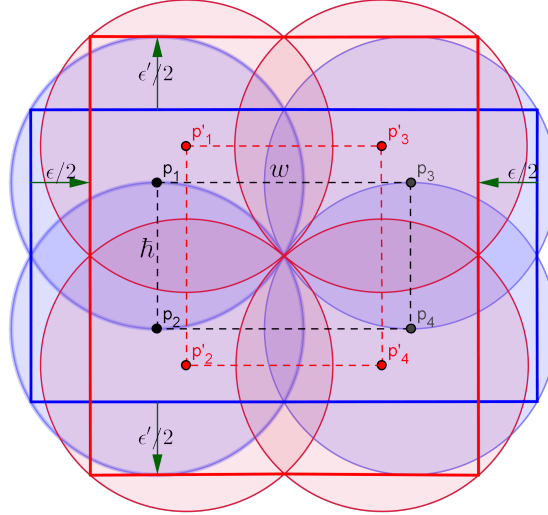


Figure 3.4: Points $\{p_1, \dots, p_4\}$ are originally placed at the four corners of an axis-aligned $1/2 \times \sqrt{3}/2$ square (drawn in dashed black). The corresponding disks cover an area including a $1 \times \sqrt{3}$ rectangle (drawn in solid blue). As $|q_{xy}q_{xz}|$ decreases, the horizontally aligned pairs (also the vertically aligned pairs) of points are shifted $\epsilon/2$ units towards (respectively $\epsilon'/2$ units away from) each other horizontally (respectively vertically), such that their corresponding disks cover an area including a $(1 + 2\epsilon') \times (\sqrt{3} + 2\epsilon)$ rectangle (drawn in solid red).

one can calculate the length of $q_{xz}q_{yz}$ as

$$\|q_{xz}q_{yz}\| < \sqrt{3} + \frac{\sigma}{\sqrt{3}} + \frac{\sigma^2}{24\sqrt{3}} + \frac{5\sigma^3}{36\sqrt{3}} + \frac{53\sigma^4}{3456\sqrt{3}}. \quad (3.3)$$

If we fix $\epsilon = \|q_{xz}q_{yz}\| - \sqrt{3}$, then the width of rectangle $\text{Rec}_{2w+\epsilon, 2h+\epsilon'}$ would be at least the length of $q_{xz}q_{yz}$. Also, in this configuration, $h_z \leq 2 - \sigma'$. Finally, using the previous equation for ϵ' , one can calculate ϵ' as

$$\epsilon' > \sigma - \frac{5\sigma^2}{8} + \frac{3\sigma^3}{4} - \frac{125\sigma^4}{128}. \quad (3.4)$$

By comparing, $2\bar{h} + \epsilon'$, height of the new rectangle with $2 - \sigma'$, the upper bound for h_z , we conclude that $2\bar{h} + \epsilon' \geq 2 - \sigma'$ when $0 < \sigma < \sqrt{3}/2$. This completes the proof. \square

3.3.1 A Parameterized Algorithm

Finally, we describe how a parameterized algorithm on arbitrary UDGs can be obtained using the tools introduced earlier. Notice that our goal here is to demonstrate a new technique and indeed the overall running-time of the algorithm can be enhanced with a more detailed analysis of various steps of the technique that is discussed below.

Imagine an optimal solution $\mathcal{C} = \{c_1, \dots, c_k\}$ for CLIQUE PARTITION on G . By Lemma 3.3.1, every maximal clique c_i , $1 \leq i \leq k$, can be represented by a constant number of convex regions in $R_{V(G)}$. To solve CLIQUE PARTITION, we run a brute-force search for a set R' of convex regions (special cliques) that forms a clique partition of size at most k . For any such a subset R' , then one can check in $\text{poly}(n, k)$ if the number of cliques corresponding to it is at most k and whether $V(G)$ is covered by those cliques as described later. We first maintain an inclusion matrix whose rows correspond to all convex regions in $R_{V(G)}$ and its columns represent the vertices of G . For every region $r \in R_{V(G)}$, in the respective row in the matrix, we set the value of elements at columns corresponding to vertices in $\mathcal{X}_G(r)$ to 1 and the rest to 0. We construct this matrix only one time for the whole graph.

DEFINITION 3.3.2. [Non-helly set]

We call a subset of 3 convex regions $\{r_1, r_2, r_3\} \subseteq R_{V(G)}$ a non-helly set, if for all 3 permutations $1 \leq i, j, l \leq 3$ and $i \neq j \neq l$, we have $Y_l = \mathcal{X}_G(r_i) \cap \mathcal{X}_G(r_j) \setminus \mathcal{X}_G(r_l) \neq \emptyset$.

If there exists a non-helly set, then there is a clique with vertex set $Y = \cup_{i=1}^3 Y_i$ in G which is not represented by any of the regions in $R_{V(G)}$. Hence, we need to manually add a new region r' with vertex set Y to the set of regions. To take care of the non-helly sets, we perform a pre-processing on the inclusion matrix. In particular, we check in any subset of 3 regions of $R_{V(G)}$ for non-helly sets. If we find one, we add a new row corresponding to the clique that the non-helly region represents. In particular, let \vec{r}_i denote the binary vector corresponding to convex region r_i in the matrix for $1 \leq i \leq 3$. To check for a non-helly set in $\{r_1, r_2, r_3\}$, we simply check if $(\vec{r}_i \wedge \vec{r}_j \wedge \neg \vec{r}_l) \neq \vec{0}$ for $1 \leq i, j, l \leq 3$ and $i \neq j \neq l$. This can be done in $\mathcal{O}(n)$ for one set of 3 regions. There are at most $|R_{V(G)}|^3 - 3|R_{V(G)}|^2 + 2|R_{V(G)}| = \mathcal{O}(|R_{V(G)}|^3)$ 3-subsets. Therefore, the pre-processing phase would take $n \cdot |R_{V(G)}|^3 = \mathcal{O}(n^7)$ time for the whole matrix. Notice that any time a non-helly set is detected, at most one new entry is added to the set of regions. Therefore, after the pre-processing there can be at most

$|R_{V(G)}|^3$ regions.

Knowing that there are at most $n^2 - n + 1$ regions in $R_{V(G)}$ (by Proposition 3.3.1), there are at most

$$\binom{(n^2 - n + 1)^3}{k'}$$

subsets of size k' for $k' \leq k$ to be considered. Therefore, total number of subsets of size at most k that we need to consider is:

$$\sum_{k' \leq k} \binom{(n^2 - n + 1)^3}{k'} \leq ((n^2 - n + 1)^3 + 1)^k \leq n^{6k} \text{ for } n > 2. \quad (3.5)$$

Given any of the above subset R' of convex regions, we finally check if R' corresponds to a valid clique partition in the following way: Using the inclusion matrix, we can check if R' includes every vertex in $V(G)$ by simply summing up the corresponding rows in the binary matrix. If that is the case, we answer ‘Yes’. This can be done using $n \cdot |R'| = \mathcal{O}(kn)$ binary operations. If after checking all subsets, none satisfied the criteria, we output ‘No’. Therefore, the overall running-time of our algorithm would be:

$$n^{6k} \cdot \mathcal{O}(kn) + \mathcal{O}(n^7) = \mathcal{O}(kn^{6k+1}) = \mathcal{O}(n^{6k+2}) \text{ for } n > 2. \quad (3.6)$$

THEOREM 3.3.1. *There is an algorithm that given any integer k , decides in $\mathcal{O}(n^{6k+2})$ time, if there exists a clique partition of size at most k for an arbitrary UDG.*

To obtain a clique partition of minimum cardinality, then it is enough to perform a binary search over the range $1 \leq k \leq n$. This can be done in total time $\mathcal{O}(n^{6n+2} \log n)$.

COROLLARY 3.3.1. *There is an algorithm that in $\mathcal{O}(n^{6n+2} \log n)$ time constructs an optimal solution for MINIMUM CLIQUE PARTITION for any UDG.*

Although to the best of our knowledge parameterization of the problem has not been explicitly studied with respect to UDGs, Dumitrescu & Pach, in [54, Section 2] calculate an exact algorithm of running-time $\mathcal{O}(n^{80q})$ for the problem on UDGs whose points are located inside a square of known length. Here, q is an upper bound on the number of cliques in the partition and is a function of the area of distribution of points and therefore it is not straight-forward to extend their results with the same time-complexity to the arbitrary UDGs.

3.4 Data Reduction

We finally propose data reduction rules for CLIQUE PARTITION on arbitrary UDGs. In particular, we use the properties of UDG to get rid of some parts of the graph in a pre-processing phase. This can later be combined with a detailed analysis and other techniques in order to obtain fixed-parameter algorithms for the problem.

The intuition behind Rule 1 and Rule 2 is that since every vertex of the graph has to be included in a clique of an optimal clique partition, if for a vertex v of the graph, there is only one clique c which covers v , then c has to be in any optimal solution. Similarly in Rule 3, for $u, v \in V(G)$, if for every maximal clique c either both u and v are included in c or both are excluded from c , every clique in an optimal solution which includes u also includes v and vice versa. Therefore, one can safely remove v , as long as u is kept.

Rule 1. Delete an isolated vertex or a vertex adjacent only to covered vertices in G and decrease k by 1.

Rule 2. For a vertex included in one maximal clique c in G only, mark every vertex in c as covered and decrease k by 1.

Rule 3. For a set of maximal cliques $\mathcal{C} = \{c_1, \dots, c_\gamma\}$ in G , let $I = \bigcap_{i=1}^\gamma V(c_i)$. Assume that $|I| \geq 2$ and $N[I] = \bigcup_{i=1}^\gamma V(c_i)$; that is, no other maximal cliques in G intersect I . We replace I in G with a vertex x and make x adjacent to $N(I)$.

A data reduction rule is *correct* if the new instance after an application of this rule is a *yes*-instance if and only if the original instance is a *yes*-instance.

THEOREM 3.4.1. *Rules 1 to 3 are correct for CLIQUE PARTITION on a UDG and any of the rules can be performed in $\mathcal{O}(n^{2.3727} + nm)$ time on any graph G with n vertices and m edges.*

Proof. Given a UDG G , in the form of an adjacency matrix, for Rule 1, finding isolated vertices or those only adjacent to covered vertices can be done by checking every edge of G in $\mathcal{O}(m)$ time for the whole graph. Let G' be the graph obtained after applying Rule 1 to a vertex v in G , then the fact that G is a UDG implies that G' is a UDG. Now we prove the correctness of Rule 1. Suppose (G, k) is a *yes* instance and consider an optimal solution OPT for k -CLIQUE PARTITION on G . Then $G[v] \in \text{OPT}$. Given any optimal solution OPT' for G' , $\text{OPT}' \cup G[v]$ is a solution for G . If $|\text{OPT}'| \leq |\text{OPT}| - 1$, then OPT' is an optimal solution for

G and we are done. Assume $|\text{OPT}'| > |\text{OPT}| - 1$. However, $\text{OPT} \setminus G[v]$ is a solution for G' and $|\text{OPT} \setminus G[v]| < |\text{OPT}'|$, a contradiction.

For Rule 2, in order to find out whether a vertex is included in one maximal clique c of G only, we need to check for every vertex $v \in V(G)$ if $G[N[v]]$ is a clique. This can be done naively by checking the presence of all edges in $G[N[v]]$ in $\deg^2(v)$ time for v . In this way, one round of application of Rule 2 to G would take $\mathcal{O}(nm)$ time. Notice that the vertices removed according to Rule 1 and Rule 2 are *simplicial* vertices and hence recognizing all such vertices can be done using fast matrix multiplication [83]. In particular, let \mathbf{A} be the adjacency matrix of G with 1 entries on its diagonal. To recognize all simplicial vertices, one needs to calculate \mathbf{A}^2 . This can be performed in $\mathcal{O}(n^{2.3727})$ using the algorithm of V. Vassilevska Williams in 2012 [114]. Once a simplicial vertex x is removed from the graph, one can recalculate \mathbf{A}^2 in $\mathcal{O}(\deg^2(x))$ time. Therefore, exhaustive application of Rules 1 and 2 takes $\mathcal{O}(n^{2.3727} + nm)$ in total.

Rule 2 also does not change the structure of the graph. Let G' be the graph obtained after applying Rule 2 to G . Consider an optimal solution OPT for k -CLIQUE PARTITION on G and let c_v be the clique in OPT that includes v . Let OPT' be an optimal solution for G' . Then $\text{OPT}' \cup c_v$ is a solution for G . If $|\text{OPT}' \cup c_v| \leq |\text{OPT}|$, then we are done. Assume $|\text{OPT}' \cup c_v| > |\text{OPT}|$. Since $\text{OPT} \setminus c_v$ is a solution for G' and $|\text{OPT}'| > |\text{OPT} \setminus c_v|$, we get a contradiction.

For Rule 3, we need to construct the set I in the neighborhood of every vertex v . This can be done by checking for a maximum subset of vertices $I \subseteq N[v]$ such that for every vertex $x \in I$, $N[x] = N[v]$. Given G in form of an adjacency matrix, one can construct this subset in the following way. Initially, we set $I = \{v\}$. For every 1 entry in the row corresponding to vertex v , we check if the row corresponding to that entry is exactly the same as the one for v and if it is the case, we add the corresponding vertex to I . This can be done in $\deg^2(v)$ time for v and in $\mathcal{O}(nm)$ time for the entire graph. Similarly, this step can be done using fast matrix multiplication. Let \mathbf{A} be the adjacency matrix of G with 1 entries on its diagonal. Notice that for every pair of vertices $x, y \in V(G)$, $(\mathbf{A}^2)_{x,y} = |N[x] \cap N[y]|$. To decide if x and y are in I , it is sufficient to check if $|N[x] \cap N[y]| = |N[v]| = |N[y]|$.

Vertices x and y are in I , if $(\mathbf{A}^2)_{x,y} = (\mathbf{A}^2)_{x,x} = (\mathbf{A}^2)_{y,y}$. Contracting I into a single

vertex can be done in constant time. Hence, Rule 3 takes $\mathcal{O}(n^{2.3727})$ using the matrix multiplication algorithm reported in [114]. Notice that if Rules 1 and 2 are applied exhaustively to the input graph, application of Rule 3 does not generate any new reducible vertices (with respect to any of the reduction rules) and hence only one round of the application of Rule 3 is sufficient.

It remains to show that after applying the reduction Rule 3, the graph can still be represented as a UDG. To see this, notice that $G[I]$ is a clique and hence contracting I to a vertex corresponds to removing all the vertices in I except for one. The UDG for the reduced graph is then obtained after removing the respective points.

If Rule 3 is applied, then there exists a collection of maximal cliques $\mathcal{C} = \{c_1, \dots, c_\gamma\}$ in G such that $N[I] = \cup_{i=1}^\gamma V(c_i)$. Let G' be the graph obtained after contracting $G[I]$ into a vertex x and OPT be an optimal solution for CLIQUE PARTITION on G . Notice that the size of every clique in $\mathcal{C}/G[I]$ is at least 2. We prove that $\text{OPT}' = \text{OPT}/G[I]$ is an optimal solution for G' . For the sake of contradiction, assume that G' has a solution OPT^* with $|\text{OPT}^*| < |\text{OPT}'|$ and let $\mathcal{C}^\dagger = \{c_j \in \mathcal{C} \mid c_j/G[I] \in \text{OPT}^*\}$. Then one can construct a solution $\text{OPT}^\ddagger = \text{OPT}^* \setminus (\mathcal{C}^\dagger/G[I]) \cup \mathcal{C}^\dagger$ for G such that $|\text{OPT}^\ddagger| = |\text{OPT}^*|$. However, since $|\text{OPT}| = |\text{OPT}'|$, we get $|\text{OPT}^\ddagger| < |\text{OPT}|$, a contradiction.

Now assume that OPT' is an optimal solution for G' and let $c_x \in \text{OPT}'$ be the clique that includes x . Then $G[(V(c_x) \setminus x) \cup I]$ is a clique in G . We prove that $\text{OPT}^* = (\text{OPT}' \setminus c_x) \cup G[(V(c_x) \setminus x) \cup I]$ is an optimal solution for G . That is, $|\text{OPT}^*| \leq |\text{OPT}|$. Assume for the contradiction that $|\text{OPT}^*| > |\text{OPT}|$. Then $\text{OPT}^\ddagger = \text{OPT}/G[I]$ is a solution for G' and $|\text{OPT}^\ddagger| = |\text{OPT}| < |\text{OPT}'|$, a contradiction to the assumption that OPT' is an optimal solution for G' . \square

COROLLARY 3.4.1. *Let G' be the graph obtained from G after repeated application of Rules 1-3. Then G' is an induced subgraph of G .*

Recall that given a graph $G = (V, E)$ and $S \subseteq V$, $\partial G(S)$ is defined as the set of vertices in S that have a neighbor in $V \setminus S$. For a set $S \subseteq V$ the neighborhood of S is $N_G(S) = \partial G(V \setminus S)$.

LEMMA 3.4.1. *Let G be an arbitrary UDG, for any maximal clique c of size $t > 1$ in the reduced graph G' it holds that $|N_{G'}(V(c))| \geq \lfloor \sqrt{t} \rfloor$*

Proof. First of all notice that for every vertex $v \in V(c)$, there is a vertex $v' \in N_{G'}(V(c))$ adjacent to v . This is since, if v was not adjacent to any vertex outside c , then c would have

been the only maximal clique covering v and hence c would have been removed by Rule 2. Since G' is reduced with respect to Rule 3, for every pair of vertices $v, w \in V(c)$, it holds that $N[v] \neq N[w]$. Because otherwise, $G'[\{v, w\}]$ would have been contracted to one vertex by Rule 3. Also since v and w are vertices of the clique c , $N[v] \cap V(c) = N[w] \cap V(c)$. It follows that $N[v] \cap N_{G'}(V(c)) \neq N[w] \cap N_{G'}(V(c))$. Consider a set of circles of radius 1 centered at vertices of $N_{G'}(V(c))$. Vertices v, w in $V(c)$ have different neighborhood iff the line segment vw intersects at least one of the circles. Since this holds for every pair of vertices of c , $|V(c)| \leq |R_{N_{G'}(V(c))}|$. Let $s = |N_{G'}(V(c))|$. By Proposition 3.3.1, there can be at most $s^2 - s + 1$ regions created by the above circles. That is, $t \leq s^2 - s + 1$. Thus, $s \geq \lfloor \frac{1}{2} (\sqrt{4t - 3} + 1) \rfloor \geq \lfloor \sqrt{t} \rfloor$. \square

3.5 Summary

We introduced a framework for parameterization of clique covering problems on unit disk graphs (UDGs). Our exact parameterized algorithm for CLIQUE PARTITION on arbitrary UDGs relies on a novel idea and a geometric theorem that demonstrates the use of convex regions for guessing clique partitions. This results in an algorithm of running-time $\mathcal{O}(n^{6k+2})$ on arbitrary UDGs. The only previously known singly-exponential exact algorithm for the problem is restricted to UDGs whose points are within a square of known length and has the running-time of $\mathcal{O}(n^{80q})$ where q is an upper bound on the size of clique partition [54]. Finally, we came up with a first-time data reduction rules for the CLIQUE PARTITION on UDGs that shrinks the input graph in $\mathcal{O}(n^{2.3727} + mn)$ time to a reduced instance. We further proved structural properties for the reduced graph.

Chapter 4

FPT Algorithms on Precision and Quasi-precision UDGs

4.1 Introduction

It is known that the complexity of many graph optimization problems remains NP-hard even when restricted to the inputs that can be realized as UDGs [26]. Similarly, many such problems are believed not to admit polynomial-size kernelization or fixed-parameter algorithms on UDGs. In an attempt to study the problems on more tractable classes of UDGs, ϵ -precision [5] and bounded-area UDGs [78] were introduced in the literature. Yet, in many cases such subclasses appear to be too trivial. For example, ϵ -precision UDGs immediately translate to graphs of bounded degree (See the discussion in Section 4.2). On the other hand, bounded-area UDGs are known to have bounded clique partition size [78] and as a result, a number of problems such as k -COLORING and HAMILTONIAN CIRCUIT are easily solvable on UDGs of bounded area.

In this chapter, we describe how one can get the best of both worlds by introducing the new class of (α, λ) -quasi-precision UDGs as UDGs which are only partially λ -precision, yet their imprecision components have bounded area. We show that (α, λ) -quasi-precision UDGs are *ubiquitous* in the UDG class in the sense that any UDG is quasi-precision for some values of λ and α . In particular, precision UDGs and bounded-area UDGs are derived as trivial cases of quasi-precision class. Quasi-precision UDGs are also *non-trivial* meaning that in their general setting, many of the known graph parameters are proved to be unbounded

on this class.

In order to demonstrate the usefulness of quasi-precision UDGs, we describe an approach for obtaining fixed-parameter algorithms on quasi-precision UDGs. We showcase this by studying the problem of CLIQUE PARTITION. We start building up the techniques by first devising fixed-parameter algorithms for the problem on the more restricted and well-known class of precision UDGs. Later, we use the approach on precision UDGs as a base and extend the techniques in order to obtain fixed-parameter results for the problem on (relaxed)-quasi-precision UDGs. Our framework further applies when the reduced graph (rather than the input graph) belongs to the discussed subclasses of UDGs.

We further study the structure of quasi-precision UDGs and show structural obstructions for obtaining FPT algorithms under our framework. In particular, we prove that UDGs that do not admit FPT under our framework, have to include as subgraph an arbitrary large instance of the obstruction in the reduced graph.

4.2 Precision UDGs

In this section, we study the problem of CLIQUE PARTITION with respect to ϵ -precision UDGs. In particular, we prove that if the reduced graph is ϵ -precision, it is possible to obtain fixed-parameter algorithms for the problem.

A UDG G is ϵ -precision if for any two vertices of G , their pairwise Euclidean distance is at least ϵ . Notice that for an ϵ -precision UDG, the maximum degree of graph and hence the size of any clique is upper-bounded by $\mathcal{O}(1/\epsilon^2)$. This leads to a trivial bound on the size of the input graph in terms of k . However, if the input graph is used directly, since the number of maximal cliques can be exponential in $(1/\epsilon)$ and k , a straight-forward algorithm for the problem which relies on the enumeration of the cliques would require a running-time which is doubly-exponential in k .

In our approach, on the other hand, we circumvent this issue by instead enumerating sets of special cliques corresponding to the convex regions (generated by disks of radius $1/2$ centered at vertices of the input graph). However, this leaves us with the problem of bounding the number of convex regions by a function of the parameter k . We later show that our approach leads to construction of a fixed-parameter algorithm of running-time $2^{\mathcal{O}(k \log k)} n + \mathcal{O}(mn)$.

EXAMPLE: We give a construction with an exponential number of maximal cliques in $(1/\epsilon)$

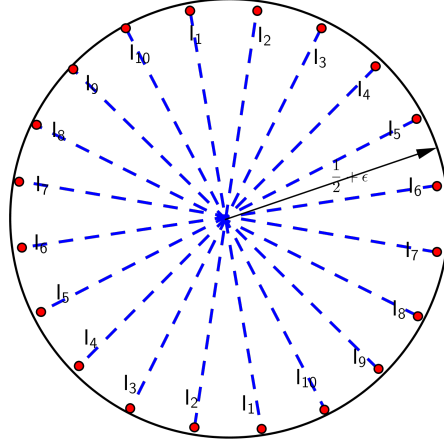


Figure 4.1: A circle of radius $\frac{1}{2} + \epsilon$ and a set of points distributed inside the circle close to its boundary. Every pair of points that form an independent set in UDG are connected with a dashed blue line in the illustration.

for an ϵ -precision UDG. Assume a disk of radius $1/2 + \epsilon$ and distribute d vertices evenly around its perimeter inside the disk. If the points are distributed close enough to the perimeter, then without loss of generality one can assume that for each point p there is at least another point q at distance larger than 1 from p . Let Q_p be the set of all such points. In the respective UDG, p and $q \in Q_p$ would form an independent set. Also, ϵ can be chosen such that $|Q_p| = 1$. Therefore, the UDG on d points is $d/2$ -partite with each of the independent sets having cardinality 2. See Figure 4.1. Observe that choosing exactly one vertex from each of the $d/2$ independent sets would result in a maximal clique. Therefore, we get $2^{\frac{d}{2}}$ different cliques for every d vertices. Set $d = \mathcal{O}(1/\epsilon)$. Then the construction is ϵ -precision. One can place n vertices in $\mathcal{O}(\epsilon n)$ disconnected copies of such constructions each with roughly $2^{\frac{1}{2\epsilon}}$ maximal cliques. Therefore, the number of maximal cliques would be linear in n and exponential in $(1/\epsilon)$ for an ϵ -precision UDG on n points.

Precision UDGs as an interesting subclass of UDGs, were first studied by Alber et al. [5] and are formally defined as follows:

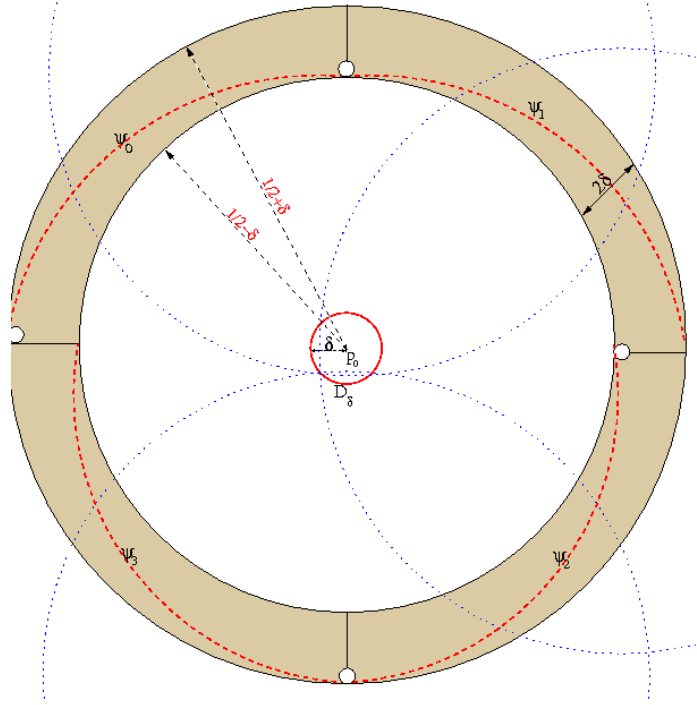


Figure 4.2: \mathcal{D}_δ is depicted as a disk with red boundary in the center of the figure. The filled ring around \mathcal{D}_δ shows annulus Ψ . The dashed red line in Ψ is the possible location of points for a scenario with maximum number of regions in \mathcal{D}_δ .

REMINDER: (λ -precision) [5] A UDG with disk centers $P = \{p_1, \dots, p_n\}$ is called λ -precision, if for every pair of distinct points $p_i, p_j \in P$, the Euclidean distance between the two points $\mathbf{dist}(p_i, p_j)$ is at least λ . The class of such graphs is denoted UDG_λ .

LEMMA 4.2.1. Let \mathcal{D}_δ be a disk of radius δ centered at an arbitrary point p_0 in the plane. Let η_δ denote the maximum number of convex regions in $R_{V(G)} \cap \mathcal{D}_\delta$ obtained after adding \mathcal{D}_δ to the drawing. If $\delta \leq (\sqrt{3}/4)\epsilon$, then $\eta_\delta = \mathcal{O}(1/\delta^2)$.

Proof. Consider the annulus Ψ enclosed between the two circles $\mathcal{D}_{(1/2-\delta)}$ and $\mathcal{D}_{(1/2+\delta)}$ of radii $1/2 - \delta$ and $1/2 + \delta$ respectively, both centered at p_0 . Clearly, the boundary of any disk of radius $1/2$ centered at an arbitrary point $p_x \in \Psi = \mathcal{D}_{(1/2+\delta)} \setminus \mathcal{D}_{(1/2-\delta)}$ has to cross \mathcal{D}_δ . Therefore, any arc intersecting \mathcal{D}_δ has to belong to the boundary of a disk centered at some point say $p_{x'}$ in Ψ . See Figure 4.2. First, we count the maximum number of possible such

disk centers (the set of vertices) that can fall in Ψ , with the restriction that every two such points have to be at least ϵ apart from each other. Let $\delta = (\sqrt{3}/4)\epsilon$. One can place in Ψ , at most $\frac{(1+2\delta)\pi}{\arcsin \frac{\epsilon}{1+2\delta}}$ points, such that every two points are at least ϵ apart. Next, we count all possible convex regions within radius of δ of p_0 in $R_{V(G)} \cap \mathcal{D}_\delta$. Let us denote the maximum number of such regions by η_δ . In order to take into account the set of possible convex regions that have points outside \mathcal{D}_δ , we also add the imaginary disk \mathcal{D}_δ to the drawing letting the boundaries of \mathcal{D}_δ contribute to possible regions. Draw an arbitrary line l through p_0 as well as its perpendicular at p_0 . Call it l' . Label the intersection of sectors of angle $\pi/2$ (formed after intersecting l and l' with $\mathcal{D}_{(1/2+\delta)}$) with Ψ , in an arbitrary circular order ψ_0, ψ_1, ψ_2 and ψ_3 . Clearly, $\cup_{i=0}^3 \psi_i = \Psi$. One can place the points in Ψ such that the disks centered at points in ψ_i and those in $\psi_{(i+2) \bmod 4}$ form non-crossing convex regions in \mathcal{D}_δ . Let us call this set of regions $\mathfrak{R}_{(i \bmod 2)}$. In particular, one can place the points such that every region $r \in \mathfrak{R}_0$ intersects every region in \mathfrak{R}_1 . This forms $|\mathfrak{R}_1|$ convex regions. The dotted arcs in Figure 4.2 shows the possible location of disk centers in Ψ for obtaining such an arrangement of regions in \mathcal{D}_δ . The maximum number of such regions is then obtained as $|\mathfrak{R}_0| \cdot |\mathfrak{R}_1|$. But since ' ψ_i 's were chosen arbitrarily, the number of such convex regions can be at most

$$\eta_\delta = \left(\frac{(1+2\delta)\pi}{4 \arcsin \frac{\epsilon}{1+2\delta}} \right)^2 \leq 7/\epsilon^2. \quad (4.1)$$

Clearly, if $\delta < (\sqrt{3}/4)\epsilon$, the argument still holds. \square

DEFINITION 4.2.1. [Face distance]

Let Γ be a plane graph and $F(\Gamma)$ be the set of all faces of Γ . The face distance between two faces $f, f' \in F(\Gamma)$, denoted $d^*(f, f')$ is the minimum number of faces to cross to get to f' from f in Γ . In other words, $d^*(f, f')$ is the length of a shortest path between vertices v_f^* and $v_{f'}^*$, corresponding to faces f and f' , in Γ^* , the planar dual of Γ . We denote by $F_m(f) \subseteq F(\Gamma)$ the set of all faces f' of Γ such that $d^*(f, f') = m$ for an integer $m \geq 0$.

DEFINITION 4.2.2. [Clique intersection graph]

Let G be the graph obtained after applying the reduction rules and consider \vec{G} , we construct the clique intersection graph, $H = (\mathcal{S}, \mathcal{E})$ in the following way. The vertices in H correspond to sink vertices (convex regions alternatively) \mathcal{S} in \vec{G} . Two vertices s_1 and s_2 are adjacent in H iff $\mathcal{X}_G(s_1) \cap \mathcal{X}_G(s_2) \neq \emptyset$.

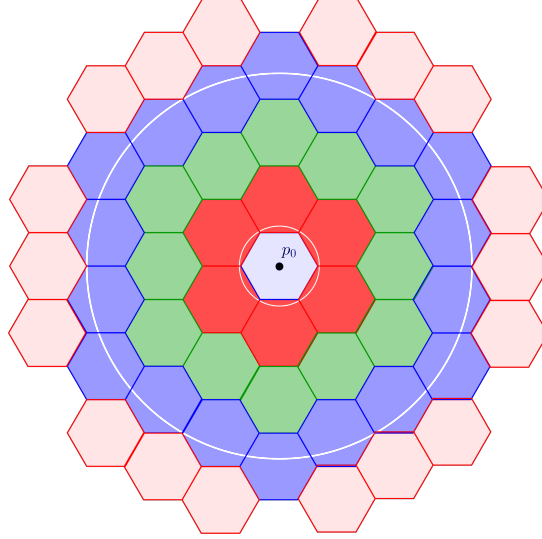


Figure 4.3: A hexagonal mesh is placed over the plane. Hexagons with face distance 1, 2 and 3 from p_0 are colored red, green and blue respectively.

Before we perform an analysis on the size of graph H , we prove an upper bound on the maximum degree of H . To do so, we count the number of possible convex regions at most unit distance away from a given region. This is argued in the following lemma and the proof idea relies on the result of Lemma 4.2.1 and a packing argument.

LEMMA 4.2.2. *Let $\Delta(H)$ denote the maximum degree of the clique intersection graph H . Then $\Delta(H) \leq 181/\epsilon^4$.*

Proof. In order to upper-bound $\Delta(H)$, we need to find the maximum number of convex regions in $R_{V(G)}$ that can share a disk with a given convex region (corresponding to a vertex in H). We show that for a given vertex in H , all such regions can be enclosed in a disk $\mathcal{D}_{1+\delta}$ and use a tiling argument to calculate an upper bound on the number of disks of radius δ (\mathcal{D}_δ) required to cover $\mathcal{D}_{1+\delta}$. Problem of covering a circle with a smaller circles of a fixed radius is a known problem in geometry, studied for example by Toth in [108]. In particular, suppose the smaller circles have radii of 1, let $R(n)$ be the radius of largest circle that n circles of radii 1 can cover. Here we are interested in calculating the minimum integer m

such that $R(m) \geq \frac{1+\delta}{\delta}$. There is no known equation that calculates $R(n)$. Although $R(n)$ is known for small fixed values of n . For example, $R(11) = 2.631$ and $R(12) = 2.769$ according to the work of Melissen [94]. Therefore, instead of an exact calculation, we consider approximation covering using polygonal meshes. Consider a hexagonal mesh of edge length δ put over the entire plane. Let f be any arbitrarily chosen face of the mesh, then $|F_m(f)| = 6m$ for any $m \geq 1$. See Figure 4.3. Draw the diagonals of f and call the intersection point of the diagonals p_0 . Let $r(m)$ be the radius of the largest circle centered at p_0 crossing the faces in $F_m(f)$ only, $r(m) = (\frac{3}{2}m + \frac{1}{2}) \cdot \delta$ for $m \geq 1$.

Notice that the length of diagonals of every face in the mesh is 2δ and therefore the respective hexagon is inscribed in some disk \mathcal{D}_δ . We have already calculated the maximum number of possible convex regions in \mathcal{D}_δ in the previous lemma. Next, we do an over-counting in order to cover a disk of radius $1 + \delta$ with \mathcal{D}_δ . Let r_0 be a convex region in $R_{V(G)}$ and suppose $p_0 \in r_0$. Also let v_0 be the vertex in H corresponding to r_0 . Then by the definition of graph H , the regions corresponding to $N_H(v_0) \subseteq \mathcal{S}$ in $R_{V(G)} \cap \mathcal{D}_\delta$ are all enclosed in a circle of radius at most $1 + \delta$ centered at p_0 in the plane. Let μ_r be the number of hexagonal faces inside or crossing the boundaries of a circle of radius r centered at a point p_0 and L_r be the maximum face distance of such faces. We can obtain:

$$L_r = \left\lceil \frac{2}{3} \left(\frac{r}{\delta} - 2 \right) \right\rceil + 1. \quad (4.2)$$

Also, since at each face distance ℓ , there are 6ℓ faces in the drawing, we obtain the number of faces μ_r by adding up the number of faces at every possible face distance. That is,

$$\mu_r = \sum_{\ell=1}^{L_r} 6\ell.$$

Hence, we get:

$$\mu_{1+\delta} = 3 \left\lceil \frac{2}{3} \left(\frac{1}{\delta} - 1 \right) + 1 \right\rceil \left\lceil \frac{2}{3} \left(\frac{1}{\delta} - 1 \right) + 2 \right\rceil \quad (4.3)$$

as an upper bound for the number of disks \mathcal{D}_δ required to cover a disk of radius $1 + \delta$ in the plane – where regions corresponding to $N_H(v_0)$ can appear. Finally, $\Delta(H)$ is obtained after multiplying $\mu_{1+\delta}$ by the equation obtained in Lemma 4.2.1. \square

Lastly, we upper-bound the number of vertices of H .

THEOREM 4.2.1. *Let H be the clique intersection graph as defined earlier. Then $|V(H)| \leq 728k/\epsilon^4$ for $\epsilon \leq 1$.*

Proof. Without loss of generality, we assume that G is connected. As if it is otherwise, an upper bound on the size of clique intersection graph for each connected component would imply an upper bound on the size of H .

Since G is reduced, any vertex of H has a degree of at least 2. Let $\text{OPT} = \{c_1, \dots, c_k\}$ be a solution for CLIQUE PARTITION on G such that each c_i is maximal. We call a vertex $s \in V(H)$ covered by c_i , if $\mathcal{X}_G(s) \subseteq V(c_i)$ for some $1 \leq i \leq k$. For a covered vertex s , we call all edges of H incident to s , covered. For $s' \in V(H)$, if s' is not incident to any covered edge, then there is a vertex in $\mathcal{X}_G(s')$ which is not included in any clique in OPT . Therefore, each vertex of H is incident to at least one covered edge. By Lemma 3.3.1, for each clique $c \in \text{OPT}$, there can be at most 4 vertices and hence at most $4\Delta(H)$ edges of H covered by c . Therefore, in total, the number of vertices that are incident to at least one covered edge (by any clique in OPT) is at most $4(\Delta(H) + 1)k$ and at least $|V(H)|$. Combining this with Lemma 4.2.2, we get $|V(H)| \leq 4\Delta(H)k + 4k \leq 728k/\epsilon^4$ for $\epsilon \leq 1$. \square

4.2.1 An FPT Algorithm on Precision UDGS

We use a similar brute-force search algorithm as the one described in the previous chapter for arbitrary UDGS to obtain a fixed-parameter tractable algorithm for the problem on ϵ -precision UDGS. By Lemma 3.3.1, every clique in an optimal clique partition can be represented by at most a constant number of convex regions in $R_{V(G)}$. Also by Theorem 4.2.1, after applying the reduction rule there are only $\mathcal{O}(k/\epsilon^4)$ convex regions left. We perform the same process as described in the previous chapter in order to take care of non-helly sets. Notice that, this time, we originally have only $|V(H)| = 728k/\epsilon^4$ rows in the binary matrix described before. Therefore, the time required to pre-process the matrix for non-helly sets would be $\mathcal{O}(|V(H)|^3 n)$ and after pre-processing at most $|V(H)|^3$ regions are in the set. The pre-processing is only performed once for the whole matrix and hence for $k > 0$ this factor is dominated by the time required for guessing the subsets and checking the validity of the clique partitions.

In order to decide whether $|\text{OPT}(G)| \leq k$, we enumerate all sets of at most k regions and for each such a set, check whether it corresponds to a clique partition. Thus the number of

such subsets could be at most

$$\sum_{k' \leq k} \binom{|V(H)|^3}{k'} \leq (|V(H)|^3 + 1)^k = \mathcal{O}(2^{3k \log |V(H)|}). \quad (4.4)$$

Taking into account the running-time of $\mathcal{O}(mn)$ for applying the reduction rules and $\mathcal{O}(kn)$ time for checking the validity of each candidate clique partition, we get the overall time complexity as:

$$\mathcal{O}(2^{3k \log |V(H)|} kn + |V(H)|^3 n + mn) = \mathcal{O}(2^{(3k+1) \log k + 12k \log \frac{1}{\epsilon} + 30k} n + mn). \quad (4.5)$$

THEOREM 4.2.2. *For any UDG G and an integer k such that the reduced graph of G is ϵ -precision, there is an algorithm that decides in $\mathcal{O}(2^{(3k+1) \log k + 12k \log \frac{1}{\epsilon} + 30k} n + mn)$ time, if there exists a clique partition of size at most k for G .*

4.3 Quasi-precision UDGs

In this section, we describe how to extend the results of previous section to the (λ, α) -quasi-precision UDGs. In particular, we show that if the graph obtained after applying the reduction rules is a (λ, α) -quasi-precision UDG with $\alpha \leq 1/2$, fixed-parameter algorithms for CLIQUE PARTITION are still possible. We first provide a definition of the class.

Given a graph $G = (V, E)$ with the location of disk centers (vertices) given in the plane and a constant $\lambda \leq 1$, we define $V_\lambda = \{v \in V \mid \forall v' \in V \setminus v, \|vv'\| \geq \lambda\}$. By this definition, $G[V_\lambda]$ is then a λ -precision UDG. $G[V_\lambda]$ is also called the *sparse subgraph* of G with respect to λ .

Let $E_{<\lambda} = \{e \in E \mid \lambda > \|e\|\}$ be the λ -short edges and $V_{<\lambda} \subseteq V$ be the set of vertices incident to $E_{<\lambda}$ (also referred to as λ -close vertices). Then the *dense subgraph* of G with respect to λ is similarly defined as $G_{<\lambda} = (V_{<\lambda}, E_{<\lambda})$. Let X_1, \dots, X_κ be the vertices of connected components of $G_{<\lambda}$. We call every component X_i a λ -dense island of G .

DEFINITION 4.3.1. **$[(\lambda, \alpha)$ -quasi-precision]**

For non-negative parameters $\lambda \leq 1$ and α , a UDG G is called (λ, α) -quasi-precision, if there exists an integer ς and a subset $S \subset V(G)$ with $|S| = \varsigma$ such that the connected components of $G[V_{<\lambda}] \setminus S$ are in a set of non-overlapping circles of radii at most α . We denote the class of all such graphs by $\text{UDG}_{(\lambda, \alpha)}$.

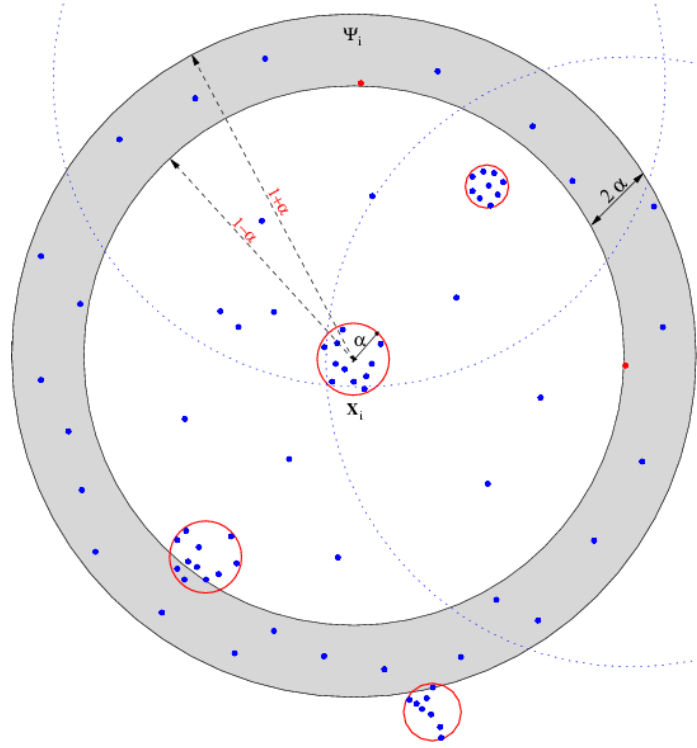


Figure 4.4: A possible decomposition of the points of a UDG to dense islands. The respective graph is (λ, α) -quasi-precision for some value of λ and α .

For any given UDG, one can choose the values of λ and α such that the graph is (λ, α) -quasi-precision. For example, ϵ -precision UDGs are $(\epsilon, 0)$ -quasi-precision while the class UDG_α of bounded-area UDGs [78] are $(c, \mathcal{O}(\sqrt{\alpha}))$ -quasi-precision for any $c > 0$.

Given a (λ, α) -quasi-precision graph $G = (V, E)$ with $\alpha \leq 1/2$, let $\mathcal{X} = V_{<\lambda} \setminus S$ and X_1, \dots, X_κ be a partition of \mathcal{X} into dense islands such that each X_i is bounded in a circle ζ_i of radius $\alpha_i \leq \alpha$ for $1 \leq i \leq \kappa$. In other words, every X_i is the vertex set of a connected component of $G[\mathcal{X}]$.

We first prove that for a $(\lambda, 1/2)$ -quasi-precision UDG which is reduced according to the reduction rules only a bounded number of vertices exist in every dense island.

For a set of vertices Y , let us define $N^\cap(Y) = \cap_{y \in Y} N(y) \setminus Y$ and $\overline{N^\cap(Y)} = N(Y) \setminus N^\cap(Y)$. The basic idea in our analysis is that in the reduced graph, the number of vertices of a dense island X_i can be bounded from above by a function of $|\overline{N^\cap(X_i)}|$ while, as we show in the next lemma, $\overline{N^\cap(X_i)}$ is restricted to an object and hence for a quasi-precision UDG it has a bounded number of vertices.

LEMMA 4.3.1. *For a connected component X_i of $G[\mathcal{X}]$, let ζ_i be a bounding circle for X_i of radius α_i . Further, let Ψ_i be the annulus bounded between circles of radii $1 + \alpha_i$ and $1 - \alpha_i$ concentric with ζ_i . Then $\overline{N^\cap(X_i)} \subseteq \Psi_i$.*

Proof. First, notice that since ζ_i is a bounding circle, any vertex inside a disk $\mathcal{D}_{1-\alpha_i}$ of radius $1 - \alpha_i$ concentric with ζ_i is adjacent to all vertices in X_i . Therefore $\overline{N^\cap(X_i)} \cap \mathcal{D}_{1-\alpha_i} = \emptyset$. Also for any vertex y at a distance further than $1 + \alpha_i$ from the center of ζ_i , $y \notin N(X_i)$. Hence $\overline{N^\cap(X_i)} \subseteq \Psi_i$. \square

For a subset \mathcal{P} of the plane, we define $\mu(\mathcal{P}, d)$ as the maximum number of points that can be placed in \mathcal{P} with the restriction that every pair of points are at pairwise Euclidean distance of at least d .

OBSERVATION: Since G is (λ, α) -quasi-precision and by the result of Lemma 4.3.1, $|\overline{N^\cap(X_i)}| \leq \mu(\Psi_i, \lambda) + |S|$.

Next, we show that if the reduction rule are applied to the graph, the number of vertices in each dense island has to be bounded.

LEMMA 4.3.2. *For a (λ, α) -quasi-precision graph with $\alpha \leq 1/2$ which reduced with respect to the reduction rules, $|X_i| = \mathcal{O}((\mu(\Psi_i, \lambda) + |S|)^2)$ for all $1 \leq i \leq \kappa$.*

Proof. Draw an imaginary circle of radius 1 at each vertex $v \in \overline{N^\cap(X_i)}$. Let C be the set of all such circles. Notice that every such a circle intersects ζ_i in at least two points on the boundary. Let Φ^* be the union of all these circles and $\mathcal{D}(\zeta_i)$ denote the disk with ζ_i as its boundary. Consider the set of all regions R in $\mathcal{D}(\zeta_i) \setminus \Phi^*$ formed after intersecting the boundary of circles with $\mathcal{D}(\zeta_i)$.

We show that for any $r \in R$, if there are more than one vertices in $V \cap \bar{r}$, then Rule 3 is applicable to G by setting $I = V \cap \bar{r}$. First notice that for every vertex $v \notin \overline{N^\cap(X_i)}$,

either $I \subseteq N[v]$ or $I \cap N[v] = \emptyset$. Also since r is a region, for any vertex $x \in \overline{N^\cap(X_i)}$, either $I \subseteq N(x)$ or none of vertices in I are adjacent to x . Thus, I can be replaced by a single vertex by Rule 3. As a result, in the reduced graph, there can be at most one vertex in each region of R . In particular, $|X_i| \leq |R|$.

To complete the proof, we upper-bound $|R|$. However, $\overline{N^\cap(X_i)} \subseteq \Psi_i$ by Lemma 4.3.1 and hence we can safely use an argument similar to the one in Lemma 4.2.1 to count the number of regions in R knowing that there are at most $\mu(\Psi_i, \lambda) + |S|$ vertices in Ψ_i .

Consider the graph obtained from $\mathcal{D}(\zeta_i) \cap \Phi^* \cup \zeta_i$. Intersection point of every two circles in $C \cup \zeta_i$ forms a vertex of the graph. Two vertices are connected by an edge in this graph if their corresponding points are on the same circle in $C \cup \zeta_i$.

Observe that the vertices which are located on the circle ζ_i have degree 3 and there are at most $2(\mu(\Psi_i, \lambda) + |S|)$ such vertices. This is while all other vertices have degree 4 because they are formed by intersecting two circle in C . Let there be n^* such vertices. The faces in the graph correspond to the regions R . Notice that we can get the number of edges in the above planar graph as half of sum of the vertex degrees as $2n^* + 3(\mu(\Psi_i, \lambda) + |S|)$. Using the Euler formula for planar graphs, we can find the number of faces (also $|R|$) as $n^* + \mu(\Psi_i, \lambda) + |S| + 2$. In the worst case when every two circles in C intersect at one point in $\mathcal{D}(\zeta_i)$, we get $n^* = \mathcal{O}(\mu(\Psi_i, \lambda) + |S|)^2$. Lastly, since $|X_i| \leq |R|$, the statement of the lemma holds. \square

LEMMA 4.3.3. *Let $\mu(\Psi_i, \lambda)$ be the maximum number of points in Ψ_i with the minimum pairwise distance λ . Then $\mu(\Psi_i, \lambda) = \mathcal{O}(\alpha_i/\lambda^2)$.*

Proof. We use an approach similar to the one in Lemma 4.2.2 to bound the function. Let $\Psi(r, r')$ be the annulus between two circles of radii r and r' . That is $\Psi(r, r') = \mathcal{D}_r \setminus \mathcal{D}_{r'}$. Then $\mu(\Psi(r, r'), \lambda) \leq \mu(\mathcal{D}_r, \lambda) - \mu(\mathcal{D}_{r'}, \lambda) + \pi / \arcsin \frac{\lambda}{2r'}$. This is since in the maximum packing of \mathcal{D}_r with $\mathcal{D}_{\lambda/2}$, cutting along boundary of \mathcal{D}_r would destroy at most $\pi / \arcsin \frac{\lambda}{2r'}$ of disks $\mathcal{D}_{\lambda/2}$. Using this argument, we get $\mu(\Psi_i, \lambda) \leq \mu(\mathcal{D}_{1+\alpha_i}, \lambda) - \mu(\mathcal{D}_{1-\alpha_i}, \lambda) + \pi / \arcsin \frac{\lambda}{2(1-\alpha_i)}$. Now we use Lemma 4.2.2 to get the upper bounds on $\mu(\mathcal{D}_{1+\alpha_i}, \lambda)$ and $\mu(\mathcal{D}_{1-\alpha_i}, \lambda)$.

$$\begin{aligned} \mu(\mathcal{D}_{1+\alpha_i}, \lambda) &\leq 3 \left\lceil \frac{4}{3} \left(\frac{1+\alpha_i}{\lambda} - 1 \right) + 1 \right\rceil \left\lceil \frac{4}{3} \left(\frac{1+\alpha_i}{\lambda} - 1 \right) + 2 \right\rceil, \\ \mu(\mathcal{D}_{1-\alpha_i}, \lambda) &\leq 3 \left\lceil \frac{4}{3} \left(\frac{1-\alpha_i}{\lambda} - 1 \right) + 1 \right\rceil \left\lceil \frac{4}{3} \left(\frac{1-\alpha_i}{\lambda} - 1 \right) + 2 \right\rceil. \end{aligned}$$

Let $x = \frac{1+\alpha_i}{\lambda}$, and $y = \frac{1-\alpha_i}{\lambda}$. Then,

$$\begin{aligned} \mu(\Psi_i, \lambda) &\leq \frac{16}{3} \left[\left(x + \frac{1}{2}\right) \left(x + \frac{5}{4}\right) - \left(y + \frac{1}{2}\right) \left(y + \frac{5}{4}\right) \right] + 2\pi y \\ &\leq \frac{22\alpha_i}{\lambda^2} + \frac{(19 - 2\pi)\alpha_i}{\lambda} + \frac{2\pi}{\lambda} \\ &\leq 24\alpha_i/\lambda^2 = \mathcal{O}(\alpha_i/\lambda^2). \end{aligned}$$

□

In the next step, we upper-bound the number of convex regions inside an arbitrary disk of small radius, similar to Lemma 4.2.1. To do so, we first calculate the maximum number of vertices inside an arbitrary annulus in the plane.

LEMMA 4.3.4. *Let G' be the reduced graph and Ψ be the annulus between two disks of radii $1/2+\delta$ and $1/2-\delta$ centered at an arbitrary point p_0 in the plane then $|V(G') \cap \Psi| = \mathcal{O}(\delta/\lambda^4)$.*

Proof. If p_0 is chosen arbitrarily, then there can be many dense islands intersecting Ψ . To get an upper bound on $|V(G') \cap \Psi|$, we cover Ψ with disks of radius α and consider the case where some of the disks intersect dense islands. Since G' is reduced, even if a disk intersects a dense island, we have an upper bound on the number of points inside it. In particular, one can repeat the approach of Lemma 4.3.3 to cover Ψ with at most $\mathcal{O}(\delta/\alpha^2)$ disks of radius α . Now for each disks \mathcal{D}_α in Ψ , we get an upper bound on the number of points inside from Lemma 4.3.2 as $|V(G') \cap \mathcal{D}_\alpha| = \mathcal{O}(\alpha^2/\lambda^4 + |S|^2)$. Therefore,

$$\begin{aligned} |V(G') \cap \Psi| &\leq |V(G') \cap \mathcal{D}_\alpha| \cdot \mathcal{O}(\delta/\alpha^2) \\ &= \mathcal{O}(\alpha^2/\lambda^4 + |S|^2) \cdot \mathcal{O}(\delta/\alpha^2) = \mathcal{O}(\delta/\lambda^4 + \delta|S|^2/\alpha^2). \end{aligned}$$

Given that $|S| = \mathcal{O}(1)$, the statement of the lemma follows. □

LEMMA 4.3.5. *Let r be a convex region in a disk \mathcal{D}_δ of radius δ and $\mathcal{X} = V_{<\lambda} \setminus S$ (as defined before). We further define $\mathcal{Y}_G(r) = \{v \in V(G) \mid \partial\mathcal{D}(v) \cap \partial r \neq \emptyset\}$. If $\mathcal{Y}_G(r) \cap \mathcal{X} \neq \emptyset$ then $\mathcal{Y}_G(r) \cap \mathcal{X}$ belongs to only one connected component of $G[\mathcal{X}]$.*

Proof. Suppose for contradiction that r is a convex region and let X_1 and X_2 be the vertex sets of two connected components of $G[\mathcal{X}]$, however $\mathcal{Y}_G(r)$ has vertices $x_1 \in X_1$ and $x_2 \in X_2$. Since r is convex, $\mathcal{D}(x_1) \cap \mathcal{D}(x_2) \neq \emptyset$. Therefore, $\mathbf{dist}(x_1, x_2) \leq 1$ and x_1x_2 is an edge in $G[\mathcal{X}]$ contradicting the fact that x_1 and x_2 are in different connected components. □

Similar to previous section, we use the bound on the number of vertices in Ψ to derive an upper bound on the number of convex regions in an arbitrary disk \mathcal{D}_λ of radius λ .

LEMMA 4.3.6. *Let η_λ denote the number of convex regions in a disk \mathcal{D}_λ of radius λ centered at an arbitrary point p_0 in the plane. Then $\eta_\lambda = \mathcal{O}(1/\lambda^6)$.*

Proof. For a constant $\delta \leq 1/2$, consider the annulus Ψ enclosed between two circles of radii $1 + \delta$ and $1 - \delta$ centered at p_0 . Since the vertices in Ψ are the only vertices in $V(G')$ whose disks cross $\partial\mathcal{D}_\delta$, for any convex region r in \mathcal{D}_δ , it holds that $r \in R_{V(G') \cap \Psi}$. We can use an idea similar to the one in Lemma 4.2.1 to upper-bound the number of convex regions in \mathcal{D}_δ in terms of $|V(G') \cap \Psi|$. In particular, let $\delta = \lambda$. Then by Lemma 4.3.4, $|V(G') \cap \Psi| = \mathcal{O}(1/\lambda^3 + \lambda|S|^2/\alpha^2)$. Therefore, the maximum number of convex regions is upper-bounded as follows:

$$\eta_\lambda \leq \max_{1 \leq i \leq \kappa} \{|V(G'_i) \cap \Psi|^2\} = \mathcal{O}(1/\lambda^6 + \lambda^2|S|^4). \quad (4.6)$$

Given that $|S| = \mathcal{O}(1)$, we get $\eta_\lambda = \mathcal{O}(1/\lambda^6)$. \square

Lastly, we can bound the maximum degree of the clique intersection graph in a similar way as the previous section using the results of Lemma 4.3.6 and Lemma 4.2.2.

THEOREM 4.3.1. *Let $\Delta(H)$ be the maximum degree of the clique intersection graph of a (λ, α) -quasi-precision UDG reduced with respect to the Rules 1-3, with $\alpha \leq 1/2$. Then $\Delta(H) = \mathcal{O}(1/\lambda^8)$.*

Proof. We use Lemma 4.2.2 to bound the number of possible disks of radius λ required to cover a disk of radius $1 + \lambda$. In particular, $\mu_{1+\lambda} \leq 14/\lambda^2$. After multiplying this by the result of Lemma 4.3.6, we get:

$$\Delta(H) \leq \mu_{1+\lambda} \cdot \eta_\lambda \leq 14/\lambda^2 \cdot \eta_\lambda = \mathcal{O}(1/\lambda^8 + |S|^4). \quad (4.7)$$

Given that $|S| = \mathcal{O}(1)$, we get $\Delta(H) = \mathcal{O}(1/\lambda^8)$. \square

COROLLARY 4.3.1. *Let H be the clique intersection graph for a (λ, α) -quasi-precision UDG which is reduced according to our reduction rules for $\alpha \leq 1/2$. Then $|V(H)| = \mathcal{O}(k/\lambda^8)$.*

4.3.1 An FPT Algorithm on Quasi-precision UDGs

In order to extend the fixed-parameter results to quasi-precision graphs, we run the same process used for ϵ -precision UDGs and arbitrary UDGs as explained in the previous sections. The algorithm is analogous to the one for the ϵ -precision UDGs and hence for the sake of brevity, we avoid to repeat it here. The time complexity of the exact algorithm for CLIQUE PARTITION on quasi-precision graphs with $\alpha \leq 1/2$ would be:

$$\mathcal{O}(|V(H)|^{3k}kn + |V(H)|^3n + mn) = 2^{\mathcal{O}(k \log \frac{k}{\lambda})}n + \mathcal{O}(mn). \quad (4.8)$$

THEOREM 4.3.2. *Given an integer k , there is an algorithm that decides in $2^{\mathcal{O}(k \log \frac{k}{\lambda})}n + \mathcal{O}(mn)$ time, if there exists a clique partition of size at most k for any UDG such that its reduced graph is (λ, α) -quasi-precision with $\alpha \leq 1/2$.*

4.4 Structure of Quasi-precision UDGs

Finally, we show that if the size excluded subset S in the definitions of quasi-precision UDGs is allowed to be $\mathcal{O}(k)$, the resulted graph would include a large co-bipartite graph as a sub-graph.

We first define another general class of UDGs based on the definition of quasi-precision UDGs.

DEFINITION 4.4.1. **(λ, α) -relaxed-quasi-precision**

For non-negative parameters $\lambda \leq 1$ and α , a UDG G is called (λ, α) -relaxed-quasi-precision, if there exists a subset $S \subset V(G)$ of size $\mathcal{O}(\kappa)$ such that the connected components of $G[V_{<\lambda}] \setminus S$ are in a set of κ non-overlapping circles of radii at most α .

Based on the above definition of relaxed-quasi-precision UDGs, it is clear that quasi-precision UDGs are a special case of relaxed-quasi-precision UDGs. Yet, as we will show later, one can still obtain fixed-parameter results for this class of UDGs.

LEMMA 4.4.1. *Let G be a (α, λ) -relaxed-quasi-precision UDG for some constants $\lambda < 1$ and $\alpha \leq 1/2$ and let S be chosen such that there are κ vertex disjoint cliques in $G[V_{<\lambda}] \setminus S$. Then $\kappa \leq k$ where k is the size of optimal clique partition on G .*

Proof. We first prove that given any induced subgraph G'' of G , $|\text{OPT}(G'')| \leq |\text{OPT}(G)|$ where $\text{OPT}(G)$ is the optimal clique partition on G . For a vertex $v \in V(G) \setminus V(G'')$ if $G[N[v]]$ is a clique of size greater than 1, removing v does not increase the solution size. Otherwise, let $I \subset N(v) \setminus v$ be a maximum independent set of $G[N[v]]$. Then $|I| \geq 2$. Again $|\text{OPT}(G \setminus v)| = |\text{OPT}(G)|$. If v is an isolated vertex, then removing v decreases the size of optimal solution by one. Therefore, $|\text{OPT}(G'')| \leq |\text{OPT}(G)|$.

Finally, since $G^* = G[V_{<\lambda}] \setminus S$ is obtained from G after removing vertices in $S \cup V_\lambda$, G^* is an induced subgraph of G and by the above argument the size of optimal solution for G^* is at most $|\text{OPT}(G)| = k$. However, since $\alpha \leq 1/2$, G^* consists of clique components only and $|\text{OPT}(G^*)| = \kappa$. Therefore $\kappa \leq k$. \square

One can use the above lemma along with Lemma 4.3.1 in order to obtain similar parameterization results on the class of relaxed-quasi-precision UDGs.

COROLLARY 4.4.1. *Let H be the clique intersection graph for a (λ, α) -relaxed-quasi-precision UDG which is reduced according to our reduction rules for $\alpha \leq 1/2$. Then $|V(H)| = \mathcal{O}(k/\lambda^8 + k^5)$.*

Hence, one can obtain the following result.

THEOREM 4.4.1. *Given an integer k , there is an algorithm that decides in $2^{\mathcal{O}(k \log \frac{k}{\lambda})} n + \mathcal{O}(mn)$ time, if there exists a clique partition of size at most k for any UDG whose reduced graph is (λ, α) -relaxed-quasi-precision with $\alpha \leq 1/2$.*

Let $K_{i,i}$ be a complete bipartite graph with bipartite sets each on i vertices and let \mathcal{M} be an arbitrary perfect matching in $K_{i,i}$. We define the dense bipartite graph as $L_i = K_{i,i} \setminus \mathcal{M}$. A *clique-prism* is then defined as \bar{L}_i (the complement of graph L_i). See Figure 4.5 for an example of such a construction.

OBSERVATION: Given any choice of integers i , any set of at least two maximal cliques in \bar{L}_i has at most *one* common vertex.

COROLLARY 4.4.2. *Let $G = \bar{L}_i$ and G' be the graph obtained from G after repeated application of Rules 1-3 (See Section 3.4). Then $G = G'$.*

We call \bar{L}_i an *obstruction* to the reduction Rules 1-3. Observe that the size of a minimum clique partition on \bar{L}_i is 2. Removing any of edges in the matching would result in a subgraph of \bar{L}_i which is reducible with respect to Rules 1-3. Likewise, removing an edge in any

of the two cliques would result in a subgraph for which the size of minimum clique partition is larger (3 in this case).

Intuitively, an obstruction to a reduction rule is the structure expected to be present (as a subgraph) in the reduced graph G' . However, to discover all such obstructions is not straightforward. In order to illustrate the strength of our framework, we show that relaxed-quasi-precision UDGs are indeed the largest class that admit FPT under our reduction rules. Particularly, we prove that any UDG that is not $(\lambda, 1/2)$ -relaxed-quasi-precision for any value of λ must admit an arbitrarily large clique-prism as subgraph.

THEOREM 4.4.2. *Let the reduced graph G' be an \bar{L}_ι -free UDG for some fixed integer $\iota > 1$. Then there are constants $\lambda \leq 1$ and $\alpha \leq 1/2$ such that G' is (λ, α) -relaxed-quasi-precision with $|S| \leq 3(\iota - 1)\kappa' - 6\iota + 6$.*

Proof. To prove the theorem, we describe how a (λ, α) -relaxed-quasi-precision graph is constructed. Set $\lambda = \frac{1}{2\iota}$. In the first step, we check the distance between every two adjacent vertices in the drawing against the value of λ and for every λ -short edge $e = \{u, v\}$, we draw a bounding circle with line segment uv being the diameter. We call each such a circle a *basic circle* and the edge $e = \{u, v\}$ the *diameter edge* of the basic circle. For a circle ζ , a basic circle ζ' is called an *expansion candidate* of ζ if $\zeta' \neq \zeta$ and exactly one end vertex of the diameter edge of ζ' is covered by ζ . In the next step, starting with an arbitrary circle ζ , if ζ has an expansion candidate ζ' , we *grow* ζ (we replace ζ and ζ' by a new circle of minimum diameter) to cover all vertices previously covered by ζ as well as the end vertices of the diameter edge of ζ' .

At every grow step, the radius of the circle increases by at most λ while at least one extra vertex is added to the circle. We continue this procedure until the grown circle does not have an expansion candidate. If that is the case, we mark the circle as *final* and start with another basic circle. We continue this until all circles are marked *final*. For any pair of final circles if the graph induced on the vertices covered by them is a clique, we merge them into a single circle by replacing the two circles by a new circle of minimum radius that covers all the vertices covered by the two former circles.

Next, we bound the radius of circles obtained in the last step. Notice that since G' is

\bar{L}_ι -free, inside any circles of radius $1/2$ there can be at most 2ι vertices. Because otherwise G' would have a subgraph isomorphic to \bar{L}_ι . We claim that the radius of every circle obtained above is smaller than $1/2$. Suppose for the sake of contradiction that one of the circles has a radius of at least $1/2$. Let X^\dagger be the set of vertices inside the circle. We argue that $|X^\dagger| \geq \lceil 1/\lambda \rceil$. In particular, we use induction to prove that at every step when we obtain a new circle of radius r , at least $2r/\lambda + 1$ vertices are covered by the circle.

For the basic circles constructed in the first step, the radius is $\lambda/2$ and there are two vertices in the circle. Therefore, the base of induction holds. Consider an intermediate step when we grow a circle of radius r and $2r/\lambda + 1$ vertices to include an intersecting basic circle. The new circle would have a radius $r' < r + \lambda/2$ and $2r/\lambda + 2$ vertices. Notice that $2r/\lambda + 2 > 2r'/\lambda + 1$. Therefore, the axiom of induction holds and $|X^\dagger| \geq \lceil 1/\lambda \rceil$. Hence, $|X^\dagger| \geq 2\iota$ and G' has \bar{L}_ι as a subgraph: A contradiction. Therefore, by the approach we have chosen the circles, the radii of all circle are smaller than $1/2$.

Finally, we show that there is a vertex set S of size $\mathcal{O}(\kappa)$ in $V(G')$ whose removal disconnects G' into κ connected components. Let κ' be number of circles obtained in the final step of the algorithm above. To bound the cardinality of S , we construct the following auxiliary graph. For each circle, we place a node in the plane. The set of vertices covered by the circle is called the associated vertex set of the node. Two nodes with associated vertex sets X_1 and X_2 are connected with a link if $G'[X_1 \cup X_2]$ is a connected graph. Observe that the above graph is planar. This is since the circles were originally placed in the plane and for any two circles with associated vertex sets X_1 and X_2 , $G'[X_1 \cup X_2]$ is not a clique. Let $\varsigma = \iota - 1$. Next, we prove that for every link in the above graph with its end nodes having associated vertex sets X_1 and X_2 , at most ς vertices is enough to disconnect $G'[X_1 \cup X_2]$. If there are at most ς vertices in either of X_1 or X_2 , then trivially the condition is true. Therefore, we assume both X_1 and X_2 have at least $\varsigma + 1$ vertices. For each link e in the above graph with end links having associated vertex sets X_1 and X_2 , we claim that there exists a cut set of ς vertices in $G'[X_1 \cup X_2]$. Suppose for contradiction that a minimum vertex cut set for $G'[X_1 \cup X_2]$ has size $\varsigma + 1$. Consider the sets $I_2 = N(X_1) \cap X_2$ and $I_1 = N(N(X_1) \cap X_2) \cap X_1$. Notice that both sets are vertex cuts in $G'[X_1 \cup X_2]$. Therefore, both have at least $\varsigma + 1$ vertices. Also, since the radii of both circles are at most $1/2$, $G'[I_1]$ and $G'[I_2]$ are cliques. Also for every vertex $v \in I_1$, there is a vertex in I_2 which is incident to v with an edge and vice versa (for I_2). Therefore, G' has a $\bar{L}_{\varsigma+1}$ as a subgraph

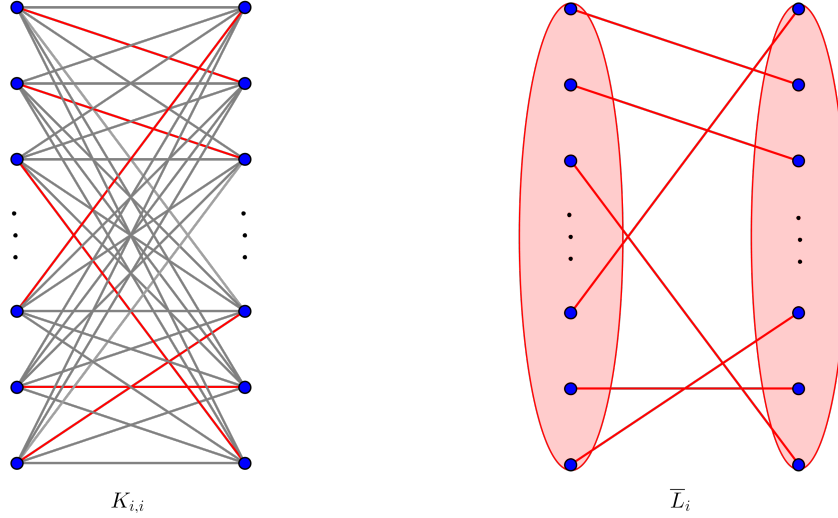


Figure 4.5: Graph $K_{i,i}$ is depicted on the left side. The edges highlighted in red form a perfect matching in $K_{i,i}$. Clique-prism \bar{L}_i corresponding to the selected matching in $K_{i,i}$ is pictured on the right side.

a contradiction to the fact that G' is \bar{L}_i -free. Hence, $G'[X_1 \cup X_2]$ has a vertex cut of size at most ς . Lastly, for every edge of the above auxiliary graph, we add the minimum vertex cut to the set S . Using the Euler formula for planar graphs, we get $|S| \leq 3\varsigma\kappa' - 6\varsigma$.

It follows that if we choose $\lambda \leq \frac{1}{2i}$, $\alpha < 1/2$ and G' is (λ, α) -relaxed-quasi-precision. \square

COROLLARY 4.4.3. *Let G' be a UDG such that it is not (λ, α) -relaxed-quasi-precision for any value of λ independent of n and for $\alpha \leq 1/2$. Then for any integer i independent of n , G' admits \bar{L}_i as subgraph.*

Another interesting insight in this line of research comes from the work of Erdős, Lovász and Vesztergombi [56]. Given a set S of points in the plane with the different distances between pairs of points given as $d_1 > d_2 > \dots$, they define the *graph of large distances* $G(S, k)$ as a graph on vertices S . An edge is present between a pair of points in $G(S, k)$, if their pairwise Euclidean distance is at least d_k for k an arbitrary integer. They study the

chromatic number of $G(S, k)$ under different configurations of the point set S . Let G be the UDG defined on points S . Notice that there exists an integer \bar{m} such that $G(S, \bar{m}) = \bar{G}$, where \bar{m} is upper-bounded by the number of edges missing in UDG G . Similarly, for any k , $G(S, k)$ is the complement of a UDG defined by the set of points S' which is obtained from S after scaling the distances by $1/(d_k - \delta)$ for $\delta < d_k - d_{k+1}$ a small constant. Deciding the *chromatic number* of $G(S, k)$, denoted $\chi(G(S, k))$, is thus equivalent to deciding CLIQUE PARTITION of a UDG. In search for conditions under which $\chi(G(S, k))$ is bounded, Erdős et al. [56] posed the following related question: “Given $t \geq 3$, what is the largest s such that $G(S, k)$ can contain a complete bipartite graph $K_{t,s}$?” Notice that this statement *almost* translates to finding the smallest integer i such that the discussed UDG avoids obstruction \bar{L}_i as subgraph.

4.5 Summary

We proposed a novel approach for deriving FPT algorithms for CLIQUE PARTITION on precision UDGs. In particular, we used the reduction rules introduced earlier for CLIQUE PARTITION on arbitrary UDGs coupled with a detailed analysis in order to design a fixed-parameter algorithm of running-time $\mathcal{O}(2^{(3k+1)\log k + 12k \log \frac{1}{\epsilon} + 30k} n + mn)$ for the problem when the reduced graph is ϵ -precision UDG.

In an attempt to explore the boundaries of fixed-parameter tractability on UDGs, we introduced (α, λ) -quasi-precision UDGs as UDGs which are only partially λ -precision, yet their imprecision components have bounded area. We showed that (α, λ) -quasi-precision UDGs are *ubiquitous* in the UDG class in the sense that any UDG is quasi-precision for some values of λ and α . In particular, precision UDGs and bounded-area UDGs are derived as trivial cases of quasi-precision class. Quasi-precision UDGs are also *non-trivial* meaning that in their general setting, none of the known graph parameters are proved to be bounded when considered on the class. Later, we demonstrated how parameterization results obtained in our work can be extended to the case where the reduced graph is a (relaxed)-quasi-precision UDG. In particular, we obtained fixed-parameter algorithms of running-time $2^{\mathcal{O}(k \log \frac{k}{\epsilon})} n + \mathcal{O}(mn)$ for the problem when the reduced graph is a (ϵ, α) -(relaxed)-quasi-precision UDG for $\alpha \leq 1/2$.

In order to demonstrate the strength of our approach, we proved structural obstructions for the parameterization of the problem under our framework. In particular, we showed that UDGs which do not admit FPT algorithms under our framework include as subgraph a loosely-connected co-bipartite graph. The proposed obstruction classifies the cases where the clique removal is not feasible. Fomin et al. [*SODA 2012*] conjectured that the main obstacle for obtaining subexponential algorithms on UDGs is the design of efficient “clique cleaning” procedures [63]. Our work further affirms their conjecture for obtaining FPT results on UDGs. To the best of our knowledge, this is the first work studying fixed-parameter algorithms for CLIQUE PARTITION on UDGs.

Chapter 5

Improved PTAS for MINIMUM CLIQUE PARTITION

In the light of complexity results for problems on UDGs as discussed in the previous chapters, even on the restricted UDG classes, the most efficient exact algorithms require a running-time which is not practical when considering the size of real-world inputs. A last resort approach in such situations is to look for approximation algorithms which can be tuned to be as close as possible to the exact solution. Formally, this is achieved through construction of PTAS or efficient PTAS (EPTAS) algorithms.

Yet, as Theorem 2.1.1 states, a EPTAS result for a problem would imply fixed-parameter tractability results for the parameterized version of that problem. This seems unlikely for CLIQUE PARTITION considering the parameterized results of the previous chapter. Therefore, PTAS is, in a sense, the best one can hope for the problem unless CLIQUE PARTITION proves to be in FPT in the UDG setting. In this chapter, we propose an approach for obtaining PTAS algorithms for MINIMUM CLIQUE PARTITION on UDGs.

5.1 Introduction

MINIMUM CLIQUE PARTITION.

Instance: An undirected graph $G = (V, E)$

Question: Find a set \mathcal{C} of cliques of minimum cardinality in G such that for every vertex $v \in V$, there exists at least a clique $c \in \mathcal{C}$ with $v \in c$.

Let us recall that on general graphs, MINIMUM CLIQUE PARTITION is equivalent to VERTEX COLORING of the complement graph and hence is not fixed-parameter tractable and is inapproximable within $n^{1-\delta}$ for any $\delta > 0$ [54]. Yet, on the class of UDGs, there are PTAS algorithms for the problem due to Dumitrescu & Pach in 2011 [54] and also the work of Pirwani & Salavatipour in 2010 [102]. The best known PTAS produces a $(1 + \delta)$ -approximate solution for the problem in $\mathcal{O}(n^{1/\delta^2})$ time [54] while the best approximation ratios for practical and randomized algorithms are 3 and 2.16 due to Cerioli et al. [22] and Dumitrescu & Pach [54] both running in $\mathcal{O}(n^2)$ time.

Recent PTAS result of Pirwani and Salavatipour [102] for MINIMUM CLIQUE PARTITION on unit disk graphs relies on the separability property of an optimal clique partition. This property was first established by Capoleas et al. [21] over two decades ago.

Two convex polygons A and B in the plane are said to be *overlapping* if $\mathbf{area}(A \cap B) > 0$, and *non-overlapping* otherwise.

THEOREM 5.1.1. [21]

Let S be a finite point set in the plane. There exists an optimal clique partition of the UDG on points S where the convex hulls of the cliques are non-overlapping.

Dumitrescu and Pach [54], on the other hand, use a stronger variant of the above theorem which is stated as below:

THEOREM 5.1.2. [54]

Let S be a finite point set in the plane. There exists an optimal clique partition of the UDG on points S in which the (convex hulls) of the cliques are pairwise disjoint.

The above theorems, set up a separability property for the cliques which as discussed later is crucial for bounding the approximation ratio of the obtained solution.

5.2 Plane Decomposition

A PTAS for MINIMUM CLIQUE PARTITION is devised in the following three general steps:

STEP 1. Partition the plane using a randomly shifted grid whose cells are squares of size $l \times l$ for $l = \mathcal{O}(1/\epsilon)$.

STEP 2. Compute an optimal clique partition for each cell of the grid and return the union of these cliques as a solution.

STEP 3. Repeat steps 1 and 2 in $\mathcal{O}(\ln \frac{1}{\delta})$ independent trials for any given $0 < \delta < 1$ and pick the smallest solution.

Analysis.

Dumitrescu & Pach [54] show that using this approach, in Step 3 with probability at least $1 - \delta$, a $(1 + \epsilon)$ solution for MINIMUM CLIQUE PARTITION is obtained. We provide a short proof of the argument here.

LEMMA 5.2.1. *If l is large enough, then the probability of a fixed clique c_i in an optimal partition being cut by a grid chosen in Step 1 is $\mathcal{O}(1/l)$.*

Proof. Consider a cell τ of the grid (τ is a $l \times l$ square). Notice that in the worst case the convex hull of c_i can be inscribed in a circle of radius $1/2$. Let us call this circle ζ_i . Consider the placement of ζ_i in τ . For ζ_i not to intersect the boundaries of τ , its center needs to be located in a $(l-1) \times (l-1)$ square inside τ . Therefore the probability of ζ_i intersecting the boundaries of τ (alternatively c_i being cut by the grid) is at most

$$\frac{l^2 - (l-1)^2}{l^2} = \frac{2l-1}{l^2} < \frac{2}{l}.$$

□

Let Opt be the size of an optimal solution. By the argument provided in Lemma 5.2.1, the probability of a grid cutting a clique in optimal solution is at most $2/l$. We consider two different cases: A) If the clique c_i only intersects with either a horizontal or a vertical line of the grid, then in the solution we obtain, we have at most two cliques representing c_i . B) Otherwise, if c_i intersects both a horizontal and a vertical line of a grid, then in the solution obtained in Step 2, c_i can be represented by at most 4 cliques. Yet the probability of this event happening is only π/l^2 . This is since case B only happens if the center of ζ_i (in the argument used in Lemma 5.2.1) is located in four sectors of angle $\pi/2$ inside τ each centered at one corner of τ and have the total area of π .

Notice that this probability has already been included in case A while calculating the probability of c_i being cut by the grid and hence we only need to count those cliques at most two additional times. Therefore, expected size of the solution obtained in Step 2 is $(1 + 2/l + 2\pi/l^2) \cdot Opt$.

LEMMA 5.2.2. *In Step 3, with probability at least $1 - \delta$, a solution of size $(1 + \epsilon) \cdot Opt$ is obtained.*

Proof. Let X be a random variable indicating the size of the solution obtained in Step 2. By Markov inequality the probability that the size of solution is larger than $(1 + \epsilon)$ can be obtained as:

$$Pr[X > (1 + \epsilon) \cdot Opt] \leq \frac{(1 + \frac{2}{l} + \frac{2\pi}{l^2}) \cdot Opt}{(1 + \epsilon) \cdot Opt} \leq \frac{1}{1 + \epsilon} + \frac{2}{l(1 + \epsilon)} + \frac{2\pi}{l^2(1 + \epsilon)}.$$

For large values of l , the probability that the above holds for all of the j trials would then be:

$$\left(\frac{1}{1 + \epsilon}\right)^j.$$

We require this value to be smaller than δ . Therefore, we obtain:

$$j \cdot \ln\left(\frac{1}{1 + \epsilon}\right) \leq \ln \delta.$$

After multiplying both sides by -1 and dividing by $\ln(1 + \epsilon)$, we get:

$$j \geq \frac{\ln \frac{1}{\delta}}{\ln 1 + \epsilon}.$$

Assuming that $\epsilon < 0.1$, it follows that $\ln(1 + \epsilon) \leq 0.9\epsilon$. Therefore $j = \mathcal{O}(\frac{1}{\epsilon} \ln \frac{1}{\delta})$. □

LEMMA 5.2.3. [54]

Let τ be a UDG whose points are restricted to a $l \times l$ square. Then $|OPT(\tau)| \leq 2l^2 + 3l$.

Proof. Place a grid with cell size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$ over τ . Clearly, the number of cells would be at most $\lceil \sqrt{2} l \rceil^2 \leq (\sqrt{2} l + 1)^2 = 2l^2 + 2\sqrt{2} l + 1 \leq 2l^2 + 3l$ for $l > 5.83$. Since the length of diagonals of a square of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$ is 1, any set of points inside such a square forms a clique. This brings us to the conclusion that an optimal clique partition consists of at most $2l^2 + 3l$ cliques. □

5.3 Exact Solution for UDG with Restricted Geometry

5.3.1 Dumitrescu and Pach's Approach

Knowing that there can be at most $q_{\max} = 2l^2 + 3l$ cliques in a solution for a cell τ , the algorithm can check for $q = 1, \dots, q_{\max}$, if there exists a clique partition of size q for the points in $V \cap \tau$. Now assume an unknown solution $\text{OPT}(\tau) = \{c_1, \dots, c_q\}$ and let $r_i \in c_i$ be a representative point for clique c_i and $1 \leq i \leq q$.

One can define an auxiliary graph X on points $r_1, \dots, r_q \in V \cap \tau$ in the following way: $r_i r_j \in E(X)$ if and only if $\|r_i r_j\| \leq 2$.

OBSERVATION: [54]

If $r_i r_j \notin E(X)$, and $p \in V \cap \tau$ satisfies $\|pr_i\| \leq 1$, then $p \notin c_j$.

Proof. Assume for the sake of contradiction that $p \in c_j$. Then $\|pr_i\| \leq 1$. By triangular inequality, $\|r_i r_j\| \leq \|pr_i\| + \|pr_j\| \leq 2$. This contradicts the assumption that $r_i r_j$ is not an edge in X (by the definition of the graph X). \square

LEMMA 5.3.1. $\Delta(X) \leq 54$.

Proof. Consider an arbitrary point r_i of X . Notice that the cliques in $\text{OPT}(\tau)$ whose representative points are in $N_X(r_i) \cup \{r_i\}$ can be contained in a circle ζ of radius 3 centered at r_i . To bound the degree of vertex r_i , partition ζ into pieces such that points within each piece can form one clique. In particular, [54] considers the fact that ζ can be contained in a square of length 6 aligned with τ and tiles that square with smaller rectangles of size $\frac{3}{5} \times \frac{4}{5}$. Clearly, the points inside each such small rectangle form a clique. There can be at most $\lceil 6 \cdot \frac{5}{4} \rceil \times \lceil 6 \cdot \frac{5}{3} \rceil = 8 \times 10 = 80$. Therefore, excluding itself, r_i can have at most 79 neighbors. One can however get a similar bound on the degree using a more complicated tiling argument. Particularly, assuming circle ζ of radius 3, we can cover ζ , with regular hexagons of diameter 1. Notice that any such a hexagon can be inscribed in a circle of radius $1/2$. As Lemma 4.2.2 calculates, there can be at most $\frac{16}{3}r^2 + \frac{28}{3}r + \frac{10}{3}$ such hexagons inside a circle of radius r . Hence, after replacing r by 3, it follows that 79 hexagons are enough to cover ζ . However, a closer look reveals that there can be at most $18 + 18 + 12 + 6 + 1 = 55$ such hexagon in ζ . See Figure 5.1. Therefore, $\Delta(X) \leq 54$. \square

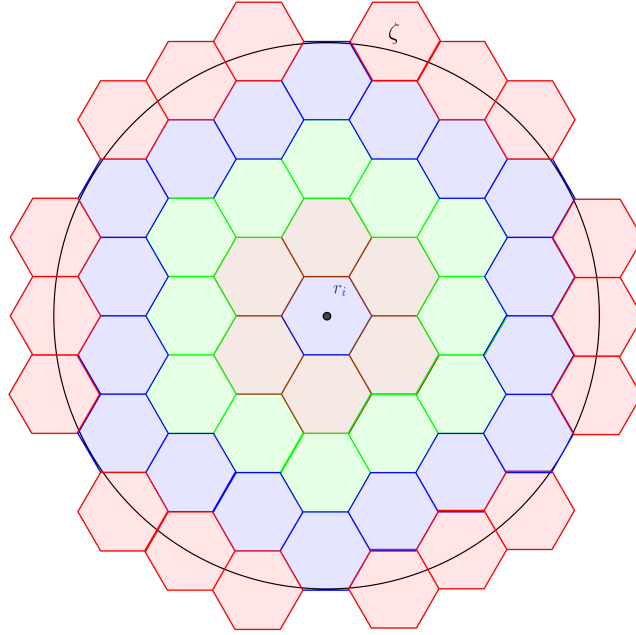


Figure 5.1: A hexagonal mesh with edge length $1/2$ is placed over a circle of radius 3 . Every such a hexagon can be inscribed in a circle of radius $1/2$.

Dumitrescu and Pach [54] use the following known geometric fact to show how the set of representative points along with the graph X can produce a clique partition of points in τ .

LEMMA 5.3.2. [54]

Given two disjoint convex polygons P and Q in the plane, there exists a separating (tangent) line determined by a pair of vertices in $V(P) \cup V(Q)$.

Notice that by the above lemma, for any pair of representative points r_i and r_j (of cliques c_i and c_j respectively), with $r_i r_j \in E(X)$, there exists a line separating c_i and c_j which passes through two points of the convex hull of c_i and c_j .

Now, assume that the algorithm is provided with the following inputs:

- A set of q representative points $r_1, \dots, r_q \in V \cap \tau$ with $r_i \in c_i$;
- Graph X with vertex set $R = \{r_1, \dots, r_q\}$, and

- The pairs of vertices (points in $V \cap \tau$) incident to each of the separating lines for the pairs $r_i r_j \in E(X)$.

Having the above information, one can verify in $\mathcal{O}(n^2)$ if it corresponds to a valid clique partition of $V \cap \tau$. An algorithm can then guess every possible combinations of input of the above form and validate if any combination corresponds to a valid solution.

Analysis.

Next, we enumerate the number of possible solutions that one need to check.

- Representative points: There can be at most $\binom{n}{q} \leq n^q$ choices of representative points.
- Graph X : Since $\Delta(X) \leq 54$ and hence there are at most $54q/2$ edges in a candidate X , for any choice of q representative points, there can be at most $2^{27q \log q + \mathcal{O}(q)}$ graphs to be considered.
- Separating lines: For every graph X with $27q$ edges, there can be at most $\binom{n}{27q} \leq n^{54q}/(54q!)$ choices for separating lines.
- Given a graph X with at most $27q$ edges and separating lines, there can be at most $27q! \leq 2^{27q \log q + \mathcal{O}(q)}$ perfect matchings between them.

Knowing that $q \leq 2l^2 + 3l$, the overall time complexity of the algorithm for calculating a minimum clique partition for a UDG on τ goes to:

$$2^{\mathcal{O}(q \log q)} \cdot n^{54q} \leq 2^{\mathcal{O}(l^2 \log l)} \cdot n^{108l^2 + \mathcal{O}(l)}. \quad (5.1)$$

This is a slight speed-up from the calculated running-time of Dumitrescu and Pach [54], $2^{\mathcal{O}(l^2 \log l)} \cdot n^{160l^2 + \mathcal{O}(l)}$ using exactly the same technique yet a stronger packing argument¹.

Finally, choosing $l = \mathcal{O}(1/\epsilon)$, we obtain the overall running-time of $(1 + \epsilon)$ -approximation algorithm as:

$$n^{(176/\epsilon)^2 + \mathcal{O}(1/\epsilon)}. \quad (5.2)$$

¹The authors of [54] mention in the manuscript that the constants in the running-time can be improved with a more detailed analysis.

THEOREM 5.3.1. *There is a randomized algorithm which runs in time $n^{(176/\epsilon)^2 + \mathcal{O}(1/\epsilon)}$ and w.h.p. obtains a $(1 + \epsilon)$ -approximation solution to MINIMUM CLIQUE PARTITION on arbitrary UDGs.*

5.3.2 Using Convex Regions

Next, we introduce a novel approach that can be used to design a PTAS with significant running-time improvement.

Imagine an optimal solution $\mathcal{C} = \{c_1, \dots, c_k\}$ for MINIMUM CLIQUE PARTITION on an $l \times l$ square τ . By Lemma 3.3.1, every maximal clique c_i , $1 \leq i \leq k$, can be represented by at most 4 convex regions in $R_{V \cap \tau}$. We run a brute-force search for a set of convex regions that can make a clique partition of size at most k . For any such a subset R' , then one can check in $\text{poly}(n, k)$ if the number of cliques corresponding to it is at most k and whether $V \cap \tau$ is covered by those cliques as described later. We first maintain an inclusion matrix whose rows correspond to all convex regions in $R_{V \cap \tau}$ and its columns represent the vertices of $V \cap \tau$. For every region $r \in R_{V \cap \tau}$, in the respective row in the matrix, we set the value of elements at columns corresponding to vertices in $\mathcal{X}_{G[V \cap \tau]}(r)$ to 1 and the rest to 0. We construct this matrix only one time for the whole graph.

REMINDER: We call a subset of 3 convex regions $\{r_1, r_2, r_3\} \subseteq R_{V(G)}$ a *non-helly set*, if for all 3 permutations $1 \leq i, j, l \leq 3$ and $i \neq j \neq l$, we have $Y_l = \mathcal{X}_G(r_i) \cap \mathcal{X}_G(r_j) \setminus \mathcal{X}_G(r_l) \neq \emptyset$. If that is the case, then there is a clique with vertex set $Y = \cup_{i=1}^3 Y_i$ in G which is not represented by any of the regions in $R_{V(G)}$. Hence, we need to manually add a new region r' with vertex set Y to the set of regions.

Calculating Non-helly Sets

To take care of the non-helly sets, we perform a pre-processing on the inclusion matrix. In particular, we check in any subset of 3 regions of $R_{V \cap \tau}$ for non-helly sets. If we find one, we add a new row corresponding to the clique that the non-helly region represents. In particular, let \vec{r}_i denote the binary vector corresponding to convex region r_i in the matrix for $1 \leq i \leq 3$. To check for a non-helly set in $\{r_1, r_2, r_3\}$, we simply check if $(\vec{r}_i \wedge \vec{r}_j \wedge \neg \vec{r}_l) \neq \vec{0}$ for $1 \leq i, j, l \leq 3$ and $i \neq j \neq l$. This can be done in $\mathcal{O}(n)$ for one set of 3 regions.

There are at most $|R_{V(G)}|^3 - 3|R_{V(G)}|^2 + 2|R_{V(G)}| = \mathcal{O}(|R_{V(G)}|^3)$ 3-subsets. Therefore, the

pre-processing phase would take $n \cdot |R_{V(G)}|^3$ time for the whole matrix. Since $|V \cap \tau| \leq n$, by Proposition 3.3.1 there are at most $n^2 - n + 1$ regions in $R_{V(G)}$ and hence the time complexity of this phase would be $\mathcal{O}(n^7)$. Observe that any time a non-helly set is detected, at most one new entry is added to the set of regions. Therefore, after the pre-processing there can be at most $|R_{V(G)}|^3$ regions.

Counting the Candidate Sets

Knowing that there are at most $2l^2 + 3l$ cliques in $R_{V \cap \tau}$ (by Lemma 5.2.3), there are at most $\binom{n^2 - n + 1}{l}$ subsets of size l for $l \leq 2l^2 + 3l$ to be considered. Therefore, the total number of subsets of size at most $2l^2 + 3l$ that we need to consider is:

$$\sum_{l' \leq 2l^2 + 3l} \binom{(n^2 - n + 1)^3}{l'} \leq \left((n^2 - n + 1)^3 + 1 \right)^{2l^2 + 3l} \leq n^{12l^2 + 18l} \text{ for } n > 2. \quad (5.3)$$

Given any of the above subset R' of convex regions, we finally check if R' corresponds to a valid clique partition in the following way: Using the inclusion matrix, we can check if R' includes every vertex in $V \cap \tau$ by simply summing up the corresponding rows in the binary matrix. This can be done using $n \cdot |R'| = \mathcal{O}(nl^2)$ binary operations. We finally find the smallest subset that satisfied the above criteria. Therefore, the overall running-time of our algorithm would be:

$$n^{12l^2 + 18l} \cdot \mathcal{O}(nl^2) + \mathcal{O}(n^7) = n^{12l^2 + 18l + 1} \cdot \mathcal{O}(l^2). \quad (5.4)$$

This is comparable to the solution of Dumitrescu and Pach [54] which derives an exact algorithm of running-time $2^{\mathcal{O}(l^2)} n^{160l^2 + \mathcal{O}(l)}$ for MINIMUM CLIQUE PARTITION on a UDG restricted to τ .

Finally, after replacing l by the value of $16/\epsilon$ (alternatively any $\mathcal{O}(1/\epsilon)$ value), by the analysis presented in the plane decomposition section, we obtain $(1 + \epsilon)$ -approximation algorithm for MINIMUM CLIQUE PARTITION of running-time

$$n^{\mathcal{O}(1/\epsilon^2)} \cdot (1/\epsilon^2). \quad (5.5)$$

THEOREM 5.3.2. *There is a randomized algorithm which runs in time $n^{(64/\epsilon)^2 + 288/\epsilon + 1} \cdot \mathcal{O}(1/\epsilon^2)$ and w.h.p. produces a $(1 + \epsilon)$ -approximation solution to MINIMUM CLIQUE PARTITION on arbitrary UDGs.*

5.4 Summary

We devised randomized $(1 + \epsilon)$ -approximation algorithms for MINIMUM CLIQUE PARTITION on UDGs. Our first PTAS is based on the technique of [54] and a new packing argument and has an improved running-time of $n^{(176/\epsilon)^2 + \mathcal{O}(1/\epsilon)}$. Furthermore, we introduced a new approach for deriving PTAS which still runs in time $n^{\mathcal{O}(1/\epsilon^2)} \cdot (1/\epsilon^2)$ yet the hidden constant in big \mathcal{O} notation is significantly (at least 13 times) smaller than the fastest previously known PTAS. In practice, this allows our algorithm to run in a reasonable time on instances of size $|I|^{13}$, if the previous algorithms could solve an instance I of size $|I|$.

Chapter 6

Parameterized Algorithms for Other Problems

This chapter, investigates the application of the framework introduced in Chapters 3 and 4 to other related problems. We showcase this by demonstrating its use for obtaining parameterization results for problems of CLIQUE COVER and WEIGHTED CLIQUE PARTITION.

6.1 CLIQUE COVER

Apart from CLIQUE PARTITION which was studied earlier, CLIQUE COVER is probably the most discussed variant within the literature of algorithms and complexity. The CLIQUE COVER problem asks for partitioning the edges of a given graph into a minimum number of cliques \mathcal{C} such that every edge is in *at least* one clique in \mathcal{C} . Comparing to CLIQUE PARTITION, CLIQUE COVER is less studied in the literature. In terms of complexity, it is proven that CLIQUE COVER is NP-complete even when restricted to planar graphs [23]. In the class of general graph, the problem is proved to be in FPT through a data reduction resulting in a kernel of size $\mathcal{O}(2^k)$ [70]. The hardness of kernelization of the problem is discussed in [31] where the authors show that indeed the problem does not admit a polynomial-size kernel on general graphs unless the polynomial hierarchy collapses to its third level. For any graph G , the cardinality of a minimum clique cover (clique cover number) is equivalent to the *intersection number* of G which is defined as the smallest number of elements in a representation of G as an intersection graph of finite sets [72]. Erdős, Goodman & Pósa

[57] proved that the edges of any graph on n vertices can be partitioned into at most $n^2/4$ cliques, all of which are either single edges or triangles showing an upper-bound of $n^2/4$ for the intersection number.

To the best of our knowledge there is no work discussing this problem on UDG. From the approximation point of view, it is known that CLIQUE COVER is inapproximable within a ratio $|V|^\epsilon$ of the optimal solution [89] while on planar graphs there is a PTAS for the problem [12]. There is a ratio-preserving reduction from CLIQUE COVER to the CLIQUE PARTITION problem due to Kou et al. [84] making the approximation of the problem as difficult as for the CLIQUE PARTITION.

PROPOSITION 6.1.1. [84]

Let $\text{OPT}_{cp}(G)$ and $\text{OPT}_{cc}(G)$ denote the size of optimal solution for the CLIQUE PARTITION and CLIQUE COVER problems on a given graph G respectively and c a non-negative constant. There is a polynomial-time algorithm A for the CLIQUE COVER problem such that $A(G) \leq c \cdot \text{OPT}_{cc}(G) + d$ for all G if and only if there is a polynomial-time algorithm A' for the CLIQUE PARTITION problem such that $A'(G) \leq c \cdot \text{OPT}_{cp}(G) + d'$ for all G .

6.1.1 Data Reduction

The CLIQUE COVER problem is similar to the CLIQUE PARTITION with the difference that the concept of coverage applies to the edges as opposed to the vertices in the former problem.

CLIQUE COVER.

Instance: An undirected graph $G = (V, E)$, a non-negative integer k .

Question: Is there a set \mathcal{C} of cliques in G such that for every edge $e \in E$, there exists at least one clique $c \in \mathcal{C}$ with $e \in c$ and $|\mathcal{C}| \leq k$?

For simplicity, we first consider another version of the problem named X -ANNOTATED CLIQUE COVER where a subset X of edges of the graph are already covered. The original setting of the problem is then obtained when $X = \emptyset$.

In order to be able to design parameterized algorithms for CLIQUE COVER, we initially shrink the instance by applying a set of reduction rules in a pre-processing phase. The intuition behind Rule 1 is that since every edge of the graph has to be in a clique in an optimal clique partition, if for an edge e of the graph, there is only one clique c which covers e , then c has to be in any optimal solution. Similarly in Rule 2, if there exists a set of

edges such that for every maximal clique c , either all of the edges are included in c or all are excluded from c , then every clique in any optimal solution that includes one of the edges would also include the rest. Therefore, one can safely remove all these edges except for one.

Rule 1. For an edge e included in one clique c in G only, mark every edge in c as covered and decrease k by 1.

Rule 2. For a set of maximal cliques $\mathcal{C} = \{c_1, \dots, c_\gamma\}$ in G , let $I = \cap_{i=1}^\gamma V(c_i)$. Assume that $|I| \geq 3$ and $N[I] = \cup_{i=1}^\gamma V(c_i)$; that is, no other maximal cliques in G intersect I . We replace I in G with a new edge $e' = xy$ and make x and y adjacent to $N(I)$ ¹.

THEOREM 6.1.1. *Rules 1 and 2 are correct for CLIQUE COVER on UDGs and can be performed in $\mathcal{O}(n^{2.3727} + nm)$ time on any graph G with n vertices and m edges.*

Proof. For Rule 1, in order to find out whether an edge is included in one clique c of G only, we need to check for every edge $e = (v, w) \in E$ if $G[N[v]] = G[N[w]] = K_{\deg(v)}$. This can be done by checking the presence of all edges in $G[N[v, w]]$ in $(\deg(v) + \deg(w))^2$ time for e and in $\mathcal{O}(nm)$ time for the whole graph. Notice that the vertices removed according to Rule 1 are *simplicial* vertices and hence recognizing all such vertices can be done using fast matrix multiplication [83]. Let \mathbf{A} be the adjacency matrix of G with 1 entries on its diagonal. To recognize all simplicial vertices, one need to calculate \mathbf{A}^2 . This can be performed in $\mathcal{O}(n^{2.3727})$ using the algorithm in [114]. Once a simplicial vertex x is removed from the graph, one can recalculate \mathbf{A}^2 in $\mathcal{O}(\deg^2(x))$ time. Therefore, exhaustive application of Rules 1 and 2 takes $\mathcal{O}(n^{2.3727} + nm)$ in total.

Also notice that Rule 1 does not change the structure of the graph. Therefore the graph obtained after applying the reduction rule is still realizable as a UDG. Consider an optimal solution OPT for CLIQUE COVER on G and let c_e be the maximal clique in OPT that includes e . Notice that by the condition of the rule, there is only one such a clique. Let OPT' be an optimal solution for G' . Then $\text{OPT}' \cup c_e$ is a solution for G . If $|\text{OPT}' \cup c_e| \leq |\text{OPT}|$, then we are done. Assume, $|\text{OPT}' \cup c_e| > |\text{OPT}|$. Since $\text{OPT} \setminus c_e$ is a solution for G' and $|\text{OPT}'| > |\text{OPT} \setminus c_e|$, we get a contradiction.

For Rule 2, we need to find the common clique in neighborhood of every edge $e = vw$.

¹A detailed analysis of Rule 2 revealed that despite the different definition, the rule is indeed similar to the ones introduced by Gramm et al. [70].

This can be done by checking if $N[v] = N[w]$ and then looking for a maximum subset $I \subseteq N[v]$ of size at least 3 such that for every vertex $x \in I$, $N[x] = N[v]$. Given G in form of an adjacency matrix, one can construct this subset in the following way. Initially, we set $I = \{v, w\}$. For every 1 entry in the row corresponding to vertex v , we check if the row corresponding to that entry is exactly the same as the one for v and if it is the case, we add the corresponding vertex to I . This can be done in $\deg^2(v)$ time for e and in $\mathcal{O}(nm)$ time for the entire graph. Similarly, this step can be done using fast matrix multiplication. Let \mathbf{A} be the adjacency matrix of G with 1 entries on its diagonal. Notice that for every pair of vertices $x, y \in V(G)$, $(\mathbf{A}^2)_{x,y} = |N[x] \cap N[y]|$. Hence, to decide if x and y are in I , it is sufficient to check if $|N[x] \cap N[y]| = |N[v]| = |N[y]|$. Using \mathbf{A}^2 matrix, vertices x and y are in I , if $(\mathbf{A}^2)_{x,y} = (\mathbf{A}^2)_{x,x} = (\mathbf{A}^2)_{y,y}$. Contracting I into an edge can be done in constant time. Hence, Rule 2 takes $\mathcal{O}(n^{2.3727})$ using the matrix multiplication algorithm reported in [114]. Notice that application of Rule 2 does not generate any new reducible vertices and hence only one round of the application of Rule 2 is sufficient.

It remains to show that after applying the reduction Rule 2, the graph can still be realized as a UDG. To see this, consider a common clique I . After replacing I by an edge $e' = xy$, it holds that $N(e') = N(I)$. Consider $\Lambda_I = \cap_{v \in I} \mathcal{D}(v)$. By the condition of the reduction rule, for every vertex $w \in N[I] \setminus I$ the corresponding unit disk $\mathcal{D}(w)$ has to intersect Λ_I in a non-empty region. But since Λ_I can be bounded in a unit disk, after replacing Λ_I with $\mathcal{D}(x) \cap \mathcal{D}(y)$, it is possible to redraw the disks such that $E(G[V \setminus I])$ does not change in the reduced graph.

Next, we show the correctness of the reduction rule. Suppose the reduction rule is applied. Then, there exists a collection of maximal cliques $\mathcal{C} = \{c_1, \dots, c_\gamma\}$ such that for every arbitrary edge $vw \in G[I]$ in their intersection, $N[\{v, w\}] = \cup_{i=1}^\gamma V(c_i)$. Notice that G' , the graph obtained after applying the reduction rule can be equivalently derived from G after contracting $I' = G[I] \setminus vw$ into a vertex. Also the size of every clique in \mathcal{C}/I' is at least 3.

Let OPT be an optimal solution for CLIQUE COVER on G . We prove that $\text{OPT}' = \text{OPT}/I'$ is an optimal solution for G' . For the sake of contradiction, assume that G' has a solution OPT^* smaller than OPT/I' and let $\mathcal{C}^\dagger = \{c_j \in \mathcal{C} \mid c_j/I' \in \text{OPT}^*\}$. Then one can construct a solution $\text{OPT}^\dagger = \text{OPT}^* \setminus (\mathcal{C}^\dagger/I') \cup \mathcal{C}^\dagger$ for G such that $|\text{OPT}^\dagger| = |\text{OPT}^*|$. However, since $|\text{OPT}| = |\text{OPT}'|$, we get $|\text{OPT}^\dagger| < |\text{OPT}|$, a contradiction.

Now, assume that OPT' is an optimal solution for G' and let $c_{e'} \in \text{OPT}'$ be the clique that includes e' . Then $G[(V(c_{e'}) \setminus \{x, y\}) \cup I]$ is a clique in G . We prove that $\text{OPT}^* = (\text{OPT}' \setminus c_{e'}) \cup G[(V(c_{e'}) \setminus \{x, y\}) \cup I]$ is an optimal solution for G . That is, $|\text{OPT}^*| \leq |\text{OPT}|$. Assume for the contradiction that $|\text{OPT}^*| > |\text{OPT}|$. Then $\text{OPT}^\ddagger = \text{OPT}/I'$ is a solution for G' and $|\text{OPT}^\ddagger| = |\text{OPT}| < |\text{OPT}'|$, a contradiction to the assumption that OPT' is an optimal solution for G' . \square

6.1.2 A Parameterized Algorithm on Arbitrary UDGs

Again, we can use a similar brute-force search algorithm as the one described for CLIQUE PARTITION in order to obtain a parameterized algorithm for the problem on ϵ -precision UDGs. By Proposition 3.3.1 there are at most $n^2 - n + 1$ convex regions in $R_{V(G)}$. We maintain a matrix whose rows corresponds to convex regions in $R_{V(G)}$ and whose columns are marked by the edges in $E(G)$. We perform a similar process as the one described before in order to take care of non-helly sets (See Definition 3.3.2). Therefore, the total number of guesses for a subset of the regions that can correspond to a clique cover remains the same as Formula 3.5. Yet in order to check whether a subset R' of convex regions corresponds to a valid clique cover (That is, it includes every edge in the graph), this time we need at most $m \cdot |R'| = \mathcal{O}(km) = \mathcal{O}(kn^2)$ binary operations.

Therefore, the overall running-time of the algorithm would be:

$$n^{6k} \cdot \mathcal{O}(kn^2) + \mathcal{O}(n^7) = \mathcal{O}(kn^{6k+2}) = \mathcal{O}(n^{6k+3}) \text{ for } n > 2. \quad (6.1)$$

THEOREM 6.1.2. *There is an algorithm that given any integer k , decides in $\mathcal{O}(n^{6k+3})$ time, if there exists a clique cover of size at most k for an arbitrary UDG.*

Combining this with the kernelization result of [70], we get the following.

COROLLARY 6.1.1. *There is an algorithm that given any integer k , decides in $\mathcal{O}(2^{6k^2+3k} + mn)$ time, if there exists a clique cover of size at most k for an arbitrary UDG.*

6.1.3 Analysis

We use a similar approach as the one for CLIQUE PARTITION in order to obtain parameterization results for CLIQUE COVER problem.

DEFINITION 6.1.1. [Clique-edge intersection graph]

Let G be the graph obtained after applying the reduction rules and consider \vec{G} . We construct the clique-edge intersection graph, $\mathcal{H} = (\mathcal{S}, \mathcal{E})$ in the following way. The vertices in \mathcal{H} correspond to sink vertices \mathcal{S} in \vec{G} . Two vertices s_1 and s_2 are adjacent in \mathcal{H} , if there is an edge $e = vw \in E(G)$ such that $\{v, w\} \subseteq \mathcal{X}_G(s_1) \cap \mathcal{X}_G(s_2)$.

Precision UDGs

Given two points p, q at the mutual distance d corresponding to an edge $e = vw$ in a graph $G \in \text{UDG}_\epsilon$, we denote by $\Lambda_{r,d}$ the intersection of two disks of radii r centered at p and q . Notice that $\Lambda_{r,\epsilon} \subseteq \mathcal{D}_r$. In order to bound $\Delta(\mathcal{H})$, we count the maximum number of convex regions in $\Lambda_{1/2,\epsilon} \subset \mathcal{D}_{1/2}$.

LEMMA 6.1.1. *Let $\Delta(\mathcal{H})$ denote the maximum degree of the clique-edge intersection graph \mathcal{H} , then $\Delta(\mathcal{H})$ can be bounded as follows: $\Delta(\mathcal{H}) \leq \mu_{1/2} \cdot \eta_\delta \leq 41/\epsilon^4$.*

Proof. First, we need to calculate the minimum number of disks \mathcal{D}_δ required to cover $\Lambda_{1/2,\epsilon}$. We use the fact that $\Lambda_{1/2,\epsilon} \subset \mathcal{D}_{1/2}$ and count the number of \mathcal{D}_δ required to cover $\mathcal{D}_{1/2}$ instead. We have previously calculated this number as μ_r in Lemma 4.2.2 for \mathcal{D}_r . Hence, we get:

$$\mu_{1/2} = 3 \left\lceil \frac{2}{3} \left(\frac{1}{2\delta} - 2 \right) + 1 \right\rceil \left\lceil \frac{2}{3} \left(\frac{1}{2\delta} - 2 \right) + 2 \right\rceil \leq 6/\epsilon^2. \tag{6.2}$$

After multiplying $\mu_{1/2}$ by η_δ , we get the upper bound on $\Delta(\mathcal{H})$. □

THEOREM 6.1.3. *Let $G \in \text{UDG}_\epsilon$. Then its clique-edge intersection graph \mathcal{H} has at most $168k/\epsilon^4$ vertices.*

Proof. Without loss of generality, we assume that \mathcal{H} is connected. As if it is otherwise, the solution can be independently obtained on every connected component. Also since the graph is reduced, every vertex has a degree of at least 2. Let $\text{OPT} = \{c_1, \dots, c_k\}$ be a solution for CLIQUE COVER on G . We call a vertex $s \in V(\mathcal{H})$ covered if $\mathcal{X}_G(s) \subseteq E(c_i)$ for some $1 \leq i \leq k$. For a covered vertex s , we call all edges of \mathcal{H} incident to s , covered. For $s' \in V(\mathcal{H})$, if s' is not incident to a covered edge, then there is an edge of $G[\mathcal{X}_G(s')]$ not included in any clique in OPT . Therefore, each vertex of \mathcal{H} is incident to at least one covered edge. By Lemma 3.3.1, each clique $c \in \text{OPT}$ can be replaced by at most 4 vertices

from $V(\mathcal{H})$. This means that there are at most $4\Delta(\mathcal{H})$ vertices of \mathcal{H} incident to a covered edge. Therefore, $k \geq |V(\mathcal{H})|/4(\Delta(\mathcal{H}) + 1)$. Combining this with Lemma 6.1.1, we get:

$$|V(\mathcal{H})| \leq 4k(\Delta(\mathcal{H}) + 1) \leq 164k/\epsilon^4 + 4k. \quad (6.3)$$

Therefore for $\epsilon \leq 1$, $|V(\mathcal{H})| \leq 168k/\epsilon^4$. □

Quasi-precision UDGs

It is easy to observe that a result parallel to Lemma 4.3.2 also holds for CLIQUE COVER.

LEMMA 6.1.2. *For a (λ, α) -quasi-precision UDG with $\alpha \leq 1/2$ which is reduced with respect to the reduction rules for CLIQUE COVER, $|X_i| = \mathcal{O}(\mu(\Psi_i, \lambda)^2 + |S^2|)$ for all $1 \leq i \leq \kappa$.*

Proof. Draw an imaginary circle of radius 1 at each vertex $v \in \overline{N^\cap(X_i)}$. Let C be the set of all such circles. Notice that every such a circle intersects ζ_i in at least two points on the boundary. Let Φ^* be the union of all these circles and $\mathcal{D}(\zeta_i)$ denote the disk with ζ_i as its boundary. Consider the set of all regions R in $\mathcal{D}(\zeta_i) \setminus \Phi^*$ formed after intersecting the boundary of circles with $\mathcal{D}(\zeta_i)$. We show that for any $r \in R$, if there are more than two vertices in $V \cap \bar{r}$, then Rule 2 is applicable to G by setting $I = V \cap \bar{r}$. First notice that for every vertex $v \notin \overline{N^\cap(X_i)}$, either $I \subseteq N[v]$ or $I \cap N[v] = \emptyset$. Also since r is a region, for any vertex $x \in \overline{N^\cap(X_i)}$, either $I \subseteq N(x)$ or none of vertices in I are adjacent to x . Thus, I can be replaced by a single edge by Rule 2. As a result, in the reduced graph, there can be at most two vertices left in each region of R . In particular, $|X_i| \leq 2|R|$. To complete the proof, we upper-bound $|R|$. However, $\overline{N^\cap(X_i)} \subseteq \Psi_i$ by Lemma 4.3.1 and hence we can safely use an argument similar to the one in Lemma 4.2.1 to count the number of regions in R knowing that there are at most $\mu(\Psi_i, \lambda) + |S|$ vertices in Ψ_i . Consider the graph obtained from $\mathcal{D}(\zeta_i) \cap \Phi^* \cup \zeta_i$. Intersection point of every two circles in $C \cup \zeta_i$ forms a vertex of the graph. Two vertices are connected by an edge in this graph if their corresponding points are on the same circle in $C \cup \zeta_i$. Observe that the vertices which are located on the circle ζ_i have degree 3 and there are at most $2(\mu(\Psi_i, \lambda) + |S|)$ such vertices. This is while all other vertices have degree 4 because they are formed by intersecting two circle in C . Let there be n^* such vertices. The faces in the graph correspond to the regions R . Notice that we can get the number of edges in the above planar graph as half of sum of the vertex degrees as $2n^* + 3(\mu(\Psi_i, \lambda) + |S|)$. Using the Euler formula for planar graphs, we can find the number

of faces (also $|R|$) as:

$$|R| = n^* + \mu(\Psi_i, \lambda) + |S| + 2. \tag{6.4}$$

In the worst case when every two circles in C intersect at one point in $\mathcal{D}(\zeta_i)$, we get $n^* = \mathcal{O}(\mu(\Psi_i, \lambda) + |S|)^2$. Now since $|X_i| \leq 2|R|$, the statement of the lemma holds. \square

Finally, using the result of Lemma 6.1.3, one can prove that the size of edge-clique intersection graph \mathcal{H} is a polynomial in k even when the reduced graph is (λ, α) -quasi-precision UDG with $\alpha \leq 1/2$. The details of the approach are very similar to the one for CLIQUE PARTITION and hence are omitted here to avoid redundancy.

COROLLARY 6.1.2. *Let $G' \in \text{UDG}_{(\lambda, 1/2)}$. Then its clique-edge intersection graph \mathcal{H} has $\mathcal{O}(k/\lambda^8)$ vertices.*

6.1.4 Fixed-parameter Algorithms on Precision and Quasi-precision UDGs

Again, we can use a similar brute-force search algorithm as the one described for CLIQUE PARTITION in order to obtain a fixed-parameter tractable algorithm for the problem on ϵ -precision UDGs. By Theorem 6.1.3 (respectively, Corollary 6.1.2), reduced precision (resp. a quasi-precision) UDG only $\mathcal{O}(k/\epsilon^4)$ (resp. $\mathcal{O}(k/\lambda^8)$) convex regions remain. We maintain a matrix whose rows corresponds to convex regions in $R_{V(G)}$ and whose columns are marked by the edges in $E(G)$. We perform a similar process as the one described before in order to take care of non-helly sets. Notice that this time, we originally have only $|V(\mathcal{H})| = \mathcal{O}(k)$ rows in the binary matrix described before. Therefore, the time required to pre-process the matrix for non-helly sets would be $\mathcal{O}(|V(\mathcal{H})|^3 n)$ and after pre-processing at most $|V(\mathcal{H})|^3$ regions are in the set. The pre-processing is only performed once for the whole matrix and hence for $k > 0$ this factor is dominated by the time required for guessing the subsets and checking the validity of the clique cover.

In order to decide whether $|\text{OPT}(G)| \leq k$, we enumerate all sets of at most k regions and for each such a set, check whether it corresponds to a clique cover. The number of such subsets could hence be at most

$$\sum_{k' \leq k} \binom{|V(\mathcal{H})|^3}{k'} \leq (|V(\mathcal{H})|^3 + 1)^k = \mathcal{O}(2^{3k \log |V(\mathcal{H})|}). \tag{6.5}$$

Taking into account the running-time of $\mathcal{O}(mn)$ for applying the reduction rules and $\mathcal{O}(km)$ time for checking the validity of each clique cover, we get the overall time complexity of the algorithm on ϵ -precision UDGs as:

$$\mathcal{O}(2^{3k \log |V(\mathcal{H})|} km + |V(\mathcal{H})|^3 n + mn) = \mathcal{O}(2^{(3k+1) \log k + 12k \log \frac{1}{\epsilon} + 24k} m + mn). \quad (6.6)$$

THEOREM 6.1.4. *There is an algorithm that given any integer k , decides in $\mathcal{O}(2^{(3k+1) \log k + 12k \log \frac{1}{\epsilon} + 24k} m + mn)$ time, if there exists a clique cover of size at most k for a UDG whose reduced graph is ϵ -precision.*

Similarly, the time complexity of the exact algorithm for CLIQUE COVER on quasi-precision graphs with $\alpha \leq 1/2$ would be:

$$\mathcal{O}(|V(\mathcal{H})|^{3k} km + |V(\mathcal{H})|^3 n + mn) = 2^{\mathcal{O}(k \log \frac{k}{\lambda})} m + \mathcal{O}(mn). \quad (6.7)$$

THEOREM 6.1.5. *There is an algorithm that given any integer k , decides in $2^{\mathcal{O}(k \log \frac{k}{\lambda})} m + \mathcal{O}(mn)$ time, if there exists a clique cover of size at most k for any UDG whose reduced graph is (λ, α) -relaxed-quasi-precision with $\alpha \leq 1/2$.*

6.2 WEIGHTED CLIQUE PARTITION

Another interesting variant is the WEIGHTED CLIQUE PARTITION where given a graph with weighted vertices, the goal is to find a partition of vertices of the graph into a set of cliques such that the sum of minimum weights of vertices in cliques is minimized. WEIGHTED CLIQUE PARTITION has been studied in the context of circular arc graphs [10] and interval graphs [59, 68]. Pirwani and Salavatipour [102] study approximation algorithms for the problem on UDGs.

Given a graph $G = (V, E)$ with each vertex $v \in V(G)$ assigned a positive integer weight $Wt(v)$. For every clique c of G , we define weight of c , $Wt(c) = \max_{v \in V(c)} Wt(v)$.

WEIGHTED CLIQUE PARTITION.

Instance: An undirected graph $G = (V, E)$, a non-negative integer k .

Question: Is there a set \mathcal{C} of cliques in G such that for every vertex $v \in V$, there exists at least a clique $c \in \mathcal{C}$ with $v \in c$ and $\sum_{c \in \mathcal{C}} Wt(c) \leq k$?

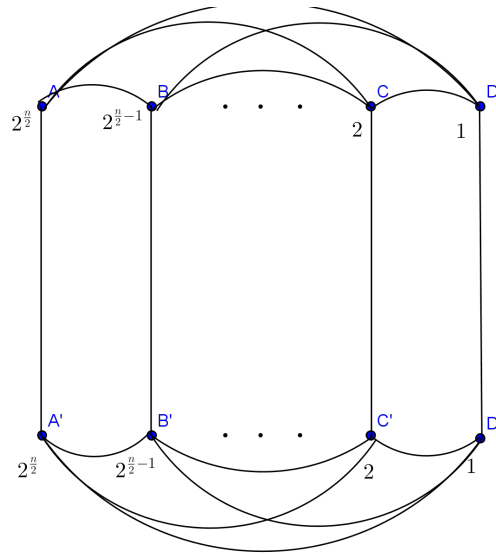


Figure 6.1: A weighted graph on n vertices consisting of two cliques and a perfect matching. Choosing the two cliques of size $n/2$ as the partition would result in a solution of weight $2^{n/2+1}$ while the optimal solution for Weighted Clique Partition is the set of $n/2$ matchings.

WEIGHTED CLIQUE PARTITION distinguishes itself from CLIQUE PARTITION in two ways. For example, the separability property of cliques discussed in Lemmas 5.1.1 and 5.1.2 does not hold for the problem. One immediate implication is that in an optimal solution for the problem, there can be minimal weighted cliques such that their convex hull overlap. Secondly, the number of cliques in an optimal solution for a UDG in a bounded radius (area) region is not bounded by the radius (area) of the region.

In particular, it is easy to construct examples for which the *clique partition* with minimum cardinality is not a *weighted clique partition* with minimum weight. As an example, the graph in Figure 6.1 consists of two cliques of $n/2$ vertices each as well as a perfect matching between the cliques. Choosing the cliques on $n/2$ vertices would result in a solution of weight $2^{n/2+1}$ while the set of matchings has a smaller weight of $\sum_{i=1}^{n/2} 2^i = 2^{n/2+1} - 1$.

6.2.1 Data Reduction

In order to be able to devise fixed-parameter algorithms for WEIGHTED CLIQUE PARTITION, we first get rid of some parts of the graph in a pre-processing data reduction phase. The intuition behind the reduction rules is similar to the ones for CLIQUE PARTITION as discussed in Section 3.4.

Rule 1. If v is an isolated vertex or a vertex adjacent only to covered vertices in G then remove v from G decrease k by $Wt(v)$.

Rule 2. For a set of vertices included in one clique c of G only, mark every vertex in c as covered and decrease k by $Wt(v)$.

Rule 3. For a set of maximal cliques $\mathcal{C} = \{c_1, \dots, c_\gamma\}$ in G , let $I = \bigcap_{i=1}^\gamma V(c_i)$. Assume that $|I| \geq 2$ and $N[I] = \bigcup_{i=1}^\gamma V(c_i)$; that is, no other maximal cliques in G intersect I . We replace I in G with a vertex x with weight $Wt(x) = \max_{v \in I} Wt(v)$ and make x adjacent to $N(I)$.

THEOREM 6.2.1. *The reduction rules are correct for WEIGHTED CLIQUE PARTITION UDG and can be performed in time $\mathcal{O}(n^{2.3727} + mn)$.*

Proof. Given a UDG G , in the form of an adjacency matrix, for Rule 1, finding isolated vertices or those only adjacent to covered vertices can be done by checking every edge of G in $\mathcal{O}(m)$ time for the whole graph. Let G' be the graph obtained after applying Rule 1 to a vertex v in G , then the fact that G is a UDG implies that G' is a UDG. Now we prove the correctness of Rule 1. Suppose (G, k) is a yes instance and consider an optimal solution OPT for WEIGHTED CLIQUE PARTITION on G . Then $G[v] \in \text{OPT}$. Given any optimal solution OPT' for G' , $\text{OPT}' \cup G[v]$ is a solution for G . If $Wt(\text{OPT}') \leq Wt(\text{OPT}) - Wt(v)$, then OPT' is an optimal solution for G and we are done. Assume $Wt(\text{OPT}') > Wt(\text{OPT}) - Wt(v)$. However, $\text{OPT}' \setminus G[v]$ is a solution for G' and $Wt(\text{OPT}' \setminus G[v]) < Wt(\text{OPT}')$, a contradiction.

For Rule 2, in order to find out whether a vertex is included in one maximal clique c of G only, we need to check for every vertex $v \in V(G)$ if $G[N[v]]$ is a clique. This can be done by checking the presence of all edges in $G[N[v]]$ in $\text{deg}^2(v)$ time for v and in $\mathcal{O}(nm)$ time for the whole graph. Notice that the vertices removed according to Rule 1 and Rule 2, are *simplicial* vertices and hence recognizing all such vertices can be done using fast matrix

multiplication [83]. Let \mathbf{A} be the adjacency matrix of G with 1 entries on its diagonal. To recognize all simplicial vertices, one needs to calculate \mathbf{A}^2 . This can be performed in $\mathcal{O}(n^{2.3727})$ time [114]. Once a simplicial vertex x is removed from the graph, one can recalculate \mathbf{A}^2 in $\mathcal{O}(\deg^2(x))$ time. Therefore, exhaustive application of Rules 1 and 2 takes $\mathcal{O}(n^{2.3727} + nm)$ in total.

Rule 2 also does not change the structure of the graph. Let G' be the graph obtained after applying Rule 2 to G . Consider an optimal solution OPT for WEIGHTED CLIQUE PARTITION on G and let c_v be the clique in OPT that includes v . Let OPT' be an optimal solution for G' . Then $\text{OPT}' \cup c_v$ is a solution for G . If $Wt(\text{OPT}' \cup c_v) \leq Wt(\text{OPT})$, then we are done. Assume $Wt(\text{OPT}' \cup c_v) > Wt(\text{OPT})$. Then $Wt(\text{OPT}') > Wt(\text{OPT}) - Wt(c_v)$. Yet, $\text{OPT} \setminus c_v$ is a solution for G' of size $Wt(\text{OPT}) - Wt(c_v)$, A contradiction.

For Rule 3, we need to construct the set I in the neighborhood of every vertex v . This can be done by checking for a maximum subset of vertices $I \subseteq N[v]$ such that for every vertex $x \in I$, $N[x] = N[v]$. Given G in form of an adjacency matrix, one can construct this subset in the following way. Initially, we set $I = \{v\}$. For every 1 entry in the row corresponding to vertex v , we check if the row corresponding to that entry is exactly the same as the one for v and if it is the case, we add the corresponding vertex to I . This can be done in $\deg^2(v)$ time for v and in $\mathcal{O}(nm)$ time for the entire graph. Similarly, this step can be performed using fast matrix multiplication in time $\mathcal{O}(n^{2.3727})$ using the matrix multiplication [114]. Replacing I with a vertex can be done in constant time.

It remains to show that after applying the reduction Rule 3, the graph can still be represented as a UDG. To see this, notice that $G[I]$ is a clique and hence contracting I to a vertex corresponds to removing all the vertices in I except for one. The UDG for the reduced graph is then obtained after removing the respective points.

If Rule 3 is applied, then there exists a collection of maximal cliques $\mathcal{C} = \{c_1, \dots, c_\gamma\}$ in G such that $N[I] = \cup_{i=1}^\gamma V(c_i)$. Let G' be the graph obtained after contracting $G[I]$ into a vertex x and OPT be an optimal solution for WEIGHTED CLIQUE PARTITION on G . Notice that the size of every clique in $\mathcal{C}/G[I]$ is at least 2. We prove that $\text{OPT}' = \text{OPT}/G[I]$ is an optimal solution for G' . For the sake of contradiction, assume that G' has a solution OPT^* with $Wt(\text{OPT}^*) < Wt(\text{OPT}')$ and let $\mathcal{C}^\dagger = \{c_j \in \mathcal{C} \mid c_j/G[I] \in \text{OPT}^*\}$. Then one can construct a solution $\text{OPT}^\dagger = \text{OPT}^* \setminus (\mathcal{C}^\dagger/G[I]) \cup \mathcal{C}^\dagger$ for G such that $Wt(\text{OPT}^\dagger) = Wt(\text{OPT}^*)$.

However, since $Wt(\text{OPT}) = Wt(\text{OPT}')$, we get $Wt(\text{OPT}^\dagger) < Wt(\text{OPT})$, a contradiction.

Now assume that OPT' is an optimal solution for G' and let $c_x \in \text{OPT}'$ be the clique that includes x . Then $G[(V(c_x)\setminus x) \cup I]$ is a clique in G and $Wt(G[(V(c_x)\setminus x) \cup I]) = Wt(c_x)$. We prove that $\text{OPT}^* = (\text{OPT}' \setminus c_x) \cup G[(V(c_x)\setminus x) \cup I]$ is an optimal solution for G . That is, $Wt(\text{OPT}^*) \leq Wt(\text{OPT})$. Assume for the contradiction that $Wt(\text{OPT}^*) > Wt(\text{OPT})$. Then $\text{OPT}^\ddagger = \text{OPT}/G[I]$ is a solution for G' and $Wt(\text{OPT}^\ddagger) = Wt(\text{OPT}) < Wt(\text{OPT}')$, a contradiction to the assumption that OPT' is an optimal solution for G' . \square

6.2.2 Analysis

A closer look at the reduction rules introduced for WEIGHTED CLIQUE PARTITION reveals that they are analogous to the ones introduced in Chapter 3 for CLIQUE PARTITION in terms of criteria for the application of the rules and the modification that each rule applies to the structure of the graph with the only difference being the weight assigned to the vertices in the reduced instance. It follows that one can use a similar analysis as the one performed for CLIQUE PARTITION to show that the size of clique intersection graph for the reduced instance is $\mathcal{O}(k)$ when the input graph is provided as a precision or a quasi-precision UDG.

LEMMA 6.2.1. *Let $\text{OPT}(G')$ the optimal solution for MINIMUM CLIQUE PARTITION on the reduced graph G' and k a parameter for WEIGHTED CLIQUE PARTITION such that G is a yes-instance. Then $|\text{OPT}(G')| \leq k$.*

Proof. Notice that $Wt(v) \geq 1$ for every vertex v of G . Also since the reduction rules do not create a vertex with a larger weight than the original vertices, the weight of every vertex of G' is at least 1. Now assume that there exists a solution \mathcal{C} for WEIGHTED CLIQUE PARTITION on G' of weight k . Clearly, \mathcal{C} is also a solution for CLIQUE PARTITION on G' . Also for every clique $c \in \mathcal{C}$, by the argument provided earlier $Wt(c) \geq 1$. Therefore, $|\mathcal{C}| \leq Wt(\mathcal{C}) = \sum_{c \in \mathcal{C}} Wt(c) = k$. Finally, by the definition of $\text{OPT}(G')$, it follows that $|\text{OPT}(G')| \leq |\mathcal{C}|$. Thus $|\text{OPT}(G')| \leq k$. \square

Given a graph G , let $\text{OPT}(G)$ be an optimal solution for MINIMUM CLIQUE PARTITION. Clearly if we set $k = |\text{OPT}(G)|$, then G would be a yes-instance for CLIQUE PARTITION and hence Theorems 4.3.1 and 4.2.1 hold. Therefore, as a consequence of Lemma 6.2.1, the following corollaries hold for WEIGHTED CLIQUE PARTITION.

COROLLARY 6.2.1. *Let H be the clique intersection graph for an ϵ -precision UDG which is reduced according to the reduction rules for WEIGHTED CLIQUE PARTITION. Then if G is a yes-instance, $|V(H)| \leq 728k/\epsilon^4$ for $\epsilon \leq 1$.*

COROLLARY 6.2.2. *Let H be the clique intersection graph for a (λ, α) -quasi-precision UDG with $\alpha \leq 1/2$ which is reduced according to the reduction rules for WEIGHTED CLIQUE PARTITION. Then if G is a yes-instance, $|V(H)| = \mathcal{O}(k/\lambda^8)$.*

6.2.3 Fixed-parameter Algorithms

Next we describe how to use convex regions in order to obtain an FPT for WEIGHTED CLIQUE PARTITION on the precision and quasi-precision UDGs.

We use an approach similar to the one for CLIQUE PARTITION in Section 6.2.3. The only difference is that here for every guessed set of convex regions, we need to calculate the size of the solution with respect to the weight of the clique that it represents. Given a subset of convex regions R' , this can be performed in $\mathcal{O}(n)$. Since every vertex in G and hence G' has a weight larger than 1, as before, there can be at most k cliques in every feasible solution. The overall time-complexity of the algorithm can therefore be calculated similar to Section as:

$$\mathcal{O}(2^{(3k+1) \log k + 12k \log \frac{1}{\epsilon} + 30k} n + mn). \tag{6.8}$$

for ϵ -precision UDGs and as:

$$2^{\mathcal{O}(k \log \frac{k}{\lambda})} n + \mathcal{O}(mn) \tag{6.9}$$

for (λ, α) -quasi precision UDG with $\alpha \leq 1/2$.

THEOREM 6.2.2. *There is an algorithm that given any integer k , decides in $\mathcal{O}(2^{(3k+1) \log k + 12k \log \frac{1}{\epsilon} + 30k} n + mn)$ time, if there exists a clique partition of weight at most k for a UDG whose reduced graph is ϵ -precision.*

THEOREM 6.2.3. *There is an algorithm that given any integer k , decides in $2^{\mathcal{O}(k \log \frac{k}{\lambda})} n + \mathcal{O}(mn)$ time, if there exists a clique partition of weight at most k for a UDG G whose reduced graph is (λ, α) -(relaxed)-quasi-precision with $\alpha \leq 1/2$.*

6.3 Summary and Future Work

In an attempt to generalize the parameterization approach of Chapter 3 for CLIQUE PARTITION on UDGs, to other clique covering problems, we study the problems of CLIQUE COVER and WEIGHTED CLIQUE PARTITION on UDGs. Despite the similarities in their definitions to CLIQUE COVER, the latter problems exhibit very different structural and computational complexity behaviors. Introducing a set of reduction rules for CLIQUE COVER, we show how the framework in Chapter 3 can be adapted in order to obtain parameterization algorithms for CLIQUE COVER on UDGs. Particularly, when considering arbitrary UDGs as input, using a similar approach as the one for CLIQUE PARTITION, we construct a brute-force algorithm of running-time $\mathcal{O}(n^{6k+3})$ for the problem. We further devise FPT algorithms for CLIQUE COVER on the classes of ϵ -precision and (λ, α) -(relaxed)-quasi-precision UDGs of running-times $\mathcal{O}(2^{(3k+1)\log k + 12k \log \frac{1}{\epsilon} + 24k} m + mn)$ and $2^{\mathcal{O}(k \log \frac{k}{\lambda})} m + \mathcal{O}(mn)$ respectively.

Our solution for the WEIGHTED CLIQUE PARTITION is more straightforward. Particularly, our introduced set of reduction rules for the problem are structurally analogous to those of CLIQUE PARTITION in Section 3.4. Therefore, using the analysis of Chapter 3, we can first obtain parameterization results with respect to a parameter which is the size of optimal MINIMUM CLIQUE PARTITION of the instance. We complete the proof by proving a reduction from WEIGHTED CLIQUE PARTITION to MINIMUM CLIQUE PARTITION. This leads to fixed-parameter algorithms of running-times $\mathcal{O}(2^{(3k+1)\log k + 12k \log \frac{1}{\epsilon} + 30k} n + mn)$ and $2^{\mathcal{O}(k \log \frac{k}{\lambda})} n + \mathcal{O}(mn)$ for WEIGHTED CLIQUE PARTITION on the classes of ϵ -precision UDGs and (λ, α) -(relaxed)-quasi-precision UDGs with $\alpha \leq 1/2$, respectively. To the best of our knowledge, this is the first parameterization results for the above problems on UDGs.

The framework introduced here might be able to be tailored to obtain parameterization results for other related problems. For the readers' interest, we also mention a few other related problems below. Problems of BICLIQUE VERTEX PARTITION/COVER are studied in [60] where the goal is to partition/cover the vertex set of a given graph into minimum number of complete bipartite subgraphs (*bicliques*). When parameterized by the number of bicliques in the solution, unlike CLIQUE PARTITION/COVER, BICLIQUE VERTEX PARTITION/COVER problems are not in FPT even when the input graph is restricted to be bipartite.

Another version of the problems where instead of V the *edge set* of the graph is required to be partitioned/covered is referred to as BICLIQUE PARTITION/COVER in [60]. It is known that BICLIQUE PARTITION/COVER to admit exponential kernels of at most 3^k vertices in

the class of general graphs and at most 2^{k+1} vertices in the class of bipartite graphs and hence is fixed-parameter tractable [60]. The vertex cover number, is a trivial upper bound on the size of the parameters for the two latter problems.

Chapter 7

A General Kernelization Framework on Quasi-precision UDGs

In this chapter, we propose a general framework for polynomial-size kernelization of problems on subclasses of UDGs.

7.1 Introduction

DEFINITION 7.1.1. [Neighborhood equivalent]

Given a graph G and $u, v \in V(G)$, u is called neighborhood equivalent of v , denoted $u \equiv_N v$, if $N[v] = N[u]$.

Notice that \equiv_N is an equivalence relation. Given a vertex $v \in V(G)$, we denote by $[v]_N$ the equivalence class representing v under the neighborhood equivalence relation.

DEFINITION 7.1.2. [UDG edge contraction]

An edge $e = xy$ of $E(G)$ is UDG contractible if the graph obtained after contracting e is still realizable as a UDG. If $x \equiv_N y$ then $e = xy$ is a UDG contractible edge of G . In particular, G/e is a UDG obtained by removing either of the disks $\mathcal{D}(x)$ or $\mathcal{D}(y)$ from G . This is since the graph representation of $G \setminus x$ (likewise $G \setminus y$) is the same as the graph representation of G/e as $N[x] = N[y]$.

DEFINITION 7.1.3. [Replicate-redundant]

A parameterized problem Π is called replicate-redundant if for every instance $(G, k) \in \Pi$, any vertex $v \in V(G)$ and any subset $S \subseteq [v]_N$ of neighborhood equivalent vertices it holds that $(G/G[S], k) \in \Pi$. If the above condition holds for any $S \subseteq [v]_N$, then Π is called (vertex) replicate-redundant otherwise if the condition holds for any $S \subset [v]_N$ (but not for $S = [v]_N$), Π is called (edge) replicate-redundant.

It is easy to observe that in the above definition, (vertex) replicate-redundant is a stronger requirement. Hence, any (vertex) replicate-redundant problem is also (edge) replicate-redundant (yet, the converse is not true). A few examples of (vertex) replicate-redundant problems are DOMINATING SET, CLIQUE PARTITION, MINIMUM FILL-IN and INDEPENDENT SET while CLIQUE COVER is an (edge) replicate-redundant problem.

7.2 Data Reduction

We propose the following generic data reduction rule for the replicate-redundant problems.

Generic reduction rule.

Let v be a vertex in UDG G and $[v]_N$ denote the equivalence class of v under the neighborhood equivalence relation. Contract $G[[v]_N]$ to K_c for c a small constant depending on the problem (typically $c = 1$ or $c = 2$).

OBSERVATION: For an arbitrary vertex v , the set of neighborhood equivalent vertices $[v]_N$ induces a clique in G which is not maximal unless the connected component containing v is a clique.

REMINDER: A data reduction rule is called *correct* if the new instance after an application of the rule is a *yes*-instance if and only if the original instance is a *yes*-instance.

THEOREM 7.2.1. *The generic reduction rule is correct for a replicate-redundant problem and can be performed in time $\mathcal{O}(n^{2.3727})$ on a UDG with n vertices.*

Proof. For a replicate-redundant problem, by definition, contracting a set of neighborhood equivalent vertices into a single vertex does not change the size of an optimal solution.

Therefore, the generic reduction rule is correct if the rule is applied with $c = 1$ for a (vertex) replicate-redundant problem and likewise with $c = 2$ for an (edge) replicate-redundant problem. Also notice that after applying the generic reduction rule, the graph is still realizable as a UDG. This is since by Definition 7.1.2, contracting the set of neighborhood equivalent vertices $G[[v]_N]$ to K_c is equivalent to removing all except for c of vertices from $[v]_N$. Therefore, one can equivalently obtain a disk representation of the graph by removing all except for c of the disks for the points located in the region representing $[v]_N$.

Next, we analyze the time complexity of the reduction for a UDG. First, we need to find the neighborhood equivalence class of every vertex v , denoted $[v]_N$. Given G in form of an adjacency matrix \mathbf{A} (with entries on the diagonal set to 1), one can construct this subset in the following way. We maintain a set N and initially let $N = \{v\}$. For every 1 entry (a vertex in $N[v]$) in the row in \mathbf{A} corresponding to vertex v , we check if the row corresponding to that entry is exactly the same as the one for v and if it is the case, we add the corresponding vertex to N . Finally set $[v]_N = N$. In the worst case, we should perform this search for every vertex $v \in V(G)$. This can be done in $\deg^2(v)$ time for v and in $\mathcal{O}(nm)$ time for a graph with m edges.

This step can alternatively be performed using fast matrix multiplication. Notice that for every pair of vertices $u, v \in V(G)$, $(\mathbf{A}^2)_{u,v} = |N[u] \cap N[v]|$. To decide if $u \equiv_N v$, it is sufficient to check if $|N[u] \cap N[v]| = |N[u]| = |N[v]|$. Hence, $u \equiv_N v$, if $(\mathbf{A}^2)_{u,v} = (\mathbf{A}^2)_{u,u} = (\mathbf{A}^2)_{v,v}$. To contract $G[[v]_N]$ to K_c , we remove all except for c of the points arbitrarily from the points within the region corresponding to $[v]_N$. This can be done in a constant time. Therefore, application of the reduction rule to G takes $\mathcal{O}(n^{2.3727})$ time using the matrix multiplication algorithm reported in [114]. \square

THEOREM 7.2.2. *Let Π be a replicate-redundant problem such that Π admits a polynomial-size kernel on bounded-degree UDGS. Then Π admits a polynomial-size kernel on a UDG such that its reduced graph with respect to the generic reduction rule is (λ, α) -quasi-precision with $\alpha \leq 1/2$.*

We provide a proof for this theorem in the next section. As we shall see later, Theorem 7.2.2 applies to a number of important NP-hard problem including DOMINATING SET, CONNECTED DOMINATING SET, MAXIMUM INDEPENDENT SET, CLIQUE PARTITION and CLIQUE COVER.

We also obtain another interesting result as a consequence.

CLUSTER VERTEX DELETION.

Instance: An undirected graph $G = (V, E)$, and a non-negative integer k .

Question: Is there a vertex set $S \subseteq V$ with $|S| \leq k$ such that deleting all vertices in S from G results in a graph where every connected component is a complete graph?

Cluster vertex deletion number of a graph G is the minimum integer k for which (G, k) is a yes-instance for **CLUSTER VERTEX DELETION**

COROLLARY 7.2.1. *Let Π be a replicate-redundant problem such that Π admits a polynomial-size kernel on bounded-degree UDGs. Then Π admits a polynomial-size kernel on arbitrary UDGs when parameterized by cluster vertex deletion number.*

This is in accordance with the recent work of Doucha & Kratochvíl [52] which proposes cluster vertex deletion number as a parameter under which many problems admit FPT.

7.3 Analysis

We recall that, given a (λ, α) -quasi-precision graph $G = (V, E)$ with $\alpha \leq 1/2$, we define $\mathcal{X} = V_{<\lambda} \setminus S$ and let X_1, \dots, X_κ be a partition of \mathcal{X} into dense islands such that each X_i is bounded in a circle ζ_i of radius $\alpha_i \leq \alpha$ for $1 \leq i \leq \kappa$. Further, Ψ_i is the annulus bounded between circles of radii $1 + \alpha_i$ and $1 - \alpha_i$ concentric with ζ_i . Also for a subset \mathcal{P} of the plane, we define $\mu(\mathcal{P}, d)$ as the maximum number of points that can be placed in \mathcal{P} with the restriction that every pair of points are at pairwise Euclidean distance of at least d .

The following crucial lemma is a direct result of Lemma 4.3.1 and the definition of generic reduction rule.

LEMMA 7.3.1. *Let G' , the graph obtained after applying the generic reduction rule, be (λ, α) -quasi-precision UDG with $\alpha \leq 1/2$, then $|X_i| = \mathcal{O}((\mu(\Psi_i, \lambda) + |S|)^2)$ for all $1 \leq i \leq \kappa$.*

Proof. Draw an imaginary circle of radius 1 at each vertex $v \in \overline{N^\cap(X_i)}$. Let C be the set of all such circles. Notice that every such a circle intersects ζ_i in at least two points on the boundary. Let Φ^* be the union of all these circles and $\mathcal{D}(\zeta_i)$ denote the disk with ζ_i as its boundary. Consider the set of all regions R in $\mathcal{D}(\zeta_i) \setminus \Phi^*$ formed after intersecting the

boundary of circles with $\mathcal{D}(\zeta_i)$. We show that for any $r \in R$ and a vertex $w \in V \cap \bar{r}$, if $|V \cap \bar{r}| > c$, then the generic reduction rule is applicable to G' by setting $[w]_{\mathbb{N}} = V \cap \bar{r}$. First, notice that for every vertex $v \notin \overline{N^\cap(X_i)}$, either $V \cap \bar{r} \subseteq N[v]$ or $V \cap \bar{r} \cap N[v] = \emptyset$. Also since r is a region, for any vertex $x \in \overline{N^\cap(X_i)}$, either $V \cap \bar{r} \subseteq N(x)$ or none of vertices in $V \cap \bar{r}$ are adjacent to x . Therefore, vertices in $V \cap \bar{r}$ are neighborhood equivalent. Thus, $G'[[w]_{\mathbb{N}}]$ can be contracted to K_c for a constant c . Therefore, after applying the generic reduction rule, at most c vertices remain in each region in R . In other words, $|X_i| \leq c|R|$. To complete the proof, we upper-bound $|R|$. However, $\overline{N^\cap(X_i)} \subseteq \Psi_i$ by Lemma 4.3.1 and hence we can safely use an argument similar to the one in Lemma 4.2.1 to count the number of regions in R knowing that there are at most $\mu(\Psi_i, \lambda) + |S|$ vertices in Ψ_i . Consider the graph obtained from $\mathcal{D}(\zeta_i) \cap \Phi^* \cup \zeta_i$. Intersection point of every two circles in $C \cup \zeta_i$ forms a vertex of the graph. Two vertices are connected by an edge in this graph if their corresponding points are on the same circle in $C \cup \zeta_i$. Observe that the vertices which are located on the circle ζ_i have degree 3 and there are at most $2(\mu(\Psi_i, \lambda) + |S|)$ such vertices. This is while all other vertices have degree 4 since they are formed by intersecting two circle in C . Let there be n^* such vertices. The faces in the graph correspond to the regions R . Notice that we can get the number of edges in the above planar graph as half of sum of the vertex degrees as $2n^* + 3(\mu(\Psi_i, \lambda) + |S|)$. Using the Euler formula for planar graphs, we can find the number of faces (also $|R|$) as $n^* + \mu(\Psi_i, \lambda) + |S| + 2$. In the worst case when every two circles in C intersect at one point in $\mathcal{D}(\zeta_i)$, we get $n^* = \mathcal{O}(\mu(\Psi_i, \lambda) + |S|^2)$. Since $|X_i| \leq c|R|$, the statement of the lemma holds. \square

COROLLARY 7.3.1. $|X_i| \leq 576c\alpha_i^2/\lambda^4$.

Proof. To calculate $|X_i|$, we simply replace the value of $\mu(\Psi_i, \lambda)$ in the result of Lemma 7.3.1. Notice that $\mu(\Psi_i, \lambda)$ has been previously calculated in Lemma 4.3.3 as $24\alpha_i/\lambda^2$. \square

LEMMA 7.3.2. *Let $\Delta(G')$ be the maximum degree of the reduced graph obtained after applying the generic reduction rule to G . Then, $\Delta(G') = \mathcal{O}(1/\lambda^4)$.*

Proof. We consider a disk \mathcal{D} of radius 1 centered at an arbitrary point in the plane. To get the bound on $\Delta(G')$, we use an approach similar to Lemma 4.2.2. We count the maximum number of points that can be inside or on the boundary of \mathcal{D} . To do so, we cover \mathcal{D} with disks of radius α and consider the case where some of the disk contain dense islands. In this way, one can repeat the approach of Lemma 4.3.3 to get a cover with at most $\mu(\mathcal{D}, 2\alpha)$ disks

of radius α . Also, assuming the worst case scenario where each disks \mathcal{D}_α is inside dense island, we can get the bound on the number of vertices inside the disk from Corollary 7.3.1. Therefore,

$$|V' \cap \mathcal{D}| \leq |V' \cap \mathcal{D}_\alpha| \cdot \mu(\mathcal{D}, 2\alpha) \leq \frac{576\alpha^2}{\lambda^4} \cdot \frac{6}{\alpha^2} \leq \frac{3456}{\lambda^4}. \quad (7.1)$$

□

Finally, we show a proof for Theorem 7.2.2 based on the previous lemmas.

Proof of Theorem 7.2.2.

Let G' be the graph obtained after applying the generic reduction rule repeatedly to the input graph. To prove the kernel bound, we show that if $\alpha \leq 1/2$ then the size of G' can be upper-bounded by a polynomial in $(1/\lambda)$ and the parameter k . First of all notice that by the definition of a replicate-redundant problem, the generic reduction rule is correct for Π . Let c be the constant in the generic reduction rule. By Lemma 7.3.2, the maximum degree of G' is bounded by $\mathcal{O}(1/\lambda^4)$. Let $f(k, \Delta) = \mathcal{O}(k^\beta \Delta^\gamma)$ be the kernel-size for the problem Π on any graph with maximum degree Δ . Then the size of G' can be bounded by $\mathcal{O}(k^\beta/\lambda^{4\gamma})$. □

7.4 Application

Finally we demonstrate how the discussed framework on quasi-precision UDGs can be applied to a number of interesting combinatorial problems. In particular, for each problem studied in this section, in order to show that the framework applies, its enough to prove that the problem is replicate-redundant and admits polynomial-size kernels on bounded-degree graphs.

7.4.1 DOMINATING SET

DOMINATING SET.

Instance: An undirected graph $G = (V, E)$ and a non-negative integer k .

Question: Is there a subset $D \subseteq V$ of vertices with $|D| \leq k$ such that every vertex in $V \setminus D$ is adjacent to *at least* one vertex in D ?

DOMINATING SET problem is $W[2]$ -complete on arbitrary graphs and $W[1]$ -hard on bounded-area UDGs and hence is not kernelizable but admits kernels of $\mathcal{O}((j+1)^{2(i+1)}k^{2i^2})$ vertices on graphs excluding biclique $K_{i,j}$ as a subgraph [101].

LEMMA 7.4.1. *DOMINATING SET is (vertex) replicate-redundant and admits a linear-size kernel on bounded-degree graphs.*

Proof. We first prove that the problem is (vertex) replicate-redundant, that is, given any set N of neighborhood equivalent vertices and an input graph G , contracting $G[N]$ to a vertex x does not change the size of any optimal solution OPT for DOMINATING SET on G . Let $G' = G/G[N]$. We claim that $\text{OPT}' = \text{OPT}/G[N]$ is an optimal solution for G' . Since the vertices in N are neighborhood equivalent and $G[N]$ is a clique (by Observation 7.2), $N_G[N]$ is dominated by any vertex in N such as x . Therefore, $|\text{OPT} \cap N| \leq 1$ and $|\text{OPT}| = |\text{OPT}/G[N]|$. Suppose for contradiction that OPT' is not an optimal solution for G' and there is a solution OPT'^\dagger with $|\text{OPT}'^\dagger| < |\text{OPT}'|$. Then $\text{OPT}'^\dagger \cap N_{G'}[x]$ dominates $N_G[N]$. Therefore OPT'^\dagger is also a solution for G and $|\text{OPT}'^\dagger| < |\text{OPT}|$. A contradiction.

Now assume an optimal solution OPT' for G' . We claim that OPT' is also an optimal solution for G . Suppose for contradiction that there is a solution OPT^* for G with $|\text{OPT}^*| < |\text{OPT}'|$. Then $\text{OPT}^*/G[N]$ is a solution for G' and as shown before $|\text{OPT}^*/G[N]| = |\text{OPT}^*|$. Therefore, $|\text{OPT}^*/G[N]| < |\text{OPT}'|$. A contradiction.

One can get a trivial kernel for the problem on graphs of bounded-degree in the following way. Since the degree of every vertex in G is bounded by $\Delta(G)$, any vertex in OPT can dominate at most $\Delta(G) + 1$ vertices (including itself). Therefore we need at least $n/(\Delta(G) + 1)$ vertices in an optimal solution. That is, $n \leq k \cdot (\Delta(G) + 1)$. \square

7.4.2 CONNECTED DOMINATING SET

CONNECTED DOMINATING SET.

Instance: An undirected graph $G = (V, E)$ and a non-negative integer k .

Question: Is there a subset $D \subseteq V$ of vertices with $|D| \leq k$ and $G[D]$ connected such that every vertex in $V \setminus D$ is adjacent to *at least* one vertex in D ?

CONNECTED DOMINATING SET problem is $W[2]$ -complete on arbitrary graphs and remains NP-complete when restricted to UDGs. However, it is known to admit PTASs on this class

of graphs [25].

LEMMA 7.4.2. *CONNECTED DOMINATING SET is (vertex) replicate-redundant and admits a linear-size kernel on bounded-degree graphs.*

Proof. We first prove that the problem is (vertex) replicate-redundant, that is, given any set N of neighborhood equivalent vertices and an input graph G , contracting $G[N]$ to a vertex x does not change the size of any optimal solution OPT for **CONNECTED DOMINATING SET** on G . Let $G' = G/G[N]$. We claim that $\text{OPT}' = \text{OPT}/G[N]$ is an optimal solution for G' . Since the vertices in N are neighborhood equivalent and $G[N]$ is a clique (by Observation 7.2), $N_G[N]$ is dominated by any vertex in N such as x and $G[\text{OPT}']$ is connected. Therefore, $|\text{OPT} \cap N| \leq 1$ and $|\text{OPT}| = |\text{OPT}/G[N]|$. Suppose for contradiction that OPT' is not an optimal solution for G' and there is a solution OPT^\dagger with $|\text{OPT}^\dagger| < |\text{OPT}'|$. Then $\text{OPT}^\dagger \cap N_{G'}[x]$ dominates $N_G[N]$. Therefore OPT^\dagger is also a solution for G and $|\text{OPT}^\dagger| < |\text{OPT}|$. A contradiction.

Now assume an optimal solution OPT' for G' . We claim that OPT' is also an optimal solution for G . Suppose for contradiction that there is a solution OPT^* for G with $|\text{OPT}^*| < |\text{OPT}'|$. Then $\text{OPT}^*/G[N]$ is a solution for G' and as shown before $|\text{OPT}^*/G[N]| = |\text{OPT}^*|$. Therefore, $|\text{OPT}^*/G[N]| < |\text{OPT}'|$. A contradiction.

One can get a trivial kernel for the problem on graphs of bounded-degree in the following way. Since the degree of every vertex v in G is bounded by $\Delta(G)$, any vertex in OPT has at most $\Delta(G) - 1$ vertices adjacent to it which are not included in OPT . This is since $G[\text{OPT}]$ is connected and hence at least one neighbor of v is in OPT . Therefore we need at least $n/\Delta(G)$ vertices in an optimal solution. That is, $n \leq k \cdot \Delta(G)$. \square

7.4.3 INDEPENDENT SET

INDEPENDENT SET.

Instance: An undirected graph $G = (V, E)$ and a non-negative integer k .

Question: Is there a subset $I \subseteq V$ of vertices with $|I| \leq k$ such that no two vertices in I are adjacent?

The problem of **INDEPENDENT SET** remains in $W[1]$ even on the class of bounded-area UDGs [78].

LEMMA 7.4.3. INDEPENDENT SET is (vertex) replicate-redundant and admits a linear-size kernel on bounded-degree graphs.

Proof. We first prove that the problem is (vertex) replicate-redundant, that is, given any set N of neighborhood equivalent vertices and an input graph G , contracting $G[N]$ to a vertex x does not change the size of any optimal solution OPT for INDEPENDENT SET on G . Let $G' = G/G[N]$. We claim that $\text{OPT}' = \text{OPT}/G[N]$ is an optimal solution for G' . Since the vertices in N are neighborhood equivalent and $G[N]$ is a clique (by Observation 7.2), $|\text{OPT} \cap N| \leq 1$ and $|\text{OPT}| = |\text{OPT}/G[N]|$. Suppose for contradiction that OPT' is not an optimal solution for G' and there is a solution OPT^\dagger with $|\text{OPT}^\dagger| > |\text{OPT}'|$. Then $\text{OPT}^\dagger \cap N_{G'}[x]$ is an independent set for $N_G[N]$. Therefore OPT^\dagger is also a solution for G and $|\text{OPT}^\dagger| > |\text{OPT}|$. A contradiction.

Now assume an optimal solution OPT' for G' . We claim that OPT' is also an optimal solution for G . Suppose for contradiction that there is a solution OPT^* for G with $|\text{OPT}^*| > |\text{OPT}'|$. Then $\text{OPT}^*/G[N]$ is a solution for G' and as shown before $|\text{OPT}^*/G[N]| = |\text{OPT}^*|$. Therefore, $|\text{OPT}^*/G[N]| > |\text{OPT}'|$. A contradiction.

If $\Delta(G)$ is known, one can get an easy lower bound on the size of any independent set in G . Consider a greedy approach, where repeatedly an arbitrary vertex w is added to the independent set while removing $N[w]$ from G . Clearly, at each step one vertex is added to the independent set and the graph shrinks by at most $\Delta(G) + 1$ vertices. Therefore, the size of any independent set is larger than $n/(\Delta(G) + 1)$. Therefore, $n = \mathcal{O}(k \cdot \Delta(G))$. \square

7.4.4 CLIQUE COVER

CLIQUE COVER.

Instance: An undirected graph $G = (V, E)$, a non-negative integer k .

Question: Is there a set \mathcal{C} of cliques in G such that for every edge $e \in E$, there exists at least one clique $c \in \mathcal{C}$ with $e \in c$ and $|\mathcal{C}| \leq k$?

CLIQUE COVER is NP-complete even when restricted to planar graphs and does not admit a polynomial-size kernel on arbitrary graphs [31].

LEMMA 7.4.4. CLIQUE COVER is (edge) replicate-redundant and admits a linear-size kernel on bounded-degree graphs.

Proof. We first prove that the problem is (edge) replicate-redundant. That is, given any set N of neighborhood equivalent vertices and an input graph G , contracting $G[N]$ to an edge $e = vw$ does not change the size of any optimal solution OPT for CLIQUE COVER on G . Let $\mathcal{C} = \{c_1, \dots, c_\gamma\}$ be the set of maximal cliques in G that intersect N . That is, $N \subseteq V(c_i)$ for $1 \leq i \leq \gamma$. Notice that in Rule 2 in Section 6.1.1, we defined $I = \bigcap_{i=1}^\gamma V(c_i)$. Clearly, set I is a neighborhood equivalent set in G and hence the generic reduction rule for CLIQUE COVER would translate to Rule 2 as discussed in Section 6.1.1. Yet for the sake of completeness, we restate the proof here.

Let $N' = G[N] \setminus vw$. Then $G' = G/N'$ is the graph obtained after performing the contraction operation. We first show that $\text{OPT}' = \text{OPT}/N'$ is an optimal solution for G' . For the sake of contradiction, assume that G' has a solution OPT^* smaller than OPT/N' and let $\mathcal{C}^\dagger = \{c_j \in \mathcal{C} \mid c_j/N' \in \text{OPT}^*\}$. Then one can construct a solution $\text{OPT}^\ddagger = \text{OPT}^* \setminus (\mathcal{C}^\dagger/N') \cup \mathcal{C}^\dagger$ for G such that $|\text{OPT}^\ddagger| = |\text{OPT}^*|$. However, since $|\text{OPT}| = |\text{OPT}'|$, we get $|\text{OPT}^\ddagger| < |\text{OPT}|$, a contradiction.

Now, assume that OPT' is an optimal solution for G' and let $c_e \in \text{OPT}'$ be the clique that includes e . Then $G[(V(c_e) \setminus \{x, y\}) \cup N]$ is a clique in G . We prove that $\text{OPT}^* = (\text{OPT}' \setminus c_e) \cup G[(V(c_e) \setminus \{x, y\}) \cup N]$ is an optimal solution for G . That is, $|\text{OPT}^*| \leq |\text{OPT}|$. Assume for the contradiction that $|\text{OPT}^*| > |\text{OPT}|$. Then $\text{OPT}^\ddagger = \text{OPT}/N'$ is a solution for G' and $|\text{OPT}^\ddagger| = |\text{OPT}| < |\text{OPT}'|$, a contradiction to the assumption that OPT' is an optimal solution for G' . The proof of existence of polynomial-size kernel for the problem on bounded-degree graphs is trivial and is inferred from the discussions in Chapter 6. \square

7.4.5 CLIQUE PARTITION

CLIQUE PARTITION.

Instance: An undirected graph $G = (V, E)$, a non-negative integer k .

Question: Is there a set \mathcal{C} of cliques in G such that for every vertex $v \in V$, there exists at least a clique $c \in \mathcal{C}$ with $v \in c$ and $|\mathcal{C}| \leq k$?

For general graphs, CLIQUE PARTITION is equivalent to VERTEX COLORING of the complement graph and hence is not kernelizable and is inapproximable within $n^{1-\delta}$ for any $\delta > 0$ [54]. It remains NP-complete even when restricted to unit coin graphs [22] – a subclass of precision UDGs with maximum degree of 5.

LEMMA 7.4.5. CLIQUE PARTITION *is (vertex) replicate-redundant and admits a linear-size kernel on bounded-degree graphs.*

Proof. We first prove that the problem is (vertex) replicate-redundant, that is, given any set N of neighborhood equivalent vertices and an input graph G , contracting $G[N]$ to a new vertex x does not change the size of any optimal solution OPT for CLIQUE PARTITION on G . Let $\mathcal{C} = \{c_1, \dots, c_\gamma\}$ be the set of maximal cliques in G that intersect N . That is, $N \subseteq V(c_i)$ for $1 \leq i \leq \gamma$. Notice that in Rule 3 in Section 3.4, we defined $I = \bigcap_{i=1}^\gamma V(c_i)$. Clearly, set I is a neighborhood equivalent set in G and hence the generic reduction rule for CLIQUE PARTITION would translate to Rule 3 as discussed in Section 3.4. Yet for the sake of completeness, we restate the proof here.

Let $G' = G/G[N]$. We claim that $\text{OPT}' = \text{OPT}/G[N]$ is an optimal solution for G' . For the sake of contradiction, assume that G' has a solution OPT^* with $\text{OPT}^* < \text{OPT}'$ and let $\mathcal{C}^\dagger = \{c_j \in \mathcal{C} \mid c_j/G[N] \in \text{OPT}^*\}$. Then one can construct a solution $\text{OPT}^\ddagger = \text{OPT}^* \setminus (\mathcal{C}^\dagger/G[N]) \cup \mathcal{C}^\dagger$ for G such that $|\text{OPT}^\ddagger| = |\text{OPT}^*|$. However, since $|\text{OPT}| = |\text{OPT}'|$, we get $|\text{OPT}^\ddagger| < |\text{OPT}|$, a contradiction.

Now, assume that OPT' is an optimal solution for G' and let $c_x \in \text{OPT}'$ be the clique that includes the new vertex x . Then $G[(V(c_x) \setminus x) \cup N]$ is a clique in G . We prove that $\text{OPT}^* = (\text{OPT}' \setminus c_x) \cup G[(V(c_x) \setminus x) \cup N]$ is an optimal solution for G . That is, $|\text{OPT}^*| \leq |\text{OPT}|$. Assume for the contradiction that $|\text{OPT}^*| > |\text{OPT}|$. Then $\text{OPT}^\ddagger = \text{OPT}/G[N]$ is a solution for G' and $|\text{OPT}^\ddagger| = |\text{OPT}| < |\text{OPT}'|$, a contradiction to the assumption that OPT' is an optimal solution for G' .

The proof of existence of polynomial-size kernel for the problem on bounded-degree graphs is trivial and is inferred from the discussion in Section 4.2. \square

COROLLARY 7.4.1. DOMINATING SET, CONNECTED DOMINATING SET, MAXIMUM INDEPENDENT SET, CLIQUE PARTITION *and* CLIQUE COVER *admit kernels linear in the solution size and polynomial in λ when the reduced graph with respect to the generic reduction rule (alternatively the input graph) is (λ, α) -quasi-precision with $\alpha < 1/2$.*

COROLLARY 7.4.2. DOMINATING SET, CONNECTED DOMINATING SET, MAXIMUM INDEPENDENT SET, CLIQUE PARTITION *and* CLIQUE COVER *admit polynomial-size kernels on*

arbitrary UDGs when parameterized with respect to cluster vertex deletion number.

7.5 Summary

We designed a general framework for polynomial kernelization of problems on quasi-precision unit disk graphs (UDGs) through the introduction of a simple generic data reduction rule. Specifically, we proved that when λ is set as a parameter, polynomial-size kernels and hence FPT results are attainable for numerous problems on a large class of (λ, α) -quasi-precision UDGs. Our framework indeed applies in a stronger form when the reduced graph with respect to the generic reduction rule (rather than the input instance) is (λ, α) -quasi-precision with $\alpha \leq 1/2$.

Furthermore, we characterized the problems for which a polynomial-size kernels are derivable through the application of this framework by providing sufficient criteria for the correctness of the generic reduction rule. The problems covered in this framework include many classical NP-hard problem for which no positive kernelization results were known beforehand on any non-trivial graph class. Among those are DOMINATING SET, CLIQUE COVER, CLIQUE PARTITION and INDEPENDENT SET.

Chapter 8

Conclusion and Future Work

8.1 Summary of Contributions

We designed a framework obtaining parameterization and fixed-parameter tractable algorithms on unit disk graphs (UDGs). Our approach is based on the novel idea of using convex regions as a model for maximal cliques in UDGs and relies on non-trivial geometric results that relate the convex regions to cliques in UDG. An important step in our approach is to efficiently apply a set of data reduction rules to the input instance in a polynomial-time preprocessing phase. Hence the design of efficient reduction rules and proving their correctness for each problem is an essential yet significant part of our work.

Studying the problem of CLIQUE PARTITION, we obtained parameterized algorithms of running-time $\mathcal{O}(n^{6k+2})$ for the problem on arbitrary UDGs which significantly improves the previously known algorithms. The only previously known singly-exponential algorithm for the problem is restricted to UDGs whose points are within a square of known length and has the running-time of $\mathcal{O}(n^{80q})$ where q is an upper bound on the size of clique partition. On the classes of ϵ -precision and (λ, α) -(relaxed)-quasi-precision UDGs with $\alpha \leq 1/2$, we were able to obtain first-time fixed-parameter algorithms of running-times $\mathcal{O}(2^{(3k+1)\log k + 12k\log \frac{1}{\epsilon} + 30k}n + mn)$ and $2^{\mathcal{O}(k\log \frac{k}{\lambda})}n + \mathcal{O}(mn)$ for CLIQUE PARTITION.

Later, we demonstrated the generality of our framework by showing how our technique can be fine-tuned to apply to a number of other interesting problems. Particularly, studying the problem of WEIGHTED CLIQUE PARTITION, we came up with a first-time set of reduction rules for the problem and followed a similar approach as before in order to construct parameterized algorithms with the same running-time as the ones for CLIQUE PARTITION

for the problem of WEIGHTED CLIQUE PARTITION on the mentioned classes of UDGs.

We further considered an adaptation of our framework to the problem of CLIQUE COVER and obtained parameterized algorithms of running-times $\mathcal{O}(n^{6k+3})$, $\mathcal{O}(2^{(3k+1)\log k + 12k\log \frac{1}{\epsilon} + 24k}m + mn)$ and $2^{\mathcal{O}(k\log \frac{k}{\lambda})}m + \mathcal{O}(mn)$ on the classes of arbitrary UDGs, ϵ -precision UDGs and (λ, α) -(relaxed)-quasi-precision UDGs with $\alpha \leq 1/2$, respectively. To the best of our knowledge, no parameterized algorithms were previously known for the problems of CLIQUE COVER and WEIGHTED CLIQUE PARTITION on UDGs or any of its subclasses. Our framework further applies in a stronger form when the graph reduced with respect to the reduction rules (rather than input instance) belongs to the discussed subclasses of UDGs.

Studying the optimization version of CLIQUE PARTITION, we designed randomized $(1 + \epsilon)$ -approximation algorithms for MINIMUM CLIQUE PARTITION on arbitrary UDGs. Our first PTAS adopts the approach of [54] along with a new packing argument and has an improved running-time of $n^{(176/\epsilon)^2 + \mathcal{O}(1/\epsilon)}$. Furthermore, we proposed a novel approach for deriving PTAS which relies on the geometric theorem on coverage of cliques by convex regions. The new PTAS still runs in time $n^{\mathcal{O}(1/\epsilon^2)} \cdot (1/\epsilon^2)$, yet the hidden constant in big \mathcal{O} notation, in power of n , is significantly (at least 13 times) smaller than the fastest previously known PTAS. In practice, this allows the algorithm to run in a reasonable time on instances that are significantly larger in size.

In an attempt to explore the boundaries of fixed-parameter tractability of problems on UDGs, we introduced (α, λ) -quasi-precision class as unit disk graphs which are only partly λ -precision, yet their areas of imprecision have bounded area. We showed that our definition of (α, λ) -quasi-precision UDGs is *ubiquitous* in the sense that any UDG is quasi-precision for some values of λ and α . For quasi-precision UDGs, maximum degree and the size of clique-partitions can be unbounded.

We later demonstrated the use of (λ, α) -(relaxed)-quasi-precision UDGs by describing how the kernelization and parameterization frameworks obtained in our work can be extended to apply to this subclass when $\alpha \leq 1/2$. We further studied the structure of quasi-precision UDGs and proved structural obstructions for the parameterization and kernelization of problems under our framework. In particular, we showed that UDGs which do not admit FPT algorithms under our framework include as subgraph a loosely-connected co-bipartite graph. The proposed obstruction classifies the cases where the clique removal is not feasible. Fomin et al. [SODA 2012] conjectured that the main obstacle for obtaining

subexponential algorithms on UDGs is the design of efficient “clique cleaning” procedures [63]. Our work further affirms their conjecture for obtaining FPT and kernelization results for problems on UDGs.

We also described a general framework for kernelization of problems on quasi-precision UDGs. Through the introduction of a simple generic data reduction rule that reduces the input instance in polynomial time to smaller instance and a detailed analysis, we proved that when λ is set as a parameter, polynomial-size kernels are attainable for numerous problems on a large class of (λ, α) -quasi-precision UDGs. Furthermore, we characterized the problems for which polynomial-size kernels are derivable through the application of this framework. In particular, we provide the sufficient criteria for the problems under which the generic reduction rule remains as a correct reduction for the problem. Finally, we exhibited the use of this framework by showing how it can be applied to obtain linear-size kernels for a number of classical combinatorial problems for many of which no positive kernelization results were known beforehand on any non-trivial UDG subclass. Among those are DOMINATING SET, CONNECTED DOMINATING SET, CLIQUE COVER, CLIQUE PARTITION and INDEPENDENT SET.

8.2 Future Work

It is worthwhile investigating the application of the frameworks introduced here to other related problems on the discussed classes of UDGs. The approach presented in this work is very general and thus for many problems the application of the frameworks might be possible with only slight modifications.

It might also be possible to benefit from other interesting geometric properties of UDGs in order to design frameworks for obtaining data reduction and parameterization results on a completely different set of problems on UDGs. Yet a fundamental challenge in this line of research would be to identify and generalize the required criteria and techniques.

Another interesting and yet challenging direction is to identify structural properties of UDGs whose presence, or lack of which, affects the parameterized tractability or kernelization bounds for a number of attractive problems. Such works could in turn lead to discovery of other useful subclasses of geometric graphs under which the candidate problems exhibit nice computational complexity behaviors.

As we showcased in Chapter 5, another avenue would be to combine the parameterization

results on UDGs with partitioning and approximation techniques in order to obtain faster PTAS algorithms for problems on UDGs.

Appendix A

Proofs

A.1 Proof of Formula (3.3) for calculating $|q_{xz}q_{yz}|$

Proof. Let $\|xy\| = \sigma$ as before. Also let us denote the distance between x (respectively y) and z , by σ' (respectively σ''). First we prove that σ' has to be larger than $1 - \sigma^2$. To see the proof, notice that since the Helly property does not hold for the 3 vertices, there must be a point inside the triangle Δxyz whose distance from all sides is at least $1/2$ (e.g. the circumcenter of Δxyz , denoted o_c satisfies the above condition). Then one can obtain a lower bound on $\sigma' = \|xz\|$ using easy calculations as follows:

$$\sigma' \geq \sqrt{\left(\frac{1}{2} + \frac{\sqrt{1-\sigma^2}}{2}\right)^2 + \frac{\sigma^2}{4}} = \sqrt{1 + \sqrt{1-\sigma^2}/2} > 1 - \sigma^2. \quad (\text{A.1})$$

Next, we obtain an upper bound on the size of the line segment $q_{xz}q_{yz}$ as $\|q_{xz}q_{yz}\| \leq \sqrt{3 - 2\sigma + 2\sigma^2 - 2\sigma^3 - \sigma^4}$. To see a proof, consider the intersection point of line segments $q_{xz}x$ and $q_{yz}y$. Call it w . Let h be the perpendicular of triangle $\Delta q_{xz}q_{xy}w$ to the edge $q_{xz}q_{xy}$. Also, assume that $\|q_{xz}w\| \leq \|q_{yz}w\|$ (and hence $\|xw\| \geq \|yw\|$).

$$\|q_{xz}q_{yz}\| \leq \sqrt{\|q_{xz}w\|^2 - \|h\|^2} + \sqrt{\|q_{yz}w\|^2 - \|h\|^2}. \quad (\text{A.2})$$

But $\|xw\| + \|yw\| \geq \|xy\| = \sigma$. Then $\|xw\| \geq \sigma/2$. Also, by definition $\|q_{xz}x\| = \|q_{xz}w\| + \|xw\| = 1$. Therefore, $\|q_{xz}w\| \leq 1 - \sigma/2$. Also, notice that $2\|h\| + \|yw\| \geq \sigma'$ and $\sigma > \|yw\|$. Hence, $2\|h\| \geq \sigma' - \sigma \geq 1 - \sigma - \sigma^2$. Finally, by replacing the above two inequalities in the former equation, we get

$$\|q_{xz}q_{yz}\| \leq \sqrt{3 - 2\sigma + 2\sigma^2 - 2\sigma^3 - \sigma^4}. \quad (\text{A.3})$$

When $0 < \sigma < 1$,

$$\|q_{xz}q_{yz}\| < \sqrt{3} + \frac{\sigma}{\sqrt{3}} + \frac{\sigma^2}{24\sqrt{3}} + \frac{5\sigma^3}{36\sqrt{3}} + \frac{53\sigma^4}{3456\sqrt{3}}. \quad (\text{A.4})$$

□

Bibliography

- [1] F. Abu-Khzam. Kernelization algorithms for d -hitting set problems. *Algorithms and Data Structures*, pages 434–445, 2007. → pages 13
- [2] J. Alber, H.L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33(4):461–493, 2002. → pages 13
- [3] J. Alber, H. Fan, M.R. Fellows, H. Fernau, R. Niedermeier, F. Rosamond, and U. Stege. A refined search tree technique for dominating set on planar graphs. *Journal of Computer and System Sciences*, 71(4):385–405, 2005. → pages 13
- [4] J. Alber, M.R. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating set. *Journal of the ACM (JACM)*, 51(3):363–384, 2004. → pages 13
- [5] J. Alber and J. Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *Journal of Algorithms*, 52(2):134–151, 2004. → pages 4, 34, 54, 56, 57
- [6] N. Alon and S. Gutner. Linear time algorithms for finding a dominating set of fixed size in degenerated graphs. *Algorithmica*, 54(4):544–556, 2009. → pages 25
- [7] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k -trees. *Discrete Applied Mathematics*, 23(1):11–24, 1989. → pages 11
- [8] B.S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM (JACM)*, 41(1):153–180, 1994. → pages 16
- [9] B. Balasundaram and S. Butenko. Optimization problems in unit-disk graphs. *Encyclopedia of Optimization*, pages 2832–2844, 2009. → pages 2, 29
- [10] Luca Becchetti, Peter Korteweg, Alberto Marchetti-Spaccamela, Martin Skutella, Leen Stougie, and Andrea Vitaletti. Latency constrained aggregation in sensor networks. *Algorithms-ESA 2006*, pages 88–99, 2006. → pages 93
- [11] J Bhasker and Tariq Samad. The clique-partitioning problem. *Computers & Mathematics with Applications*, 22(6):1–11, 1991. → pages 37

- [12] M. Blanchette, E. Kim, and A. Vetta. Clique cover on sparse networks. *ALLENEX*, pages 93–102, 2012. → pages 86
- [13] H. Bodlaender. Dynamic programming on graphs with bounded treewidth. *Automata, Languages and Programming*, pages 105–118, 1988. → pages 11
- [14] Hans L Bodlaender. Kernelization: New upper and lower bound techniques. In *Parameterized and Exact Computation*, pages 17–37. Springer, 2009. → pages 12
- [15] Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (meta) kernelization. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS '09*, pages 629–638, 2009. → pages 20, 21, 22
- [16] Béla Bollobás. *The art of mathematics: Coffee time in Memphis*. Cambridge University Press, 2006. → pages 41
- [17] H. Breu. *Algorithmic aspects of constrained unit disk graphs*. PhD thesis, university of british columbia, 1996. → pages 37
- [18] H. Breu and D.G. Kirkpatrick. Unit disk graph recognition is NP-hard. *Computational Geometry*, 9(1-2):3–24, 1998. → pages 29
- [19] L. Cai, J. Chen, R.G. Downey, and M.R. Fellows. Advice classes of parameterized tractability. *Annals of pure and applied logic*, 84(1):119–138, 1997. → pages 12
- [20] Liming Cai and Xiuzhen Huang. Fixed-parameter approximation: conceptual framework and approximability results. *Parameterized and Exact Computation*, pages 96–108, 2006. → pages 11
- [21] Vasilis Capoyleas, Günter Rote, and Gerhard Woeginger. Geometric clusterings. *Journal of Algorithms*, 12(2):341–356, 1991. → pages 76
- [22] M.R. Cerioli, L. Faria, T.O. Ferreira, and F. Protti. On minimum clique partition and maximum independent set on unit disk graphs and penny graphs: complexity and approximation. *Electronic Notes in Discrete Mathematics*, 18:73–79, 2004. → pages 37, 76, 110
- [23] M.S. Chang and H. Müller. On the tree-degree of graphs. In *Graph-Theoretic Concepts in Computer Science*, pages 44–54. Springer, 2001. → pages 85
- [24] J. Chen, H. Fernau, I.A. Kanj, and G. Xia. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. *STACS 2005*, pages 269–280, 2005. → pages 13
- [25] Xiuzhen Cheng, Xiao Huang, Deying Li, Weili Wu, and Ding-Zhu Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, 2003. → pages 108

- [26] B.N. Clark, C.J. Colbourn, and D.S. Johnson. Unit disk graphs. *Discrete mathematics*, 86(1-3):165–177, 1990. → pages 4, 29, 30, 31, 54
- [27] B. Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and computation*, 85(1):12–75, 1990. → pages 21
- [28] B. Courcelle. The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues. *ITA*, 26:257–286, 1992. → pages
- [29] B. Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. *Handbook of graph grammars and computing by graph transformations*, 1:313–400, 1997. → pages
- [30] B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science*, 109(1-2):49–82, 1993. → pages 21
- [31] Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, and Magnus Wahlström. Clique cover and graph separation: new incompressibility results. In *Proceedings of the 39th international colloquium conference on Automata, Languages, and Programming-Volume Part I*, pages 254–265. Springer-Verlag, 2012. → pages 85, 109
- [32] H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 251–260. ACM, 2010. → pages 12
- [33] E.D. Demaine and M.T. Hajiaghayi. Equivalence of local treewidth and linear local treewidth and its algorithmic applications. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 840–849. Society for Industrial and Applied Mathematics, 2004. → pages 15
- [34] E.D. Demaine and M.T. Hajiaghayi. Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 682–689. Society for Industrial and Applied Mathematics, 2005. → pages
- [35] E.D. Demaine, M.T. Hajiaghayi, N. Nishimura, P. Ragde, and D.M. Thilikos. Approximation algorithms for classes of graphs excluding single-crossing graphs as minors. *Journal of Computer and System Sciences*, 69(2):166–195, 2004. → pages
- [36] E.D. Demaine, M.T. Hajiaghayi, and D.M. Thilikos. The bidimensional theory of bounded-genus graphs. *SIAM Journal on Discrete Mathematics*, 20(2):357–371, 2007. → pages 15

- [37] Erik Demaine, MohammadTaghi Hajiaghayi, and Bojan Mohar. Approximation algorithms via contraction decomposition. *Combinatorica*, 30:533–552, 2010. → pages 17, 18
- [38] Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Fixed-parameter algorithms for the (k, r) -center in planar graphs and map graphs. In *acm transactions on algorithms*, pages 829–844, 2003. → pages 15
- [39] Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Bidimensional parameters and local treewidth. *SIAM Journal on Discrete Mathematics*, 18:501–511, 2004. → pages 15, 16
- [40] Erik D. Demaine, Fedor V. Fomin, MohammadTaghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on graphs of bounded-genus and H -minor-free graphs. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '04, pages 830–839. Society for Industrial and Applied Mathematics, 2004. → pages
- [41] Erik D. Demaine and Mohammad T. Hajiaghayi. Diameter and Treewidth in Minor-Closed Graph Families, Revisited. *Algorithmica*, 40(3):211–215, 2004. → pages
- [42] Erik D Demaine and MohammadTaghi Hajiaghayi. Bidimensionality, map graphs, and grid minors. *arXiv preprint cs/0502070*, 2005. → pages 15
- [43] Erik D. Demaine and MohammadTaghi Hajiaghayi. Bidimensionality: new connections between FPT algorithms and PTASs. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '05, pages 590–601, 2005. → pages 15, 23
- [44] Erik D Demaine and MohammadTaghi Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, 51(3):292–302, 2008. → pages 15, 23
- [45] Erik D. Demaine, Mohammadtaghi Hajiaghayi, and Ken ichi Kawarabayashi. Algorithmic Graph Minor Theory: Decomposition, Approximation, and Coloring. In *46th Annual IEEE Symposium on Foundations of Computer Science*, pages 637–646. Press, 2005. → pages 15
- [46] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Contraction decomposition in H -minor-free graphs and algorithmic applications. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 441–450, 2011. → pages 18, 20
- [47] Erik D. Demaine, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors. *Algorithmica*, 41:245–267, 2005. → pages 15

- [48] M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and IDs. *Automata, Languages and Programming*, pages 378–389, 2009. → pages 27, 32, 33
- [49] F. Dorn. Dynamic programming and fast matrix multiplication. *Algorithms–ESA 2006*, pages 280–291, 2006. → pages 13
- [50] F. Dorn, F.V. Fomin, and D.M. Thilikos. Catalan structures and dynamic programming in h-minor-free graphs. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 631–640. Society for Industrial and Applied Mathematics, 2008. → pages 20
- [51] Frederic Dorn, Eelko Penninkx, Hans L Bodlaender, and Fedor V Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut branch decompositions. In *Algorithms–ESA 2005*, pages 95–106. Springer, 2005. → pages 14
- [52] Martin Doucha and Jan Kratochvíl. Cluster vertex deletion: a parameterization between vertex cover and clique-width. In *Mathematical Foundations of Computer Science 2012*, pages 348–359. Springer, 2012. → pages 104
- [53] R.G. Downey and M.R. Fellows. *Parameterized complexity*, volume 5. Springer New York, 1999. → pages 9, 10, 12, 13, 25
- [54] A. Dumitrescu and J. Pach. Minimum Clique Partition in Unit Disk Graphs. *Graphs and Combinatorics*, 27(3):399–411, 2011. → pages 3, 5, 37, 48, 53, 76, 77, 78, 79, 80, 81, 83, 84, 110, 115
- [55] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000. → pages 17
- [56] P Erdős, László Lovász, and Katalin Vesztergombi. On the graph of large distances. *Discrete & Computational Geometry*, 4(1):541–549, 1989. → pages 72, 73
- [57] Paul Erdős, Adolph W Goodman, and Louis Pósa. The representation of a graph by set intersections. *Canad. J. Math*, 18:106–112, 1966. → pages 86
- [58] M. Fellows, G. Fertin, D. Hermelin, and S. Vialette. Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. *Automata, Languages and Programming*, pages 340–351, 2007. → pages 27
- [59] Gerd Finke, Vincent Jost, Maurice Queyranne, and András Sebő. Batch processing with interval graph compatibilities between tasks. *Discrete Applied Mathematics*, 156(5):556–568, 2008. → pages 93
- [60] H. Fleischner, E. Mujuni, D. Paulusma, and S. Szeider. Covering graphs with few complete bipartite subgraphs. *Theoretical Computer Science*, 410(21-23):2045–2053, 2009. → pages 99, 100

- [61] Jörg Flum and Martin Grohe. *Parameterized complexity theory*, volume 3. Springer Heidelberg, 2006. → pages 11
- [62] Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Bidimensionality and EPTAS. In *SODA*, pages 748–759, 2011. → pages 15
- [63] Fedor V Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1563–1575. SIAM, 2012. → pages 40, 74, 116
- [64] Fedor V Fomin and Yngve Villanger. Subexponential parameterized algorithm for minimum fill-in. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1737–1746. SIAM, 2012. → pages 29
- [65] F.V. Fomin, D. Lokshtanov, S. Saurabh, and D.M. Thilikos. Bidimensionality and kernels. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 503–510. Society for Industrial and Applied Mathematics, 2010. → pages 22, 23, 24, 34
- [66] F.V. Fomin, G. Philip, Y. Villanger, S. Chakraborty, and A. Kumar. Minimum Fill-in of Sparse Graphs: Kernelization and Approximation. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011)*, volume 13, pages 164–175. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011. → pages 28, 29
- [67] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman & Co., 1979. → pages 13
- [68] Dion Gijswijt, Vincent Jost, and Maurice Queyranne. Clique partitioning of interval graphs with submodular costs on the cliques. *RAIRO-Operations Research*, 41(03):275–287, 2007. → pages 93
- [69] P. Golovach and Y. Villanger. Parameterized complexity for domination problems on degenerate graphs. In *Graph-Theoretic Concepts in Computer Science*, pages 195–205. Springer, 2008. → pages 25
- [70] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Data reduction and exact algorithms for clique cover. *Journal of Experimental Algorithmics (JEA)*, 13:2, 2009. → pages 85, 87, 89
- [71] M. Grigni and P. Sissokho. Light spanners and approximate TSP in weighted graphs with forbidden minors. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 852–857. Society for Industrial and Applied Mathematics, 2002. → pages 20
- [72] Jonathan L Gross and Jay Yellen. *Graph theory and its applications*. CRC press, 2006. → pages 85

- [73] Q. Gu and N. Imani. Connectivity is not a limit for kernelization: planar connected dominating set. *LATIN 2010: Theoretical Informatics*, pages 26–37, 2010. → pages 14
- [74] J. Guo and R. Niedermeier. Linear problem kernels for NP-hard problems on planar graphs. *Automata, languages and programming*, pages 375–386, 2007. → pages 32
- [75] Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007. → pages 12
- [76] T.W. Haynes, S.M. Hedetniemi, S.T. Hedetniemi, and M.A. Henning. Domination in graphs applied to electric power networks. *SIAM Journal on Discrete Mathematics*, 15(4):519–529, 2002. → pages 13
- [77] T.W. Haynes, S.T. Hedetniemi, and P.J. Slater. *Fundamentals of domination in graphs*, volume 208. CRC, 1998. → pages 13
- [78] H. Ito and M. Kadoshita. Tractability and intractability of problems on unit disk graphs parameterized by domain area. In *Proceedings of the 9th International Symposium on Operations Research and Its Applications (ISORA10)*, pages 120–127, 2010. → pages 4, 34, 35, 54, 63, 108
- [79] B. Jansen. Polynomial kernels for hard problems on disk graphs. In *Algorithm Theory-SWAT 2010: 12th Scandinavian Workshop on Algorithm Theory, Bergen, Norway, June 21-23, 2010. Proceedings*, volume 6139, pages 310–321. Springer-Verlag New York Inc, 2010. → pages 32, 33, 34
- [80] B.M.P. Jansen and H.L. Bodlaender. Vertex cover kernelization revisited: Upper and lower bounds for a refined parameter. In *28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011)*, volume 9, pages 177–188, 2011. → pages 12, 13
- [81] I. Kanj and F. Zhang. 3-hitting set on bounded degree hypergraphs: upper and lower bounds on the kernel size. *Theory and Practice of Algorithms in (Computer) Systems*, pages 163–174, 2011. → pages 13
- [82] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 1972. → pages 37
- [83] Ton Kloks, Dieter Kratsch, and Haiko Müller. Finding and counting small induced subgraphs efficiently. *Information Processing Letters*, 74(3):115–121, 2000. → pages 50, 87, 96
- [84] L.T. Kou, L.J. Stockmeyer, and C.K. Wong. Covering edges by cliques with regard to keyword conflicts and intersection graphs. *Communications of the ACM*, 21(2):135–139, 1978. → pages 86

- [85] Xiang-Yang Li. Algorithmic, geometric and graphs issues in wireless networks. *Wireless Communications and Mobile Computing*, 3(2):119–140, 2003. → pages 14
- [86] Kevin M Lillis, Sriram V Pemmaraju, and Imran A Pirwani. Topology control and geographic routing in realistic wireless networks. In *Ad-Hoc, Mobile, and Wireless Networks*, pages 15–31. Springer, 2007. → pages 36
- [87] R.J. Lipton and R.E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, pages 177–189, 1979. → pages 16, 17
- [88] Daniel Lokshtanov, Matthias Mnich, and Saket Saurabh. A linear kernel for a planar connected dominating set. *Theoretical Computer Science*, 412(23):2536–2543, 2011. → pages 14
- [89] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM (JACM)*, 41(5):960–981, 1994. → pages 86
- [90] Weizhong Luo, Jianxin Wang, Qilong Feng, Jiong Guo, and Jianer Chen. Improved linear problem kernel for planar connected dominating set. *Theoretical Computer Science*, 2013. → pages 14
- [91] M.V. Marathe, H. Breu, H.B. Hunt III, S.S. Ravi, and D.J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25(2):59–68, 1995. → pages 30
- [92] Michal Pilipczuk Marek Cygan, Marcin Pilipczuk and Jakub Onufry Wojtaszczyk. Kernelization Hardness of Connectivity Problems in d -Degenerate Graphs. In *WG'10*, pages 147–158, 2010. → pages 25, 26, 27, 28
- [93] Dániel Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008. → pages 11
- [94] H. Melissen. *Packing and Covering with Circles*. PhD thesis, Universiteit Utrecht, the Netherlands,, 1997. → pages 60
- [95] H. Moser. A problem kernelization for graph packing. *SOFSEM 2009: Theory and Practice of Computer Science*, pages 401–412, 2009. → pages 34
- [96] E. Mujuni and F. Rosamond. Parameterized complexity of the clique partition problem. In *Proceedings of the fourteenth symposium on Computing: the Australasian theory-Volume 77*, pages 75–78. Australian Computer Society, Inc., 2008. → pages 37
- [97] George L Nemhauser and Leslie E Trotter Jr. Vertex packings: structural properties and algorithms. *Mathematical Programming*, 8(1):232–248, 1975. → pages 12
- [98] Saurav Pandit and Sriram V Pemmaraju. Finding facilities fast. In *Distributed Computing and Networking*, pages 11–24. Springer, 2009. → pages 36

- [99] Sriram V Pemmaraju and Imran A Pirwani. Energy conservation via domatic partitions. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 143–154. ACM, 2006. → pages 36
- [100] Sriram V Pemmaraju and Imran A Pirwani. Good quality virtual realization of unit ball graphs. In *Algorithms–ESA 2007*, pages 311–322. Springer, 2007. → pages 36
- [101] G. Philip, V. Raman, and S. Sikdar. Solving dominating set in larger classes of graphs: FPT algorithms and polynomial kernels. *Algorithms-ESA 2009*, pages 694–705, 2009. → pages 25, 28, 107
- [102] I. Pirwani and M. Salavatipour. A weakly robust PTAS for minimum clique partition in unit disk graphs. *Algorithm Theory-SWAT 2010*, pages 188–199, 2010. → pages 37, 76, 93
- [103] Rolf Rees. Minimal clique partitions and pairwise balanced designs. *Discrete mathematics*, 61(2):269–280, 1986. → pages 37
- [104] N. Robertson and P.D. Seymour. Graph minors. XX. Wagner’s conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004. → pages 15
- [105] Neil Robertson and Paul D Seymour. Graph minors. I. Excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983. → pages 11
- [106] Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986. → pages 11
- [107] K.J. Supowit. *Topics in computational geometry*. PhD thesis, University of Illinois at Urbana-Champaign, 1981. → pages 37
- [108] G.F. Toth. Thinnest covering of a circle by eight, nine, or ten congruent circles. *Combinatorial and computational geometry*, 52:361, 2005. → pages 59
- [109] Vijay V Vazirani. *Approximation algorithms*. springer, 2004. → pages 10
- [110] K. Wagner. Über eine Eigenschaft der ebenen Komplexe. *Mathematische Annalen*, 114(1):570–590, 1937. → pages 14
- [111] P.J. Wan, K.M. Alzoubi, and O. Frieder. A simple heuristic for minimum connected dominating set in graphs. *International Journal of Foundations of Computer Science*, 14(2):323–333, 2003. → pages 13
- [112] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Englewood Cliffs, 2001. → pages 8
- [113] A. Wiese and E. Kranakis. Local PTAS for independent set and vertex cover in location aware unit disk graphs. *Distributed Computing in Sensor Systems*, pages 415–431, 2008. → pages 32

- [114] Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th symposium on Theory of Computing*, pages 887–898. ACM, 2012. → pages 50, 51, 87, 88, 96, 103

Index

- H*-minor-free graph, 18
- $\gamma(G)$, 13
- $\gamma_c(G)$, 14
- λ -close vertex, 62
- λ -short edge, 62
- p*-MAX-CMSO problem, 21
- p*-MIN-CMSO problem, 21
- (edge) replicate-redundant, 102
- (vertex) replicate-redundant, 102

- extended protrusion, 22

- apex graph, 24

- basic circle, 70
- bidimensional, 16
- boundary arc, 41
- bounded-area UDG, 34
- bounded-genus graph, 17

- clique intersection graph, 58
- clique-edge intersection graph, 90
- clique-prism, 69
- cluster vertex deletion number, 104
- color class, 18
- compact problem, 21
- complete bipartite graph, 14
- composition algorithm, 27
- composition problem, 27
- contraction, 9
- contraction bidimensional, 16
- contraction decomposition, 17
- contraction-closed problem, 17
- convex hull, 76
- convex region, 38
- correct reduction rule, 102

- Counting Monadic Second Order (CMSO), 20

- degenerate graph, 25
- dense island, 62
- dense subgraph, 62
- diameter edge, 70
- dominated vertex, 14
- domination number, 13

- edge-weighted graph, 20
- efficient PTAS, 10
- EPTAS, 10
- especial clique, 40
- Euler genus, 17
- expansion candidate, 70

- face distance, 58
- feedback vertex set number, 12
- finite integer index, 23
- fixed-parameter algorithm, 9
- FP-approximation, 11
- FPT, 9
- FPT-time., 9

- genus, 18

- hamiltonian cycle, 19
- Helly property, 31
- hitting set, 13

- induced subgraph, 8
- isomorphic graphs, 8

- kernel size, 10
- kernelization, 10

- minimum spanning tree, 20
- minor, 9
- minor bidimensional, 15
- minor operation, 15
- minor-closed problem, 17
- minor-free, 18

- neighborhood equivalent, 101
- non-contractible curve, 18
- non-helly set, 47, 82
- non-overlapping cliques, 76

- parameterized problem, 9
- penny graph, 34
- planar graph, 14
- polynomial hierarchy, 12
- polynomial parameter reducible, 25
- polynomial parameter transformation, 25
- polynomial time approximation scheme, 10
- precision UDG, 57
- problem kernel, 2, 10
- Problems index:
 - H*-MATCHING, 33
 - X*-ANNOTATED CLIQUE COVER, 86
 - α -INDEPENDENT SET, 35
 - k*-COLORING, 35
 - k*-CUT, 19
 - 3-HITTING SET, 13
 - BICLIQUE PARTITION/COVER, 99
 - BICLIQUE VERTEX PARTITION/COVER, 99
 - CHORDAL GRAPH COMPLETION, 29
 - CLIQUE COVER, 86, 109
 - CLIQUE PARTITION, 36, 110
 - CLUSTER VERTEX DELETION, 104
 - COLORFUL GRAPH MOTIF, 27
 - CONNECTED DOMINATING SET, 14, 15, 28, 107
 - CONNECTED EDGE DOMINATING SET, 15
 - CONNECTED FEEDBACK VERTEX SET (CFVS), 26
 - CONNECTED ODD CYCLE TRANSVERSAL, 26
 - CONNECTED VERTEX COVER, 26, 32
 - DOMINATING SET, 13, 15, 35, 106
 - EDGE CLIQUE PARTITION, 37
 - EDGE DOMINATING SET, 15
 - FACE COVER, 15
 - FEEDBACK VERTEX SET, 15
 - HAMILTONIAN CIRCUIT, 34, 35
 - HITTING SET, 13
 - INDEPENDENT SET, 35, 108
 - MAX CLIQUE, 10
 - MINIMUM CLIQUE PARTITION, 48, 75, 76
 - MINIMUM FILL-IN, 28, 29
 - MINIMUM MAXIMAL MATCHING, 15
 - RED-BLUE DOMINATING SET, 32, 33
 - STEINER TREE, 26
 - UNWEIGHTED TSP TOUR, 15
 - VERTEX COVER, 15
 - WEIGHTED CLIQUE PARTITION, 93, 96
 - WEIGHTED TRAVELING SALESMAN, 15
 - WEIGHTED TSP, 15, 19, 20
- protrusion, 22
- PTAS, 18, 20, 75

- quasi-compact problem, 21
- quasi-precision UDG, 62, 101

- radial distance, 20
- reduction rule, 22
- region coverage graph, 38
- relaxed-quasi-precision UDG, 68
- replicate-redundant, 102

- separable, 23
- separation, 16
- separation property, 23
- separator, 16
- set of representatives, 23
- simplicial vertex, 50, 95
- sink vertex, 40
- spanner, 18
- spanning subgraph, 20

sparse subgraph, 62
strip graph, 37
subgraph, 8

terminal, 22
terminal graph, 22
tree decomposition, 11
treewidth, 11

UDG contractible, 101
unit coin graph, 34, 37

W[1], 10
W[1]-hard, 33
W[2], 13
W[2]-hard, 25
Wagner's theorem, 14
weighted graph, 18

XP, 10

yes-instance, 49